Abstract of "Probabilistic approaches for rigorous and efficient analysis of statistical properties of large datasets" by Lorenzo De Stefani, Ph.D., Brown University, May 2020.

The analysis of the statistical properties of data is a core goal in computer science. The challenges encountered in this task have evolved through time due to the explosion of the size of the datasets being considered and to the development of new analysis tools. Rather than computing the exact quantities of interest, computing high-quality estimates allows for a considerable reduction of computational effort while retaining meaningful and reliable insight into the statistical properties of the data.

In this thesis, we study the applications of probabilistic analysis and statistical learning theory to the analysis of large datasets. Using these concepts allows us to rigorously characterize the relationship between the amount of input *training* data, the *quality*, in terms of *confidence* and *precision*, of the approximations achievable for a certain task, and the *complexity* of the task itself.

In the first part of this work, we study the application of statistical learning tools such as Rademacher Complexity and VC dimension, to the classical *hypothesis testing* problem. We introduce RadaBound, a rigorous, efficient, and practical procedure for controlling the generalization error when using a holdout sample for multiple adaptive testing. Further, in the standard (nonadaptive) multiple hypothesis setting, our RadeFWER (resp., RadeFDR) procedure achieves control of the Family Wise Error Rate -FWER- (resp., False Discovery Rate -FDR-). The statistical power of our procedures decreases with respect to the *actual complexity* of the considered hypotheses, captured in a *data-dependent* way by estimation of the Rademacher Complexity.

In the second part of this thesis, we study massive dynamic graphs being observed as an adversarial stream of edges insertions and deletions towards counting the number of occurrences of a given small sub-graph pattern or a "*motif*" of interest: we present Trièst, a suite of one-pass streaming algorithms for triangle counting based on *reservoir sampling* and the *random pairing* sampling schemes, and Tiered Sampling, an extension of Trièst which employs multiple reservoir sample tiers in order to accurately estimate the count of rare and complex patterns.

Probabilistic approaches for rigorous and efficient analysis of statistical properties of large datasets

by
Lorenzo De Stefani
B. S., University of Padova - Italy, 2009
M. S., University of Padova - Italy, 2012

A thesis proposal submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2020

This dissertation by Lorenzo De Stefani is accepted in its present form by the Department of Computer Science as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date _____ _____

Professor Eli Upfal, Director

Recommended to the Graduate Council

Date _____ _____

Professor Michael Littman, Reader
(Brown University)

Date _____ _____

Tim Kraska, Reader
(MIT)

Approved by the Graduate Council

Date _____ _____

Andrew G. Campbell
Dean of the Graduate School

Acknowledgements

I would like to thank my advisor, Professor Eli Upfal, for his patient and inspiring guidance over these years. I would like to thank Professor Michael Littman and Professor Tim Kraska for being part of my thesis committee. I would like to thank the Computer Science Department at Brown University for accepting me in the program, and, in particular, Professor Ugur Centitemel for offering me the opportunity to teach CS1010.

I would like to thank all my academic siblings, my collaborators and the other talented students I had the opportunity to meet at Brown. I learn from them more that I can tell. I would like to thank Professor Gianfranco Bilardi, for encouraging me to pursue graduate studies in the United States.

I would like to thank my mother, Ivana, and my father, Antonio, for supporting me in this adventure from afar. Without their encouragement and nurture, I would not have embarked in this enterprise. Finally, I want to thank my beloved wife, Megumi, without whose support, understanding and patience I would not be here today.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The analysis of the statistical properties of data is a core goal in computer science. The challenges and approaches to this task have evolved through time due to the explosion of the size of the datasets being considered and due the development of new analysis tools. Data availability is thus a crucial driver of the most important challenges in computer science. Analysts are often faced with the necessity of tackling a wide array of situations: in some cases, they are called to extract insight while analyzing very limited available data; in other cases, they are tasked to analyze massive, dynamic datasets towards synthesizing insights. In both cases, the size of the available data shapes the most critical aspects of the task at hand.

In the case of low data availability, the analyst needs to put in place rigorous analysis in order to ensure that any insight he/she may infer will be indeed valid for the entire population from whom the data is drawn, that is, to ensure that observations made on the available data do in fact *generalize* to the larger population. Without such careful analysis, the analyst may easily incur the risk of erroneously attributing statistical relevance to phenomena that are instead only peculiar to the specific small dataset being used. In practice, this may have very serious consequences: for example, a financial trader may infer from the short-term behavior of the market a long time trend which is not adequately supported by historical data and, thus, using such erroneous insight to commit to a risky investment. Even more critically, outcomes of a clinical trial in which the analyst does not opportunely *correct* for the number of patients in the study and the different elements of the patients' clinical condition may lead to wrong conclusions, which may, in turn, lead to the approval of an instead dangerous medication, or, vice versa, the rejection of an actually effective drug.

While for massive datasets the availability of data does not present challenges regarding the *generalizability* or the *statistical relevance* of possible insight, the analyst must generally face some computational difficulties: Handling massive datasets is often unpractical, as it requires both high computation time and high availability of computational resources such as memory, GPA availability, and cloud storage. While modern computing systems are generally equipped with considerable storage and computing power, the rate of growth of the technology can not keep up with the

amount of available data for many aspects of great importance and interest such as social networks, telecommunication data, market data, consumer information, traffic and supply routes. Further, even as more efficient storage and computation technology become available, the time, computation cost, and energy cost of moving data appear unavoidable. This fact is particularly relevant as data communication is known to be the main bottleneck of algorithmic performance. This trend appears to be destined to increase as, even as technology approaches *physical limitations* such as the maximum number of computational or memory units which can be printed on a unit of surface, and as the speed of data communication approaches the speed of light, some limitation of the computational power and minimum latency for data movement appears unavoidable. Vice versa, data availability is projected to be only ever increasing in both sheer size and complexity at an unparalleled rate. Further, in many instances, such as the ones previously presented as examples, the nature of such massive datasets is intrinsically dynamic as the datasets are continuously updated. This fact poses additional challenges as it makes computationally intensive procedures that require to be run from scratch each time the dataset is updated particularly unappealing, if not outright nonsensical. Instead, analysts should pursue algorithmic techniques that allow them to study such *streams* of updates and which produce results that can be updated continuously and efficiently.

In this thesis, we study how the application of probabilistic modeling and sampling techniques may address all the mentioned concerns and challenges in the different possible scenarios of data analysis. We argue how, rather than computing the exact quantities of interest, computing high-quality estimates often allows for a considerable reduction in terms of computational effort while retaining meaningful and reliable insight in the statistical properties of the data. Our solutions aim to be efficient both in terms of data requirement (for the low data availability scenario) and in terms of the use of computational resources and effort.

In the most common scenario for the low data availability setting, while the goal of the analysis is to study the statistical properties of a large underlying population, only a *small sample* of the population is available to the analyst. Such limited availability may depend on different factors: In some scenarios, such as clinical trials, acquiring input points may be very expensive. For the massive dataset setting, operating on a small sample drawn from the available data is useful in order to reduce the computational cost of the operations involved in the analysis. While this generally leads to the loss of the ability of computing *exact values* for the quantity of interest, trough a rigorous analysis it is possible to show that, for opportune sampling methods, such loss is guaranteed (with high probability) to be tolerable without compromising the statistical obtained by analyzing only a small sample. Therefore, regardless of the circumstance, operating on the sample allows to obtain *estimates* and *approximations* of the quantities of interest for the analysis. In order for such estimates to be *significant*, it is desirable that they are *representative* of the corresponding actual *ground truth* values with respect to the entire population. In this thesis, we discuss how, using opportunely chosen sampling methods and probabilistic modeling for the data generation process, it is possible to ensure such *representativeness*.

Both Classical Statistical theory and Learning Theory offer various tools that allow relating

quantities and properties evaluated on the sample to the entire population. The popular paradigm of "*data exploration*" has however introduced new challenges which manifest themselves in the necessity to evaluate a very high number of hypotheses or queries using the same sample dataset while controlling either the number of *false positives* (in the case of hypotheses) or the distribution of the errors of the estimates. In the classical setting, the analyst aims to verify whether a certain *belief* or *hypothesis* regarding the distribution is indeed correct. This is generally achieved by evaluating the given sample dataset with an opportune procedure. However, the paradigm of *data exploration* is based on a different setup and objective, which could be summarized as asking "*is there anything interesting that we can detect from the sample?*". Rather than verifying a small group of given hypotheses, the goal of *data exploration* is to *uncover* possible interesting patterns and relations with respect to the original underlying distribution by making use of the sample data. While such a task may appear similar in nature, its actual realization corresponds to evaluating a high number of possible hypotheses and beliefs and then reporting on those which appear to be of actual statistical relevance. Such an approach leads to an explosion of the number of properties or hypotheses to be evaluated: the more thorough the exploration, the higher the number of hypotheses that are to be evaluated.

This setting is generally referred to in the literature as *Multiple Hypothesis Testing* or *Multi Comparison Testing*. Intuitively, as the number of properties to be tested increases, so does the probability of one such property exhibiting a *peculiar statistical behavior*. Without opportune statistical tools, it is, however, problematic to decide whether such phenomena actually correspond to *statistically sound* patterns, or if they are an effect of the randomness and the use of a finite sample (i.e., they are *false positives*). The necessity of controlling the likelihood and the number of false positives in data exploration led to a very rich literature of desired statistical testing procedures aimed to control, among others, the probability of false discoveries (FWER) and the expected ratio of false positives (FDR). Despite the richness and variety of such procedures, all appear to suffer a considerable loss in statistical power as the number of properties being evaluated grows. In our work, we build on results from probabilistic analysis and statistical learning theory in order to overcome these challenges. Using these concepts allows us to rigorously characterize the relationship between the amount of input *training* data, the *quality*, in terms of *confidence* and *precision*, of the approximations achievable for a certain task, and the *complexity* of the task itself. We claim that by using these results, it is possible to obtain *rigorous* procedures, which exhibit a much higher *statistical power* compared to traditional techniques.

Further, exploratory data analysis is usually executed as an *adaptive process*, where the choice of the properties to be considered at a given step may, in general, depend on the outcome of the analysis in the previous steps. Such a form of adaptive exploration is notorious for easily leading to *overfit*. That is a situation for which observations obtained on the sample do not *generalize* to the entire population. In this work we build on uniform convergence bound from statistical learning theory to produce a procedure which allows quantifying the *error* (i.e., the overfit) accumulated after a certain number of adaptive exploration steps, and to, therefore, halt the exploration if it is

no longer possible to meet the desired guarantees.

Data visualization is a tool of widespread use and great importance in data analysis. Using histograms, heatmaps, and graph representation of data is useful to provide to analysts and users an immediate overview of complex phenomena and can highlight interesting relations between features of the population being observed, or interesting difference in the behavior of sub-populations. Such tools are particularly useful in the context of the "data exploration" task discussed before as they provide an intuitive and user-friendly tool to delve into the analysis of otherwise complex data. While such systems are growing in popularity [231, 43] they do hide some possible threats for inexpert users: whenever data visualizations are used to extract insight on statistical properties of the data they should be treated as hypotheses and, thus, treated opportunely in order to avoid erroneous insights (i.e., False Positives or False Negatives). In our work, we build on the *Alpha Investing* technique [80] to obtain procedures to ensure control of the *risk of erroneous insights*, measured by means of the *marginal False Discovery Rate* (mFDR). Such procedures are meant to provide assistance to the analysis in order to warn him/her on weather a phenomenon being displayed by a visualization may not be indicative of an actual, statistically significant phenomenon in the overall population. We develop a number of different procedures suited for different degrees of prior information on the data from the side of the analyst.

Visualizations are also used to guide data exploration by means of *recommendations*, that is, given a *starting visualization* of some data property (e.g., the distribution of a subpopulation with respect to the values) a *visual recommendation system* will suggest other visualizations that are of interest with respect to the starting visualization. While there is a large degree of freedom in the definition of interest, in many cases of practical importance, a *candidate visualization* is it deemed the more interesting, the more the population distribution it represents differs from that in the original starting visualization. While the visual appearance of such difference may be useful to guide the user in the data exploration, it may also lead to erroneous conclusions as in some case the difference observed between two visualizations may be the consequence of the specific composition of the data sample being used in the representation (i.e., an overfit to the sample) rather than a representative of a *general* statistical phenomenon for the overall population. We develop VizRec a framework for validating visual data recommendations. Our approach tests each pair of starting and candidate recommendations, and it determines whether the available data allows us to conclude that any observed difference is indeed statistically relevant, that is, it corresponds to an actual large difference with respect to the entire population with high probability. Our procedure estimated the values of the quantities represented by the visual representation using a sample of the population and uses such estimates to estimate the level of *interest* of a candidate recommendation measured as the difference between the estimates with respect to the values of the starting visualization. To ensure the significance of the estimates, we characterize the quality of the obtained estimated by building on uniform convergence bounds based on the notion of Vapnik–Chervonenkis dimension of the class of predicate functions corresponding to the selection criteria for the data appearing in the visualization. Our method allows for the control of the Family Wise Error Rate of the probability

of recommending visualizations which are not actually of interest with respect to the starting visualization. As our procedure corrects for all possible combinations of parameters corresponding to possible visualization, it allows for control of probability of error for procedures which adaptively explore the space of possible visualizations.

In the third part of this thesis, we focus on the use of sampling approaches for the analysis of massive datasets. As mentioned earlier, in this setting, the main challenges have to do with the amount of computational effort and resources required to extract insight from massive datasets. We consider the analysis of massive dynamic networks. Graph models are pervasive in may aspects of data science. Some notable examples include protein interaction networks [160], Social Networks, and relations between consumers and products. Further, these graphs are intrinsically dynamic as new relations (i.e., edges) are continuously added or removed from the network. Such evolving nature introduces several new challenges: running expensive algorithms based on post-processing is not possible as there is *no end* to the graph evolution, re-running onerous algorithms for every update is unpractical due to the frequencies of updates. Finally, as the size of graphs of interest may be in the hundreds of billions of edges (e.g., Facebook, Twitter), maintaining the entirety of such graphs in memory, or making multiple passes over the data. In our work, we focus on the task of counting the number of occurrences of subgraphs, such as triangles and 4-cliques, in massive dynamic graphs modeled as a stream of edge updates. We show how by maintaining a sample of the entire graph and studying the properties of such a sample, it is possible to estimate the number of triangles and other motifs of interest in the entire graph. Our first approach, TRIÉST, builds on *reservoir sampling* to obtain a one-pass, sampling-based method for efficiently estimating the number of triangles in a graph stream using a fixes memory. Further, we discuss how this approach can be extended to *fully-dynamic* edge streams, for which edges can be deleted as well as added. This extension is achieved by integrating the *random pairing* sampling scheme, a variation of the reservoir sampling scheme, into the paradigm of TRIÉST. For all TRIÉST algorithms, we present a rigorous analysis of the statistical properties of the obtained estimates: we show that the estimates are *unbiased*, (i.e., heir value corresponds *in expectation* to the exact triangle count), we obtain bounds on the variance of such estimates, and corresponding *concentration bounds*. The analysis is made challenging by the fact that while the reservoir sampling scheme ensures that each edge of the graph is uniformly likely to be included in the sample, the events corresponding to each edge being included in the sample are not independent. While this may, at first glance, appear to be a minor complication, it snowballs introducing many additional challenges which require, in some case, to break down all possible order of arrivals for edges belonging to pairs of triangles sharing an edge. Regardless, the effort is worthwhile as the TRIÉST algorithms exhibit low variance for many graphs of real-world interest and with billions of edges while using a sample whose size is less than 0.1% of the size of the graph.

Towards pushing further these results to the task of counting more complex motifs (e.g., 4 and 5-cliques) we develop the Tiered Sampling framework. While edge-only sampling algorithms such

as TRIÉST can, in principle, be extended to more complex patterns, they generally exhibit poor performance as the variance of the obtained estimates tend to increase *exponentially* with respect to the number of edges in the pattern of interest. Tiered Sampling addresses this issue by breaking the task of detecting occurrences of a subgraph of interest in two parts, the detection of a *prototype* sub-pattern, and its combination with a few remaining edges to detect a full occurrence of the subgraph whose count we are interested in. Thus, the *fixed size* available memory is correspondingly divided into two *tiers* one used to maintain a uniform sample of edges observed on the stream, and the other used to maintain a sample of the prototype sub-patterns observed so far. Both such samples are maintained using the reservoir sampling scheme. Tiered Sampling retains many of the desirable properties of TRIÉST: it is a one-pass algorithm, it yields unbiased estimates, it exhibits low experimental variance, it uses a finite size memory, and it does not require any additional prior information on the properties of the overall network (e.g., node degrees, degrees distribution, diameter). The use of multiple tiers does, however, introduce an additional layer of dependence, which, in turn, further complicates the analysis of the statistical properties of Tiered Sampling estimates. Finally, discuss how to further generalize this framework by simplifying the analysis using opportune approximates.

In the last chapter of this thesis, we study the problem of aggregating rankings produced by users. We establish a probabilistic model for the generation of such rankings for which given a *central ranking* corresponding to the *ground truth*, which we aim to reconstruct, other rankings are observed with a probability which depends on *how different* those rankings are with respect to the central ranking. We measure such difference in terms of the AP correlation statistic for which discrepancies in highest-ranked positions of a pair of rankings are weighted more heavily than discrepancies in lower positions. This is aimed to model a rather intuitive phenomenon according to whom users belonging to the same group are most likely to agree on most of the *important decisions*. We show how under this probabilistic model, it is possible to reconstruct the central from a sample of the population whose cardinality depends on the quality of the desires approximation without requiring any additional knowledge of the ground truth ranking.

All mentioned results share a common foundation given by probabilistic modeling of the population, or the phenomena, being observed via the available data. Our methods are based on sampling, either used as a way to gain insight into a large unavailable population or as a means of handling massive datasets while containing computational effort. Our methods rely on obtaining *estimates* of the quantities of interest from such a sample while rigorously analyzing the quality and *reliability* of such estimates building on tools from statistical learning theory and probabilistic analysis. Our methods aim to be efficient and to provide statistically rigorous guarantees for the insight achievable through their use.

Organization of the presentation of the content of the thesis

In the first part of this work, we study the application of statistical learning tools such as Rademacher Complexity and VC dimension, to the classical *hypothesis testing* problem. We explore the application of the Rademacher Complexity uniform convergence bounds to the setting of multiple hypothesis testing. Traditional methods offering control of the Family Wise Error Rate (FWER) do not scale well as their *statistical power* considerably decreases as the number of hypotheses being considered scales the power. In our RadeFWER approach, the statistical power of the procedure decreases with respect to the *actual complexity* of the considered hypotheses, captured by a *data-dependent* way by estimation of the Rademacher Complexity. We verify the correctness of our approach and the achieved increase of statistical power compared to classical FWER control procedures experimentally. As an additional result, we introduce RadaBound, a rigorous, efficient, and practical procedure for controlling the generalization error when using a holdout sample for multiple adaptive testing. Practical data analysis and machine learning are usually iterative and adaptive processes. Based on the results of the hypotheses (or models) tested so far, the user selects a new hypothesis (or model) to test until an adequate model is found. Standard statistical inference techniques and machine learning generalization bounds assume that tests are run on data selected independently of the hypotheses. To evaluate the performance of such an iterative analysis process, we need to quantify the accumulated errors in running a sequence of non-independent tests on the same data. Our solution is based on a new application of the *Rademacher Complexity* generalization bounds adapted to dependent tests. We demonstrate the statistical power and practicality of our method through extensive simulations and comparisons to alternative approaches.

In the second part of this thesis, we investigate the importance of rigorous statistical control of multiple hypothesis testing in the context of the visual exploration of large datasets. We argue that data representation should indeed be considered hypotheses, and the statistical significance of the observed phenomenon should correspondingly rigorously be ascertained. We build upon the *Alpha Investing* testing scheme to obtain a procedure which allows for adaptive visual exploration while controlling the *marginal False Discovery Rate* (mFDR), that is the ratio between the expected number of false positives and the expected number of discoveries. As additional result, we consider the problem commonly referred to as *visualization recommendation*. In this setting, we aim to provide the analyst some assistance while recommending data visualizations which may be of *interest*. The challenge of this scenario is given by the fact that it often the case that differences observed in the evaluation of a given sample may not correspond to differences with respect to the entire population. In this work, we use statistical learning analysis tools in order to both obtain strong FWER guarantees when validating the recommendations of "*interesting*" visualization. Our VizRec system is is based on the analysis of the Vapnik–Chervonenkis (VC) dimension of the class of queries corresponding to the visualizations being considered.

In the third, and final, part of this proposal, we study massive dynamic graphs being observed

ad an adversarial stream of edges insertions and deletions. The goal of our analysis is to count the number of occurrences of a given small sub-graph pattern or a "*motif*" of interest.

We present TRIÈST, a suite of one-pass streaming algorithms based on *reservoir sampling* and the *random pairing* sampling schemes. TRIÈST algorithms, which yield high-quality unbiased estimate of the number of *triangle motifs* in fully dynamic edges stream while using a small local memory in which a small and random sample of the edges is maintained. We present a complete analysis of the variance of the proposed algorithms and corresponding large deviations bounds. We also present TIERED SAMPLING, an extension of the TRIÈST paradigm which employs multiple reservoir sample tiers in order to accurately estimate the count of rare and complex patterns such as 4 and 5 cliques. We evaluate the performance of both approaches thought extensive experimental analysis on real world graphs with up to five billions of edges while maintaining only a small fraction ($\leq 0.5\%$) of such graphs as a sample.

Part I

Hypothesis testing and statistical learning

Chapter 2

Family Wise Discovery Rate using Rademacher Complexity uniform convergence bounds

Modern data analysis tools allow users to easily examine many hypotheses on the same dataset, significantly increasing the risk of making false discoveries. While this issue, commonly known in statistics as the "*multiple hypothesis testing error*", is well understood, classical statistical methods offering control of the Family Wise Error Rate (FWER) or False Discovery Rate (FDR) do not provide satisfying solutions for large scale data analysis as their *statistical power* considerably decreases as the number of hypotheses being considered increases. In this Chapter, we present a novel approach for addressing this problem, building on applications of statistical learning theory and Rademacher Complexity to hypotheses testing. In our approaches, the statistical power of the procedure decreases with respect to the "*actual complexity*" of the family of hypotheses being tested, captured in a *data-dependent* way by estimation of the Rademacher Complexity, rather than to its sheer cardinality. Our methods do not rely on any assumption on the correlation of the hypotheses being considered. We verify experimentally the correctness of our approach and the achieved increase of statistical power compared to classical FWER and FDR control

2.1 Introduction

As advances in technology allow for the collection, storage, and analysis of vast amounts of data, the task of screening and assessing the significance of discoveries is becoming a major challenge in data analysis applications. In particular, modern technologies allow for easy testing of large numbers of hypotheses on the same data, increasing the risk of claiming *false positive discoveries*. While this issue, commonly known as the *multiple hypothesis testing error*, is well known, standard frequentist statistical tests do not provide satisfying solutions for large scale data analysis.

To appreciate the challenge, consider the following simple data analysis example: We obtain DNA samples from 100 patients with a common phenotype (disease) and 100 patients in a control group without the phenotype. The goal is to identify a DNA marker for the phenotype, i.e., a mutation that increases the likelihood of the phenotype. Assume that we check one DNA location and observe that it has a mutation in 27 patients with the phenotype and in only 13 patients in the control group. The probability of such an observation when the occurrences of the mutations are independent of the phenotype is less than 0.02. Therefore, we can conclude that a mutation at that location is a valid marker for the phenotype. Now assume that this location was one of 100 different DNA locations tested for mutations. Even if mutations in all these locations are not related to the phenotype, we expect to see a discrepancy that is as large (as $14 = 27 - 13$) or larger in at least two of the tested locations. Thus, an observation that indicated a good marker when tested in isolation cannot be considered valid when it is one of a large number of tests. Accurate large scale exploratory data analysis requires efficient tools for distinguishing between spurious and meaningful discoveries within a large number of observations.

For concreteness, we focus here on standard hypotheses about expectations of functions or probabilities of events. Given a set of hypotheses evaluated on the same sample, the goal is to avoid erroneous discoveries (i.e., *false positives*), while not compromising the *statistical power* of the test (i.e., minimizing *false negatives*). The classic, frequentist statistics, solution for this problem typically consists of two stages. First, each hypothesis is evaluated in isolation to compute the p-value of the corresponding observed statistic in the data. Then, a "multi-comparison" control procedure evaluates the collection of the p-values to select the rejected null hypotheses, satisfying the required FWER (*Family Wise Error Rate*) or FDR (*False Discovery Rate*) bound. A number of multi-comparisons control methods have been proposed (see Section 2.2) but in general, these approaches do not scale well to large numbers of hypotheses, rapidly losing statistical power.

In this Chapter, we develop a novel alternative approach for large scale multi-hypotheses testing problem, building on modern machine learning techniques for uniform convergence bounds. The statistical power of our procedures scales with respect of the *actual complexity* (or expressiveness) of the family of hypotheses being tested, here evaluated in terms of its Rademacher Complexity, rather than its sheer cardinality. This leads to a substantial improvement in terms of statistical power for settings in which there are strong correlations between the hypotheses being tested.

Our contributions: We introduce RadeFWER and RadeFDR, a pair of statistical procedures which achieve control of, respectively, FWER and FDR. Both procedures ensure *strong control* (i.e., not only under the *global null hypothesis*) and do not require any assumption regarding the independence or the type of correlation of the test statistics associated with the hypotheses being considered. We compare our procedures with classical statistical control procedures (i.e., Holm's [106] and Hochberg's [103] procedures for FWER, and the Benjamini-Hochberg [15] and the Benjamini-Yakuteli [18] procedures for FDR control), and we show that, in a variety of settings, our novel approach outperforms the standard methods both on synthetic and real-world data.

Depending on the specific approach used to bound the generalization error, our methods lead

to procedures with different guarantees and requirements. In this work, we focus on the use of a bound obtained using the Central Limit Theorem for Martingales (MCLT) which allows us to obtain asymptotic bounds (more similar to those obtained in the classical hypotheses testing literature). Further, we show that by bounding the generalization error using Bernstein's inequality for martingales, we can obtain a *finite-sample bound* which, albeit clearly much more conservative than it asymptotic counterpart, leads to a procedure with stronger guarantees.

2.2 Standard multi-comparisons control procedures

For a given set $\mathcal{H}$ of null hypotheses to be tested, with $|\mathcal{H}| = m$, let m_0 denote the number of false null hypotheses (i.e, true discoveries). Further, let V denote the number of null hypotheses rejected by a statistical testing procedure Π, and let R (resp., S) denote the number of false (resp., true) positives. Out of these, only m and V are *observable*.

Definition 2.0.1 (FWER control)**.** *Given $\delta \in (0, 1]$, a statistical testing procedure ensures Family Wise Error Rate (FWER) control at level δ if $Pr(R > 0) \leq \delta$.*

Definition 2.0.2 (FDR control)**.** *Given $\delta \in (0, 1]$, a statistical testing procedure ensures False Discovery Rate (FDR) control at level δ if $E\left[R/V\right] \leq \delta$.*

Any procedure which controls FWER at level δ also controls FDR at the same level. FWER is, in general, a much stricter criterion than FDR, with the latter being introduced mainly in order to retain statistical power. There is an extremely rich literature on statistical testing procedure achieving control of FWER and FDR. Here we provide a partial overview, with a focus on the methods used as direct terms of comparison for our proposed methods.

FWER control procedures: As our proposed RadeFWER control procedure controls FWER *strongly* (not only under the *global null hypothesis*), henceforth we focus on comparisons with procedures controlling FWER strongly. Holm's procedure [7] is *uniformly more powerful* than Bonferroni's [28] and Tukey's [228] procedure. While Hommel's procedure is more powerful than Holm's, it holds only under non-negative dependence of the test statistics associated with the hypotheses being tested. Hommel's procedure [107] is uniformly more powerful than other procedures which require the same, or more restrictive, assumptions on the dependence between hypotheses, such as Sǐdak's [212], Holm-Sǐdak's, Hochberg's [103]. Tukey's and Dunnett's tests are used only as follow-up tests to ANOVA and hold assuming that the calculated means of the sets of observations are independently normally distributed with the same variance.

- **Bonferroni - Holm's FWER control procedure [106]:**
  - Sort the p-values from lowest to highest as $P_{(1)}, P_{(2)}, \ldots, P_{(m)}$, and let $H_{(i)}$ denote the null hypothesis associated with the p-value $P_{(i)}$;
  - Let k denote the smallest value such that $P_{(k)} > \frac{\delta}{m-k+1}$;

– Reject the null hypotheses $H_{(1)}, \ldots, H_{(k-1)}$. If $k = 1$, no null hypothesis is rejected.

- **Hochberg's FWER control procedure [103]:**
  - Sort the p-values from lowest to highest as $P_{(1)}, P_{(2)}, \ldots, P_{(m)}$, and let $H_{(i)}$ denote the null hypothesis associated with the p-value $P_{(i)}$;
  - Let k denote the highest value such that $P_{(k)} \leq \frac{\delta}{m-k+1}$;
  - Reject the null hypotheses $H_{(1)}, \ldots, H_{(k)}$. If no value for k satisfies the criteria, no null hypothesis is rejected.

FDR control procedures: As our RADEFDR technique controls FDR under arbitrary dependence between the hypotheses, we compare it with the *Benjamini–Yekutieli procedure* [18] (BY), which generalizes the Benjamini-Hochberg procedure (BH) (which holds under independence or positive dependence between hypotheses) to arbitrary dependence.

- **Benjamini-Yekutieli FDR control procedure (BY) [18]:**
  - Sort the p-values from lowest to highest as $P_{(1)}, P_{(2)}, \ldots, P_{(m)}$, and let $H_{(i)}$ denote the null hypothesis associated with the p-value $P_{(i)}$;
  - Let k denote the highest value such that $P_{(k)} \leq \delta \frac{k}{mc(m)}$;
  - If the tests are independent or positively correlated (as in the Benjamini-Hochberg (BH) procedure [15]) $c(m) = 1$. Under arbitrary dependence $c(m) = \sum_{i=1}^{m} i^{-1}$;
  - Reject the null hypotheses $H_{(1)}, \ldots, H_{(k-1)}$. If $k = 1$, no null hypothesis is rejected.

We do not compare our methods with weighted versions of the mentioned procedures as we do not assume prior knowledge of the hypotheses being tested. Knowledge of the ratio of true nulls m_0/m may lead to improvement of the mentioned FWER (resp., FDR) control procedures. However, m_0/m is not generally known, and *adaptive* variations of the previously mentioned procedures for controlling FWER (resp., FDR), among whom [104] [75][93] [202] (resp., [16][23] [84][201][221]) only *approximately* control FWER, or strongly control FWER under additional assumptions such as the p-values of the true null hypotheses being iid as $U(0, 1)$, or positive dependence.

As our proposed method RADEFWER (resp., RADEFDR) does not require *any* assumption on the distribution or the dependence structure of the hypotheses, Holm's (resp., the Benjamini-Yakuteli) procedure is a more appropriate term of comparison. To gauge the effectiveness of our methods, we show however that, in opportune settings, RADEFWER (resp., RADEFDR) achieves improvements even with comparison to Hochberg's (resp., the Benjamini-Hochberg) procedure.

All mentioned methods hold under the critical assumption that the p-value of true null hypotheses are distributed uniformly in $[0, 1]$. However, in the finite sample setting, such guarantee can only be achieved asymptotically, thus, such procedures actually provide asymptotic control of the FWER for $|\bar{x}| \to \infty$. Such an assumption is not required by our methods.

We do not compare RADEFWER with *resampling* procedures for FWER and FDR control, such as *bootstrapping* (e.g., Romano and Wolf FWER control [195], and the FDR control of Romano et al. [194]) and *permutations* methods (e.g., Westfall and Young method [236]. These methods require additional assumptions such as *subset pivotality* [236] (which is itself stronger than the assumption on p-values being distributed as $U(0,1)$), or assumptions regarding the convergence of the estimate of the unknown distribution obtainable via bootstrap [195][194]. Further, such resampling approaches appear to be most suited for situations with small samples and a limited number of hypotheses, due to the conspicuous computational overhead which does not scale for massive amounts of hypotheses, and the complications in constructing null-invariant permutations [108]. Finally, we do not compare our procedure with Bayesian approaches as we assume no prior knowledge of the likelihood of hypotheses being true or false nulls.

2.3 Setting

Let $\bar{x}$ denote the input dataset with $|\bar{x}| = m$, we assume that $\mathcal{X}$ is composed by m i.i.d. samples from an unknown distribution $\mathcal{D}$. Assume each $x \in \mathcal{X}$ to be a tuple of $d+1$ values. For $i = 1, \ldots, d$, we denote as $x[i]$ the value of the i-th feature of x. We consider families of null hypotheses such that each null hypothesis corresponds to a statistical counting query (e.g.,"*predicate queries*"), characterized by a predicate binary function $f_h : \mathcal{X} \to 0, 1$:

$$f_h(x) = \mathbb{1}\left(predicate(x)\right).$$

As an example, consider the null hypothesis "h = there is no correlation between the sign of feature j and that of the feature k". The corresponding predicate function f_h would be defined as:

$$f_h(x) = \mathbb{1}\left(\texttt{sign}(x[j]) == \texttt{sign}(x[k])\right).$$

In the remainder of the presentation, we discuss how to use our approach as if the individual null hypotheses being considered are evaluated according to the *one-sample student t-test*. Our approach can, however, be extended to variations of this setting such as those for which the expected null-values are not given, for different types of hypotheses, and for two-sample testing. Regardless of the specific variation, our approaches do not require knowledge of the variance of the test statistic. With a slight abuse of notation we denote the average of the evaluations of f_h on the points in the sample dataset $\bar{x}$ as:

$$f_h(\bar{x}) = |\bar{x}|^{-1} \sum_{x_i \in \bar{x}} f_h(x_i)$$

Clearly, $\mathrm{E}_{\mathcal{D}}\left[f_h(x)\right] = \mathrm{E}_{\mathcal{D}^m}\left[f_h(\bar{x})\right]$. In the following, we refer to the family of functions corresponding to the family $\mathcal{H}$ of hypothesis as $\mathcal{F}_{\mathcal{H}} = f_h, \forall h \in \mathcal{H}$. Let

$$p_i = \mathrm{E}_{\mathcal{D}}\left[\mathbb{1}\left(\texttt{sign}\left(x[j]\right) == \text{“+”}\right)\right];$$

$$p_k = \mathrm{E}_{\mathcal{D}}\left[\mathbb{1}\left(\texttt{sign}\left(x[k]\right) == \text{“}-\text{”}\right)\right].$$

If, as stated in the null hypothesis h, there is no correlation between the signs of the i-th and the k-th features, we would have:

$$\mathrm{E}_{\mathcal{D}}\left[f_h(x)\right] = p_i p_j + (1-p_i)(1-p_j) = \mu_h,$$

where μ_h denotes the expected value of $f_h(x)$ according to the null hypothesis. In the classic one sample t-test the value μ_h would correspond to the population mean specified by the null hypothesis. Such values may either be specified by the analyst or, for opportune choices of the null hypothesis (e.g., null hypotheses such as h in the previous example), be automatically estimated from the data. In the following, in order to simplify the presentation, we assume the values μ_h to be given.

Rather than relying on the evaluation of the p-values, the core idea of our approach is to use estimates of the value $\mathrm{E}_{\mathcal{D}}\left[f_h(x)\right]$ obtained from $\bar{x}$ in order to make decisions regarding whether the associated hypothesis h should be rejected or not. In particular, given a uniform convergence bound holding *simultaneously* for all $f_h \in \mathcal{F}_{\mathcal{H}}$, we can achieve FWER control without any further correction for the cardinality of $\mathcal{H}$. Further, such a procedure holds for *any possible dependence* among the hypotheses in $\mathcal{H}$. Our approach proceeds by characterizing the distribution of the *generalization error* for functions in $\mathcal{F}_{\mathcal{H}}$:

$$\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right) = \sup_{f_h \in \mathcal{F}_{\mathcal{H}}} |f_h(\bar{x}) - \mathrm{E}_{\mathcal{D}}\left[f_h(x)\right]|. \tag{2.1}$$

The tighter the bound on the generalization error, the tighter the control of FWER. In the next section, we discuss how to obtain tight and data dependent bounds on $\Psi\left(\mathcal{F}_{\mathcal{H}}\right)$ based on Rademacher Complexity theory.

2.4 Uniform bound on generalization error with Rademacher Complexity

Statistical learning theory offers various tools to characterize the distribution of $\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right)$. In this work, we use generalization bounds based on the *Rademacher Complexity* of $\mathcal{F}_{\mathcal{H}}$, defined as follows.

Definition 2.0.3. [165] *Let $\bar{\sigma} = (\sigma_1, \ldots, \sigma_m)$ be a vector of m independent* Rademacher random variables*, such that for all i, $Pr(\sigma_i = 1) = Pr(\sigma_i = -1) = 1/2$. The Empirical Rademacher Complexity of a class of function $\mathcal{F}$ with respect to a sample $\bar{x} = \{x_1, \ldots, x_m\}$, with $\bar{x} \sim \mathcal{D}^m$ is*

$$\hat{R}_m\left(\mathcal{F}, \bar{x}\right) := E_{\bar{\sigma}}\left[\sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} f\left(x_i\right))\sigma_i\right].$$

The Rademacher Complexity of $\mathcal{F}$ for samples of size m is defined as

$$R_m\left(\mathcal{F}\right) := E_{\bar{x} \sim \mathcal{D}^m}\left[R_{\bar{x}}^{\mathcal{F}}\right].$$

The relation between $\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right)$ and the Rademacher Complexity of $\mathcal{F}_{\mathcal{H}}$ is given by the following results [1]:

Lemma 2.1. [208, Lemma 26.2] $E_{\bar{x} \sim \mathcal{D}^m}\left[\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right)\right] \leq 2R_m\left(\mathcal{F}_{\mathcal{H}}\right)$.

Lemma 2.2. [165, Theorem 14.21] *Assume that for all $x \in \mathcal{X}$ and $f \in \mathcal{F}$ we have $f(x) \in [0,1]$, then, for $\epsilon \in (0,1]$:*

$$Pr\left(\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right)) > 2R_m\left(\mathcal{F}_{\mathcal{H}}\right) + \epsilon\right) \leq e^{-2m\epsilon^2}. \tag{2.2}$$

[1] In our setting, in order apply the result with absolute value we implicitly assume that for any $f \in \mathcal{F}_{\mathcal{H}}$ we also have $-f \in \mathcal{F}_{\mathcal{H}}$, that is, we assume that $\mathcal{F}_{\mathcal{H}}$ is *closed under negation*.

The result in Lemma 2.2 allows us to characterize the distribution of generalization errors of *all* functions in $\mathcal{F}_{\mathcal{H}}$ as they are bounded by $\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right)$. This leads to confidence intervals holding *simultaneously* for all functions in $\mathcal{F}_{\mathcal{H}}$. Leveraging this observation, we obtain an FWER control procedure whose performance scales according to $R_{m,\mathcal{F}}$ rather than to the *sheer number* of hypotheses in $\mathcal{H}$.

Rademacher Complexity estimate

In order to actually use the Rademacher Complexity results for $\mathcal{F}_{\mathcal{H}}$, it is necessary to estimate $R_m\left(\mathcal{F}_{\mathcal{H}}\right)$ using the given sample $\bar{x} \sim \mathcal{D}^m$. Let $\bar{\sigma}_1, \ldots, \bar{\sigma}_\ell$ be ℓ independent vectors each composed by m independent Rademacher random variables (i.e., $\bar{\sigma}_j = \sigma_{j,1}\sigma_{j,2}\ldots\sigma_{j,m}$). Consider the estimate of $R_m^{\mathcal{F}_{\mathcal{H}}}$ computed as:

$$\hat{R}_{\bar{x},\ell}\left(\mathcal{F}_{\mathcal{H}}\right) = \frac{1}{\ell}\sum_{j=1}^{\ell}\sup_{f\in\mathcal{F}_{\mathcal{H}}}\frac{1}{m}\sum_{i=1}^{m} f(x_i)\sigma_{j,i}. \tag{2.3}$$

Clearly, $\mathrm{E}_{\bar{x},\bar{\sigma_1},\ldots,\bar{\sigma}_\ell}\left[\hat{R}_{\bar{x},\ell}\left(\mathcal{F}_{\mathcal{H}}\right)\right] = R_m\left(\mathcal{F}_{\mathcal{H}}\right) R^{\mathcal{F}_{\mathcal{H}}}$.

Such an estimate can be used to characterize the distribution of $\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right)$. In this work, we use the following bounds obtained using applications of, respectively, the Martingale Central Limit Theorem (MCLT) and Bernstein's Inequality for Martingales.

Lemma 2.3 (MCLT-based bound on $\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right)$)**.** *Let $\Phi(x)$ denote the cumulative distribution function for the standard normal distribution. Assume that for all $x \in \mathcal{X}$ and $f \in \mathcal{F}_{\mathcal{H}}$, we have $f(x) \in [0,1]$, then for $\epsilon \in (0,1]$,*

$$\lim_{m\to\infty} Pr\left(\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right) - 2\hat{R}_{\bar{x},\ell}\left(\mathcal{F}_{\mathcal{H}}\right) > \epsilon\frac{\sqrt{\ell + 4\sqrt{\ell} + 20}}{2\sqrt{\ell m}}\right) < 1 - \Phi\left(\epsilon\right), \tag{2.4}$$

where $\Phi(x)$ denotes the cumulative distribution function for the standard normal distribution.

Lemma 2.4 (BIM-based bound on $\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right)$)**.** *Assume that for all $x \in \mathcal{X}$ and $f \in \mathcal{F}_{\mathcal{H}}$, we have $f(x) \in [0,1]$, then for $\epsilon \in (0,1]$,*

$$Pr\left(\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right) > 2\hat{R}_{\bar{x},\ell}\left(\mathcal{F}_{\mathcal{H}}\right) + \epsilon\right) < e^{-\epsilon^2\left(\frac{\ell+4\sqrt{\ell}+20}{2m\ell} + \frac{4\epsilon}{3m}\right)^{-1}}. \tag{2.5}$$

While the result in Lemma 2.4 is a *finite sample bound*, in practical applications, it may be preferable to use the result based of applying central limit asymptotic bounds such as the one in Lemma 2.3 as it allows for a less restrictive criterion. Note that using *asymptotic statistical tools* is an established tenet of classical hypothesis testing. The standard assumption that p-values are uniformly distributed between $[0,1]$ holds only for *large datasets*, due to the *central limit theorem*.

2.5 Proofs of uniform convergence bounds

Given a sample $\bar{x} \sim \mathcal{D}^n$ and ℓ independent Rademacher vectors $\bar{\sigma}_1, \ldots, \bar{\sigma}_\ell$, each composed of n independent Rademacher random variables (i.e., $\bar{\sigma}_j = \sigma_{j,1}\sigma_{i,2}\ldots\sigma_{j,n}$), we estimate $R_n^{\mathcal{F}_\mathcal{H}}$ with

$$\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_\mathcal{H}} = \frac{1}{\ell}\sum_{j=1}^{\ell}\sup_{f\in\mathcal{F}_\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n} f(x_i)\sigma_{j,i}. \tag{2.6}$$

Clearly, $\mathrm{E}_{\bar{x},\bar{\sigma_1},\ldots,\bar{\sigma}_\ell}\left[\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_\mathcal{H}}\right] = R_n^{\mathcal{F}_\mathcal{H}}$. To bound the generalization error $\Psi\left(\mathcal{F}_\mathcal{H}, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_\mathcal{H}}$, we model the process as a *Doob supermartingale* ([165, Chapter 13.1]) as follows:

$$C_i = E[\Psi\left(\mathcal{F}_\mathcal{H}, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_\mathcal{H}} \mid Y_1, \ldots, Y_i] \text{ for } i = 0, \ldots, n(\ell+1),$$

where the $Y_1, \ldots, Y_{n(\ell+1)}$ are the random variables that determinate the value of the estimate $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_\mathcal{H}}$. The first n variables Y_i's correspond to the values of the sample $\bar{x}$, i.e. for $1 \le i \le n$, $Y_i = X_i$, and the remaining $n\ell$ Y_i's correspond to the Rademacher random variables, $Y_i = \sigma_{\lfloor i/n\rfloor, i-\lfloor i/n\rfloor}$. It is easy to verify that $C_0 = 0$, and $C_{n(\ell+1)} = R_n^{\mathcal{F}} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_\mathcal{H}}$.

Next, we define a *martingale difference sequence* $Z_i = C_i - C_{i-1}$ with respect to the martingale $C_0, C_1, \ldots C_{m(\ell+1)}$, and note that $\sum_{t=1}^{n(\ell+1)} Z_t = C_{n(\ell+1)} = R_n^{\mathcal{F}_\mathcal{H}} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_\mathcal{H}}$.

2.5.1 Uniform convergence bound based on the Martingale Central Limit Theorem

We adapt the following version of the MCLT [2]:

Theorem 2.5 (*Corollary 3.2,* [97])**.** *Let $Z_0, Z_1, \ldots$ be a difference martingale with bounded absolute increments. Assume that (1) $\sum_{i=1}^{n} Z_i^2 \xrightarrow{p} V^2$ for a finite $V > 0$, and (2) $E\left[\max_i Z_i^2\right] \le M < \infty$, then $\sum_{i=1}^{n} Z_i / \sqrt{\sum_{i=1}^{n} E\left[Z_i^2\right]}$ converges in distribution to $N(0,1)$.*

The proof of Lemma 2.3 relies on the careful analysis of $\mathrm{E}\left[Z_i^2\right]$ for the martingale difference sequence $Z_i = C_i - C_{i-1}$.

Lemma 2.3. Consider the Doob supermartingale:

$$C_i = \mathrm{E}\left[\Psi\left(\mathcal{F}, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}} | Y_1, \ldots Y_i\right] \text{ for } i = 0, \ldots, n(\ell+1).$$

Let us define the corresponding martingale difference sequence $Z_i = C_i - C_{i-1}$. For each $i \in \{1, \ldots, n(\ell+1)\}$, due to linearity of expectation, we have $Z_i = A_i - 2B_i$, where:

$$A_i = \mathrm{E}\left[\Psi\left(\mathcal{F}_\mathcal{H}, \bar{x}\right) | Y_1, \ldots Y_i\right] - \mathrm{E}\left[\Psi\left(\mathcal{F}_\mathcal{H}, \bar{x}\right) | Y_1, \ldots Y_{i-1}\right];$$

$$B_i = \mathrm{E}\left[\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_\mathcal{H}} | Y_1, \ldots Y_i\right] - \mathrm{E}\left[\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_\mathcal{H}} | Y_1, \ldots Y_{i-1}\right].$$

In order apply the Martingale Central Limit Theorem, we need an upper-bound $a \ge |Z_i|$ for $1 \le i \le n(\ell+1)$ and an upper-bound L, such that $L \ge \sum_{i=1}^{n(\ell+1)} \mathrm{E}\left[Z_i^2\right]$. Given our definition of Z_i, we have that for every i, $\mathrm{E}\left[Z_i\right] = \mathrm{E}\left[A_i\right] = \mathrm{E}\left[B_i\right] = 0$, and thus: $\mathrm{E}\left[Z_i^2\right] = \mathrm{Var}\left[Z_i\right] \le \mathrm{Var}\left[A_i\right] + 4\mathrm{Var}\left[B_i\right] + 4\mathrm{Cov}\left[A_i, B_i\right]$. From the properties of covariance, we have $|\mathrm{Cov}\left[A_i, B_i\right]| \le \sqrt{\mathrm{Var}\left[A_i\right]\mathrm{Var}\left[B_i\right]}$, and thus, $\mathrm{E}\left[Z_i^2\right] = \mathrm{Var}\left[Z_i\right] \le \mathrm{Var}\left[A_i\right] + 4\mathrm{Var}\left[B_i\right] + 4\sqrt{\mathrm{Var}\left[A_i\right]\mathrm{Var}\left[B_i\right]}$.

We consider the cases for $1 \le i \le n$ and $mni \le m(\ell+1)$ separately:

[2] Formally, the asymptotic is defined on a triangle array, where rows are samples of growing sizes. We also assume that all expectations are well-defined in the corresponding filtration.

- $1 \leq i \leq n$: In our setting $\forall x \in \mathcal{X}$ and $\forall f \in \mathcal{F}_{\mathcal{H}}$, $f(x) \in [0,1]$, changing the value of any of the n points in $\bar{x}$ can change $f(\bar{x})$ by at most $1/n$. Therefore, $|A_i| \leq 1/n$, and $A_i \in [\alpha, \beta]$ with $\beta - \alpha \leq 1/n$. From *Popoviciu's Inequality on variance* [184], we have that the variance of a random variable which takes values in $[\alpha, \beta]$ is bounded from above by $(\beta-\alpha)^2/4$. Hence, by applying Popoviciu's Inequality, we have: $\text{Var}[A_i] \leq 1/4n^2$. For $1 \leq j \leq \ell$, let us consider

$$B_i^{(j)} = \text{E}\left[\sup_{f \in \mathcal{F}_{\mathcal{H}}} \frac{1}{n} \sum_{i=1}^{n} f(x_i)\sigma_{j,i} | Y_1, \ldots Y_i\right] - \text{E}\left[\sup_{f \in \mathcal{F}_{\mathcal{H}}} \frac{1}{n} \sum_{i=1}^{n} f(x_i)\sigma_{j,i} | Y_1, \ldots Y_{i-1}\right].$$

  According to our definitions, we have

$$B_i = \frac{1}{\ell} \sum_{j=1}^{\ell} B_i^{(j)}.$$

  Since $\forall x \in \mathcal{X}$ and $\forall f \in \mathcal{F}_{\mathcal{H}}$, $f(x) \in [0,1]$, changing the value of any of the n points in $\bar{x}$ can change $\sup_{f \in \mathcal{F}_{\mathcal{H}}} \frac{1}{n} \sum_{i=1}^{n} f(x_i)$ by at most $1/n$, and thus we have $|B_i| \leq 1/n$, and $B_i^{(j)} \in [\alpha, \beta]$ with $\beta - \alpha \leq 1/n$. By applying Popoviciu's Inequality, we have that $\text{Var}\left[B_i^{(j)}\right] \leq 1/(4n^2)$.

  As we are considering the expectation over the unassigned values of the Rademacher random variables, and as we are averaging over the values obtained using ℓ independent and identically distributed vectors of Rademacher random variables, we can conclude that $|B_i| \leq \frac{1}{\ell} \sum_{j=1}^{\ell} |B_i^{(j)}| \leq 1/n$, and $\text{Var}[B_i] = \frac{1}{\ell^2} \sum_{j=1}^{\ell} \text{Var}\left[B_i^j\right] \leq 1/(4n^2\ell)$.

- $n < i \leq n(\ell+1)$: Changing the value of any of the Rademacher random variables does not change the value of $\Psi(\mathcal{F}_{\mathcal{H}}, \bar{x})$. Hence, $\text{Var}[A_i] = \text{E}\left[A_i^2\right] = 0$.

  Given fixed values for the random variables corresponding to the points in $\bar{x}$, changing the value of one Rademacher random variable can change the value of $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{\mathcal{H}}}$ by at most $2/\ell n$. Thus, $|B_i| \leq \frac{2}{\ell n}$, and $B_i \in [\alpha, \beta]$ with $\beta - \alpha \leq 2/\ell n$. By applying Popoviciu's Inequality, we have:

$$\text{Var}[Z_i] = 4\text{Var}[B_i] \leq \frac{4}{n^2\ell^2}.$$

We, therefore, have $|Z_i| \leq \frac{2}{n}$ for all $1 \leq i \leq n(\ell+1)$, and $\sum_{i=1}^{n(\ell+1)} \text{E}\left[Z_i^2\right] \leq \frac{\ell + 4\sqrt{\ell} + 20}{4n\ell}$. Applying the MCLT, we have that as n goes to infinity, $\sum_{i=1}^{n(\ell+1)} Z_i / \sqrt{\sum_{i=1}^{n(\ell+1)} \text{E}\left[Z_i^2\right]}$ converges in distribution to $N(0,1)$, and thus:

$$\lim_{n\to\infty} \Pr\left(\sum_{i=1}^{\ell(n+1)} Z_i \left(\sqrt{\sum_{i=1}^{n(\ell+1)} \text{E}\left[Z_i^2\right]}\right)^{-1} > \epsilon\right) < 1 - \Phi(\epsilon),$$

$$\lim_{n\to\infty} \Pr\left(\sum_{i=1}^{\ell(n+1)} Z_i > \epsilon \sqrt{\frac{\ell + 4\sqrt{\ell} + 20}{4n\ell}}\right) < 1 - \Phi(\epsilon),$$

By linearity of expectation, and by applying Theorem 2.1:

$$\begin{aligned}\sum_{i=1}^{\ell(n+1)} Z_i &= \Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right) - \mathrm{E}\left[\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right)\right] - 2\left(\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{\mathcal{H}}} - R_n^{\mathcal{F}_{\mathcal{H}}}\right) \\ &\geq \Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right) - 2R_n^{\mathcal{F}_{\mathcal{H}}} - 2\left(\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{\mathcal{H}}} - R_n^{\mathcal{F}_{\mathcal{H}}}\right); \\ &\geq \Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{\mathcal{H}}};\end{aligned}$$

The statement follows. □

Due to its asymptotic nature, it is not possible to compare directly the tightness of the bound in Theorem 2.3 with that of finite sample bounds such as the one in Theorem 2.4. Still, this bound is of great interest in many practical scenarios as it allows for a much tighter bound for the generalization error.

2.5.2 Application of Bernstein's Inequality for Martingales

We use the following version of Bernstein's Inequality for Martingales (BIM) due to Freedman [82], as presented in [63] and adapted to one-sided error.

Theorem 2.6. [63][82] *Let $Z_1, \ldots, Z_t$ be a martingale difference sequence with respect to a certain filtration $\{\mathscr{F}_i\}_{i=0,\ldots,t}$.*

Thus, $E\left[Z_i|\mathscr{F}_{i-1}\right] = 0$ for $i = 1, \ldots, t$. The process $\sum_{i=1}^{t} Z_i$ is thus a martingale with respect to this filtration. Further, assume that $|Z_i| \leq a$ for $i = 1, \ldots, t$, and that the conditional variance $\sum_{i=1}^{t} E\left[Z_i^2\right] \leq L$. For $\epsilon \in (0, 1)$, we have:

$$Pr\left(\sum_{i=1}^{t} Z_i > \epsilon\right) \leq e^{-\epsilon^2\left(2L + 2a\epsilon/3\right)^{-1}}. \tag{2.7}$$

Note that the bound presented here is slightly different from the one in [82] as for our purposes we only require a one-sided bound.

A careful analysis of $E[Z_i^2]$ in our application allows us to obtain a significantly stronger bound than the one obtained using McDiarmid's Inequality [13, 208], which depends on the maximum variation of the martingale.

Proof. of Lemma 2.4 Consider the Doob supermartingale:

$$C_i = \mathrm{E}\left[\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{\mathcal{H}}} | Y_1, \ldots Y_i\right] \quad \text{for } i = 0, \ldots, n(\ell+1),$$

where for $1 \leq i \leq m$, $Y_i = X_i$, and the remaining Y_i correspond to the $n\ell$ independent Rademacher random variables in the ℓ vectors; that is, $Y_{j(n)+i} = \sigma_{j,i}$ for $1 \leq j \leq \ell$ and $1 \leq i \leq n$. It is easy to verify that $C_{n(\ell+1)} = \Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{\mathcal{H}}}$. Further, $C_0 = \mathrm{E}\left[\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right)\right] - 2R_n^{\mathcal{F}_{\mathcal{H}}}$, and due to Theorem 2.1, $C_0 \leq 0$. Let us define the corresponding martingale difference sequence $Z_i = C_i - C_{i-1}$.

In order apply Bernstein's Inequality, we need an upper-bound $a \geq |Z_i|$ for $1 \leq i \leq n(\ell+1)$ and an upper-bound L, such that $L \geq \sum_{i=1}^{n(\ell+1)} \mathrm{E}\left[Z_i^2\right]$. Note that the sequence Z_i defined here corresponds to the martingale difference sequence by the same name that we studied in the proof of Theorem 2.3. Thus, we have $\sum_{i=1}^{n(\ell+1)} \mathrm{E}\left[Z_i^2\right] \leq \frac{\ell + 4\sqrt{\ell} + 20}{4n\ell}$, and $|Z_i| \leq 2/n$ for all $1 \leq i \leq n(\ell+1)$.

By linearity of expectation, and by applying Theorem 2.1:

$$\begin{aligned}\sum_{i=1}^{\ell(n+1)} Z_i &= \Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right) - \mathrm{E}\left[\Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right)\right] - 2\left(\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{\mathcal{H}}} - R_n^{\mathcal{F}_{\mathcal{H}}}\right) \\ &\geq \Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right) - 2R_n^{\mathcal{F}_{\mathcal{H}}} - 2\left(\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{\mathcal{H}}} - R_n^{\mathcal{F}_{\mathcal{H}}}\right); \\ &\geq \Psi\left(\mathcal{F}_{\mathcal{H}}, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{\mathcal{H}}};\end{aligned}$$

The statement follows by applying BIM (Theorem 3.3). □

This result can be used in RadeFWER or RadeFDR in place of the bound given by Lemma 2.3.

2.6 FWER control with Rademacher Complexity

ALGORITHM 1 RadeFWER

Input: $\bar{x}$ input sample; $\mathcal{H}$ class of tested null hypotheses; δ significance threshold;
ℓ number of Rademacher vectors used in Rademacher Complexity estimation.
Output: R set of rejected null hypotheses.

procedure RadeFWER($\bar{x}, \mathcal{H}, M_0, \delta, \ell$)
  for $j \in \{0, 1, \ldots, \ell\}$ **do** ▷ Initialization Rademacher Complexity Estimate
    $\sigma_j \leftarrow$ vector of m iid Rademacher RVs
  for each hypothesis $h_i \in \mathcal{H}$ **do** ▷ Main
    $f_{h_i} \leftarrow$ predicate function corresponding toh_i
    for $j \in \{1, \ldots, \ell\}$ **do** ▷ Rademacher Complexity estimation update
      $R_j^{\mathcal{F}_\mathcal{H}} \leftarrow \max\{R_j^{\mathcal{F}_\mathcal{H}}, \frac{1}{m}\sum_{k=1}^{|\bar{x}|} f_{h_i}(x_k)\sigma_{j,k}\}$
    $f_{h_i}(\bar{x}) \leftarrow \frac{1}{|\bar{x}|}\sum_{k=1}^{|\bar{x}|} f_{h_i}(x_k)$
  $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_\mathcal{H}} \leftarrow \frac{1}{\ell}\sum_{j=1}^{\ell} R_j^{\mathcal{F}_\mathcal{H}}$
  $\epsilon' \leftarrow \Phi^{-1}(1-\delta)\sqrt{\frac{\ell+4\sqrt{\ell}+20}{4\ell m}}$ ▷ Control with MCLT
  for $h_i \in \mathcal{H}$ **do**
    if $\left|f_{h_i}(\bar{x}) - \mu_{h_i}\right| - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_\mathcal{H}} > \epsilon'$ **then** ▷ Decision
      $R \leftarrow R \cup \{h_i\}$ ▷ Reject null h_i
  retun R

Our proposed RadeFWER procedure uses the uniform bound on the generalization error for functions in $\mathcal{F}_\mathcal{H}$ (i.e., Lemma 2.4 or Lemma 2.3) to evaluate the hypotheses in $\mathcal{H}$ while controlling FWER at level δ. Given $h \in \mathcal{H}$ and the corresponding predicate function $f_h \in \mathcal{F}_\mathcal{H}$, let μ_h be the expected value of f_h according to the null hypothesis h. If the distribution of $f_h(\bar{x})$ corresponds to the one hypothesized by h with expected value μ_h, we would have that the discrepancy between $f_h(\bar{x})$ and the assumed expected value μ_h (according to the null hypothesis) would be upperbounded by $\Psi(\mathcal{F}_\mathcal{H}, \bar{x})$, whose distribution can be characterized by means of Lemma 2.4 or Lemma 2.3. For each $h \in \mathcal{H}$, RadeFWER computes $f_h(\bar{x})$ and evaluates the deference from expectation according to the null hypothesis $|f_h(\bar{x}) - \mu_h|$. RadeFWER then determines the likelihood of the difference $|f_h(\bar{x}) - \mu_h|$ being observed just due to noise, as an indication of the fact that $\mathrm{E}\left[f_h(\bar{x})\right] \neq \mu_h$, and, hence, h is not a true null.

Given the desired FWER control level δ, and the estimate of the Rademacher Complexity of $\mathcal{F}_\mathcal{H}$ computed according to (2.3), RadeFWER computes the maximum value ϵ' such that $\Pr\left(\Psi(\mathcal{F}_\mathcal{H}, \bar{x})) > 2R_m(\mathcal{F}_\mathcal{H}) + \epsilon'\right) > \delta$. That is, ϵ' denotes the maximum value of $\Psi(\mathcal{F}_\mathcal{H}, \bar{x})) - 2R_m(\mathcal{F}_\mathcal{H})$ which can be observed with probability higher than δ. Hence, the probability of observing a difference between the empirically evaluated $f_h(\bar{x})$ and its expectation for *any* $f_h \in \mathcal{F}_\mathcal{H}$ is at most δ. The actual computation of ϵ' depends on the method used for characterizing the distribution of $\Psi(\mathcal{F}_\mathcal{H}, \bar{x}))$. When using the bound in Lemma 2.3 we have:

$$\epsilon' = \Phi^{-1}(1-\delta)\sqrt{\left(\ell + 4\sqrt{\ell} + 20\right)/(4\ell m)}. \tag{2.8}$$

When using the bound in Lemma 2.4 we have:

$$\epsilon' = \sqrt{\left(8\ell \ln\left(\delta^{-1}\right)\right)^2 + 4\left(6\ell m\right)\ln\left(\delta^{-1}\right)\left(3\ell + 12\sqrt{\ell} + 60\right)} \\ \frac{8\ln\left(\delta^{-1}\right)}{12\ell m}. \tag{2.9}$$

RadeFWER then uses the value ϵ' by checking for every function $f_h \in \mathcal{F}_{\mathcal{H}}$ whether $|f_h(\bar{x}) - \mu_h| > \epsilon' - 2R^{\mathcal{F}_{\mathcal{H}}}_{\bar{x},j}$. If this is the case, then the probability of the observed phenomenon given the null hypothesis is below the given FWER control threshold δ, and RadeFWER rejects the corresponding null hypothesis. Otherwise, the null hypothesis is accepted. This ensures that the probability of rejecting true null hypotheses is bounded by δ. As the uniform bound being used characterizes simultaneously the distribution of *all* $f_h \in \mathcal{F}_{\mathcal{H}}$, we can conclude that RadeFWER does indeed control FWER for $\mathcal{H}$. These arguments provide a proof of the correctness of RadeFWER, as stated in the following theorem.

Theorem 2.7 (Correctness of RadeFWER)**.** *Let $\mathcal{H}$ be a class of hypotheses which can be computed by using a class of statistical queries $\mathcal{F}_{\mathcal{H}}$. Given $\delta \in (0, 1]$, using* RadeFWER *to evaluate $\mathcal{H}$ ensures FWER control at level δ.*

2.7 FDR control with Rademacher Complexity

As in RadeFWER, for each null hypothesis $h \in \mathcal{H}$ RadeFDR evaluates the corresponding predicate function $f_h(\bar{x})$ and computes the absolute difference $|f_h(\bar{x}) - \mu_h|$. The procedure then proceeds in a number of phases during which null hypotheses from $\mathcal{H}$ are rejected with decreasing confidence. The procedure leverages the fact that a noticeable number of hypotheses may, in fact, be rejected with higher confidence in order to reject a larger number of hypotheses while controlling the FDR at level δ. The algorithm is characterized by: (a) a parameter $\eta > 1$ such that $\delta' = \delta/\eta$; (b) an ordered sequence $(\theta_1, \theta_2, \ldots, \theta_{i'})$ such that (i) $\theta_i \in [0, 1]$ for all $j = 1, \ldots, i'$; (ii) for all $j = 1, \ldots, i'$ we have $\theta_j \geq \theta_{j+1}$; and (iii) we have $\delta' \sum_{j=1}^{i^*} \theta_j \leq \delta$, where i^* denotes the index of the last possible phase. We can always assume such phase exists and that $i^* \lceil \delta/\eta \rceil$.

The number of non-zero *threshold* values θ_j determines the number of total phases. While any choice of the threshold values which satisfy the previous criteria is admissible, it is in general advisable to set the thresholds with low index to high values (i.e., close or equal to 1) and then chose progressively lower values. In particular, setting $\theta_i = 1$ for $i = 1, 2, \ldots, \eta$ for $\eta \in \mathbb{N}$, ensures that RadeFDR marks as discoveries all and only the hypotheses which would be marked as such by the RadeFWER. Hence, in order to achieve any possible improvement in terms of recall, it will be necessary to set some of the first η thresholds lower than 1. While in the worst case this may lead to a lower recall compared to the more restrictive RadeFWER criterion, it allows identifying as discoveries hypotheses that would not be detected by the RadeFWER procedure.

We present the pseudocode for RadeFDR in Algorithm 2.7. In the i-th phase, for $i = 1, 2, \ldots$, RadeFDR evaluates which of the hypothesis in $\mathcal{H}$ which were not marked as discoveries in the previous $i - 1$ phases (except for $i = 1$) can be rejected with FWER control at level $i\delta'$ while

ALGORITHM 2 RadeFDR - False Discovery Rate control with Rademacher Complexity control

Input: $\bar{x}$ input sample; H class of null hypotheses being tested; δ significance threshold; ℓ Rademacher vector used in Rademacher Complexity estimation; η, $(\theta_1, \theta_2, ..., \theta_i)$ algorithm parameters
Output: R set of rejected null hypotheses.

procedure RadeFDR($\bar{x}, \mathcal{H}, M_0, \delta, \ell, \eta$)
  $m \leftarrow |\bar{x}|$ ▷ Size of the input sample
  $fval \leftarrow []$ ▷ Empty array for hypotheses evaluation
  $D \leftarrow \emptyset$ ▷ Empty array for discoveries
  $d \leftarrow []$ ▷ Empty counters for discoveries per phase
  $\delta' \leftarrow \delta/\eta$ ▷ Confidence step used by the algorithm
  $\gamma \leftarrow 0$ ▷ Counter for algorithm FDR control
  for $j \in \{0, 1, \ldots, \ell\}$ **do** ▷ Initialization estimator for Rademacher Complexity
    $\sigma_j \leftarrow$ vector of m iid Rademacher RVs
    $R^{\mathcal{F}_\mathcal{H}}_{\bar{x},j} \leftarrow 0$
  for each hypothesis $h_i \in \mathcal{H}$ **do** ▷ Main execution body
    $f_{h_i} \leftarrow$ predicate function corresponding toh_i
    $\mu_{h_i} \leftarrow$ expected value of f_{h_i} according to h_i
    for $j \in \{0, 1, \ldots, \ell\}$ **do** ▷ Rademacher Complexity estimation update
      $R^{\mathcal{F}_\mathcal{H}}_{\bar{x},j} \leftarrow \max\{R^{\mathcal{F}_\mathcal{H}}_{\bar{x},j}, \frac{1}{m}\sum_{k=1}^{m} f_{h_i}(x_k)\sigma_{j,k}\}$
    $fval[i] \leftarrow \left|\frac{1}{m}\sum_{k=1}^{m} f_{h_i}(x_k) - \mu_{h_i}\right|$
  $\tilde{R}^{\mathcal{F}_\mathcal{H}}_{\bar{x},\ell} \leftarrow \frac{1}{\ell}\sum_{j=1}^{\ell} R^{\mathcal{F}_\mathcal{H}}_{\bar{x},j}$
  $i \leftarrow 0$
  $D' \leftarrow \emptyset$
  while $|\mathcal{H}| > 0$ **do**
    $i \leftarrow i + 1$
    $\alpha \leftarrow \Phi^{-1}\left(1 - i\delta'\right)\sqrt{\frac{\ell + 4\sqrt{\ell} + 20}{4\ell m}}$ ▷ Control with MCLT
    for $h_i \in \mathcal{H}$ **do**
      if $fval[i] - 2\tilde{R}^{\mathcal{F}_\mathcal{H}}_{\bar{x},\ell} > \max\{0, \alpha\}$ **then** ▷ Decision
        $D' \leftarrow D' \cup \{h_i\}$ ▷ Reject null h_i
        $\mathcal{H} \leftarrow \mathcal{H} \setminus \{h_i\}$ ▷ Accept null h_i
    $|d_i| \leftarrow |D'|$
    for $j = 1, 2, \ldots, i$ **do**
      if $\frac{\sum_{\ell=j}^{i} d[\ell]}{\sum_{\ell=1}^{i} d[\ell]} > \theta_j$ **then** ▷ Test FDR control
        retun D ▷ Interruption of the procedure
    $D \leftarrow D \cup D'$
  retun D

controlling FWER at level $\delta' = \delta\eta^{-1}$. The procedure used to determine such hypotheses follows the same steps as RadeFWER. Let D_i (resp., V_i) be the random variable whose value corresponds to the number of null (resp., true null) hypotheses rejected in i-th phase. By Theorem 2.7, we have $\Pr(V_i > 0) \leq i\delta'$.

For each execution of RadeFDR the values of the D_i's are observable as they depend uniquely on $\bar{x}$, the known null expected values μ_h associated with each null hypothesis, and the values $i\delta'$ (that is not, however, the case for the $V_i's$). Let d_i denote the observed values of a realization of D_i. Let i^* be the maximum value of i such that either $\sum_{\ell=1}^{i} d_\ell = 0$, or for all $j = 1, 2, \ldots, i$ we have $\sum_{\ell=j}^{i} d_\ell / \sum_{\ell=1}^{i} d_\ell \leq \theta_j$. RadeFDR returns as discoveries all the hypotheses marked as discoveries during the first i^* phases.

Theorem 2.8 (Correctness of RadeFDR)**.** *Let $\mathcal{H}$ be a class of hypotheses which can be computed by using a class of statistical queries $\mathcal{F}_{\mathcal{H}}$. Given $\delta \in (0,1]$, using* RadeFDR *to evaluate $\mathcal{H}$ ensures False Discovery Rate control at level δ.*

Proof. Let δ be the desired FDR control level. Let $\eta \geq 1$ denote the selected algorithm parameter, and let $\delta' = \delta/\eta$. The value δ' is used by RadeFDR to define its execution phases.

The proof proceeds by partitioning the outcomes of the experiment based on the values of:

$$\psi = \sup_{h_i \in \mathcal{H}_0} |\frac{1}{n} \sum_{x_i \in \bar{x}} f_{h_i}(x_i) - \mu_{h_i}|, \tag{2.10}$$

where $\frac{1}{n}\sum_{x_i \in \bar{x}} f_{h_i}(x_i)$ denotes the empirical average of the values of the function f_{h_i} of the function associated with the hypothesis h_i, and μ_{h_i} denotes the expected value of f_{h_i} according to the null hypothesis.

As, in general, it is not known which of the hypotheses in $\mathcal{H}$ belong in $\mathcal{H}_0$ rather than $\mathcal{H}_1$, we shall instead consider the quantity:

$$\psi' = \sup_{h_i \in \mathcal{H}} |\frac{1}{n} \sum_{x_i \in \bar{x}} f_{h_i}(x_i) - \mu_{h_i}|.$$

As $\mathcal{H}_0 \subseteq \mathcal{H}$, we clearly have $\psi' \geq \psi$, and thus $\Pr\left(\psi' > \epsilon\right) \geq \Pr\left(\psi > \epsilon\right)$, for any $\epsilon \in (0,1)$.

As described, RadeFDR proceeds in phases during which RadeFDR verifies whether it is possible to mark some of the hypotheses as *discoveries* with decreasing confidence (i.e., in the i-th phase the confidence level will be $i\delta'$). In particular, in the i-th phase the algorithm determines the minimum value ϵ_i such that

$$\Pr\left(\psi' \geq \epsilon_i + 2\tilde{R}^{\mathcal{F}_{\mathcal{H}}}_{\bar{x},\ell}\right) < i\delta', \tag{2.11}$$

where $\tilde{R}^{\mathcal{F}_{\mathcal{H}}}_{\bar{x},\ell}$ denotes the estimation of the Rademacher Complexity of the class of functions $\mathcal{F}_{\mathcal{H}}$ computed using the sample $\bar{x}$.

The algorithm then identifies the functions for which the difference between the measured empirical average $\frac{1}{n}\sum_{x_i \in \bar{x}} f_{h_i}(x_i)$ the expected value according to the null hypothesis is higher than ϵ_i. The corresponding hypotheses are then marked as discoveries, provided that they were not marked as such during previous phases.

Let D_i denote the random variable whose value corresponds to the number of hypotheses marked as discoveries during the i-th phase, and let V_i denote the random variable whose value corresponds of false positives rejected during the i-th phase. Clearly $V_i \leq D_i$. The total number of discoveries (resp., false positives) is therefore given by $D = \sum D_i$ (resp., $V = \sum V_i$).

The goal of the proof is to verify that RadeFDR guarantees:

$$\mathrm{E}\left[\frac{V}{D} | D > 0\right] \Pr\left(D > 0\right) \leq \delta.$$

Consider the events: $E_i :=$"*the first false discovery occurs in the i-th phase*" for $i = 1, \ldots, i^*$, where i^* denotes the maximum possible index for a phase. We can always assume that such value exists and that $i^* \leq \lceil \eta/\delta \rceil$. Clearly such events are disjoint. Our desired property can be rewritten as:

$$\mathrm{E}\left[\frac{V}{D}\right] = \sum_{i=1}^{i^*} \mathrm{E}\left[\frac{V}{D} | E_i\right] \Pr\left(E_i\right) \tag{2.12}$$

Let us now consider the events:

$$E'_i := \begin{cases} \psi - 2\tilde{R}^{\mathcal{F}_\mathcal{H}}_{\bar{x},\ell} > \epsilon_1 & for\ i = 1; \\ \psi > \epsilon_i - 2\tilde{R}^{\mathcal{F}_\mathcal{H}}_{\bar{x},\ell} \wedge \psi - 2\tilde{R}^{\mathcal{F}_\mathcal{H}}_{\bar{x},\ell} \leq \epsilon_{i-1} & for\ i > 1; \end{cases}$$

By the construction of RadeFDR, whenever E_i is verified so is E'_i but not vice versa. This is due to the fact that the algorithm may halt before reaching the i-th phase for a given value of i. That is, unless the algorithm observes an opportune amount of rejections in the initial, "*high confidence*" phases (i.e., the phases whose index i is such that we have $i\delta' \leq \delta$), it will not mark hypotheses as discoveries in the later, "*low confidence*" phases. Hence, we have $\Pr\left(E'_i\right) \geq \Pr\left(E_i\right)$ for $i = 1, \ldots, i^*$. Thus, we can bound the FDR control as:

$$\mathrm{E}\left[\frac{V}{D}\right] \leq \sum_{i=1}^{i^*} \mathrm{E}\left[\frac{V}{D}|E_i\right] \Pr\left(E'_i\right) \tag{2.13}$$

From the basic property according to which in any given phase the maximum number of false discoveries correspond to the number of total discoveries in the phase itself, we have:

$$\mathrm{E}\left[\frac{V}{D}|E_i\right] \leq \mathrm{E}\left[\frac{D - \sum_{j=1}^{i-1} D_j}{D}|E_i\right] \tag{2.14}$$

We can thus rewrite (2.13) as:

$$\mathrm{E}\left[\frac{V}{D}\right] \leq \sum_{i=1}^{i^*} \mathrm{E}\left[\frac{D - \sum_{j=1}^{i-1} D_j}{D}|E_i\right] \Pr\left(E'_i\right) \tag{2.15}$$

Let us now consider the each of the $\mathrm{E}\left[\frac{\sum_{\ell=j}^{i^*} D_\ell}{D}|E_i\right]$ terms of the summation. Given *any* assignment of the random variables $D_\ell = d_\ell$, by construction RadeFDR ensures that:

$$\sum_{\ell=j}^{i^*} d_\ell/d \leq \theta_j, \tag{2.16}$$

for all the values $j = 1, 2, \ldots, i^*$. As the RadeFDR halts at the first i'-th phase for which it is not possible to guarantee that the condition in (2.16) is verified. Hence we can bound the right-hand side of (2.15) as :

$$\mathrm{E}\left[V/D\right] \leq \sum_{i=1}^{i^*} \theta_i \Pr\left(E'_i\right) \tag{2.17}$$

From (2.11) and by definition of the events E'_i we have:

$$\Pr\left(E'_i\right) := \begin{cases} < \delta' & for\ i = 1; \\ i\delta' - \sum_{j=1}^{i-1} \Pr\left(E'_j\right) & for\ i > 1; \end{cases}$$

By construction, for $j = 1, 2, \ldots$ we have that $\theta_j \geq \theta_{j+1}$ As the terms which multiply the $\Pr\left(E'_i\right)$ increase as i increases , the right hand side of (2.15) is maximized by upper bounding the values $\Pr\left(E'_i\right)$ under the constraint that (by construction and by the disjointedness of the E'_i events)

$$\Pr\left(E'_i\right) < i\delta' - \sum_{j=1}^{i-1} \Pr\left(E'_j\right).$$

That is, (2.15) is maximized by upper bounding $\Pr\left(E'_i\right) \leq \delta'$ for all $i = 1, \ldots, i^*$.

We can therefore say:

$$\begin{aligned}
\mathrm{E}\left[\frac{V}{D}\right] &\leq \sum_{i=1}^{i^*} \mathrm{E}\left[\frac{D - \sum_{j=1}^{i-1} D_j}{D}\right] \Pr\left(E_i'\right) \\
&\leq \sum_{i=1}^{i^*} \mathrm{E}\left[\frac{D - \sum_{j=1}^{i-1} D_j}{D}\right] \delta' \\
&\leq \delta' \sum_{i=1}^{i^*} \theta_i \leq \delta
\end{aligned}$$

where the last passage from assumption regarding the choice of the threshold values θ. □

RadeFDR (resp., RadeFWER) controls FDR (resp., FWER) under *any dependence structure* among the test statistics associated with the hypotheses in $\mathcal{H}$. The choice of the threshold values θ can heavily influence the performance in terms of recall.

2.8 Experimental analysis

As RadeFWER (resp., RadeFDR) controls FWER (resp., FDR) strongly and for *any* dependence structure between the hypotheses, we compare its performance with that of the "*Holm-Bonferroni method*" [106] (HB) (resp., of the "*Benjamin-Yekutieli procedure*" with correction for arbitrary dependence [18] (BY)), which controls FWER (resp., FDR) under analogous assumptions. To further evaluate the power of our approach, we also compare RadeFWER (resp., RadeFDR) with the "*Höchberg step-up procedure*" [103] (HOC) (resp., the *Benjamini-Hochberg procedure* [15] (BH). While uniformly more powerful than their respective counterpart, these procedures assume non-negative dependence between the hypotheses being tested. We refer the reader to Section 2.2 for a brief presentation of these procedures.

In our comparison, we focus on the one sample student t-test setting, for which, as null hypothesis, it is assumed that the test statistic associated with the hypotheses is distributed according to the normal distribution with expected value μ_0, which we assume to be known. We do not, however, assume knowledge of the variance.

2.8.1 Experiments with synthetic data

The dataset used in this experiment is composed of n vectors, each composed of m features which can each take values in $\{-1, 1\}$. Each vector is furthermore associated with one additional $m+1$-th feature which corresponds to the *label* ℓ of the record. The values of all the features for all the points are assigned randomly in such a way that, while there is no actual correlation between the values of any of the features and the value of the label, there is in fact correlation (either positive or negative) between the values of the features. To introduce a "*signal*" (i.e., a correlation between the value of the label and the value of certain features), we select $m - m_0$ out of the m features, and we modify the entries of the dataset corresponding to the selected features: Let $c = 1$ (resp.,

$c = -1$) for positive (resp., negative) correlation. For all the n records, the values of each entry corresponding to the $m - m_0$ correlated features is assigned a new value v_i such that:

$$\Pr(v_i = c\ell) = 0.5 + \texttt{bias}; \qquad \Pr(v_i = -c\ell) = 0.5 - \texttt{bias};$$

where ℓ denotes the value of the label of the record, and $\texttt{bias} \in (0, 0.5]$ denotes the strength of the correlation between the feature and the label. This setting allows us to compare the *statistical power* (resp., verify the correctness) of the testing procedures measured in terms of "*recall*" (resp., "*precision*"), that is, as the fraction of true discoveries that are recognized as such by the testing procedure (resp., the fraction of true positives among the rejected hypotheses). For the remainder, we refer to the versions of RadeFWER using the MCLT (resp., BIM) bound in Lemma 2.3 (resp., Lemma 2.4) as RadeFWER-MCLT (resp., RadeFWER-BIM). We use analogous notation for RadeFDR.

Correctness of RadeFWER and RadeFDR

We verify the correctness of RadeFWER (resp., RadeFDR) by evaluating over 50 runs using independently generated random datasets the fraction of executions for which at least one true null hypothesis is rejected (resp., the average ratio of false positives over number of rejections per each run). The used synthetic datasets are composed of 3000 points with $m = 10^i$ for $i = 4, \ldots, 7$; $m - m_0 = 500$; and $\texttt{bias} = 0.05$.

RadeFWER appears to effectively control FWER as in one of the runs (using $\delta = 0.2$) we have a false rejection when using RadeFWER-MCLT for $m = 10^4$. We have no false positives when using RadeFWER-BIM.

Similarly, RadeFDR appears to effectively control FDR as the average empirical FDR is consistently lower than the prescribed control level. In particular, using RadeFWER-MCLT we have false positives only when using $\delta = 0.2$ for $m = 10^4$ with empirical FDR = 0.001. We have no false positives when using RadeFWER-BIM. This suggests that despite the asymptotic nature of the bound in Lemma 2.3, the Rademacher Complexity based analysis of the generalization error gives a very reliable, albeit conservative, criteria.

Statistical power of RadeFWER and RadeFDR

We evaluate the statistical power of the procedures by comparing their respective recall, that is the fraction of the true-alternative hypotheses (i.e., the false nulls) marked as discoveries. First, we compare the performance of RadeFWER-MCLT (resp., RadeFDR-MCLT) with the classical statistical testing procedure, as they share asymptotic nature and ensure *asymptotic FWER (resp., FDR) control*. We then evaluate the performance RadeFWER-BIM (resp., RadeFDRBIM), which provide finite-sample guarantees.

Comparison of FWER control procedures: When comparing the performance of FWER controlling procedures, we average the recall observed over 50 independent runs on synthetic data each weighted as 1 if no false positive was detected, or 0 otherwise. In Figures 2.1(a) and 2.1(b), we present a comparison of the performance of HB, HOC and RadeFWER when considering an increasing

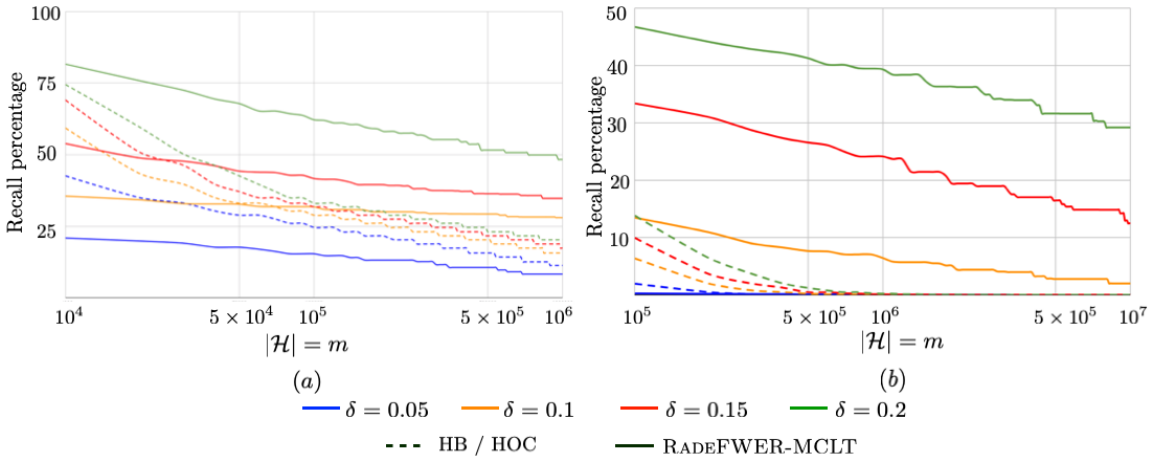


Figure 2.1: Recall comparison between RadeFWER, HB and HOC for different values of FWER control level δ for synthetic data. In setting (a): $|\bar{x}| = 2000$, $m - m_0 = 1000$, `bias` $= 0.05$. In setting (b): $|\bar{x}| = 3000$,$m - m_0 = 1000$, `bias` $= 0.04$.

number of null hypotheses in $\mathcal{H}$. As in these results, the performance of HB and HOC are almost equivalent for $|\mathcal{H}| > 2 \times 10^4$, we report only the performance of HOC in order to allow for better graphical representation.

We observe that as the number of hypotheses being tested increases the recall of the Holm and Hóchberg procedure decreases consistently, albeit with some fluctuations due to the randomness in the dataset generation. In contrast, the recall of our RadeFWER-MCLT methods decreases at a much slower pace, as it depends on how the Rademacher complexity of the associated $\mathcal{F}_{\mathcal{H}}$ increases rather than the sheer number of hypotheses in $\mathcal{H}$. In particular, the *asymptotic* bound provided by RadeFWER-MCLT exhibits improved performances over the classic FWER control procedures in most both settings for most of the desired control levels δ. The difference between the performance of the classical methods and RadeFWER is stronger especially when there is a high correlation between the behaviors of the features being considered. The higher such correlation, the lower the Rademacher Complexity of $\mathcal{F}_{\mathcal{H}}$, and, hence, the tighter the approximation achieved by our algorithm. While all approaches exhibit an improvement in accuracy given the increase of the strength of the bias, our approach based on the MCLT application seems to achieve stronger improvements.

Comparison of FDR control procedures: When comparing the performance of FWER controlling procedures, we average the recall F_1 score observed over 50 independent runs on synthetic data. The F_1 score is a measure of performance which incorporates the *precision* and the *recall* of the procedure as:$F_1 = 2(Precision \times Recall)/(Precision + Recall)$. In our setting, we have $Precision = 1 - F\hat{D}R$, where $F\hat{D}R$ denotes the observed empirical FDR. In Figure 2.2(a) (resp., Figure 2.2(b)), we present a comparison of the performance of BY, BH and RadeFDR, when considering an increasing number of null hypotheses in $\mathcal{H}$. We observe that while the less conservative FDR guarantee allows in general for greater recall, the performance of BY and BH steadily decrease as the number of tested

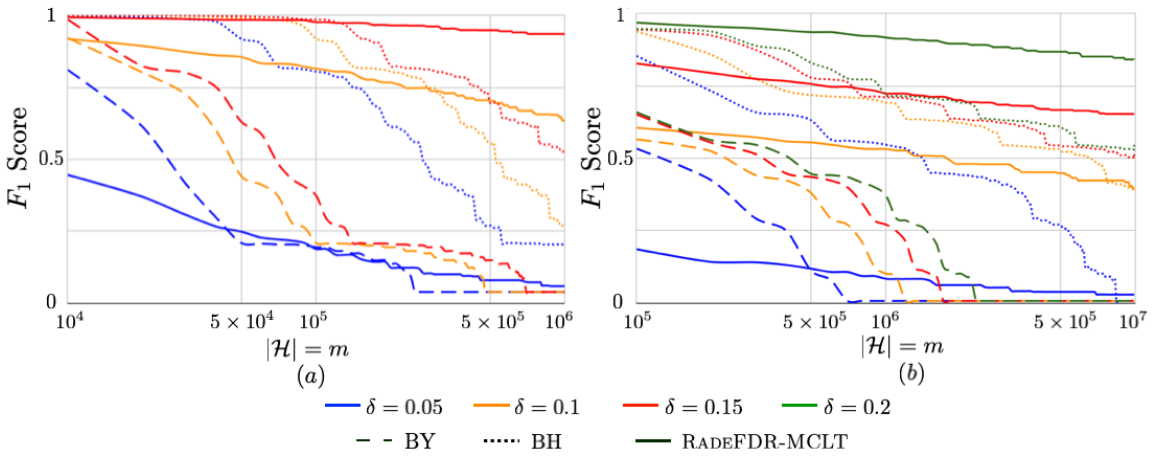


Figure 2.2: F_1 **score comparison between RadeFWER, HB and HOC for multiple values of FDR control level** δ **on synthetic data. In setting** (a)**:** $|\bar{x}| = 2000$, $m - m_0 = 100$, $\texttt{bias} = 0.05$. **In setting** (b)**:** $|\bar{x}| = 3000$, $m - m_0 = 500$, $\texttt{bias} = 0.04$.

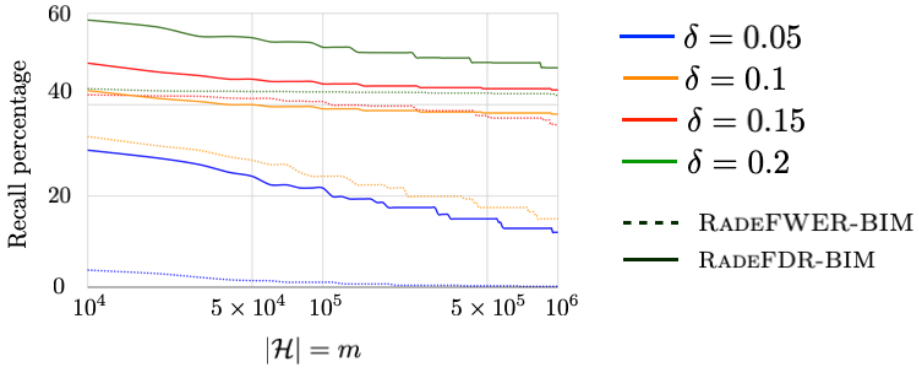


Figure 2.3: Average percentage recall achieved by RadeFWER--BIM and RadeFDR--BIM on synthetic data: $|\bar{x}| = 4000$, $m - m_0 = 1000$, $\texttt{bias} = 0.06$.

null hypotheses increases. RadeFDR performance decreases at a much slower pace, especially for high values of δ. The comparison with the BY procedure, our most correct term of comparison, is particularly noticeable as RadeFDR outperforms BY in virtually all the considered settings for $|\mathcal{H}|$ large enough.

Finite sample FWER/FDR control with RadeFWER--BIM *and* RadeFDR--BIM*:* We evaluate the performance, in terms of recall[3] for our finite-sample procedures RadeFWER--BIM and RadeFDR--BIM. We report in Figure 2.3 the average performance over 50 independent runs on synthetic data.

While compared with the asymptotic methods previously evaluated, in the considered settings our BIM approaches require an input sample of higher size and are able to detect only strong signals,

[3] As in the experiments RadeFDR--BIM does not reject any false positive, we have that its F_1 measure and recall are identical.

Table 2.1: FWER control hypothesis testing on BreastCancer dataset: number of discoveries for considered procedures.

	Order 1 hyp: $m = 60$				Order 2 hyp.: $m = 174000$				Order 3 hyp.: $m = 3248000$			
δ	HB	HOC	BER	MCLT	HB	HOC	BER	MCLT	HB	HOC	BER	MCLT
0.025	33	33	4	8	29	29	3	12	3	3	1	7
0.05	37	37	5	10	29	29	5	19	3	3	1	15
0.075	41	41	5	11	32	32	5	27	3	3	1	19
0.1	42	42	5	12	32	32	5	29	3	3	1	23
0.15	45	45	6	14	34	34	6	35	4	4	3	29
0.2	48	48	6	19	35	35	7	40	4	4	3	39

it is important to remark that the methods are not directly comparable as the latter ensures actual *finite sample* FWER and FDR control. The experiment confirms that, just as the MCLT version, the statistical power of our procedures decreases gradually as the complexity of the considered hypotheses grows, and not as a consequence of the sheer size of $\mathcal{H}$.

2.8.2 Experiments with real data

We evaluate the performance of our methods compares to classical procedures while studying the correlation between features and labels on the "*Breast Cancer Wisconsin (Diagnostic) Data Set*" dataset [222] (obtained from [57]) with $n = 569$ records. Each record is composed of $d = 30$ real-valued features (plus the identifier), which correspond to measured values for gene expressions, and a binary label which takes value "1" patient associated with the record suffers from breast cancer, or "0" otherwise.

We construct our queries by partitioning the range of the values of each feature in 20 intervals, and we then evaluate the relation between the fact that the value of a feature is within an interval and whether the record corresponds to a parent affected by breast cancer or not. We then construct more complex hypotheses by considering multiple of such "*predicate conditions*" (over different features) and by composing them using the "`and`" logical operator (any other operator could be used here). We refer to the queries obtained by composing i such predicate conditions as "*order i hypotheses*".

We compare the methods in terms of the number of hypotheses being returned as discoveries.[4] As in the previous section, we evaluate the hypotheses by standard t-tests for the classical procedures and with the corresponding method in Algorithm 1 (resp., Algorithm 2) for FWER (resp., FDR) control. The null-expected values used in the experiment are computed based on the marginal frequencies of the labels and of the records satisfying the predicate condition associated with the hypothesis being tested.

We report in Table 2.1 (resp., Table 2.2) the results for the comparison for FWER (resp., FDR) controlling procedures for different values of the desired control level δ. As in the synthetic experiments, we see that as the number of hypotheses being considered increases the classical statistic

[4] We cannot directly compare precision and recall as the information regarding which of the hypotheses being tested is, in fact, a *true alternative* is not available.

Table 2.2: FDR control hypothesis testing on BreastCancer dataset: number of discoveries for considered procedures.

	Order 1 hyp: $m = 60$				Order 2 hyp.: $m = 174000$				Order 3 hyp.: $m = 3248000$			
δ	BH	BY	BER	MCLT	BH	BY	BER	MCLT	BH	BY	BER	MCLT
0.025	65	45	4	9	47	31	4	15	4	3	0	9
0.05	88	49	5	11	49	33	5	25	5	3	1	17
0.075	94	53	5	12	54	35	5	29	6	3	1	22
0.1	107	53	5	13	58	35	6	33	6	3	1	28
0.15	126	59	6	19	63	39	8	41	8	3	2	40
0.2	134	67	7	20	67	41	9	49	9	3	4	54

approaches become extremely conservative, and unable to identify discoveries. This is particularly evident when considering order 3 queries: while classical procedures are virtually unable to retain any recall, our procedures are still able to detect some hypothesis as discovery. Further, RadeFWER and RadeFDR exhibit a greater relative increase in the number of identified discoveries when increasing the threshold δ of desired FWER or FDR control compared to the alternative testing procedures. This is of particular interest for this setting, given the small sample size and the high number of hypotheses being considered.

2.9 Conclusion

We introduced RadeFWER (resp., RadeFDR), a statistical procedure for multiple hypotheses testing controlling the Family Wise Error Rate (resp., the False Discovery Rate) which builds on Rademacher Complexity uniform convergence bounds. Our experimental analysis highlights the benefit of our approaches over classical procedures, especially when a very high number of hypotheses is being tested. Due to the generality of the approach, it appears amenable to extensions to other types of statistical tests.

Chapter 3

A Rademacher Complexity Based Method for Controlling Power and Confidence Level in Adaptive Statistical Analysis [1]

While standard statistical inference techniques and machine learning generalization bounds assume that tests are run on data selected independently of the hypotheses, practical data analysis and machine learning are usually iterative and adaptive processes where the same holdout data is often used for testing a sequence of hypotheses (or models), which may each depend on the outcome of the previous tests on the same data. In this Chapter, we present RADABOUND a rigorous, efficient and practical procedure for controlling the generalization error when using a holdout sample for multiple adaptive testing. Our solution is based on a new application of the Rademacher Complexity generalization bounds, adapted to dependent tests. We demonstrate the statistical power and practicality of our method through extensive simulations and comparisons to alternative approaches.

3.1 Introduction

The goal of data analysis and statistical learning is to model a stochastic process, or distribution, that explain an observed data. A major risk in statistical learning is *overfitting*, that is, learning a model that fits well with the observed data but does not predict new data. The standard practice in machine learning is to split the data into *training* and *holdout* (or testing) sets. A learning algorithm then learns a model using the training data and tests the model on the holdout set to

[1] The results presented in this chapter Were published in the proceedings of the 6th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2019). This is joint work with Professor Eli Upfal.

obtain a confidence interval for the expected error or for the value of the loss function of the model. If the process halts after a single iteration, then the statistical analysis is easy. However, in most cases, the learning process is iterative and adaptive. One uses successive tests for model selection, feature selection, parameter tuning, etc., and the choice of the tests themselves often depends on the outcomes of previous tests.

Ideally, each hypothesis should be tested on a fresh data sample. However, it is common practice to reuse the same holdout data to evaluate a sequence of hypotheses. While widespread, this practice is known to lead to *overfitting*; that is, the learned model becomes representative of the sample rather than the actual process. Evaluating the accumulated error in testing a sequence of related hypothesis on the same data set is a major challenge in both machine learning and modern statistics. In machine learning, the problem of "*preventing overfit*", is usually phrased and analyzed in terms of bounding the generalization error [208]. In inference statistics, the goal is controlling the Family Wise Error Rate (FWER), or the False Discovery Rate (FDR) of a sequence of hypothesis tests [15].

Our Results: We develop and analyze RadaBound, a rigorous, efficient, and practical procedure for online evaluation of the accumulated generalization error in a sequence of statistical inferences applied to the same sample. RadaBound can evaluate fully adaptive sequences of tests. The choice of a test may depend on the information obtained from previous tests, and the total number of tests is not fixed in advance.

One way to quantify the risk of overfitting after k queries is by considering the probability of the condition defined by the results of the first k queries. If the probability of such condition is close to 1, the results of the queries evaluated so far do not significantly restrict the sample space, and there is, therefore, no risk of overfitting. Viceversa, if the probability of the observed condition is small, the sample space defined by the true distribution conditioned on the results of the queries is noticeably different from the true distribution, and there is thus a significant risk of overfitting. In general, it is hard to bound the probability of the observed condition as the true distribution over the samples is unknown. However, in the special case for which the queries being considered correspond to evaluating the average of functions (such as evaluating the average risk or loss functions of alternative learning procedures), we can design an adaptive process based on an empirical estimate of the Rademacher Complexity of the set of queries which correctly bounds this probability and correspondingly halts the procedure when the risk of overfitting exceeds a certain threshold fixed by the user.

Our method builds on the concept of Rademacher Complexity [12, 133] that has emerged as a powerful alternative to VC-dimension and related uniform convergence methods for characterizing generalization error and sample complexity. A fundamental advantage of the Rademacher Complexity approach in contrast to standard uniform convergence tools, such as VC-dimension, that capture the complexity with respect the worst case input distribution, is that it yields a data-dependent bound as it is computed with respect to the input (sample) distribution, and can be efficiently approximated from the sample.

Our solution employs three major components: (1) For a set of functions chosen independent of the sample, the Rademacher Complexity [12, 133] provides a powerful and efficient bound on the error in estimating the expectations of all these function using one sample; (2) As long as the outcome of the sequence of tests does not significantly overfit to the sample, conditioning on these outcomes has only a minor effect on the distribution; and (3) The Rademacher Complexity of a sequence of tests can be estimated efficiently form a given sample, requiring similar computation time as running the actual tests.

To fully utilize our technique, we need computationally efficient methods for rigorously estimating the Rademacher Complexity from a sample. We introduce two novel methods based on Bernstein's inequality for martingales [82] and the Martingale Central Limit Theorem [97]. Our analysis and extensive experiments prove and demonstrate that our method guarantees statistical validity while retaining statistical power and practical efficiency.

Related Work: Classic statistics offers a variety of procedures for controlling the Family Wise Error Rate (FWER), ranging from the simple Bonferroni [28] to Holm's step-down [106] and Hochberg's step-up procedures [103] in the context of multiple hypotheses testing. While controlling the FWER under weak assumptions about the hypotheses, these methods are too conservative, giving many false negative results, in particular for large sets of hypotheses. Less conservative procedures, such as Benjamini and Hochberg [15], which control the False Discovery Rate (FDR) (i.e., the expected fraction of false discoveries), still do not scale up well for a very large number of hypotheses. However, all these procedures cannot be applied in the adaptive setting, as they require for the set of hypotheses to be fixed at the beginning of the testing procedure (i.e., before any data evaluation).

A series of recent papers [58, 59, 60] explored an interesting relation between "*Differential Privacy*" [62] and overfit prevention in adaptive analysis. The basic idea is to limit the user access to the holdout data so that the answers to the sequence of queries is differentially private. A differentially private access to the holdout data limits the risk of overfitting to that data set. Unfortunately, the practical application of this elegant mathematics is limited. Differential privacy is achieved through random perturbation of the data (or the reply to the queries). The higher the number of adaptive queries, the larger the required perturbation. However, the amount of perturbation is limited by the need to preserve the actual signal in the data. As a result, rigorous application of this approach is either limited to a small number of queries or is computationally intractable [60], making it less useful than alternative methods [100, 218]. Our experiments in Section 5 show that RadaBound allows orders of magnitude reduction of the required holdout dataset compared to Dwork et al.'s method [58] while offering the same guarantees. Further, our technique is much simpler as it does not require any introduction of additional noise. We discuss in detail the advantages of our solution compared to [58] in Section 3.7. A more practical solution for a restricted setting inspired by machine learning competitions was presented in [24]. Their solution, "*the Ladder*", provides a loss estimate only for those that made a significant improvement over the previous best. This restricted setting allows to sidestep the hardness results discussed in [100, 218].

3.2 The RadaBound

The process: For concreteness, we focus on the following setup. We have an holdout sample composed by m independent observations $\bar{x} = (x_1, \ldots, x_m)$, each from a distribution $\mathcal{D}$, and parameters $\epsilon, \delta \in (0, 1)$ fixed by the user. In an iterative process, at each step, the user (or an adaptive algorithm) submits a function f and receives an estimate $\tilde{\mathrm{E}}_{\bar{x}}[f] = \frac{1}{m}\sum_{i=1}^{m} f(x_i)$ of the "*ground truth value*" $\mathrm{E}_{\mathcal{D}}[f]$. The user has no *direct access* to the sample $\bar{x}$. That is, he can *only* acquire information regarding $\bar{x}$ from the confidence intervals $\tilde{\mathrm{E}}_{\bar{x}}[f] \pm \epsilon$ for the expectation $\mathrm{E}_{\mathcal{D}}[f]$, which the testing procedure has returned as answer to the queries considered so far. Let $\mathcal{F}_k = \{f_1, \ldots, f_k\}$ denote the set of the first k functions evaluated during the adaptive process. The maximum error in estimating the expectations of the k functions is given by:

$$\begin{aligned}\Psi(\mathcal{F}_k, \bar{x}) &= \sup_{f \in \mathcal{F}_k} |\frac{1}{m}\sum_{i=1}^{m} f(x_i) - \mathrm{E}_{\mathcal{D}}[f]| \\ &= \sup_{f \in \mathcal{F}_k} |\tilde{\mathrm{E}}_{\bar{x}}[f] - \mathrm{E}_{\mathcal{D}}[f]|.\end{aligned}$$

In this work, we refer as "*overfittig*" as follows: A given set of functions $\mathcal{F}_k$ is said to *overfit the sample* $\bar{x}$ if for any $f \in \mathcal{F}$ the value $\tilde{\mathrm{E}}_{\bar{x}}[f]$ evaluated on $\bar{x}$ differs from the *true value* $\mathrm{E}_{\mathcal{D}}[f]$ by more than the user given threshold ϵ. Our adaptive testing process halts at the first k-th step for which for which it cannot guarantee that the probability of overfitting is at most δ, that is, when $\Pr\left(\Psi(\mathcal{F}_k, \bar{x}) \leq \epsilon\right) \geq 1 - \delta$.

The process is fully adaptive. The choice of the function f_{k+1} evaluated at the $k+1$-th step may depend on the information obtained during the first k steps. We make no assumptions on the processes according to which the functions are adaptively chosen to be tested, nor do we require the total number of tests to be fixed in advance. For simplicity, we assume that all functions are in the range $[0, 1]$. More general settings are discussed later in the Chapter.

Bounding the generalization error for the iterative process: The sequence of answers to the queries, $\tilde{\mathrm{E}}_{\bar{x}}[f_1] \pm \epsilon, \tilde{\mathrm{E}}_{\bar{x}}[f_2] \pm \epsilon, \ldots, \tilde{\mathrm{E}}_{\bar{x}}[f_k] \pm \epsilon$ defines a filtration $\mathcal{L} = \{\mathcal{D}_k\}_{k \geq 0}$, such that

$$\mathcal{D}_0 = \mathcal{D} \quad \text{and} \quad \mathcal{D}_k = \{\mathcal{D} \mid \tilde{\mathrm{E}}_{\bar{x}}[f_1] \pm \epsilon \wedge \cdots \wedge \tilde{\mathrm{E}}_{\bar{x}}[f_k] \pm \epsilon\}.$$

The k-th query is chosen with respect to, and is answered in the filtered distribution $\mathcal{D}_{k-1}$.

Let E_k denote the event that the answer to the k-th query was within ϵ of the correct value, that is, $E_k := |\tilde{\mathrm{E}}_{\bar{x}}[f] - \mathrm{E}_{\mathcal{D}}[f_k]| \leq \epsilon$. Then, $\Pr\left(\Psi(\mathcal{F}_k, \bar{x}) \leq \epsilon\right) = \Pr\left(\wedge_{i=1}^{k} E_k\right)$, and thus, in the filtration process,

$$\begin{aligned}&\Pr_{\mathcal{L}}\left(\Psi(\mathcal{F}_k, \bar{x}) > \epsilon\right) \\ &\quad \leq \Pr_{\mathcal{D}_0}\left(\bar{E}_1\right) + \Pr_{\mathcal{D}_1}\left(\bar{E}_2 \mid E_1\right) + \ldots \\ &\quad + \Pr_{\mathcal{D}_{k-1}}\left(\bar{E}_k \mid \wedge_{j=1}^{k-1} E_j\right)\end{aligned}$$

$$\begin{aligned} &\leq \sum_{i=1}^{k} \frac{\Pr\left(\bar{E}_i\right)}{\Pr\left(\wedge_{j=1}^{i-1} E_j\right)} \leq \frac{1-\Pr\left(\wedge_{j=1}^{k} E_j\right)}{\Pr\left(\wedge_{j=1}^{k-1} E_j\right)} \\ &= \frac{1-\Pr\left(\Psi\left(\mathcal{F}_k, \bar{x}\right) \leq \epsilon\right)}{\Pr\left(\Psi(\mathcal{F}_{k-1}, \bar{x}) \leq \epsilon\right)}, \end{aligned} \tag{3.1}$$

where $\Pr()$ with no subscript refers to probability in the un-filtered distribution $\mathcal{D}$. The fact that the distribution of the generalization error in the adaptive case, $\Pr_{\mathcal{L}}\left(\Psi\left(\mathcal{F}_k, \bar{x}\right) > \epsilon\right)$, is related to the probability of an error in the non-adaptive case, $\Pr\left(\Psi\left(\mathcal{F}_k, \bar{x}\right) > \epsilon\right)$, is not surprising. In order to fit the sample differently than the original distribution $\mathcal{D}$, the process needs to detect a pattern whose frequency is considerably different in the sample compared to the actual distribution $\mathcal{D}$. However, the first query that observes such a pattern is chosen when the process has not yet observed a significant difference between the sample and the distribution. This is due to the fact that the process halts as soon as such difference is detected. Thus, the probability of overfitting in k queries is related the to the probability that the sample gives a bad estimate for the correct value of one of the k queries in the non-adaptive case. The challenge is to compute a tight bound to $\Pr\left(\Psi\left(\mathcal{F}_k, \bar{x}\right) \leq \epsilon\right)$. We achieve this through two novel bounds on estimating the Rademacher Complexity of $\mathcal{F}_k$.

3.3 Bounding $\Psi\left(\mathcal{F}_k, \bar{x}\right)$ using Rademacher Complexity

Definition 3.0.1. *[165] Let $\bar{\sigma} = (\sigma_1, \ldots, \sigma_m)$ be a vector of m independent* Rademacher random variables*, such that for all i, $Pr(\sigma_i = 1) = Pr(\sigma_i = -1) = 1/2$. The* Empirical Rademacher Complexity *of a class of function $\mathcal{F}$ with respect to a sample $\bar{x} = \{x_1, \ldots, x_m\}$, with $\bar{x} \sim \mathcal{D}^m$ is*

$$R_{\bar{x}}^{\mathcal{F}} = E_{\bar{\sigma}}\left[\sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} f(x_i)\sigma_i\right]$$

The Rademacher Complexity *of $\mathcal{F}$ for samples of size m is defined as $R_m^{\mathcal{F}} := E_{\bar{x} \sim \mathcal{D}^m}\left[R_{\bar{x}}^{\mathcal{F}}\right]$.*

The relation between $\Psi(\mathcal{F}_k, \bar{x})$ and the Rademacher Complexity of $\mathcal{F}_k$ is given by the following results [2]:

Lemma 3.1. *[208, Lemma 26.2]*

$$\begin{aligned} E_{\bar{x} \sim \mathcal{D}^m}\left[\Psi(\mathcal{F}_k, \bar{x})\right] &= E\left[\sup_{f \in \mathcal{F}_k} |\frac{1}{m} \sum_{i=1}^{m} f(x_i) - E_{\mathcal{D}}[f]|\right] \\ &\leq 2R_m^{\mathcal{F}_k}. \end{aligned}$$

Lemma 3.2. *[165, Theorem 14.21] Assume that for all $x \in \mathcal{X}$ and $f \in \mathcal{F}_k$ we have $f(x) \in [0, 1]$, then:*

$$Pr\left(\Psi(\mathcal{F}_k, \bar{x}) > 2R_m^{\mathcal{F}_k} + \epsilon\right) \leq e^{-2m\epsilon^2}. \tag{3.2}$$

[2] In our setting, in order apply the result with absolute value we assume that for any $f \in \mathcal{F}_k$ we also have $-f \in \mathcal{F}_k$, i.e., we assume that $\mathcal{F}_k$ is *closed under negation.*

Note that in our context (a) we need a one-sided bound, and (b) for all $x \in \mathcal{X}$ and $f \in \mathcal{F}_k$ we have $f(x) \in [0,1]$.

For algorithmic applications, two important consequences of these result are that: (a) for bounded functions the generalization error is concentrated around their expectation, and (b) the Rademacher Complexity can be estimated from the sample. In order for this bound to be actually usable in practical applications, it is necessary to compute an estimate of the Rademacher Complexity given the dataset $\bar{x}$, and to bound its error. In the "*textbook*" treatment, the difference between Rademacher Complexity and its empirical counterpart is bounded using a second application of *McDiarmid's Inequality* [13, 208]. However, this bound is often too loose for practical applications such as ours.

In this work, we propose an alternative, *direct*, estimate of the Rademacher Complexity and we develop two novel and tight methods for bounding the estimation error.

3.3.1 Tight bounds on Rademacher Complexity estimate

Given a finite size sample $\bar{x} \sim \mathcal{D}^m$ and ℓ independent Rademacher vectors $\bar{\sigma}_1, \ldots, \bar{\sigma}_\ell$, each composed of m independent Rademacher random variables (i.e., $\bar{\sigma}_j = \sigma_{j,1}\sigma_{i,2}\ldots\sigma_{j,m}$), we estimate $R_m^{\mathcal{F}_k}$ with

$$\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} = \frac{1}{\ell}\sum_{j=1}^{\ell}\sup_{f\in\mathcal{F}_k}\frac{1}{m}\sum_{i=1}^{m} f(x_i)\sigma_{j,i}. \tag{3.3}$$

Clearly, $\mathrm{E}_{\bar{x},\bar{\sigma}_1,\ldots,\bar{\sigma}_\ell}\left[\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}\right] = R_m^{\mathcal{F}_k}$. To bound the error $R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$, we model the process as a *Doob martingale* [165, Chapter 13.1] as follows:

$$C_i = E[R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} \mid Y_1, \ldots, Y_i] \text{ for } i = 0, \ldots, m(\ell+1),$$

where the $Y_1, \ldots, Y_{m(\ell+1)}$ are the random variables that determinate the value of the estimate $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$. The first m variables Y_i's correspond to the values of the sample $\bar{x}$, i.e. for $1 \le i \le m$, $Y_i = X_i$, and the remaining $m\ell$ Y_i's correspond to the Rademacher random variables, $Y_i = \sigma_{\lfloor i/m \rfloor, i-\lfloor i/m \rfloor}$. It is easy to verify that $C_0 = 0$, and $C_{m(\ell+1)} = R_m^{\mathcal{F}} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$.

Next, we define a *martingale difference sequence* $Z_i = C_i - C_{i-1}$ with respect to the martingale $C_0, C_1, \ldots C_{m(\ell+1)}$, and note that $\sum_{t=1}^{m(\ell+1)} Z_t = C_{m(\ell+1)} = R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$.

Our first bound builds on Bernstein's Inequality for Martingales (BIM). We use the following version due to Freedman [82], as presented in [63] and adapted to one-sided error.

Theorem 3.3. *[63, 82] Let $Z_1, \ldots, Z_t$ be a martingale difference sequence with respect to a certain filtration $\{\mathscr{F}_i\}_{i=0,\ldots,t}$. Thus, $E\left[Z_i|\mathscr{F}_{i-1}\right] = 0$ for $i = 1, \ldots, t$. The process $\sum_{i=1}^{t} Z_i$ is thus a martingale with respect to this filtration. Further, assume that $|Z_i| \le a$ for $i = 1, \ldots, t$, and that the conditional variance $\sum_{i=1}^{t} E\left[Z_i^2\right] \le L$. For $\epsilon \in (0,1)$, we have:*

$$Pr\left(\sum_{i=1}^{t} Z_i > \epsilon\right) \le e^{-\frac{\epsilon^2}{2L+2a\epsilon/3}}. \tag{3.4}$$

A careful analysis of $E[Z_i^2]$ in our application allows us to obtain a significantly stronger bound than the one obtained using McDiarmid's Inequality [13, 208], which depends on the maximum variation of the martingale.

Theorem 3.4. *Given a sample $\bar{x} \sim \mathcal{D}^m$, a family of functions $\mathcal{F}_k$ which take values in $[0,1]$, ℓ independent vectors of Rademacher random variables, and $\epsilon, \delta \in (0,1)$, we have:*

$$Pr\left(R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} > \epsilon\right) \leq e^{-\frac{6m\ell\epsilon^2}{15+8\ell\epsilon}}. \tag{3.5}$$

Proof. Recall that we defined the Doob martingale

$$C_i = E[R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} \mid Y_1, \ldots, Y_i] \text{ for } i = 0, \ldots, m(\ell+1),$$

where $Y_i = X_i$ for $1 \leq i \leq m$, and the remaining Y_i are the ℓm independent Rademacher random variables which compose the ℓ vectors of Rademacher random variables used to compute the estimate $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$ (i.e., $Y_{j(m)+i} = \sigma_{j,i}$, for $1 \leq j \leq \ell$ and $1 \leq i \leq m$). By the definition of $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$, C_i is a function of the $m(\ell+1)$ independent random variables $\{Y_i\}$. In the following, we assume $\ell \geq 2$.

Recall also the definition of the corresponding martingale difference sequence $Z_i = C_i - C_{i-1}$. By definition, for every $i = 1, \ldots, m(\ell+1)$, we have $\mathrm{E}[Z_i] = 0$, and hence, $\mathrm{E}\left[Z_i^2\right] = \mathrm{Var}[Z_i]$. In order to apply Bernstein's Inequality, we need a bound a, such that $a \geq |Z_i|$ for $1 \leq i \leq m(\ell+1)$, and an upper-bound L to the conditional variance, such that $L \geq \sum_{i=1}^{m(\ell+1)} \mathrm{E}\left[Z_i^2\right] = \sum_{i=1}^{m(\ell+1)} \mathrm{Var}[Z_i]$.

We consider the cases for $1 \leq i \leq m$ and $m < i \leq m(\ell+1)$ separately:

- $1 \leq i \leq m$: For $1 \leq j \leq \ell$, let us consider

$$C_i^{(j)} = E[R_m^{\mathcal{F}_k} - \sup_{f \in \mathcal{F}_k} \frac{1}{m} \sum_{i=1}^{m} f(x_i)\sigma_{j,i} \mid Y_1, \ldots, Y_i],$$
$$Z_i^{(j)} = C_i^{(j)} - C_{i-1}^{(j)}.$$

  According to our definitions, we have

$$C_i = \frac{1}{\ell} \sum_{j=1}^{\ell} C_i^{(j)}$$
$$Z_i = \frac{1}{\ell} \sum_{j=1}^{\ell} Z_i^{(j)}.$$

  Since $\forall x \in \mathcal{X}$ and $\forall f \in \mathcal{F}_k$, $f(x) \in [0,1]$, changing the value of any of the m points in $\bar{x}$ can change $\sup_{f \in \mathcal{F}_k} \frac{1}{m} \sum_{i=1}^{m} f(x_i)$ by at most $1/m$, and thus we have $|Z_i^{(j)}| \leq 1/m$, and $Z_i^{(j)} \in [\alpha, \beta]$ with $\beta - \alpha \leq 1/m$.

  From *Popoviciu's Inequality on variance* [184], we have that the variance of a random variable which takes values in $[\alpha, \beta]$ is bounded from above by $(\beta - \alpha)^2/4$. Hence, by applying Popoviciu's Inequality to $Z_i^{(j)}$, we have that $\mathrm{Var}\left[Z_i^{(j)}\right] \leq 1/(4m^2)$.

  As we are considering the expectation over the unassigned values of the Rademacher random variables, and as we are averaging over the values obtained using ℓ independent and identically distributed vectors of Rademacher random variables, we can conclude that $|Z_i| \leq \frac{1}{\ell} \sum_{j=1}^{\ell} |Z_i^{(j)}| \leq 1/m$, and $\mathrm{Var}[Z_i] = \frac{1}{\ell^2} \sum_{j=1}^{\ell} \mathrm{Var}\left[Z_i^j\right] \leq 1/(4m^2\ell)$.

- $m < i \leq m(\ell+1)$: Changing the value of any of the ℓm Rademacher random variables can change the value of $\tilde{R}_{\bar{x},j}^{\mathcal{F}_k}$ by at most $2/\ell m$, and thus we have $|Z_i| \leq 2/\ell m \leq 1/m$, and $Z_i \in [\alpha, \beta]$ with $\beta - \alpha \leq 2/\ell m$. By applying Popoviciu's Inequality, we thus have $\mathrm{Var}[Z_i] \leq 1/\ell^2 m^2$.

By linearity of expectation, $\sum_{i=1}^{m(\ell+1)} Z_i = R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$. Further, we have $\sum_{i=1}^{m(\ell+1)} Z_i^2 \leq 5/4\ell m$, and $|Z_i| < 1/m$ for all $1 \leq i \leq m(\ell+1)$. The statement follows by applying Theorem 3.3. □

Note that for a sufficiently large (constant) ℓ, the term $6\epsilon\ell$ dominates the denominator of the exponent in the right hand side of (3.5), giving a *fast rate of convergence* for the estimate. Our estimate fully characterizes the benefit achieved using multiple independent vectors of Rademacher random variables in estimating $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$.

Combining the results of Theorem 3.4, Lemma 3.1, and Lemma 3.2 using the union bound, we obtain a tight empirical bound on $\Psi\left(\mathcal{F}_k, \bar{x}\right)$.

Theorem 3.5. *Given a sample $\bar{x} \sim \mathcal{D}^m$, a family of functions $\mathcal{F}_k$ which take values in $[0,1]$, ℓ independent vectors of Rademacher random variables, and $\epsilon, \delta \in (0,1)$, we have:*

$$Pr\left(\Psi(\mathcal{F}_k, \bar{x}) > 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} + \epsilon\right) < \min_{\alpha \in (0,\epsilon)} e^{-2m(\epsilon-\alpha)^2} + e^{-\frac{3m\ell\alpha^2}{30+8\ell\alpha}}. \tag{3.6}$$

Proof. From Lemmas 3.1 and 3.2, we have:

$$\Pr\left(\Psi(\mathcal{F}_k, \bar{x}) > 2R_m^{\mathcal{F}_k} + \epsilon_1\right) \leq e^{-2m\epsilon_1^2}.$$

Theorem 3.4 characterizes the quality of the estimate of the Rademacher Complexity given by $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$, computed as specified in (3.3):

$$\Pr\left(R_m^{\mathcal{F}} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} > \epsilon_2\right) \leq e^{-\frac{6m\ell\epsilon_2^2}{15+8\ell\epsilon_2}}.$$

Combining the two results, we obtain:

$$\Pr\left(\Psi(\mathcal{F}_k, \bar{x}) > 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} + \epsilon_1 + 2\epsilon_2\right) \leq e^{-2m\epsilon_1^2} + e^{-\frac{6m\ell\epsilon_2^2}{15+8\ell\epsilon_2}}.$$

By substituting $\alpha = 2\epsilon_2$ and $\epsilon = \epsilon_1 + 2\epsilon_2$ in the previous equation, we have:

$$\Pr\left(\Psi(\mathcal{F}_k, \bar{x}) > 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} + \epsilon\right) \leq e^{-2m(\epsilon-\alpha)^2} + e^{-\frac{6m\ell(\alpha/2)^2}{15+8\ell\epsilon_2}}.$$

The statement follows. □

Alternative bound with single application of Bernstein's Inequality for Martingales: We now present an alternative result to the one in Theorem 3.5, which can be achieved with a single application of BIM. This bound is tighter than the one in Theorem 3.5 when the number of independent vectors of Rademacher random variables is very high.

Theorem 3.6.

$$Pr\left(\Psi\left(\mathcal{F}_k, \bar{x}\right) > 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} + \epsilon\right) < e^{-\frac{\epsilon^2}{\frac{\ell+4\sqrt{\ell}+20}{2m\ell} + \frac{4\epsilon}{3m}}}$$

Proof. Consider the Doob supermartingale:

$$C_i = \mathrm{E}\left[\Psi\left(\mathcal{F}_k, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} | Y_1, \ldots Y_i\right] \quad \text{for } i = 0, \ldots, m(\ell+1),$$

where for $1 \leq i \leq m$, $Y_i = X_i$, and the remaining Y_i correspond to the $m\ell$ independent Rademacher random variables in the ℓ vectors; that is, $Y_{j(m)+i} = \sigma_{j,i}$ for $1 \leq j \leq \ell$ and $1 \leq i \leq m$. It is easy to verify that $C_{m(\ell+1)} = \Psi\left(\mathcal{F}_k, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$. Further, $C_0 = \mathrm{E}\left[\Psi\left(\mathcal{F}_k, \bar{x}\right)\right] - 2R_m^{\mathcal{F}_k}$, and due to Lemma 3.1, $C_0 \leq 0$.

Let us define the corresponding martingale difference sequence $Z_i = C_i - C_{i-1}$. For each $i \in \{1, \ldots, m(\ell+1)\}$, due to linearity of expectation, we have $Z_i = A_i - 2B_i$, where:

$$A_i = \mathrm{E}\left[\Psi\left(\mathcal{F}_k, \bar{x}\right) | Y_1, \ldots Y_i\right] - \mathrm{E}\left[\Psi\left(\mathcal{F}_k, \bar{x}\right) | Y_1, \ldots Y_{i-1}\right];$$
$$B_i = \mathrm{E}\left[\tilde{R}^{\mathcal{F}_k}_{\bar{x},\ell} | Y_1, \ldots Y_i\right] - \mathrm{E}\left[\tilde{R}^{\mathcal{F}_k}_{\bar{x},\ell} | Y_1, \ldots Y_{i-1}\right].$$

In order apply Bernstein's Inequality, we need an upper-bound $a \geq |Z_i|$ for $1 \leq i \leq m(\ell+1)$ and an upper-bound L, such that $L \geq \sum_{i=1}^{m(\ell+1)} \mathrm{E}\left[Z_i^2\right]$.

Given our definition of Z_i, we have that for every i, $\mathrm{E}\left[Z_i\right] = \mathrm{E}\left[A_i\right] = \mathrm{E}\left[B_i\right] = 0$, and thus: $\mathrm{E}\left[Z_i^2\right] = \mathrm{Var}\left[Z_i\right] \leq \mathrm{Var}\left[A_i\right] + 4\mathrm{Var}\left[B_i\right] + 4\mathrm{Cov}\left[A_i, B_i\right]$. From the properties of covariance, we have $\left|\mathrm{Cov}\left[A_i, B_i\right]\right| \leq \sqrt{\mathrm{Var}\left[A_i\right]\mathrm{Var}\left[B_i\right]}$, and thus, $\mathrm{E}\left[Z_i^2\right] = \mathrm{Var}\left[Z_i\right] \leq \mathrm{Var}\left[A_i\right] + 4\mathrm{Var}\left[B_i\right] + 4\sqrt{\mathrm{Var}\left[A_i\right]\mathrm{Var}\left[B_i\right]}$.

We consider the cases for $1 \leq i \leq m$ and $m < i \leq m(\ell+1)$ separately:

- $1 \leq i \leq m$: In our setting $\forall x \in \mathcal{X}$ and $\forall f \in \mathcal{F}$, $f(x) \in [0,1]$, changing the value of any of the m points in $\bar{x}$ can change $f(\bar{x})$ by at most $1/m$. Therefore, $|A_i| \leq 1/m$, and $A_i \in [\alpha, \beta]$ with $\beta - \alpha \leq 1/m$. By applying Popoviciu's Inequality, we have: $\mathrm{Var}\left[A_i\right] \leq 1/4m^2$.

  The analysis for $\mathrm{Var}\left[B_i\right]$ follows the same reasoning discussed in the proof of Theorem 3.4 for bounding $\mathrm{Var}\left[Z_i\right]$ in the case $1 \leq i \leq m$, and thus $\mathrm{Var}\left[B_i\right] \leq 1/(4m^2\ell)$. We can thus conclude:

$$\begin{aligned}\mathrm{E}\left[Z_i^2\right] = \mathrm{Var}\left[Z_i\right] &\leq \frac{1}{4m^2} + \frac{4}{4\ell m^2} + 4\sqrt{\frac{1}{4m^2}\frac{1}{4\ell m^2}} \\ &\leq \frac{\ell + 4 + 4\sqrt{\ell}}{4m^2\ell}; \\ |Z_i| &\leq \frac{2}{m}.\end{aligned}$$

- $m < i \leq m(\ell+1)$: Changing the value of any of the Rademacher random variables does not change the value of $\Psi\left(\mathcal{F}_k, \bar{x}\right)$. Hence, $\mathrm{Var}\left[A_i\right] = \mathrm{E}\left[A_i^2\right] = 0$.

  Given fixed values for the random variables corresponding to the points in $\bar{x}$, changing the value of one Rademacher random variable can change the value of $\tilde{R}^{\mathcal{F}_k}_{\bar{x},\ell}$ by at most $2/\ell m$. Thus, $|B_i| \leq \frac{2}{\ell m}$, and $B_i \in [\alpha, \beta]$ with $\beta - \alpha \leq 2/\ell m$. By applying Popoviciu's Inequality, we have:

$$\mathrm{Var}\left[Z_i\right] = 4\mathrm{Var}\left[B_i\right] \leq \frac{4}{m^2\ell^2}.$$

We, therefore, have $|Z_i| \leq \frac{2}{m}$ for all $1 \leq i \leq m(\ell+1)$, and $\sum_{i=1}^{m(\ell+1)} \mathrm{E}\left[Z_i^2\right] \leq \frac{\ell + 4\sqrt{\ell} + 20}{4m\ell}$. By linearity of expectation, and by applying Lemma 3.1:

$$\begin{aligned}\sum_{i=1}^{\ell(m+1)} Z_i &= \Psi\left(\mathcal{F}_k, \bar{x}\right) - \mathrm{E}\left[\Psi\left(\mathcal{F}_k, \bar{x}\right)\right] - 2\left(\tilde{R}^{\mathcal{F}_k}_{\bar{x},\ell} - R^{\mathcal{F}_k}_m\right) \\ &\geq \Psi\left(\mathcal{F}_k, \bar{x}\right) - 2R^{\mathcal{F}_k}_m - 2\left(\tilde{R}^{\mathcal{F}_k}_{\bar{x},\ell} - R^{\mathcal{F}_k}_m\right); \\ &\geq \Psi\left(\mathcal{F}_k, \bar{x}\right) - 2\tilde{R}^{\mathcal{F}_k}_{\bar{x},\ell};\end{aligned}$$

The statement follows by applying BIM (Theorem 3.3). □

This result can be used in RadaBound in place of the bound given by Theorem 3.5. Note that with this result, it is easier to compute the bound on the probability of overfitting (denoted as δ' in line 11: of Algorithm 1).

3.3.2 Application of the Martingale Central Limit Theorem

In practical applications, one may prefer the standard practice in statistics of applying central limit asymptotic bounds. We develop here a bound based on the Martingale Central Limit Theorem (MCLT). Our experimental results in Section 3.6 show that the bound obtained using the MCLT is more powerful while still preserving statistical validity.

We adapt the following version of the MCLT [3]:

Theorem 3.7. *[97, Corollary 3.2] Let $Z_0, Z_1, \ldots$ be a difference martingale with bounded absolute increments. Assume that (1) $\sum_{i=1}^{n} Z_i^2 \xrightarrow{p} V^2$ for a finite $V > 0$, and (2) $E\left[\max_i Z_i^2\right] \leq M < \infty$, then $\sum_{i=1}^{n} Z_i / \sqrt{\sum_{i=1}^{n} E\left[Z_i^2\right]}$ converges in distribution to $N(0,1)$.*

When applying the MCLT, there is no advantage in bounding separately $\Psi\left(\mathcal{F}_k, \bar{x}\right) - 2R_m^{\mathcal{F}_k}$ and $2R_m^{\mathcal{F}_k} - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$. Instead, we compute a bound on the distribution of $\Psi\left(\mathcal{F}_k, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$ by analyzing the *Doob supermartingale*

$$C_i = E[\Psi\left(\mathcal{F}_k, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} \mid Y_1, \ldots, Y_i]$$

for $i = 0, \ldots, m(\ell+1)$, with respect to the same $Y_1, \ldots, Y_{m(\ell+1)}$ defined as in Section 3.1.

As in the finite sample case, the following theorem relies on a careful analysis of $\mathrm{E}\left[Z_i^2\right]$ for the martingale difference sequence $Z_i = C_i - C_{i-1}$.

Theorem 3.8. *Given a sample $\bar{x} \sim \mathcal{D}^m$, a family of functions $\mathcal{F}_k$ which take values in $[0,1]$, ℓ independent vectors of Rademacher random variables, and $\epsilon, \delta \in (0,1)$, we have:*

$$\lim_{m\to\infty} Pr\left(\Psi\left(\mathcal{F}_k, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} > \epsilon \frac{\sqrt{\ell + 4\sqrt{\ell} + 20}}{2\sqrt{\ell m}}\right) < 1 - \Phi\left(\epsilon\right).$$

Where $\Phi(x)$ denotes the cumulative distribution function for the standard normal distribution.

Due to its asymptotic nature, it is not possible to compare directly the tightness of the bound in Theorem 3.8 with that of finite sample bounds such as the one in Theorem 3.5. Still, this bound is of great interest in many practical scenarios as it allows for a much tighter bound for the generalization error.

Proof of Theorem 3.8. The proof closely follows the steps of the proof of Theorem 3.6. Consider the Doob supermartingale for the function $\Psi\left(\mathcal{F}_k, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$:

$$C_i = \mathrm{E}\left[\Psi\left(\mathcal{F}_k, \bar{x}\right) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} | Y_1, \ldots Y_i\right] \text{ for } i = 0, \ldots, m(\ell+1),$$

[3] Formally, the asymptotic is defined on a triangle array, where rows are samples of growing sizes. We also assume that all expectations are well-defined in the corresponding filtration.

where for $1 \leq i \leq m$, $Y_i = X_i$, and the remaining Y_i correspond to the $m\ell$ independent Rademacher random variables in the ℓ vectors. That is, $Y_{j(m)+i} = \sigma_{j,i}$, for $1 \leq j \leq \ell$ and $1 \leq i \leq m$. Further, let us define the corresponding martingale difference sequence $Z_i = C_i - C_{i-1}$.

In order to apply the MCLT, we need to bound $\sum_{i=1}^{m(\ell+1)} \mathrm{E}\left[Z_i^2\right]$ from above, and we need to verify that $|Z_i|$ is bounded.

Note that the sequence Z_i defined here corresponds to the martingale difference sequence by the same name that we studied in the proof of Theorem 3.6. As shown in the proof of Theorem 3.6, we have $\sum_{i=1}^{m(\ell+1)} \mathrm{E}\left[Z_i^2\right] \leq \frac{\ell+4\sqrt{\ell}+20}{4m\ell}$, and $|Z_i| \leq 2/m$ for all $1 \leq i \leq m(\ell+1)$.

Applying the MCLT, we have that as m goes to infinity, $\sum_{i=1}^{m(\ell+1)} Z_i / \sqrt{\sum_{i=1}^{m(\ell+1)} \mathrm{E}\left[Z_i^2\right]}$ converges in distribution to $N(0,1)$, and thus:

$$\lim_{m\to\infty} \Pr\left(\sum_{i=1}^{\ell(m+1)} Z_i \left(\sqrt{\sum_{i=1}^{m(\ell+1)} \mathrm{E}\left[Z_i^2\right]} \right)^{-1} > \epsilon \right) < 1 - \Phi(\epsilon),$$

$$\lim_{m\to\infty} \Pr\left(\sum_{i=1}^{\ell(m+1)} Z_i > \epsilon \sqrt{\frac{\ell+4\sqrt{\ell}+20}{4m\ell}} \right) < 1 - \Phi(\epsilon),$$

By linearity of expectation, and by applying Lemma 3.1:

$$\begin{aligned}
\sum_{i=1}^{\ell(m+1)} Z_i &= \Psi(\mathcal{F}_k, \bar{x}) - \mathrm{E}\left[\Psi(\mathcal{F}_k, \bar{x})\right] - 2\left(\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} - R_m^{\mathcal{F}_k}\right) \\
&\geq \Psi(\mathcal{F}_k, \bar{x}) - 2R_m^{\mathcal{F}_k} - 2\left(\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} - R_m^{\mathcal{F}_k}\right); \\
&\geq \Psi(\mathcal{F}_k, \bar{x}) - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k};
\end{aligned}$$

The statement follows. □

Two step application of the Martingale Central Limit Theorem: It is possible to break the result in Theorem 3.8 into two parts: (1) a bound on the the generalization error $\Pr\left(\Psi(\mathcal{F}_k, \bar{x}) - \mathrm{E}\left[\Psi(\mathcal{F}_k, \bar{x})\right] > \epsilon\right)$, and (2) a bound on the tightness of the proposed approximation of the Rademacher Complexity, that is $\Pr\left(R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} > \epsilon\right)$.

Lemma 3.9.

$$\lim_{m\to\infty} Pr\left(\Psi(\mathcal{F}_k, \bar{x}) - E\left[\Psi(\mathcal{F}_k, \bar{x})\right] > \frac{\epsilon}{2\sqrt{m}}\right) \leq 1 - \Phi(\epsilon).$$

Proof. Let us consider the Doob martingale for the function $\Psi(\mathcal{F}_k, \bar{x})$

$$C_i = \mathrm{E}\left[\Psi(\mathcal{F}_k, \bar{x}) \,|X_1, \ldots, X_i\right] \quad i = 0, \ldots, m, \tag{3.7}$$

which is defined with respect to the random variables $X_1, \ldots, X_m$, which correspond each to one of the elements in the sample $\bar{x}$. Let $Z_i = C_i - C_{i-1}$, for $1 \leq i \leq m$, be the corresponding martingale difference sequence.

In our setting $\forall x \in \mathcal{X}$ and $\forall f \in \mathcal{F}_k$, $f(x) \in [0,1]$, changing the value of any of the m points in $\bar{x}$ can change $f(\bar{x})$ by at most $1/m$. Therefore, $|Z_i| \leq 1/m$, and $Z_i \in [\alpha, \beta]$ with $\beta - \alpha \leq 1/m$. Given

our definition of the difference martingale Z_i, we have that for every i, $\mathrm{E}[Z_i] = 0$, and hence, by applying Popoviciu's Inequality, we have:

$$\mathrm{E}\left[Z_i^2\right] = \mathrm{Var}[Z_i] \leq \frac{1}{4m^2}.$$

By linearity of expectation, $\sum_{i=1}^{m} Z_i = \Psi(\mathcal{F}_k, \bar{x}) - \mathrm{E}\left[\Psi(\mathcal{F}_k, \bar{x})\right]$. Further, we have $\sum_{i=1}^{m} Z_i^2 \leq 1/4m$. By applying the MCLT we therefore have that $2\sqrt{m}\left(\Psi(\mathcal{F}_k, \bar{x}) - \mathrm{E}\left[\Psi(\mathcal{F}_k, \bar{x})\right]\right)$ converges in distribution to $N(0,1)$ as m goes to infinity. The lemma follows. □

We now show an application of the MCLT which allows to characterize the distribution of $R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x}}^{\mathcal{F}_k}$.

Lemma 3.10.

$$\lim_{m\to\infty} Pr\left(R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} > \frac{\sqrt{5}\epsilon}{2\sqrt{\ell m}}\right) \leq 1 - \Phi(\epsilon). \tag{3.8}$$

Proof. The proof closely follows the steps of the proof of Theorem 3.4. In the following we assume $\ell \geq 2$. Consider the Doob martingale C_i and the martingale difference sequence Z_i defined as in the proof of Theorem 3.4.

As shown in the proof of Theorem 3.4, we have $\sum_{i=1}^{m(\ell+1)} Z_i^2 \leq 5/4\ell m$, and that $|Z_i| < 1/m$ for all $1 \leq i \leq m(\ell+1)$. By linearity of expectation, $\sum_{i=1}^{\ell(m+1)} Z_i = R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$.

By applying the MCLT, we therefore have that $2\sqrt{\frac{\ell m}{5}}\left(R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}\right)$ converges in distribution to $N(0,1)$. The lemma follows. □

3.4 Bounding the generalization error using McDiarmid's inequality

McDiarmid's inequality is a useful variation of the more general Azuma-Hoeffding inequality [165]. In the theory literature, McDiarmid's inequality is used to obtain bounds on both the generalization error (i.e., Lemma 3.2) and the difference between the Rademacher Complexity and the Empirical Rademacher Complexity for a given sample $\bar{x}$ [208].An additional error is occurred in estimating the empirical Rademacher complexity using a finite number of Rademacher vectors.

Our proposed method eliminates this additional passage by estimating *directly* the Rademacher Complexity using as $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}}$, computed according to equation (3.3).

In [13] (Theorem 11), Bartlett et al. present a bound on the quality of the approximation of the Rademacher Complexity achievable using a single vector of Rademacher random variables. Instead, the result presented here is a one-sided bound. Further, we achieve an improvement of the exponential term in the right-hand side of the bound by taking into considerations that the functions being considered are non-negative and take values in $[0,1]$, and, crucially, by using multiple vectors of Rademacher random variables.

Lemma 3.11.

$$Pr\left(R_m^{\mathcal{F}_k} - \tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k} > \epsilon\right) \leq e^{-2m\ell\epsilon^2/(\ell+4)}. \tag{3.9}$$

Proof. Clearly $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$ is a function of $m(\ell+1)$ independent Random variables Y_i. For the simplicity purposes, assume $Y_i = X_i$ for $1 \leq i \leq m$, and the remaining Y_i are the ℓm independent Rademacher random variables which compose the ℓ vectors of Rademacher random variables. In order to apply McDiarmid's inequality we need to carefully bound the maximum change of the the value of the function $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$ when changing the the value of the i-th random variable Y_i, denoted as c_i.

- $1 \leq i \leq m$: as is our setting $\forall x \in \mathcal{X}$ and $\forall f \in \mathcal{F}_k$, $f(x) \in [0,1]$, changing the value of any of the m points in $\bar{x}$ can change $f(\bar{x})$ by at most $1/m$. As we are considering the maximum deviation in expectation with respect to the value of the $m\ell$ Rademacher random variables we can conclude $c_i \leq 1/m$.

- $m+1 \leq i \leq m(\ell+1)$: changing the value of one of the Rademacher random variables can change the value of $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$ by at most $\frac{2}{\ell m}$, hence, we have $c_i \leq \frac{2}{\ell m}$.

We therefore have:

$$\sum_{i=1}^{m(\ell+1)} c_i^2 = \sum_{i=1}^{m} c_i^2 + \sum_{i=m+1}^{m(\ell+1)} c_i^2 = \frac{1}{m} + \frac{4}{\ell m} = \frac{\ell+4}{\ell m}.$$

The lemma follows by applying McDiarmid's inequality. □

Note that for high values of ℓ, the right hand side of equation (3.9), is close to $e^{-2km\epsilon^2}$. In a natural tradeoff, while using multiple vectors of Rademacher random variables does indeed allow to obtain a higher quality estimate of the Rademacher Complexity, it also introduces a higher overhead time in the computation of the estimate.

In order to improve both the time and memory space requirements of our algorithm, we can obtain an estimation of the Rademacher Complexity $R_m^{\mathcal{F}_k}$ using a single vector of m Rademacher random variables. In this case, we can obtain a slightly better guarantee on the accuracy of $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}$ than the one implied by (3.9):

Lemma 3.12.

$$Pr\left(R_m^{\mathcal{F}_k} > \tilde{R}_{\bar{x},1}^{\mathcal{F}_k} + \epsilon\right) \leq e^{-m\epsilon^2/2}.$$

The proof of this lemma mostly follows the same steps as the one for Lemma 3.11: the sharper bound is obtained by observing that $\tilde{R}_{\bar{x},1}^{\mathcal{F}_k}$ can be characterized as a function of just m independent and identically distributed random variables $y_i = x_i\sigma_i$ from the product distribution $\mathcal{D} \times \sigma$.

Using the *union bound* we can combine the results of Theorem 3.2 and Lemma 3.11 (or Lemma 3.12) and obtain:

Theorem 3.13.

$$Pr\left(\Psi(\mathcal{F}_k, \bar{x}) > 2\tilde{R}_m^{\mathcal{F}_k} + \epsilon_1 + 2\epsilon_2\right) \leq e^{-2m\epsilon_1^2} + e^{-2m\ell\epsilon^2/(\ell+4)}. \tag{3.10}$$

3.5 The RADABOUND Algorithm

The algorithm starts by drawing ℓ *independent* vectors of Rademacher variables. These vectors are fixed throughout the execution of the algorithm. The advantage of fixing the Rademacher vectors is that (1) we deal with a nested sequence of events, $\mathcal{F}_{k-1} \subseteq \mathcal{F}_k$, and (2) the actual computation of the Rademacher complexity estimate is simple and efficient.

Computing the estimate: At the end of each round k, the algorithm stores for each of the Rademacher vectors $j = 1, \ldots, \ell$, the value $\tilde{M}_{\bar{x},j}^{\mathcal{F}_k} = \max_{f \in \mathcal{F}_k} \frac{1}{|\bar{x}|} \sum_{i=1}^{|\bar{x}|} f(x_i)\sigma_{i,j}$. To update these values, at iteration $k+1$, the algorithm computes

$$\tilde{M}_{\bar{x},j}^{\mathcal{F}_{k+1}} \leftarrow \max\{\tilde{M}_{\bar{x},j}^{\mathcal{F}_k}, \frac{1}{|\bar{x}|} \sum_{i=1}^{m} f_{k+1}(x_i)\sigma_{i,j}\}, \quad j = 1, \ldots, \ell.$$

The estimate of the Rademacher Complexity at round $k+1$ is then given by $\tilde{R}_{\bar{x}}^{\mathcal{F}_{k+1}} = \frac{1}{\ell} \sum_{j=1}^{\ell} \tilde{M}_{\bar{x},j}^{\mathcal{F}_{k+1}}$.

ALGORITHM 3 RADABOUND - Adaptive data analysis with Rademacher Complexity control

procedure RADABOUND($\bar{x}, \varepsilon, \delta, \ell$)
  $m \leftarrow |\bar{x}|$ ▷ Size of the input sample
  ▷ Initialization estimator for Rademacher Complexity
  for $j \in \{0, 1, \ldots, \ell\}$ **do**
    $\sigma_j \leftarrow$ vector of m iid Rademacher RVs
    $R_{\bar{x},j}^{\mathcal{F}_0} \leftarrow 0$
  ▷ Main execution body
  while new k-th query f_k from the stream **do**
    $\mathcal{F}_{k+1} \leftarrow \mathcal{F}_k \cup \{f_k\}$
    ▷ Rademacher Average estimation update
    for $j \in \{0, 1, \ldots, \ell\}$ **do**
      $R_{\bar{x},j}^{\mathcal{F}_{k+1}} \leftarrow \max\{R_{\bar{x},j}^{\mathcal{F}_k}, \frac{1}{m} \sum_{i=1}^{m} f_k(x_i)\sigma_{j,i}\}$
    $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{k+1}} \leftarrow \frac{1}{\ell} \sum_{j=1}^{\ell} R_{\bar{x},j}^{\mathcal{F}_{k+1}}$
    ▷ Control with BIM
    $\delta' \leftarrow \min_{\epsilon' \in \left(0, \epsilon - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_k}/2\right)} e^{-2(\epsilon - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{k+1}} - 2\epsilon')^2 m} + e^{\frac{6m\ell\epsilon'^2}{15+8\ell\epsilon'}}$
    ▷ Control with MCLT- Alternative to 11:
    or $\delta' \leftarrow 1 - \Phi\left(\max\{0, \epsilon - 2\tilde{R}_{\bar{x},\ell}^{\mathcal{F}_{k+1}}\} \sqrt{\frac{4\ell m}{\ell + 4\sqrt{\ell} + 20}}\right)$
    ▷ Overfit control test
    if $\delta' \leq \delta(1-\delta)$ **then**
      return $\frac{1}{m} \sum_{x \in \bar{x}} f(x)$
    else
      Halt: Cannot guarantee the statistical validity of further queries.

Stopping rule: Given real values $\epsilon, \delta \in (0, 1)$, the procedure halts at the first k-th step for which it cannot guarantee that $\Pr_{\mathcal{L}}\left(\Psi(\mathcal{F}_{k+1}, \bar{x}) > \epsilon\right) \leq \delta$. Recall from (3.1) that

$$\Pr_{\mathcal{L}}\left(\Psi(\mathcal{F}_k, \bar{x}) > \epsilon\right) \leq \frac{\Pr\left(\Psi(\mathcal{F}_k, \bar{x}) > \epsilon\right)}{\Pr\left(\Psi(\mathcal{F}_{k-1}, \bar{x}) \leq \epsilon\right)}. \tag{3.11}$$

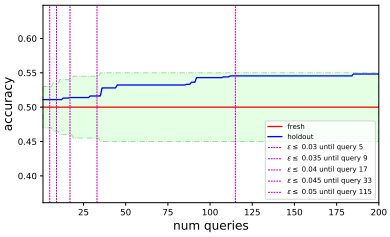
Since $\Pr(\Psi(\mathcal{F}_k, \bar{x}) > \epsilon) \geq \Pr(\Psi(\mathcal{F}_{k-1}, \bar{x}) > \epsilon)$, it is sufficient to require $\Pr\left(\Psi(\mathcal{F}_k, \bar{x}) > \epsilon\right) < \delta(1-\delta)$ to have $\Pr_{\mathcal{L}}\left(\Psi(\mathcal{F}_k, \bar{x}) > \epsilon\right) \leq \delta$, and we can use the bounds obtained in Theorem 3.5 or Theorem 3.8. Thus, we prove

Theorem 3.14. *Given a sample $\bar{x} \sim \mathcal{D}^m$, let $\mathcal{F}_k$ denote the set of functions adaptively selected during the first k steps. If* RadaBound *has not halted at step k, then*

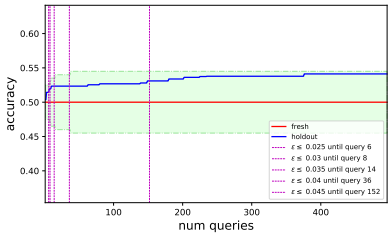
$$Pr_{\mathcal{L}}\left(\Psi(\mathcal{F}_k, \bar{x}) \leq \epsilon\right) > 1 - \delta.$$

The bound in Theorem 3.8 based on the MCLT can be used in RadaBound *as an alternative* to the bound in Theorem 3.5 (lines 11-12 in Algorithm 1). In Section 5, we present an experimental comparison of performance of RadaBound when using the two methods.

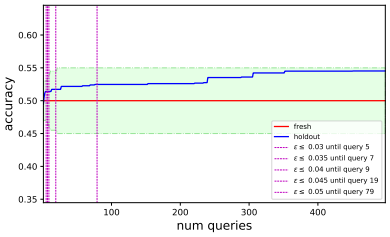
3.6 Experimental results



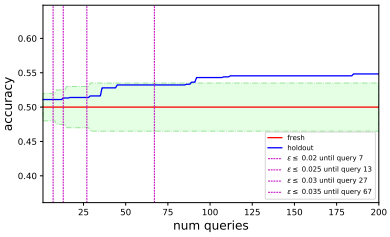
(a) Bernstein bound



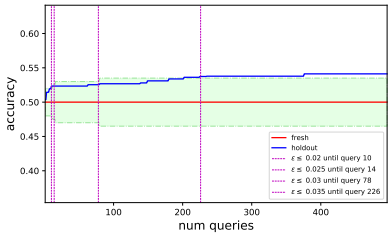
(a) Bernstein bound



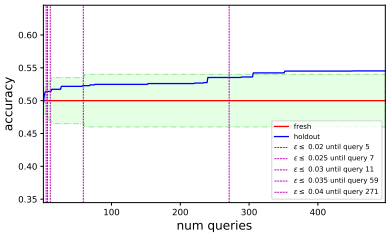
(a) Bernstein bound



(b) MCLT



(b) MCLT bound



(b) MCLT bound

Figure 3.1: No signal. Feature values from $\mathcal{N}(0,1)$. $\delta = 0.1$.

Figure 3.2: No signal. Feature values from $\mathcal{N}(0,2)$. $\delta = 0.15$.

Figure 3.3: No signal. Feature values from $\mathcal{N}(0,8)$. $\delta = 0.2$.

We demonstrate the power and efficiency of our technique through a variety of experiments. Our experimental setup is similar to the one used in [58], except that all our reported results are for ranges of parameters for which we actually have provable statistical guarantees.

We consider a learning task of classifying vectors composed by d features to the classes "-1" or "1". We consider only linear classifier vectors $\mathbf{w} \in \{-1, 0, 1\}^d$, assigning vector $\mathbf{x}$ to class $\mathbf{h}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x})$. The goal of the learning algorithm is to find a classifier with minimum expected loss for the $0,1$ *hard loss function* (0 for correct classification, 1 otherwise). To model a typical learning scenario, the learning algorithm is given two independent datasets. A *training set* X_T and an *holdout*

set X_H. We then evaluate the performance of the learning algorithm using a third, independent *fresh set*.

Our learning algorithm works as follows: In the first phase, the algorithm evaluates the correlation between the values of the features and the labels of the vectors using *only* the training dataset X_T as $c_i = \frac{1}{m}\sum_{\mathbf{x}\in X_T} \mathbf{x}[i]l(\mathbf{x})$. The features are then sorted (in descending order) according to the absolute values of their correlations $|c_i|$ to the labels.

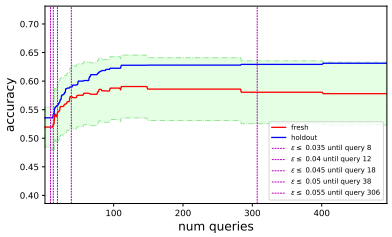
The actual adaptive analysis of the data occurs in the second phase of the algorithm using the holdout data X_H. The algorithm starts with a classifier $\mathbf{w} = 0$. It then considers features according to the order computed in the first phase. Using the holdout set X_H, the algorithm tests, for each feature, whether assigning weight -1 or 1 to it improves the performance of the current best classifier. If that is the case, the classifier is updated with the new value for the feature; otherwise, the feature is left with weight zero. Each newly tested classifier is added to the class function $\mathcal{F}_k$. RadaBound then computes a new estimate $\tilde{R}_m^{\mathcal{F}_k}$ of the Rademacher Complexity of $\mathcal{F}_k$, and uses it to determinate whether the total accumulated error is below ϵ with probability at least $1-\delta$ as discussed in Section 3.5.

Each figure above corresponds to several runs of an experiment with the same parameters but different values of ϵ (the error bound). The blue line gives the accuracy of the best classifier computed after running the corresponding number of queries on the holdout set X_H. The red line gives the accuracy of the same classifier on fresh data. The vertical bars give the computed stopping time for each value of ϵ. The shaded green area corresponds to $\pm\epsilon$ values around the "*true accuracy*" of the classifier (the red line), for the ϵ value of the next vertical bar. The green shaded area beyond the last bar uses the same ϵ as the last bar. In a correct execution of RadaBound, the blue line does not exit the green shaded area before the last vertical bar. The power of RadaBound is measured by how close is the last bar to the first time the green line exit the shaded area.

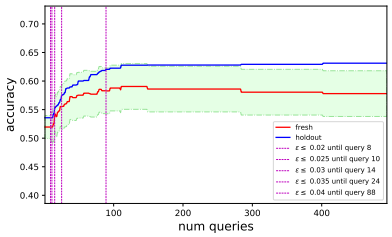
In all the experiments $|X_T| = |X_H| = 4000$. Each vector in the dataset has 500 features. The estimation of the Rademacher Complexity $\tilde{R}_{\bar{x},\ell}^{\mathcal{F}}$ is computed according to (3.3) using $\ell = 32$ vectors of Rademacher random variables. We report results using (a) Bernstein's Inequality (Section 3.3.1) and, (b) the MCLT (Section 3.3.2). We consider the two following scenarios:

No signal in the data: In this setting, each point $\mathbf{x} \in X_H$ is assigned a label independently and uniformly at random. The feature values are taken *independently* from a normal distribution with expectation 0 and various variance values. Thus, there is no correlation between the labels and the values of the features. We report the results in Figures 3.1-3.3.

Signal in the data: In this setting the ''"*strength*" of the correlation between some features and the labels is characterized by two parameters: n, the number of the queries whose value is correlated to the label, and (positive or negative) *bias* which defines the strength and sign of the correlation. We first generate datasets with no signal, like in the previous setting. We then fix a set of 50 features to be correlated with the label of their vectors. Letting $l(\mathbf{x})$ be the label of vector $\mathbf{x}$, the 50 correlated features of $\mathbf{x}$ are modified by adding $bias \times l(\mathbf{x})$ to their original value. We report the results in Figures 3.4-3.6.

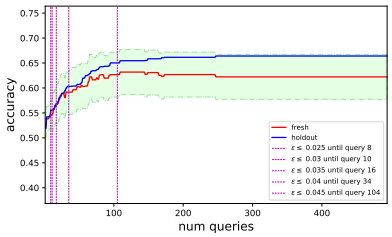


(a) Bernstein bound

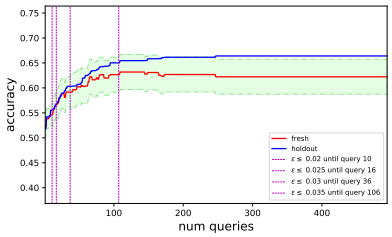


(b) MCLT bound

Figure 3.4: Signal. Feature values from $\mathcal{N}(0,4)$, $\delta = 0.1$, $bias = 0.5$.

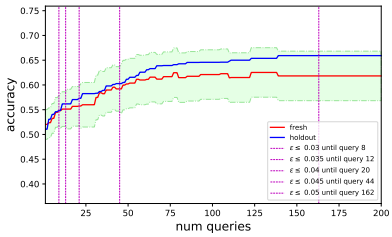


(a) Bernstein bound

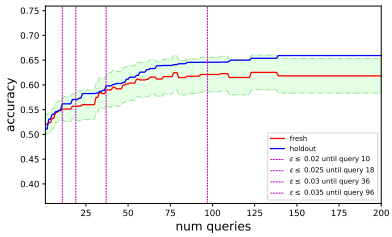


(b) MCLT bound

Figure 3.5: Signal. Feature values from $\mathcal{N}(0,2)$, $\delta = 0.15$, $bias = 0.5$.



(a) Bernstein bound



(b) MCLT bound

Figure 3.6: Signal. Feature values from $\mathcal{N}(0,2)$, $\delta = 0.2$, $bias = 0.25$.

The results demonstrate that RadaBound successfully halts the sequence of tests before overfitting for the various values of ϵ, as the green line corresponding to the values of the function evaluated on the holdout does not exit the green shaded area before the corresponding vertical bar. The statistical power of the procedure is highlighted in particular by the result of experiments for which there is actual correlation between the labels and the value of the features as the overfit control ensured by RadaBound is not achieved at the expense of detecting the signal in the data.

In several scenarios, RadaBound halts its execution very close to the first iteration for which overfit (with respect to the value of ϵ) actually occurs. RadaBound does not appear to be influenced by the distribution $\mathcal{D}$ over the data, but rather it behaves differently depending on the actual family of functions being tested.

For similar ϵ, δ parameters, the state-of-the-art `Thresholdout` algorithm [58] would require an holdout dataset of size $\sim 4 \times 10^6$ to provide answers to just 10 queries (details in Section 3.7). In contrast, our experiments show that RadaBound can provably handle such parameters with an holdout set of just 4000 samples, thus with an improvement of almost three orders of magnitude in terms of *sample complexity*. The comparison is further discussed in Section 3.7.

Using the MCLT leads to a tight analysis of $\Psi(\mathcal{F}_k, \bar{x})$, which in turn allows to test a higher number of adaptively chosen classifiers before halting and without overfitting (Figures 3.1b - 3.6b), compared to the stopping points obtained using the BIM (Figures 3.1a - 3.6a)[4].

Finally, the fact that even when using the bounds obtained using the MCLT, the procedure halts

[4]Using bounds obtained using McDiarmid's Inequality in RadaBound leads to stopping points considerably more conservative than those obtained using the BIM for the same parameters.

correctly, further suggests that, despite their "asymptotic" nature, these are actually highly reliable even when dealing with an input sample of relatively small dimension.

3.7 Comparison with methods based on Differential Privacy

The `Thresholdout` algorithm [58] provides guarantees similar to those of RadaBound. `Thresholdout` operates using two datasets: a public dataset and a private *holdout* dataset. Every time a new query is received the algorithm evaluates its value on both the public and the private dataset. If their absolute difference is within a given *threshold*, `Thresholdout` returns to the user the value observed on the public dataset after perturbing it with some noise. Viceversa, if the absolute difference is higher than a certain threshold, the algorithm detects that the query being considered is overfitting on the public dataset. In this case, `Thresholdout` may instead provide the value computed on the private dataset after perturbing it with noise. As this last operation effectively "*leaks*" information regarding the holdout it can be executed up to B times, where B must be fixed prior to the execution of the algorithm.

To characterize the number of queries for which algorithm `Thresholdout` provides provable statistical guarantees we apply Theorem 25 in [58] [5]. The theorem states that when testing up to k queries, with up to $B = 1$ of those being answered using the private "Holdout set", the size of the holdout set must be at least:

$$n \geq 96\epsilon^{-2} \ln\left(4k\delta^{-1}\right) \min\{80\sqrt{B\ln(1/\epsilon\delta)}\}, 16B\}$$

Thus, for $k = 10$, $B = 1$, $\epsilon = 0.5$, and $\delta = 0.1$: the required sample size is at least

$$n \geq 400 \times 96 \times \ln(400) \min\{80\sqrt{\ln(200)}, 16\} \geq 3.7 \times 10^6.$$

Therefore, even when requesting such, fairly loose, guarantees, `Thresholdout` requires an extremely high sample size in order to provide reliable answer to a handful of queries.

In contrast, we showed in Section 3.6 that RadaBound can provably handle problems with these, and better, parameters while using an holdout set composed by just 4000 samples. Thus RadaBound achieves an improvement of almost three orders of magnitude in terms of *sample complexity* compared to `Thresholdout`.

Further, `Thresholdout` requires that the user specifies *before* the execution the number of queries k which are going to be adaptively chosen to be tested, and the number B of maximum times that the algorithm can tolerate overfit on the public dataset by revealing information from the private "*holdout*" dataset. These requirements limit the adaptiveness of the process. In contrast RadaBound uses the holdout dataset *as much as possible* without a fixed maximum number of queries. Finally, using Rademacher Complexity in evaluating the stopping criterion of the adaptive testing procedure allows RadaBound to evaluate the properties of the actual family of functions

[5] A careful reader will notice that Figures 1-3 in [58] (the same figure appears as Figures 1 and 2 in [59]) represent an idealized illustration rather than statistically valid results. For the sample size used in these figures, the bound on the error probability of the threshold algorithm is not smaller than 1. Furthermore, the results crucially depend on a preprocessing of the two data sets (lines 95-97 in the "`runClassifier`" procedure in the python code in the supporting materials) that is not discussed in the paper.

tested so far (i.e., their *expressiveness*), rather than just its cardinality, in order to provide guarantees on the quality of the evaluations obtained in the adaptive analysis.

3.8 Conclusion

We presented a rigorous, efficient and practical method for bounding the generalization error in an adaptive sequence of queries tested on the same dataset. While the standard "rule of thumb" for responsible data analysis and machine learning is to use a test set only once, our results demonstrate that, with an appropriate control mechanism, it may be possible to use the same test set more than once without significantly reducing the validity of the results.

For concreteness, we focused here on the problem of evaluating the expectations of a set of functions in the range $[0, 1]$. This problem corresponds to the basic machine learning task of evaluating the correctness of classifiers with bounded $[0, 1]$ loss functions. We note that our methods can be extended to a more general setting, for example using new concentration bounds on sub-exponential distributions [134], and self-bounding functions [170].

Part II

Visual data analysis with quality guarantees

Chapter 4

Towards sustainable insights[1]

Have you ever been in a sauna? If yes, according to our recent survey conducted on Amazon Mechanical Turk, people who go to saunas are more likely to know that Mike Stonebraker is not a character in "The Simpsons". While this result clearly makes no sense, recently proposed tools to automatically suggest visualizations, correlations, or perform visual data exploration, significantly increase the chance that a user makes a false discovery like this one. In this Chapter, we first show how current tools mislead users to consider random fluctuations as significant discoveries. We then describe our vision and early results for QUDE, a new system for automatically controlling the various risk factors during the data exploration process.

4.1 Introduction

"A new study shows that drinking a glass of wine is just as good as spending an hour at the gym" [Fox News, 02/15]. "A new study shows how sugar might fuel the growth of cancer" [Today, 01/16]. "A new study shows late night snacking could damage the part of your brain that creates and stores memories" [Fox News, 05/16].

Over the last years we have seen an explosion of data-driven discoveries like the ones mentioned above. While several of these are indeed legit, there has also been an increase of more and more questionable findings [113]. Albeit the reasons behind this trend are manifold, we recently observed that the research community started to develop tools, like Vizdom/IDEA [48], SeeDB [231] or Data Polygamy [44], that are likely to considerably increase the number of false discoveries from data analysis. For instance, visual data exploration tools, such as Vizdom/IDEA [48] or Tableau, significantly simplify data exploration for domain experts and, more importantly, novice users. These tools allow to discover complex correlations and to test hypotheses and differences between various populations in an entirely visual manner with just a few clicks, unfortunately, often ignoring even the

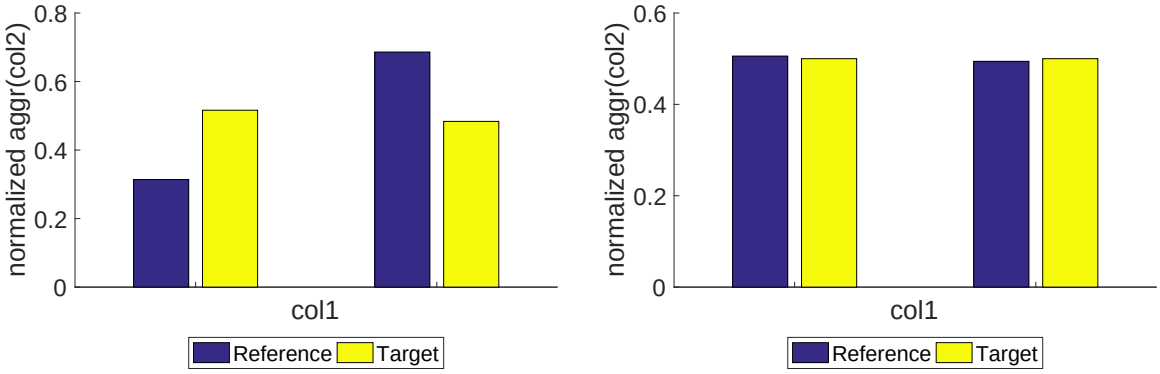
[1]The results presented in this chapter were presented in the 7th biennial Conference on Innovative Data Systems Research (CIDR 2017). This is joint work with Professor Carsten Binnig, Professor Tim Kraska, Professor Eli Upfal, Emanuel Zgraggen and Zheguang Zhao.

most basic statistical rules. For example, in a recent user study that we performed, we have asked people to explore a dataset containing information about different wines and report their findings. Using histograms showing American vs. French wines, most subjects came to the conclusion, that French wines earn higher critic ratings on average. At the same time, almost none of the participants used a statistical method to test if the visually observed difference from the histogram is actually meaningful. Similar, none of the participants, even the more statistically savvy ones, did consider that the arbitrary exploration and attempts to find interesting facts actually increases their chance to find random occurrences of seemingly significant correlations.

While these concerns are often waived under arguments such as "scientists are in desperate need for better tools, so any help is better than none", it is often not recognized that these tools not only greatly increase the likelihood of spurious discoveries but in many cases, make it also impossible to control the number of false discoveries later on. For example, recent visual recommendation systems, such as SeeDB [231] or Data Polygamy [44], are potentially checking thousands of hypotheses in just a few seconds and are smoking guns hiding as water pistols. SeeDB tries for example to find interesting visualizations and while a visualization per se does not seem like a hypothesis test, it should be treated as one. Why otherwise should a visualization be considered interesting, if the effect shown by the visualization is not relevant?

As a result, by testing thousands of visualizations it is almost guaranteed that the system will find something "*interesting*" regardless of whether the observed phenomenon is actually statistically relevant or not. Similar, Data Polygamy tries to find interesting correlations between time series data. Hypothesis tests are therefore actually performed using MC-methods relying on a fixed threshold for the p-values, without providing a correction for multiple hypotheses. Even more astonishingly, while the authors do discuss p-value adjustments using the Bonferroni correction, they then – surprisingly – disregard it. Let us assume the system has to test 100 potential correlations, 10 of them being true. Assuming a p-value of 0.05 (as suggested in [44]), and that our test has a statistical power of 0.8 (common values for a single statistical test), Data Polygamy on average will find 13 correlations of which 5 ($\approx 40\%$) are "*bogus*" (i.e., they are *false positives*). Even worse, without knowing how exactly the system tried to find the "interesting" correlations and how many correlations it tested, it is later on impossible for the user to determine what the expected false discovery rate will be across the whole data exploration session.

In this Chapter, we outline our vision and initial results for QUDE, the first system to Quantifying the Uncertainty in Data Exploration, which is part of Brown's Interactive Data Exploration Stack (BIDES). In order to better quantify the severity of the problem, we analyze existing visual recommendation systems, namely SeeDB and Data Polygamy, and discuss how they are prone to find wrong insights using simulated and real-world data. We further quantify the risk for data exploration systems like Vizdom/IDEA, which is not as severe as for automatic insight finders, but still persists because of its capability to quickly test many different hypotheses. Afterwards, we discuss QUDE and how it achieves a more sustainable data discovery process based on techniques for controlling the False Discovery Rate (FDR) [17]. Furthermore, we also show how QUDE tries



(a) Interesting visualization **(b) Uninteresting visualization**

Figure 4.1: Examples of interestingness as defined in [231].

to automatically infer the tested hypotheses based on the user interactions, and how we plan to incorporate user feedback, as well as, warn the user about potential risk factors. While QUDE is designed mainly to avoid the risk of multi-hypothesis testing, it also includes techniques to tackle other risk factors such as missing data or visually misleading results (e.g., Simpson paradox).

Our main contribution is twofold: first, we demonstrate the risk of false discoveries by using three example systems, Vizdom/IDEA [49, 48], SeeDB [231], and Data Polygamy [44] (Section 4.2). However, the insights gained from these systems apply to a large range of commercial (e.g. [98]) and research prototypes (e.g. [231]). Secondly, in Section 4.3, we present our vision of QUDE and initial techniques we use to control the amount of false discoveries.

4.2 The Risk With Today's Tools

Modern tools for interactive data exploration enable domain experts and novice users alike to efficiently analyze large amounts of data. At the same time, if not used carefully, these tools can significantly increase the risk of making spurious discoveries. In this section, we analyze different tools for data exploration and discuss how these tools amplify the risk of false discoveries. Later, in Section 4.3, we present techniques based on a statistical concept called "*False Discovery Rate*" (FDR) and how we adopt these techniques for data exploration to control the risk factors.

4.2.1 Visual Data Exploration

Visualizations are arguably the most important tool to explore, understand and find insights in data. As part of interactive data exploration, visualizations are used to skim through the data and look for interesting patterns. It comes therefore at no surprise that the database research community over the last few years focused on developing techniques (e.g., adaptive indexing, approximate query processing) to better support interactive exploratory workloads [111]. Visualization systems such as Vizdom [48] are capable of visualizing large-scale data with interactive speed. While interactivity is

key to the usability of advanced analytical tools [148], using them unfortunately also significantly increases the risk of making spurious discoveries. Such risk has two aspects:

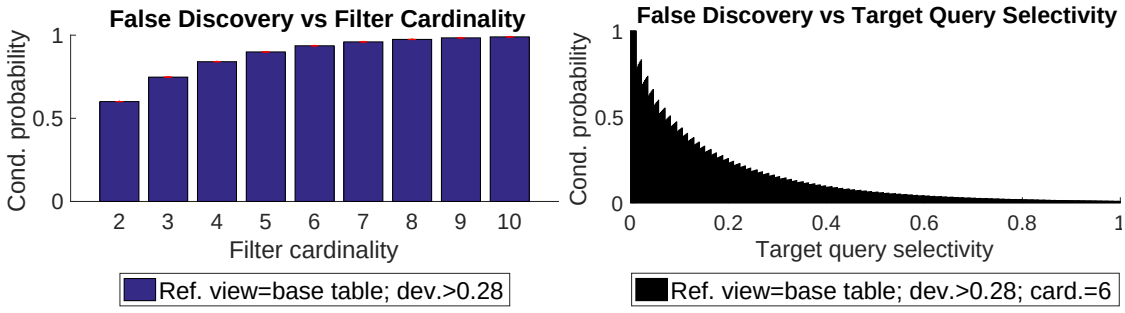
(1) The statistical significance of the visualized results is unclear.

(2) The growing number of hypotheses being tested during exploration increases with every single visualization.

The first aspect of risk is important because visualizations have the power to influence human perception and understanding by the rich information they may carry. Suppose that a salesperson of an ice cream company is exploring a data set about the sales. As the first step, she wants to get a yearly distribution of the sales figures. So she compares the sales of the last five years using a histogram of sales per year. In the second step, she is interested in learning if the the sales differ significantly across different states. She thus compares sales per state over the last five years.

Suppose the histogram shows that sales in Vermont were higher than in Rhode Island. Consider how tempting it is for an unsophisticated user to conclude that Vermonters buy more ice cream just based on the visualization. Although a statistically inclined user would formally analyze this observation by using hypothesis testing, she would have to redirect her attention to work with a different statistical tool (e.g. R [81]) before proceeding to the next data exploration step. After such efforts on context-switching, the insight might turn out completely due to random noise. At scale, the division of labor between data exploration and hypothesis testing will cause even more waste of human efforts on such spurious insights. Thus, *if a visualization provides any insight, the insights should be immediately tested for their significance.* If that would not be the case, the value of the visualization would be very limited as the user would not be allowed to make any conclusions based on the visualization. Thus if we consider a visualization as something more than a pretty picture presented to the user (i.e., more than just a listing of facts), we should always test the insight the user gains from the visualization for its significance and inform the user about it. A central challenge is of our work is the understanding of the hypothesis derived by the user given a certain data visualization. With respect to the previous example, the hypothesis derived by the user could be: (1) Vermonters buy more ice cream than Rhode Islanders, (2) Rhode Islanders buy more than Vermonters, or (3) they buy ice cream in the same amount.

The second aspect of risk is arguably even more severe. With every additional hypothesis test the chance of finding a false discovery increases. This problem is known as the "*multiple comparisons problem*" and has been studied extensively in the statistics literature [28, 15, 91, 130]. While only a decade ago it was an art left to experts, data analytics has become more and more accessible to a broader range of users with the advent of open-source data analysis systems. What has clearly changed is how easy it has become to test an hypothesis. For instance, if ten years ago it took a scientist one day to do a single test (including determining the right test statistic, collecting and cleaning data, etc.), it is now very easy to do 20 or more in a few minutes on a system such as Vizdom [48]. Assuming a significance level of 5%, for 20 tests the risk to falsely reject at least one true null hypothesis increases to $1 - (1 - 0.05)^{20} = 64\%$.



(a) Multiple Target Queries **(b) Single Target Query**

Figure 4.2: The risk analysis of false discoveries in SeeDB [231]

Data exploration on systems such as Vizdom [48] not only increase the risk of false discovery, but also change the way how statistical tests are applied. Suppose in the previous example the salesperson explores various relationships in the sales dataset through visualizations until she sees a visualization that she deems useful (e.g. significantly more ice cream sales to males in Massachusetts compared to California). With some statistics background, she validates this insight by using an appropriate test with a significance level of 5%. Suppose the observed p-value is below the significance level, she rejects the null hypothesis and believes that there is only a 5% chance that she incorrectly rejected the null hypothesis in case it was true. However, this way of applying statistical test is wrong. What the user ignores is that before she did the test she had already searched through the dataset for a while, and had observed different insights and implicitly their corresponding hypotheses, albeit untested. Thus, by the time the user applied the statistical test, she was already inadvertently trapped into the multiple comparisons problem, because the data exploration tool provided her the illusion that data exploration was not a sequence of hypotheses.

To conclude, without considering the risk of false discoveries, current interactive data exploration tools have the propensity to significantly exacerbate the problem of considering random occurrences as insights. Rather than limiting data exploration exclusively to statisticians, we believe in empowering both unsophisticated and advanced users with more intelligent systems with automatic risk control, where data-driven insights can be drawn both efficiently and safely. While it is clearly not the tool's fault that false discoveries happen, in the end it is the user's, tools like Vizdom, Tableau and many others purposefully target a broader audience of users. That is, more users without a sufficient statistic background will be using these tools and not understand the risk factors. Furthermore, even trained statisticians struggle to fully control the multi-hypothesis problem, which in theory requires keep track of every single insight every user ever made over a given dataset.

Therefore, with QUDE we plan to build a system which actively makes the user aware of these problems during the data exploration process and controls the risk of false discoveries automatically. Unfortunately, traditional techniques for multi-hypothesis testing, such as the Bonferroni correction, are both too pessimistic and require the system to know the number of hypotheses the user can explore upfront, which make them inadequate for data exploration.

4.2.2 Visual Recommendations

To automate data-driven discoveries at scale, *visualization recommender systems* such as Scagnostics [168], SeeDB [231], VizDeck [127], or Voyager [237] have been proposed. None of them however considers the risk of false discovery.

In a nutshell, visualization recommendation systems automate two dimensions of the visual data exploration process: (1) Recommend new visualizations with the goal to provide new insights (2) provide better representations for a given visualization. We focus on the first type of recommendation systems as they are similar to the systems discussed in Section 4.2.1, except for the fact that the system itself becomes the explorer of the data and is capable of checking thousands of hypotheses in just a few seconds. Without controlling the risk of false discoveries, these systems systematically increase the risk of spurious discoveries at scale. For the remainder of the discussion we use SeeDB [231] as an example, though similar observations can be made for other systems.

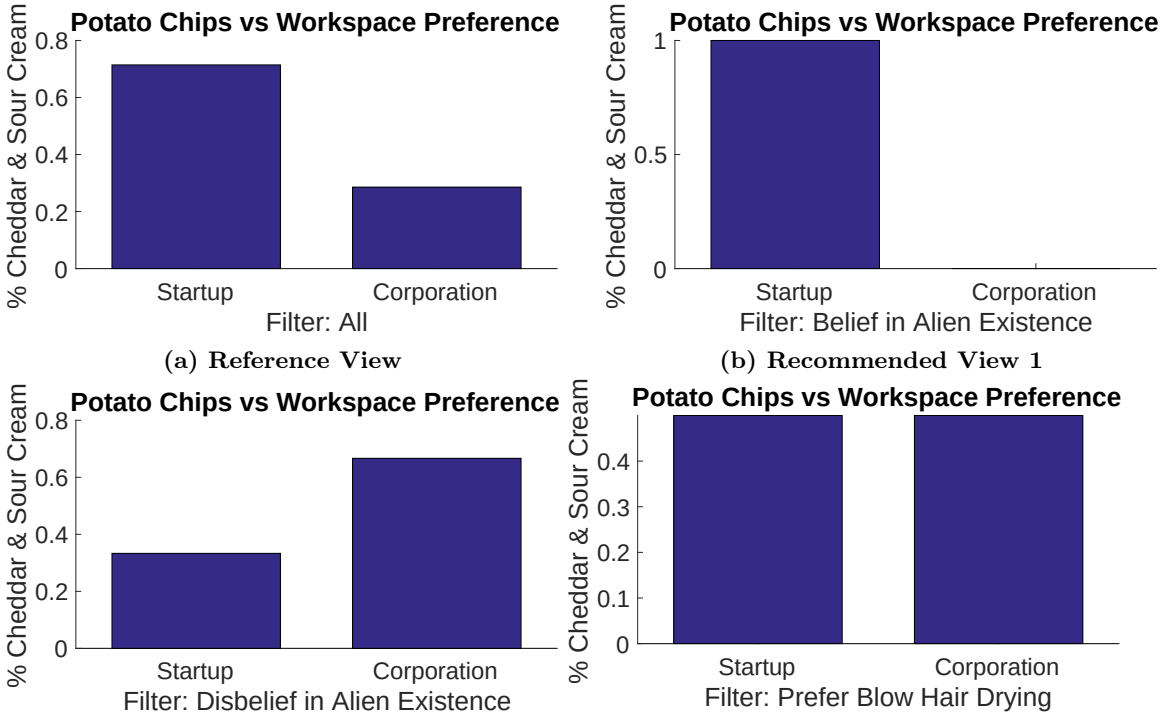
SeeDB considers the current query and according visualization of the user (i.e., the *reference query*) and offers recommendation (i.e., the *target query*) by adding/changing filtering and group-by attributes etc. To rank the recommendations, SeeDB recommends to the user the most interesting target queries based on the deviations from the given reference query. SeeDB assumes a larger deviation indicates a more interesting target query. Figure 4.1a and Figure 4.1b show examples of "*interesting*" and "*uninteresting*" target views. Furthermore, SeeDB truncates uninteresting visualizations if the deviation value is below a certain threshold.

Unfortunately, the deviation in SeeDB may just result from random noise, and thus carries no statistical significance. In [231], the authors report that the discovery rate for interesting visualizations with SeeDB is three times higher than for manual exploration tools such as Tableau [98] or Vizdom [48]. This increase of efficiency is troubling because the false discoveries also increase in an uncontrolled manner.

False Discoveries on Random Data

As a first step to show how visual recommendation systems such as SeeDB [231] suffer from false discoveries, we use a probabilistic model to analyze how likely it is that SeeDB finds a large deviation from the reference view on random data without any real correlations. We focus on SeeDB's capability of adding and changing a single filtering condition to the reference query. We ignore more advanced variations (e.g. multiple filter conditions, adding group-by's, etc.), as all of these would further increase the chance of false discoveries. For simplicity our model makes the following assumptions: (a) the aggregate function is `SUM`, the aggregate column has zero variance and the group-by column has uniformly distributed binary values; (b) the filter column is a sequence of Bernoulli trials; (c) the selectivity of attribute values on the filter column is drawn from a multinomial distribution; (d) all columns are independent; (e) we used the same deviation distance as in [231] for the recommendation threshold.

Figure 4.2 shows the risk of the SeeDB model making recommendations based on the random effects. The first simulation in Figure 4.2a varies the number of unique values in the filtering attribute



(a) Reference View **(b) Recommended View 1**

(c) Recommended View 2 **(d) Recommended View 3**

Figure 4.3: An example of SeeDB [231] on survey data.

(called filter cardinality), which corresponds to the number of target queries per reference query. The support size (i.e., number of selected tuples) of each target query is kept constant at 100. Given higher filter cardinality, more target queries are compared against the reference query, and thus the risk of false discovery increases. The second simulation in Figure 4.2b uses 1000 records, keeps the filter cardinality constant at 6 but varies the selectivity of the predicate and with it the support size. The figure shows — not surprisingly — that the lower the selectivity, the higher the chance of a false discovery because of the reduced support size.

False Discoveries on Survey Data

As a second step to verify what spurious recommendations SeeDB would make, we collected 104 answers for 69 (mostly unrelated) multiple-choice and 17 fill-in-the-blank questions on Amazon Mechanical Turk [112]. Questions range from *Who is Mike Stonebraker?* to *What is your eye color?* and *Have you ever been in a Sauna?*. Each answer is treated as an attribute.

We implemented the SeeDB recommendation algorithm, and used simple queries with one aggregated and one group-by attribute but without any filtering condition as the reference views. The deviation threshold was set according to the example in [231]'s Figure 1. As a result, our SeeDB implementation generated $2,078,608$ target views (i.e., potential recommendations) based on $9,996$ reference views, among which a stunning $708,109$ were recommended, though many of which are

statistically insignificant.

Figures 4.3a-4.3d show example spurious recommendations. Suppose the user analyzes the preference of Potato Chips (Cheddar vs. Sour Cream) based on the Workspace Preference using the reference view in Figure 4.3a, which is already a questionable finding on its own. Still SeeDB recommends three of the top-ranked target views shown in Figure 4.3b-4.3d (ranked by the deviation beyond the threshold as in [231]'s example), which are even more questionable and do not hold up in our statistical test. For instance, the recommendation in Figure 4.3c shows that the disbelief in aliens reverts the trend compared to the reference view, though the correlation between disbelief in aliens and preference of potato chips is insignificant (p-value of 0.59). On the other hand, SeeDB also recommends views based on statistically significant yet questionable correlations, such as the correlation between Saunas and Stonebraker from the abstract, which even passes our post mortem statistical test (p-value of 0.036). *Thus, even if the user would perform a statistical test after seeing the visualization, she might wrongly assume that the insight is significant as she would certainly never consider the risk the visualization recommendation system introduced by searching for an "interesting" visualization.*

4.2.3 Automatic Correlation Finders

As a last class of system, we analyze recent recommendation engines, which not just suggest visualizations but try to automatically find insights through automatic hypothesis testing. One of such systems is Data Polygamy [44], which searches for statistically significant correlations in temporal-spatial datasets. Such correlations may exist at certain time or location. For example, the wind speed may not correlate with the number of taxi trips during the year, but it may when the hurricane strikes. Data Polygamy first identifies extreme data points, then uses the F1 score to measure the relationship strength, and performs Monte Carlo permutation test to determine the statistical significance given a predefined significance level [44].

Unfortunately, Data Polygamy ignores the problem of multiple comparisons and therefore its method is only sensible for a *single* compared relationship. Suppose there are two datasets of 5 attributes each, resulting in 25 pairwise relationships to test. With a significance level of 0.05, on average at least about one such relationships would pass the significance criteria even on random input. However, data variety, on the other hand, is increasing quickly. The NYC Urban data collection has 228 features on weather monitoring, and over 1,300 data sets in the span of two years have been collected by the government agencies in NYC [44] [83]. Thus recommendation systems without controlling for multiple comparisons are not suitable for real-world datasets.

We downloaded the code of Data Polygamy and studied the number of false discoveries over random data with randomly introduced extreme data points, as summarized in Figure 4.4. Each extreme data point was sampled independently with 20% probability from a distinct uniform distribution than the normal data. With 100 records and 11 attributes per dataset, Data Polygamy found a total of 43 "bogus" relationships in 50 independent trials. Thus, without considering the risk of multiple comparisons, *Data "Polygamy can be bad for you"; it is literally an automatic p-hacking*

# Records	100
# Attributes	11
# Datasets	2
Extreme data prob.	20%

(a) Random input

# Trials	50
Significance level	0.05
# Incorrect rejection	43
# Correct acceptance	7

(b) False discoveries

Figure 4.4: Data Polygamy [44] on random extreme points.

system.

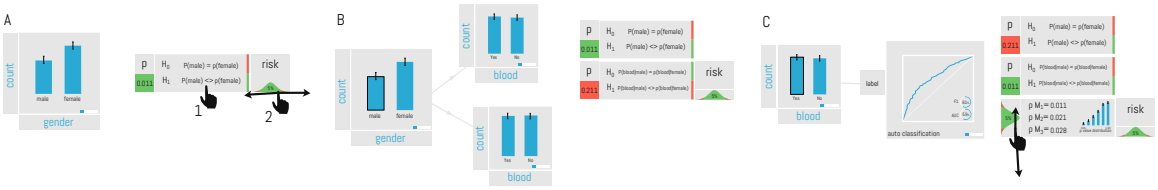


Figure 4.5: Storyboard of how a "risk controller" could look like.

4.2.4 Automatic Model Finding

Finally, systems for automatic model building and tuning in data mining or machine learning (e.g. MLBase [135]) are also victim of the risk of multiple comparisons. To demonstrate the complexity of this problem, suppose that we evaluate a sequence of 20 possible models $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_{20}$ for our observed data. We test each model using cross validation on different holdout sets, and accept a model if its estimation (prediction) error satisfies our requirement with significance level ≤ 0.05 (i.e., the probability that the model achieved that smaller level of estimation error on a random data is bounded by 0.05). However, this also implies that at least one such model on average would pass our criteria even on random data.

4.3 QUDE: A System to Quantify the Uncertainty in Data Exploration

As discussed in the previous section, the multi-hypothesis pitfall is a core problem affecting many recent systems for interactive data exploration, recommendations for visualizations and insights, as well as, automatic model building. With QUDE (pronounced "*cute*") we are building the first system to automatically **Q**uantify the **U**ncertainty in **D**ata **E**xploration. QUDE is part of Brown's Interactive Data Exploration Stack (BIDES) and consist of a risk assessment engine as well as a user facing component integrated into Vizdom, BIDES' user interface. While the main focus of QUDE is on the control of the risk of false discoveries due to the testing of multiple hypotheses, QUDE will also be able to detect other risk factors as explained at the end of this section.

4.3.1 Controlling the Exploration Risk

When a user is exploring a larger number of hypotheses based on the data, either explicitly, indirectly through visualizations, or automatically through recommendation engines, there is a growing risk of flagging a random (i.e., non "*statistically significant*") fluctuation in the data as a significant discovery. Any sustainable data exploration system should therefore effectively control the risk of such "*false discoveries*".

Multi-Hypothesis Evaluation

The risk of false discovery is known as the problem of multiple comparisons, or multi-hypothesis evaluation Two main fundamental challenges arise when attempting to automatically quantify the risk: (1) the traditional techniques do either not scale well with the number of hypothesis or can not be used in an interactive environment and (2) in many cases it is not clear which hypothesis is currently being tested through a visualization by the user (i.e., the "*user intent*"). In the following, we describe various multiple-hypothesis control techniques and how well they work to address the first challenge, whereas in Section 4.3.1 we discuss how we plan to address the user intent challenge.

Family Wise Error Rate (FWER): Traditionally, frequentist methods for multiple hypothesis testing focus on correcting for modest numbers of comparisons. A natural generalization of the significance level to multi-hypothesis testing is the *Family Wise Error Rate (FWER)*, which is the probability of incurring at least one Type I error (i.e., false positive: the null hypothesis is true, but is rejected) in any of the individual tests. The Bonferroni correction [28] was proposed to control FWER for m hypothesis tests at an upper bound α. The Bonferroni correction tests each null hypothesis with significance level α/m. However, this method is too conservative in that the power of the test is too low (i.e., the accepted significance level becomes extremely small) when m is large, resulting in many false negatives. Several methods have been proposed to improve the average power of the Bonferroni method for small to modest m; but for large number of hypotheses, all of these techniques lead to tests with low power. A review of these techniques is given in [207].

False Discovery Rate (FDR): The *False Discovery Rate (FDR)* was introduced by Benjamini and Hochberg [15] as an alternative and less conservative approach to control errors in multiple hypothesis tests. Let V be the number of Type I errors in the individual tests, and let R be the total number of null hypotheses rejected by the multiple test. FDR is defined as the expected ratio of erroneous rejections among all rejections, namely FDR $= E[V/R]$, with $V/R = 0$ when $R = 0$. Designing a statistical test that controls FDR is not simple as the FDR is a function of two random variables that depend both on the set of null hypotheses and the set of alternative hypotheses. Building on the work in [15], Benjamini and Yekutieli [17] developed a general technique for controlling the FDR in multi-hypothesis tests. Furthermore, for entirely random data, FDR controls the same error rate as FWER. That is, FDR and FWER result in the same expected number of mistakes over random data. This property makes FDR easy to explain to users (though, admittedly understanding the differences between FWER and FDR is harder in the presence of real correlations). Most importantly, recent work by Foster and Stine [80] allow to incrementally run

the hypothesis tests and thus provide a starting point for controlling the risk in interactive data exploration.

Uniform Convergence and (Structural) Risk Minimization: The uniform convergence paradigm is often used to control the risk in model selection (i.e., machine learning) and is a great candidate technique to control the risk for automatic recommendations (e.g., visualizations as in SeeDB or correlations as in Data Polygamy) and model tuning (e.g., as in MLBase). In this approach the complexity of the predefined class of all possible hypotheses under consideration is analyzed, and based on this complexity it is possible to compute an upper bound to the sample size that is sufficiently large to simultaneously evaluate the expected error of all hypotheses in the class. The approach was first proposed by [229] as the theoretical foundation for statistical learning, but it has been shown to provide practical solutions to some important data analysis problems [191, 192, 193]. The method of structural risk minimization prioritizes less complex models by assigning weights to different hypothesis classes (model) corresponding to the user's preferences. We say that a set of functions has the uniform convergence property if we can use one finite sample to estimate the expectation of all the functions in the set, with a uniform bound on the gap between the empirical mean and the true expectation that hold simultaneously over all the functions in the set. Formally, a set of functions $\mathcal{F}$ has the *uniform convergence* property with respect to a domain Z if there is a function $m(\epsilon, \delta)$ such that for any $\epsilon, \delta > 0$, $m(\epsilon, \delta) < \infty$, and for any distribution D on Z, a sample $z_1, \ldots, z_m$ of size $m = m(\epsilon, \delta)$, drawn independently & identically distributed (i.i.d.) from D satisfies

$$\Pr(\sup_{f \in \mathcal{F}} |\frac{1}{m} \sum_{i=1}^{m} f(z_i) - E_{\mathcal{D}}[f]| \leq \epsilon) \geq 1 - \delta.$$

Other Approaches: An alternative to the previous methods is a *hold-out dataset*, i.e., randomly dividing the dataset into an exploration D_1 and a validation D_2 dataset [245]. While a feasible approach for very large datasets, it can be shown that this approach significantly lowers the power (i.e., the chance of finding a real insight) especially for smaller datasets or subsets of the data. For example, if the user tries to find what distinguishes her top 100 customers from the rest, this method leads to significantly more false negatives. *Permutation tests* [89] can also be used to achieve similar control for multi-hypothesis testing. While well suited for small datasets, permutation tests on large datasets are usually too computationally intensive to be executed interactively.

Automatic Risk Control in QUDE

Our goal is to provide the user with accurate risk estimates for different types of interactions in data exploration. Our work as in QUDE is built upon the line of research on efficient FDR bounds for massive data explorations and the application of uniform convergence [130, 191, 192]. The core idea of QUDE is to assume a standard null hypothesis for any exploration the user performs, while allowing the user to customize the null hypothesis with her domain-specific prior knowledge. We are therefore currently developing a heuristic based on our user study to determine the intention of the user. For example, when the user observes that there are an equal number of men and women in the database, but the distribution of men and women is unbalanced when considering only individuals

with income over 50k, QUDE assumes the user executes a test to evaluate the significance of this difference. At the same time, QUDE's interface presents this standard hypothesis to the user and allows to overwrite the null hypothesis, if the user chooses to adjust the null hypothesis. Our assumption is that it is not crucial to be always correct with the default hypothesis, but rather that the default hypothesis is used to actively make the user aware that the insight he might have gotten should be tested for significance and to actively seek the user's feedback. Then, as the user continues exploring the data set, the system continuously calculates the risk of false discovery based on the FDR method. If a shown difference is not significant, the user is automatically warned about it. Furthermore, if the user trains a model or uses a visualization recommendation, we apply the same principle by providing an upper bound on the expected number of false recommendations using Uniform Convergence and (Structural) Risk Minimization.

While with our current QUDE we are already able to quantify the risk factors for simple workflows, we also discovered several challenges which we still need to address. Most importantly, for each interaction we need to identify the appropriate null hypothesis to compute the corresponding p-value. Besides using a default hypothesis and allowing the user to overwrite it, we plan to explore alternative approaches such as (1) asking the user what information she is looking for; (2) learning from past action of users on similar data; and (3) apply a precalculated upper bound on the number of different hypothesis answered by a given chart (otherwise a histogram with 20 bars would create over 190 tests).

Similarly, we found that current FDR methods are still not well suited for the iterative process of data exploration. The standard FDR control methods evaluate all the null hypothesis and select a subset to reject. In the iterative process instead we want a stopping criteria that depends only on the actual hypothesis evaluated by the user. There has been some preliminary work towards this direction [91, 80]. Our work builds upon [80] to provide incremental and interactive risk control of data exploration.

A last challenge is to identify classes of hypothesis (e.g., for recommending certain visualization) for which we can compute practical and efficient bounds on their sample complexity using structural risk techniques, as in our recent work [191, 192, 193] for controlling the risk in frequent itemsets mining. The idea is to group sets of primitive hypothesis into classes, and evaluate the total error with respect of the number of different classes used by the users.

QUDE User Interface

Integrating the user feedback in the data exploration process is a key factor towards avoiding false discoveries: (1) the system needs to understand the user intents to better quantify the risk and (2) the system needs to adequately warn the user about potential risk factors so that the user understands the risk of her actions. The core idea of our approach is to use an automatically derived default null hypothesis in order to obtain user feedback and (potentially) as a pessimistic lower bound. Figure 4.5 shows a storyboard of QUDE's way to to convey the risk factors to the users. In this example, **(A)** Eve drags out a histogram and the system immediately displays a"*risk controller*" on the left-hand

side where the results of a default hypothesis test for this histogram are displayed: rejected null hypotheses highlighted in green, accepted null hypotheses highlighted in red. It confirms what Eve intuitively observed from the visualization: there seems to be a significant difference between the number of female and male patients in this dataset. (1) Tapping on this default hypothesis allows Eve to manual adjust and correct what's being tested (e.g., she might want to change from two-sided to one-sided). (2) Eve can also use the "*risk*" slider to change the amount of false discoveries she is willing to accept. **(B)** Eve adds more histograms and connects them to the previous one and the system again automatically runs the appropriate hypothesis tests. Again this confirms what Eve sees visually, the chances of having a blood disease does not seem to be dependent on the gender of a person. **(C)** Afterwards, Eve decides to build a classifier that predicts if someone has a blood disease. Our system's risk controller automatically adds an entry in the risk list, which allows Eve to define a false discovery rate budget for the model search and which is then, for example, controlled with the structural risk minimization technique. Furthermore, Eve can see the p-values of the top 3 models as well as the distribution of p-values for all models that have been tested.

4.3.2 Detecting Common Statistical Pitfalls

While our main goal of QUDE is to control the multiple hypothesis error, we are also planning to implement tools to detect other common statistical errors/mistakes. In the following, we discuss some of these pitfalls and initial ideas of how to make users aware of them.

Simpson's Paradox

A well-known phenomenon in statistics is the Simpson's Paradox [128] in which a trend reverts when splitting a data set into multiple subgroups. One of the most famous examples is the gender bias among graduate school admissions to University of California, Berkeley. The overall admission figures of 1973 showed that men were more likely to be admitted than women. However, when looking at the largest departments the trend actually reverted for the majority of departments.

For QUDE, we therefore integrated algorithms that detect a Simpson's Paradox online as the user explores a data set. Figure 4.6 shows a storyboard of how a exploration session of QUDE looks like: Eve already has filtered down her dataset to look at patients from a particular demographic and with certain types of blood testing result values. Eve is now interested to see percentages of such patients that have blood diseases grouped by age groups (kids and adults). From looking at the histogram in **(A)** it seems like kids are more prone for these types of diseases. Afterwards, Eve however notices that the system displays a warning (yellow box). **(B)** By dragging out the warning, the system presents a set of visualizations showing that when accounting for the lurking variable "*digestive disease*" the trend reverses (i.e., kids are less prone for blood diseases for both, with digestive diseases and without).

Testing a dataset for the Simpson's Paradox is quite challenging as it requires to test many different attribute combinations while controlling the risk of finding a Simpson's Paradox by chance. For dealing with both these issues, we are currently developing techniques based on novel index

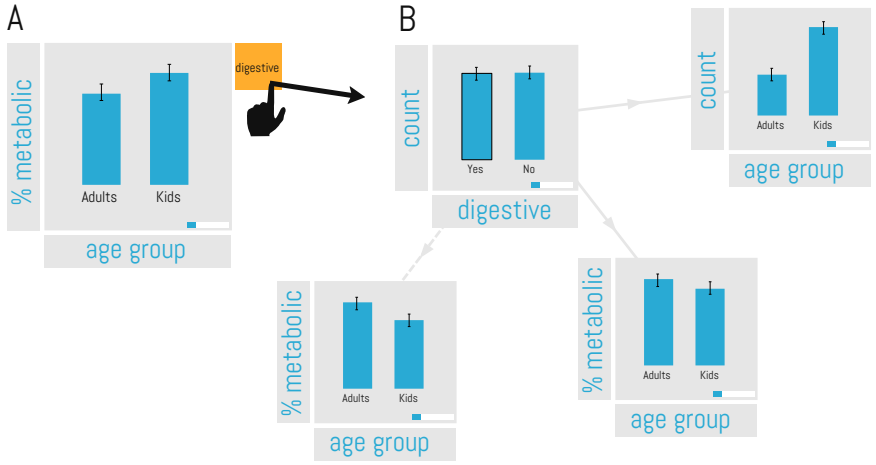


Figure 4.6: Storyboard of a Simpson's Paradox Warning

structures and the FDR method.

Other Hypothesis Testing Issues

So far we only focused our attention on Type I error on homogeneous data. However, the Type II error can be as important and (if possible) should be quantified as well. Furthermore, many test can fail on non-homogeneous data and ideally, QUDE should warn the user in those cases or suggest different types of tests. In the context of ML, similar issues, such as the *Base Rate Fallacy* or the *Imbalance of Labels*, can significantly disturb the result if the system/user does not control for it. Similar, *Pseudoreplication*, a very common problem with data collected in life-sciences, may lead to detect a false statistical significance. While it is not possible to automatically detect all of these issues as they might depend on the semantics of the data itself, it might be possible to derive some of the issues automatically based on the schema, analyzing the general data statistics (e.g., for the base rate fallacy), or testing for correlations.

4.3.3 Data Quality Issues

Finally, many issues can also arise from dirty data in form of missing, duplicate, or inconsistent records. Unfortunately, all data cleaning techniques are expensive, in both time and money (e.g., to pay humans to correct errors) [155], are often not adequate for interactive data exploration, and in almost all cases it is unrealistic to assume that a data-set is perfectly cleaned upfront.

Estimating Remaining Errors

For data exploration it is often more important to understand how many errors a dataset contains and whether these errors are systematic or random rather than trying to correct all errors. This would allows an analyst greater insight on the both the data set and the potential risk factors. While a simple question at first, it is actually extremely challenging to define data quality without knowing the ground truth [182, 41, 70, 72, 125].

A naïve approach would be to "perfectly" clean a small sample as the gold-standard data (as in [235]) and extrapolate the insight of the cleaning process to the entire data set. For example, if we found 10 new errors in a sample of 1000 records out of 1M records, we would assume that the total data set contains 10000 total errors. A very small sample of cleaned data may however not be representative of the entire dataset. Further, how can the analyst determinate whether the sample itself is actually perfectly clean without a quality metric? As part of QUDE, we have therefore started to develop alternative methods to the naïve estimator, which consider the entire data set – albeit when it is imperfectly cleaned. Our core insight is that almost any cleaning technique has diminishing returns, that is, every additional error is more difficult to detect.

Automatically Repairing Errors

As a second step, we are exploring the possibility of using our insight for the remaining errors in order to automatically correct query answers and models. For example, in previous work [46] we developed and analyzed techniques to estimate the impact of the missing data (a.k.a., "*unknown unknowns*") on simple aggregate queries. The key idea is that the overlap between different data sources enables us to estimate the number and values of the missing data items. Our main techniques are parameter-free and do not assume prior knowledge about the distribution. For future work, we plan to develop similar techniques to correct for a broader range of analytical queries and to learn repair procedures for other errors based on the history of user interactions as well as data characteristics that can either be applied automatically or simply suggested to the user during exploration.

4.3.4 Current State of QUDE

QUDE currently performs risk evaluations using default hypotheses for simple workflows. We implemented QUDE as part of Vizdom and currently evaluate different types of user feedback and warnings as outlined in our storyboards. We also already developed techniques for quantifying the impact of the unknown unknowns [46], currently evaluate the data quality metrics with several real world use cases, and developed new approximation algorithms for detecting the Simpson's Paradox. However, by no means do we claim that we solved all open issues. Rather, we believe that QUDE is just a first step towards a potential new research area focused on the control of various risk factors in all type of analytics.

4.4 Conclusion

We demonstrated that recent recommendation systems such as SeeDB [231] and Data Polygamy [44] significantly increase the risk of making false discoveries. We further presented our vision and initial ideas for QUDE, a system for automatically controlling the various risk factors in interactive data exploration, automatic model building, and insight recommendation. The goal of this work is, on one hand, to point out that the risk of false discoveries can not be ignored, and on the other, to

outline possible solutions with the hope to foster a new line of research around tools for sustainable insights.

Chapter 5

Controlling False Discoveries During Interactive Data Exploration[1]

Recent tools for interactive data exploration significantly increase the chance that users make false discoveries. They allow users to (visually) examine many hypotheses and make inference with simple interactions, and thus incur the issue commonly known in statistics as the "*multiple hypothesis testing error*." In this Chapter, we propose a solution to integrate the control of multiple hypothesis testing into interactive data exploration systems. A key insight is that existing methods for controlling the False Discovery Rate (i.e., FDR) are not directly applicable to interactive data exploration. We therefore discuss a set of new control procedures that are better suited for this task and integrate them in our system, QUDE. Via extensive experiments on both real-world and synthetic data sets we demonstrate how QUDE can help experts and novice users alike to efficiently control false discoveries.

5.1 Introduction

"Beer is good for you: study finds that suds contain anti-viral powers" [DailyNews 10/12]. "Secret to winning a Nobel Prize? Eat more chocolate" [Time, 10/12]. "Scientists find the secret of longer life for men (the bad news: Castration is the key)" [Daily Mail UK, 09/12]. "A new study shows that drinking a glass of wine is just as good as spending an hour at the gym" [Fox News, 02/15].

In recent years there has been an explosion of data-driven discoveries as the ones cited above. While some of these are likely to be legitimate, there is an increasing concern that a large amount of current published research findings may actually be false [113].

In this Chapter, we make the case that the rise of interactive data exploration (IDE) tools has the potential to worsen this situation further. Commercial systems such as *Tableau* or research prototypes such as *Vizdom* [48], *Dice* [122] or *imMens* [149], aim to empower domain experts and

[1]The results presented in this chapter were presented in the 38th ACM SIGMOD International Conference on Management of Data (SIGMOD/PODS 2017). This is joint work with Professor Carsten Binnig, Professor Tim Kraska, Professor Eli Upfal, Emanuel Zgraggen and Zheguang Zhao.

novice users alike to discover complex relationships and trends from data in an entirely visual manner. Unfortunately these systems often ignore even the most basic statistical rules. We recently performed a user study and asked people to explore the U.S. Census data [143] using such an interactive data exploration tool.[2] Within minutes, all participants were able to derive multiple insights, such as "*people with a Ph.D. earn more than people with a lower educational degree.*" However, none of the participants used a statistical procedure to determinate whether the visually observable differences in the histogram is actually meaningful (i.e., "*statistically significant*"). Further, none of the users considered that the data exploration consisting of multiple attempts to find interesting insights would considerably increase the risk of observing seemingly significant results by chance.

This problem is well known in the statistics community and referred to as the "*multiple comparisons problem*" or "*multiple hypothesis error*" and it states that the more hypothesis tests an analysts performs, the higher is the chance that apparently significant phenomenon (i.e., a "*discovery*") is actually observed just by chance. Let us assume an analyst tests 100 potential correlations each with significance level $\alpha = 0.05$. Assume further that 10 of the correlation are in fact true, and that our test has a statistical power (i.e, the likelihood to discover a real correlation) of 0.8; all very common values for a statistical testing. In this setting, the user would find ≈ 13 correlations of which 5 ($\approx 40\%$) are "bogus."

One way to lower the probability of incurring any false discovery among all the tests — known as the *family-wise error rate* (FWER) — is to use a multiple hypothesis correction procedure such as the Bonferroni correction [28]. Unfortunately, a well-known drawback of the Bonferroni correction and of may others FWER control procedures is that they lead to a significant decrease of the statistical power; the chance to detect truly significant phenomenons. Furthermore, in the context of interactive data exploration we need to cope with the ulterior complication that the hypotheses are generally unknown upfront, hence rendering any static procedures, such as the Bonferroni, correction unsuitable.

Another crucial challenge in modeling data exploration lies in the fundamental question of "*what should be considered as a hypothesis test when users interactively explore the data.*" Suppose a user sees a visualization, which shows no difference in salaries between men and women based on their education, and then decides based on this insight to look at salary differences between married men and women. Should the first, the second or both visualization be considered as a hypothesis test? The answer in most cases is *both*, as the analyst probably implicitly made a conclusion based on the first visualization, which then led to her next exploration step.

However, if she considers this visualization just as a descriptive statistic of the current dataset, and makes no inference based on it (i.e. it did not influence the decision process and no inference is made by it), then it should not be considered as a hypothesis test. The difference is subtle and usually very hard to understand for non-expert users, while it might have a profound impact on the false discovery a user makes.

[2] The results were gathered by analyzing (in retrospective) the think-aloud protocols of various user studies including the study described in [242] and entailed in total over 50 participants.

In this Chapter, we present the first end-to-end system, QUDE (short for Quantifying Uncertainty in Data Exploration), to automatically control the risk of false discovery for visual, interactive data exploration. We propose a user interface and an initial set of meaningful default hypotheses, i.e., the "*null hypotheses*," to control the ratio of false discoveries without interrupting the exploration process (Section 5.4). In Section 5.5 we discuss the control procedures based on the family-wise error rate (FWER), and explain why they are too pessimistic for interactive data exploration, and why the more modern criterion of controlling the false discovery rate (FDR) is better suited. The challenge of FDR, however, is that the standard techniques, such as the Benjamini-Hochberg procedure [15], are not incremental and require computing the p-values of all the hypotheses a priori before determining which hypotheses are significant. This clearly constitutes a problem for interactive data exploration where hypotheses are created incrementally. The recent *α-investing* technique [80] proposes an incremental procedure to control a variant of FDR, the marginal FDR (mFDR), together with a special-case investing strategy, Best Foot Forward. However, our experiment shows that this technique does not work well in interactive data exploration, because it was designed for a rather limited scenario where hypotheses are clustered. Thus, in Section 5 we propose new α-investing techniques that are designed specifically for interactive data exploration. Finally, we implement these ideas in QUDE and demonstrate how the system controls false discovery for experts and novice users alike using generated and real-world data.

It should be noted, that multiple hypothesis control is perhaps one of the most difficult and thorny issues facing modern statistics. Despite extensive work that explored a variety of approaches and techniques, there is *no single perfect solution*, thus neither do we claim one in this work. Instead, our main contribution is to combine and extend recent advances in statistics to build the first functional system which automatically assists the users in recognizing and controlling the false discovery during interactive data exploration. In summary, we make the following detailed contributions:

- We establish a connection between data visualizations and multiple hypothesis testing. We point out the risk of incurring a high number of false discoveries unless visual exploration systems employ corrections for multiple hypothesis testing.
- We propose a model of setting default (i.e., null) hypotheses during the interactive data exploration.
- We present QUDE, a novel system which automatically controls the multiple hypothesis error in visual data exploration.
- We discuss inadequacies of the existing multiple hypothesis control methods for interactive data exploration;
- Based on these observations, we develop new α-investing rules to control the marginalized false discovery rate (mFDR) that are designed specifically for interactive data exploration.
- We use Markov chain simulation and real-world datasets and workflows to show that our

methods indeed achieve automatic control of false discovery and have significantly higher power than other techniques.

Chapter organization and notation

The presentation of the contents of this Chapter is structured as follows: we discuss contributions in the literature related to ours in in Section 5.2, in Section 5.3 we discuss, by means of an example, why some visualizations should be considered hypothesis tests and what are the main challenges encountered when testing hypotheses for the IDE setting. In Section 5.4 we present QUDE's user interface and discuss how to automatically track hypotheses and how to integrate the user feedback into tracking the hypothesis. In Section 5.5 we discuss multiple hypothesis testing techniques known in literature and show how well they fit in the IDE setting. In Section 5.6 we then propose new multiple hypothesis testing procedures for IDE based on the α-investing procedure. Afterwards, in Section 5.9, we present the result of our experimental evaluation using both real-world and synthetic data. Finally, in Section 5.11, we discuss related work and present our conclusions. Table 5.1 summarizes the important symbols and notations used in this chapter.

H	The set $\{H_1, \ldots, H_m\}$ of null hypotheses.
$\mathcal{H}$	The set $\{\mathcal{H}_1, , \ldots, \mathcal{H}_m\}$ of corresponding alternative hypotheses.
R	The number of null hypotheses rejected by the testing procedure (i.e., the discoveries).
V	The number of erroneously rejected null hypotheses (i.e., false discoveries, false positives, Type I errors).
S	The number of correctly rejected null hypotheses (i.e., true discoveries, true positives,).
$R(j)$	The number of discoveries after j tested hypotheses.
$V(j)$	The number of false discoveries after j tested hypotheses.
$S(j)$	The number of false discoveries after j tested hypotheses.
m	The number of tested hypotheses.
p_j	The p-valuecorresponding to the null hypothesis H_j.
$W(0)$	Initial wealth for the α-investing procedures.
$W(j)$	Wealth of the α-investing procedures after j tests.
α	Significance level for the test with $\alpha \in (0, 1)$.
η	Bias in the denominator for $mFDR_\eta$.

Table 5.1: Notation Reference

5.2 Related Work

There has been surprisingly little work in controlling the number of false discoveries during data exploration even. This is especially astonishing as the same type of false discovery can also happen with traditional analytical SQL-queries. To our knowledge this is the first work to achieve an automatic control in tracking the user steps.

Most related to this work are all the various statistical methods for significance testing and multiple hypotheses control. Early works tried to improve the power of the Family Wide Error Rate using adaptive Bonferroni procedures such as Sĭdák [212], Holm [106], Hochberg [103], and Simes [213]. However, all these methods lack power in large scale multi-comparison tests.

The alternative False Discovery Rate measure was first proposed by Benjamini and Hochberg [15], and soon became the statistical criteria of choice in the statical literature and in large scale data exploration analysis for genomic data [156].

In the original FDR method, *all* hypotheses have to be collected and sorted by their p-values before determining the significance of each test. The Sequential FDR procedure [91] does not require the sorting of all the hypotheses, but still require to calculate *all* the tests before their corresponding significance can finalize. These procedures cannot determine the final significance of each test incrementally and hence are not applicable to interactive data exploration. The interactive data exploration motivated the study of interactive and adaptive techniques, such as α-investing [80], which can be applied in scenarios where hypotheses arrive sequentially and the testing procedure needs to decide "on the fly" whether to accept or reject each of the hypotheses before testing the next one, while maintaining a bound on the FDR.

Depending on the observed order of hypotheses, Sequential FDR can overturn previously accepted hypotheses into rejections based on the subsequent hypotheses.

α-investing procedure also has revisiting policies that can potentially overturn previous decisions. The implication is that these procedures are incremental but non-interactive, because they require observing all the hypotheses before finalizing the decisions. However, it is often infeasible to obtain all the possible hypotheses a priori. Therefore our work concerns α-investing procedure with policies that are both incremental and interactive. In addition, none of the work addresses the issue on how to automatically integrate these techniques as part of an data exploration tool.

In a recent paper [60], Dwork et al. introduce a new adaptive testing procedure for streams of hypotheses which exploits concepts and techniques from differential privacy. Although this technique can reliably test up to m adaptively chosen hypotheses it has also several practical drawbacks: the computationally efficient version of the procedure requires the size of the available sample to be proportional to $\sqrt{m}$ and knowledge of the amount of hypotheses being tested is required.

In [24] Blum and Hardt presented "*the Ladder*", an algorithmic test procedure that, given a training dataset, reliably evaluates the quality of different versions a model by adaptively tuning the parameters. While this approach does not address the general issue raised in [15, 91, 80, 60], it shows good performance in the practical context of parameter tuning for machine learning.

5.3 A Motivational Example

To motivate the various aspects of multi-hypothesis control during data exploration we present a use case that is inspired by Vizdom [48]. Similar workflows however can be achieved with other systems like Tableau [98], imMens [149] or Dice [122].

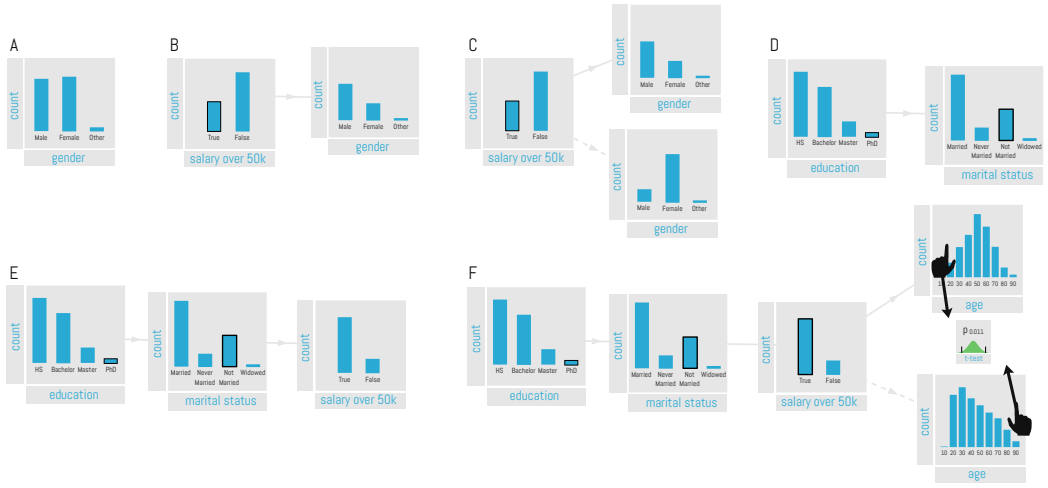


Figure 5.1: An example Interactive Data Exploration Session

Suppose Eve is a researcher at a non-profit organization and is working on a project relevant to a specific country. She obtained a new dataset containing census information and is interested in getting an overview of this data and extracting new insights.

She first considers the "*gender*" attribute and observes that the dataset contains the same number of records for men and women (Figure 5.1 A). She then moves on to a second visualization, displaying the distribution of people who earn above or below $50k a year. Eve links the two charts so that selections in the "*salary*" visualization filter the "*gender*" attribute. She notices that for salaries above $50k, the "*gender*" distribution is skewed towards men, and infers that men have higher salaries than women (B). After creating a third visualization for "*gender*" with, conversely, salaries lower than $50k (dashed line indicates inversion of selection), she confirms her finding "*Women are predominately earning less than $50k*" (C).

Eve now wants to understand what influences salaries and creates a chain of visualizations for people who have PhD degrees and are not married (D). Extending this chain using "*salary*" appears to suggest that this sub-population contains many high-earners (E). By selecting the high-earners and extending the chain with two "*age*" visualizations, she compares the age distribution of unmarried PhDs earning more than $50k to those making less. To verify that the observed visual difference is actually *statistically significant* she performs a t-test by dragging the two charts close to each other (F).

While the example contains only one hypothesis test *explicitly* initiated by the user, we argue that without accounting for other *implicit hypothesis tests* there is a significant increase of risk that the user may observe a false phenomenon during similar scenarios of data exploration. This opens up new important questions: why and when should visualizations be considered statistical hypothesis tests? How should these tests be formulated?

5.3.1 Hypothesis Testing

In this Chapter, we focus on the widely used frequentist approach. That is, to determine whether the relationship between two observations formalized as a "*research hypothesis*" or "*alternative hypothesis*' $\mathcal{H}$ is statistically relevant (i.e., not a product of data noise) we analyze its corresponding "*null hypothesis*" H which states no such relationship. The *testing procedure* will then calculate the p-value, which denotes the probability of observing an outcome at least as extreme as the one that was actually observed in the data, under the assumption that the null hypothesis H is true. If the p-valueassociated to the null hypothesis H is less than or equal to a priori chosen *significance level* α (commonly 0.05 or 0.01), the test suggests that the observed data is *inconsistent* with the null hypothesis which must thus be rejected. Respectively, if the p-valueis larger than the significance level, the null hypothesis H is accepted. This procedure guarantees for a single test, that the probability of a "*false discovery*" (also known as "*false positive*" or "*Type I error*") – wrongly rejecting the null hypothesis of no effect – is at most α. This does not imply that the alternative hypothesis is true; it just states that the observed data has the likelihood of $p \leq \alpha$ if the null hypothesis is true. In contrast, the *statistical power* is the probability that the test correctly rejects the null hypothesis H.

While the frequentist approach to hypothesis testing has been criticized [115, 167] and there has been work in developing alternative approaches, such as Bayesian tests [19], it is still widely used in practice and we consider it a good first choice to build a system which automatically controls the multiple hypothesis error as it has two advantages: (1) Novice users are more likely to have experience with standard hypothesis testing than the more demanding Bayesian testing paradigm. (2) The frequentist inference approach does not require to set a sometimes hard-to-determine *prior* as it is the case with Bayesian tests.

5.3.2 Visualizations as Hypotheses

A visualization per-se shows a descriptive statistic (e.g., the count of women or the count of men) of the dataset and is not a hypothesis. It is reasonable to assume that in step A of Figure 5.1 the user just looks at the gender distribution and simply acknowledges that the census surveys roughly the same amount of women and men. However, it becomes a hypothesis if the user draws a conclusion/inference based on the information. For example, if the user assumed that there should be more men than women in the data and therefore considering the fact that there is an equal amount as an insight. The notion of a visualization being considered as a hypothesis becomes even clearer in step (B) and (C) of the example workflow. When looking at the visualization in (B) in isolation, it just depicts a descriptive statistic. But once the user makes any inference and/or bases further exploration on an insight extracted from this visualisation, then it should be considered an hypothesis. We believe that user inference in data exploration is ubiquitous and important. First, the human analytical reasoning and sense-making process is inherently non-linear [183, 211]. The future actions are influenced by new knowledge the user discovered in previous observations.

Second, while susceptible to certain types of biases [56], the human visual system is highly optimized at registering differences in visual signals and detecting patterns [36]. An average user is very likely drawn to the changes between the gender distribution of step (A) and step (B) and might therefore infer that women earn less than men and potentially flag this as an interesting insight that deserves more investigation. This is illustrated in step (C) where the user now further drills down and visually compares the distribution of gender filtered by salary. We qualitatively confirmed this notion through a formative user study where we manually coded user-reported insights, following a think-aloud protocol similar to the one proposed in [92]. In this study we observed that users tend to pick up on even slight differences in visualizations and regard them as insights and users predominantly base future exploration paths on previously inferred insights.

We conclude two things: (1) most of the time users indeed treat visualizations as hypotheses, though there are exceptions, and (2) they often (wrongly) assume that what they see is statistical significant. The latter is particularly true if the users do not carefully check the axis on the actual count. For example, if a user starts to analyze the outliers of a billion record dataset and makes the conclusion that mainly uneducated whites are causing the outliers, the subset of the data she is referring to might be comparable small and the chance of randomness might be much higher. As part of visual data exploration tools, users often explore sub-populations, and while the original dataset might be large, the sub-population might be small. Thus, we argue that every visualization as part of a interactive data exploration tool should be treated as a hypothesis and that users should be informed about the significance of the insights they gain from the visualization. At the same time, a user should have the choice to declare a visualization as just descriptive.

5.3.3 Heuristics for Visualization Hypotheses

A core question remains: what should the hypothesis for a visualization be. Ideally, users would tell the system every single time what they are thinking so that the hypothesis is adjusted based on their assumed insight(s) they gain from the visualization. However, this is disruptive to any interactive data exploration session. We rather argue that the system should use a good default hypothesis, the user can modify (or even delete) if she so desires. For the purpose of this work, we mainly focus on histograms as shown in Figure 5.1 and acknowledge that there exist many other visualizations, which we consider as future work. We derived the following heuristics from two separate user studies where we observed over 50 participants using a IDE tool to explore various datasets.

1. *Every visualization without any filter conditions is not a hypothesis (e.g., step A in Figure 5.1) unless the user makes it one.* This is reasonable, as users usually first gain a general high-level impression of the data. Furthermore, in order to make it an hypothesis, the user would need to provide some prior knowledge/expectation, for example as discussed before, that he expected more men than women in the dataset.

2. *Every visualization with a filter condition is a hypothesis with the null hypothesis that the filter condition makes no difference compared to the distribution of the whole dataset.* For

example, in step B of Figure 5.1 the null hypothesis for the distribution of men vs. women given the high salary class of over \$50k would be that there is no difference compared to the equal distribution of men vs. women over the entire dataset (the visualization in step A). This is again a reasonable assumption as the distribution of an attribute given others is only interesting, if it shows some different effect compared to looking at the whole dataset.

3. *If two visualization with the same but some negated filter conditions are put next to each other, it is a test with the null hypothesis that there is no difference between the two visualized distributions, which supersedes the previous hypothesis.* This is the case in step C: given that the user looks explicitly at the distribution of males vs females given a salary over and under \$50k is a strong hint from the user, that he wants to compare these two distributions.

As with every heuristic it is important to note, that the heuristic can be wrong. Therefore it is extremely important to allow the user to overwrite the default hypothesis as well as delete default hypothesis if one really just acted as a descriptive statistic or was just generated as part to a bigger hypothesis test. Furthermore, there exist of course other potential null hypothesis. For example, in our workflow we assume by default that the user aims to compare distributions, which requires a χ^2-test. However, maybe in some scenarios comparing the means (i.e., a t-test) might be more appropriate as the default test. Yet, studying in detail what a good default null hypothesis is dependent on the data properties and domain, is beyond the scope of this work.

5.3.4 Heuristics Applied to the Example

For our example in Figure 5.1 the resulting hypothesis could be as follows: Step A is not an hypothesis based on rule 1 as it just visualizes the distribution of a single attribute over the whole dataset. Step B is the hypothesis m_1 if the distribution of gender is different given a salary over \$50k. Step C supersedes the previous hypothesis and replaces it with an hypothesis m_1' if the gender distribution between a salary over and under \$50k is different, which is a sightly different question. Step D creates a hypothesis m_2 if the marital status for people with PhDs is different compared to the entire dataset, whereas step-E generates a hypothesis m_3 if there is a different salary distribution given not married people with a PhD. By studying the age distribution in step F the system first generated a default hypothesis m_4 that the distribution of the ages is different given a PhD and being not married for different salary classes. However, the user overwrites immediately the default hypothesis with an hypothesis m_4' about the average age. Furthermore, as the previous visualizations in step D and E might just have been stepping stones towards creating m_4' the user might or might not delete hypothesis m_2 and m_3. However, if the insights our user gained from viewing the marital status, etc., influenced her to look at the age distribution, she might want to keep them as hypothesis.

While this is clearly a simple example, it succeeds in highlighting the general issue. Not every insight the user gains (e.g., the insight that women earn less) is explicitly expressed as a test. At the same time, the more the user explores the data the higher the chance that she finds something which looks interesting, but is actually just effect of noise in the data. In the example above, by

the time the user actually performs its first test (step F), she implicitly already tested at least one other hypothesis and potentially even four others. Assuming a targeted p-valueof $\alpha = 0.05$, the chance of a false discovery therefore increased to $1 - (1 - \alpha)^2 = 0.098$ for two hypothesis and up to $1 - (1 - \alpha)^4 = 0.185$ for four hypothesis. While the question of what should count as an hypothesis is highly dependent on the user and can never be fully controlled by any system, we can however, enable the system to make good suggestions and help users to track the risk of making false discoveries by chance. Furthermore, this short workflow also demonstrates that hypotheses are built by adding but also by removing attributes. As we will discuss later, there exist no good method so far to control the risk of making false discoveries for incremental sessions like the ones created by interactive data exploration systems. We present new methods for interactive data exploration in Section 5.6.

Finally, it should be noted, that the same problems also exist with exploratory analysis using SQL or other tools. However, we argue that the situation is becoming worse by the up-rise of visual exploration tools, like Tableau, which allow to test more hypothesis in a shorter amount of time.

5.4 The QUDE User Interface

As argued in the previous section, user feedback is essential in determining, tracking and controlling the right hypothesis during the data exploration process. With QUDE we created a system that applies our heuristic automatically to all visualizations. We designed QUDE 's user interface with a few goals in mind.

First, the user should be able to see the hypotheses the system assumed so far, their p-values, effect sizes and if they are considered significant and should be able to change, add or delete hypotheses at any given stage of the exploration.

Second, hypotheses rejection decisions should never change based on future user actions unless the user explicitly asks for it. We therefore require an incremental procedure to control the multiple hypothesis risk that does not change its rejection decisions even if more hypothesis tests are executed. For example, the system should not state that their is a significant age difference for not married highly educated people, and then later on revoke its assessment just because the user did more tests. More formally, if the system determined which hypotheses $m_1...m_n$ are significant (i.e., it rejects the null) or not and the user changes the last hypothesis or adds an hypothesis m_{n+1}, which should be the most common cases, the significance of hypotheses $m_1..m_n$ should not change. However, if the user might change, delete, or add hypothesis $k \in 1, .., n$, depending on the used procedure we might allow that the significance of hypotheses m_{k+1} to m_n might have to change as well.

Third, individual hypothesis descriptions should be augmented with information about how much data n^{H1} the user has to add, under the assumption that the new data will follow the current observed distribution of the data, to make an hypothesis significant. While sounding counter-intuitive, as one might (wrongly) imply, it is possible to make any hypothesis true by adding more data, calculating this value is in some fields already common practice. For example, in genetics scientist often

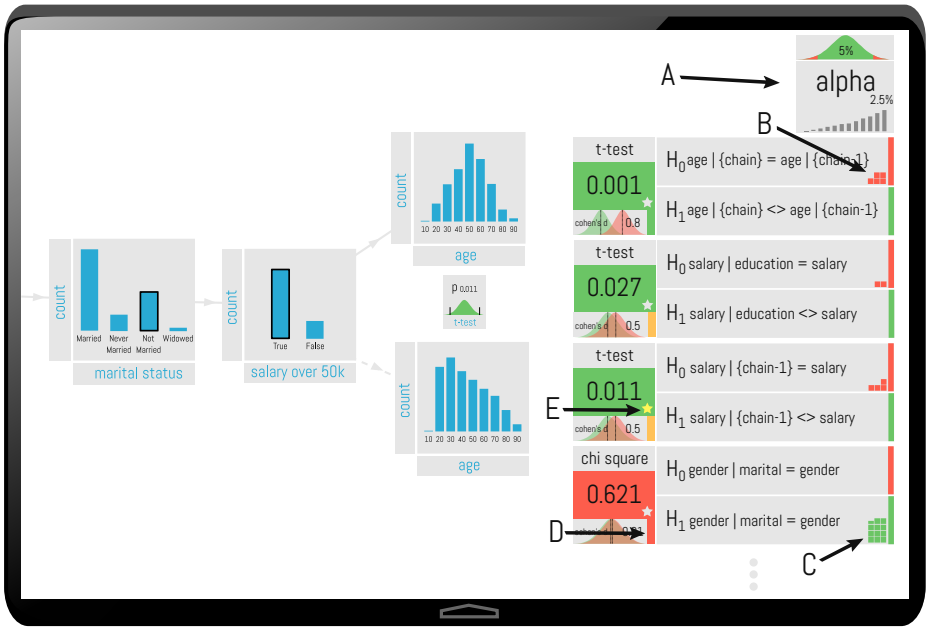


Figure 5.2: The QUDE User Interface

search (automatically) for correlations between genes and high-level effects (like cancer). If such a correlation is found, often because of the multiple hypothesis error the chance of a true discovery is tiny (i.e., the p-valueis too high). In that case the scientist works backwards and estimates how much more genes she has to to sequence in order to make the hypothesis relevant, expecting that the new data (e.g., gene sequences) follow the same distribution of the data the scientist already has. However, if the effect was just produced by chance, the new data will be more similar to the distribution of the null hypothesis and the null will not be rejected. The required value is generally easy to calculate or approximate, and are highly valuable for the end-user. A small value for n^{H1} in relation to the number of totally tested hypotheses might be an indication that the power (i.e., the chance to accept a true alternative hypothesis) of the test was not sufficiently large.

Finally, users should be able to bookmark important hypotheses. As our system uses default hypotheses, there might be more hypotheses generated by the system than those that the user actually intends to test. It may be too cumbersome for the user to correct every time to convey his true intentions. Further, some hypotheses might be more important than others; (e.g. the hypotheses the user would like to include in a presentation or show to her boss).

A key question is what is the expected number of false discoveries among those important discoveries. Figure 5.2 shows the current interface design of QUDE with a risk controller, which incorporates the above ideas, running on a tablet. The user interface features an unbounded 2D canvas where chains of visualizations (such as the one shown in Figure 5.1) can be laid out in a free form fashion. A "risk-gauge" on the right-hand side of the display (Figure 5.2 (A)) serves two purposes: it gives users a summary of the underlying procedure (e.g., the budget for the false discovery rate set to

5% with current remaining wealth of 2.5%; both explained in the next two sections) and it provides access to a scrollable list of all the hypothesis tests (implicit and explicit) that have been execute so far. Each list entry displays details about one test and its results. Textual labels describe the null and alternative hypothesis and color coded p-valuesindicate if the null hypothesis was rejected or accepted (green for rejected, red for accepted). Furthermore, it visualizes the distribution of null hypothesis and alternative hypothesis and shows its difference, included an indication of its color coded effect size (D). Tap gestures on a specific item allow users to change things like the default hypothesis or the type of test. Additionally other information such as an estimation of the size of an additional data n^{H1} that could make the observation significant can be displayed in each item. In the example this information is encoded through a set of small squares (B, C) where each square indicates the amount of data that is in the corresponding distribution. In (B) the five red squares tells us that we need 5x the amount of data from the distribution under null to flip this test form rejected to accepted or conversely in (C) 11.5x the amount of data from the alternative-distribution to rejected this hypothesis. Finally, we allow to mark important hypotheses by tapping the "star" icons (E).

5.5 Background

The previous section described how we convey the multiple hypothesis error to the user and ask for user feedback to derive the right hypothesis to be considered. In this section we describe different techniques that allow to control the risk of incurring in false discoveries and we discuss they appropriateness for the IDE setting. The notation used through this Chapter is summarized in Table 5.1.

We consider a setting, in which we evaluate the statistical relevance of hypotheses from a set $\mathcal{H} = \mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_m$, created incrementally by an IDE system in a streaming fashion. In order to verify whether any such hypothesis $\mathcal{H}_j$ corresponds to an actually statistically significant phenomenon, we consider its corresponding null hypothesis H_j. Using the appropriate statistical test (e.g., the *t-test* or the $\mathcal{X}^2$*-test*), we evaluate p-valueof H_j. Our testing procedure the uses this value to determinate whether to *accept* (resp., *reject*) a null hypothesis H_j which in turn corresponds to rejecting (resp., accepting) the corresponding *alternative hypothesis* (or *research hypothesis*) $\mathcal{H}_|$. The hypothesis according to which all null hypotheses are true is referred as the "*complete*" or "*global*" null hypothesis.

The set of null hypotheses rejected by a statistical test is called "*discoveries*"and is denoted as R. Among these we distinguish the set of *true discoveries* S, and the set of *false discoveries* or *false positives* V where $|V| + |S| = |R|$. False discoveries are commonly referred to as Type 1 errors. Null hypotheses corresponding to true discoveries are called *false null hypotheses*, whereas the others are referred as *true null hypotheses*.

5.5.1 Hold-Out Dataset

A plausible method to handle the multiple hypothesis error is to split the original dataset D into an exploration D_1 and a validation D_2 dataset [245]. D_1 is then used for the data exploration process, while the validation dataset is used to re-test all hypotheses in order to *validate* the results of the first phase. Next we provide examples to clarify why, albeit useful, the hold-out approach does not provide a solution for the multiple hypothesis testing problem.

Let us consider a null hypothesis H, and let p_D denote its associated p-valuewhen H is evaluated with respect of the entire dataset D. Lets assume we perform a test with significance-level α. In this case the probability of wrongly rejecting H is at most α Suppose now that we randomly split the dataset into two datasets D_1 and D_2. For the same null hypothesis H we evaluate the p-valuesp_{D_1} and p_{D_2} each obtained by evaluating H on D_1 or D_2 respectively. We then run a test with significance-level α (like the one discussed above) for each of the datasets. We then decide to reject H if it has been rejected by *both* the testing procedures operating on the datasets D_1 and D_2. Given that both the procedures operating on D_1 and D_2 have significance-level α, the probability that the overall procedure ends up rejecting H is at most α^2.

For the common value of $\alpha = 0.05$, the chance of a Type I error is thus reduced to $p_D = 0.0025$, which is good news. Rather than fully handling the multiple hypothesis problem, we have however only just lowered of the threshold for rejecting the null hypothesis (i.e., the significance level of the test).

This fact appears clearly in the following scenario. Suppose that the user wants to evaluate multiple hypotheses (e.g., 25) rather than just one. Assuming that these hypotheses, and their p-valuesare independent, the probability of observing at least one erroneous rejection using the test technique based on the use of the holdout dataset would be: $p_f = 1 - (1 - p_D)^{25} \approx 0.06$, which is higher than the desired α significance level. If the user would re-test 100 hypotheses on the validation dataset, p_f increases to ≈ 0.22. As it can be seen by the examples, the hold-out dataset helps to reduce the chance of a false discovery as it lowers the chance of a false positive, but does not control the multi-hypothesis error.

Unfortunately, this technique also significantly reduces the power of the testing procedure. Consider the following example scenario in which we aim to compare the means M_1 and M_2 of two samples one drawn from a population with expected value $\mu_1 = 0$ and the other from a population with $\mu_2 = 1$, both having standard deviation $\sigma = 4$. In order to determinate weather the observed difference between M_1 an M_2 is actually statistically significant, we test the null hypothesis "*there is no significant difference between* μ_1 *and* μ_2" using the *one-sided t-test* and a sample composed by 500 records from each population. Given the properties of the t-test (see [77]), the statistical power of our test would be 0.99, and the probability of erroneously accepting the null hypothesis would be at most 0.01.

Suppose now that we divide the original dataset into a training dataset for exploration and one for validation each composed by 250 records. The statistical power for each of the individual t-test executed on the two dataset is now lowered to 0.87, due to the reduction of the amount of data being

used in the individual tests. Further, recall that the procedure based on the holdout set rejects a null hypothesis only if said hypothesis is rejected by both sub-tests. This implies that the actual *overall* power of the testing procedure is $0.87 \cdot 0.87 \approx 0.76$, which is significantly lower than the 0.99 achieved by the test which uses the entire data.

In general, approaches based on hold-out datasets are considered inferior compared to testing over the entire dataset. In some scenarios, such as building machine learning models, hold-out datasets might even be the only possibility to test a model or tune parameters. In those cases, a hold-out approach (e.g., "*k-fold cross-validation*") should be considered as test on its own and, as recent work suggests [53, 131, 189], should be controlled for the multiple hypothesis error. It is however important to remark that in our work we aim to predict guarantees on the statistical significance of the statistical predictors which are instead not achievable using prediction-driven approaches such as cross-validation.

5.5.2 Family-Wise Error Rate (FWER)

Traditionally, frequentist methods for multiple comparisons testing focus on correcting for modest numbers of comparisons. A natural generalization of the significance level to multiple hypothesis testing is given by the *Family Wise Error Rate* (FWER). Given a family of hypotheses, the FWER denotes the is the probability of making at least one type I error when rejecting the corresponding null hypotheses:

$$FWER = \Pr(V \geq 1) = 1 - \Pr(V = 0) \tag{5.1}$$

If $FWER \leq \alpha$, that is the FWER is *controlled at level* α, we have that the probability of even one Type I error in evaluating a family of hypotheses is at most α.

We say that a procedure controls the FWER *in the weak sense*, if the FWER control at level α is guaranteed only when *all* null hypotheses are true (i.e. when the complete null hypothesis is true). We say that a procedure controls the FWER *in the strong sense*, if the FWER control at level α is guaranteed for any configuration of true and non-true null hypotheses (including the global null hypothesis).

Bonferroni Correction: The Bonferroni correction is simple test procedure that allows to control FWER over multiple hypotheses [28]. Let α be the critical threshold for the test. The value of α is usually selected at 0.01 or 0.05.

Let p_i the p-valuestatistic associated with the null hypothesis H_i. When testing m distinct null hypotheses using the Bonferroni correction, a null hypothesis H_i is rejected if $p_i \leq \alpha/m$. The Bonferroni procedure thus achieves control of the FWER at level α.

Using the Bonferroni correction requires knowledge of the total number of hypotheses being evaluated. This constraint make it not applicable in our IDE setting where the set of hypotheses is incrementally generated by the user over multiple data exploration steps. A possible alternative approach would be to use a variation of the Bonferroni correction, according to which the j-th null hypothesis H_j is rejected if $p_j \leq \alpha \cdot 2^{-j}$. It is possible to show that this procedure indeed controls

FWER at level α as $j \to \infty$ while not requiring knowledge of m. The crucial drawback of this approach is however given by the fact that the acceptance threshold decreases exponentially with respect to the number of hypotheses, thus resulting in a high number of false negatives.

The main common issue with all FWER techniques is that the power of the test significantly decreases as m increases due to the corresponding decrease in the acceptance threshold (α/m in the original Bonferroni or $\alpha/2^i$ in the sequential variant). While some alternative testing procedures such as those of Vǐdák [212], Holm [106], Hochberg [103], and Simes [213] offer more power while controlling FWER, the achieved improvements are generally minor (see [207] for a review of several of these techniques).

5.5.3 False Discovery Rate (FDR)

In [15] Benjamini and Hochberg proposed the notion of *False Discovery Rate (FDR)* as a less conservative approach to control errors in multiple hypotheses tests which achieves a substantial increase in the power of the testing procedure.

FDR-controlling procedures are designed to control the expected ratio of false discoveries among all discoveries returned by a procedure. In particular, the FDR of a statistical procedure is defined as:

$$FDR = E\left[Q\right] = E\left[\frac{V}{R}|R > 0\right] P(R > 0). \tag{5.2}$$

If we define FDR to be zero when $R = 0$, we can simplify 5.2 to:

$$FDR = E\left[\frac{V}{R}\right] \tag{5.3}$$

We say that a testing procedure controls FDR at level α if we have $FDR \leq \alpha$. Designing a statistical test that controls for FDR is not simple, as the FDR is a function of two random variables that depend both on the set of null hypotheses and the set of alternative hypotheses. The standard technique to control the FDR is the *Benjamini-Hochberg procedure*(BH), which operates as follows: let $p_1 \leq p_2 \leq \ldots \leq p_m$ be the sorted order of the the p-valuesfor the m tested null hypotheses. To control FDR at level α (for independent null p-values) determine the maximum k for which $p_k \leq \frac{k}{m} \cdot \alpha$, and reject the null hypotheses corresponding to the p-values$p_1, p_2, \ldots, p_k$.

Interestingly, under the complete null hypothesis, controlling the FDR at level α guarantees also "*weak control*" over the FWER: $FWER = P\left(V \geq 1\right) = E\left(\frac{V}{R}\right) = \text{FDR} \leq \alpha$. This follows from the fact that the event of rejecting at least one true null hypothesis $V \geq 1$ is exactly the event $V/R = 1$, and the event $V = 0$ is exactly the event $V/R = 0$ (recall $V/R = 0$ when $V = R = 0$). The concept of FDR control is thus relatively easy to convey to the user as under complete random data, the chance of one or more false discoveries is at most α as in FWER. However, FDR does not ensure control of the FWER if there are some true discoveries to be made (i.e., it does not ensure "*strong control*" of the FWER). In Appendix 5.10 we provide experimental comparison between FDR and FWER.

Because of its increased power, FDR appears to be a better candidate than FWER in the context interactive data exploration, where usually a larger number of hypotheses are to be considered. Unfortunately, both the original *Benjamini-Hochberg procedure* and its variation for dealing with dependent hypotheses [17] are not incremental as they require knowledge of the total number of hypotheses being tested (similar to what was discussed for the Bonferroni correction) and of the sorted list of all the p-valuescorresponding to each null hypothesis being evaluated.

An adaptation of the FDR technique to a setting for which an unspecified number of null hypotheses are observed incrementally was recently presented in [91]. The main idea behind the Sequential FDR procedure is to convert the arbitrary sequence of p-valuescorresponding to the null hypotheses observed on the stream into an sequence of increasing p-values akin to the one generated by the classical Benjamini-Hochberg procedure. The natural application for this technique is the progressive refinement of a model by considering additional features. That is, it starts constructing a model for the data with something known and general. The user then proceeds to refine the model by determining the most significant features.

One drawback of the Sequential FDR method, is given by the fact that the order according to which the hypotheses are observed on the stream heavily influences the outcome of the procedure. For example, if an hypothesis with high p-valueis observed among the first in the stream, this will harm the ability of the procedure of rejecting following null hypotheses, even if they have low p-value(see discussion in [91]). This aspect makes Sequential FDR not applicable for data exploration system for which the user is likely to explore different "*avenues*" of discovery rather than focusing on the specialization of a model.

5.5.4 Other Approaches

Although for most practical applications, FDR controlling procedures constitute the *standard de facto* for multiple hypothesis testing [65], many other techniques have been presented in the literature. Among them, Bayesian techniques are particularly noteworthy. Alternative solutions based on decision theory using "*Bayesian FDR*" are discussed in [19]. However, as often the case with Bayesian approaches, the computational cost for these procedures when applied to large datasets are significant, and the results are highly dependent on the prior model assumptions.

Another approach is correcting for the multiplicity through simulations (e.g., the *permutation test* [204]) in order to experimentally evaluate the probability of an observation in the null distribution. This approach is also not practical in large datasets because of the large number of different possible observations and the need to evaluate very small p-values for each of these distributions [120].

In this work, we elect to use a family of multiple hypothesis testing procedures know as α-investing introduced in [80] and then generalized in [1]. These procedures are especially interesting for the incremental and interactive nature of interactive data exploration. The details of α-investing and its application to our setting is extensively discussed in the next section.

5.6 Interactive Control

One drawback of the Sequential FDR procedure [91] and any adaptation of the FWER controlling techniques to the streaming setting lies in the fact that decisions regarding the rejection or acceptance of previously considered null hypotheses could potentially be overturned in latter stages due to new hypotheses being considered. Although statistically sound, this fact could appear counter intuitive and confusing to the user. The only way to adopt the Sequential FDR procedure to data exploration would be to batch all the hypotheses and only present the final decisions after the exploration halts. Thus although it is incremental, Sequential FDR is *not* suited for interactive data exploration.

In order to have both incremental and interactive multiple hypothesis control, we consider a different approach for multiple hypothesis testing based on the "α-investing " testing procedure originally introduced by Foster and Stine in [80]. Similarly to Sequential-FDR , this procedure does not require explicit knowledge of the total number of hypotheses being tested and can therefore be applied in the hypothesis streaming setting. α-investing presents however several crucial differences with respect to both traditional and sequential FDR control procedures.

In the following, we first introduce the general outline of the procedure as presented in [80], then we discuss several strategies (called policies) that we have developed for interactive data exploration.

5.6.1 Outline of the Procedure

In α-investing , the quantity being controlled is not the classic FDR but rather an alternative quantity called "*marginal FDR*" (mFDR):

$$mFDR_{\eta}(j) = \frac{E\left[V(j)\right]}{E\left[R(j)\right] + \eta} \tag{5.4}$$

where j denotes the total number of tests which have been executed, while $V(j)$ (resp., $R(j)$) denote the number of false (resp., total) discoveries after j tests using the α-investing procedure.

We say that a testing procedure controls $mFDR_{\eta}$ at level α if $mFDR_{\eta}(j) \leq \alpha$. The parameter η is introduced in order to weight the impact of cases for which the number of discoveries is limited. Common choices for η are $1, (1-\alpha)$, whereas the procedure appears to lose in power for values of η close to 0 [80].

Under the complete null hypothesis we have $V(j) = R(j)$ hence $mFDR_{\eta}(j) \leq \alpha$ implies that $E\left[V(j)\right] \leq \alpha\eta/(1-\alpha)$. If we chose $\eta = 1-\alpha$ then $E\left[V(j)\right] \leq \alpha$, and we can thus conclude that control of the $mFDR_{1-\alpha}$ at level α implies weak control fo the FWER at level α [80]. We refer the reader to the original paper of Foster and Stine [80] for an extensive discussion on the relationship between $mFDR$ and the classic FDR. A generalization of the α-investing procedure was later introduced in [1]. The α-investing procedure does not in general require any assumption regarding the independence of the hypotheses being tested, although opportune corrections are necessary in order to deal with possible dependencies. In our analysis, we however assume that all the hypotheses and the corresponding p-valuesare indeed independent (see also Section 5.8).

Intuitively the α-investing procedure works by assigning to each hypothesis test a budget α' from an initial "*α-wealth*." If the p-valueof the null hypothesis being considered is above α' the null hypothesis is accepted and some budget is lost, otherwise it is rejected and some exploration budget is gained. More formally, we denote as $W(0)$ the initial α-wealth assigned to the testing procedure. If the goal of the testing procedure is to control $mFDR_\eta$ at level α, then we shall set $W(0) = \alpha \cdot \eta$. Here, η is commonly set to $(1 - \alpha)$. We denote as $W(j)$ the amount of "*available α-wealth*" after j tests have being executed.

Each time a null hypothesis H_j is being tested, it is assigned a budget $\alpha_j > 0$. Let p_j denote the p-valueassociated with the null hypothesis H_j. This hypothesis is rejected if $p_j \leq \alpha_j$. If H_j is rejected than the testing procedure obtains a "*return*" on its investment $\omega \leq \alpha$. Instead, if the null hypothesis H_j is accepted, $\alpha_j/(1 - \alpha_j)$ alpha wealth is deducted from the available α-wealth:

$$W(t) - W(t-1) = \begin{cases} \omega & \text{if } p_j \leq \alpha_j, \\ -\frac{\alpha_j}{1-\alpha_j} & \text{if } p_j > \alpha_j \end{cases} \tag{5.5}$$

The testing procedure halts when the available α-wealth reaches 0. At that point in time, the user should stop exploring to guarantee that $mFDR \leq \alpha$. This is obviously something not desirable from the perspective of the user. We discuss this problem and potential solutions in Section 5.6.8.

The budget α_j which can be assigned to a test must be such that regardless of the outcome of the test, the available α-wealth available after the test is not negative $W(j) \geq 0$, hence $\alpha_j \leq W(j-1)/\big(1 - W(j-1)\big)$. Further we impose that $\alpha_j < 1$. While this constraint was not explicated in [80], it is indeed necessary for the correct functioning of the procedure. Setting $\alpha_j = 1$ would lead to the potential deduction of an infinite amount of α-wealth, violating the non negativity of $W(j)$. Setting $\alpha_j > 1$ would instead lead to having a positive increase of the available α-wealth *regardless* of the outcome of the test. In our analysis we will however assume that all the hypotheses being considered are indeed independent and their associated p-valuesare independent as well.

We refer as "*α-investing rule*" to the policy according to which available budget is assigned to the hypotheses that needs to be tested. Furthermore, in [80] it was shown that any α-investing policy for which $W(0) = \eta \cdot \alpha$, $\omega = \alpha$, and which obeys the rule in (5.5), controls the $mFDR$ at level α, for $\alpha, \eta \in [0, 1]$. The freedom of assigning to each hypothesis a specific level of significance independent of the order, and the possibility of "*re-investing*" the wealth obtained by previous rejections constitute great advantages with respect to the Sequential FDR procedure.

5.6.2 α-Investing for Data Exploration

While it is relatively straightforward to devise investing rules, it is difficult determinate a priori the "*best way to invest*" the available *alpha*-wealth. If α_j budged assigned to the hypothesis is too low, the statistical power of every test is greatly reduced and there is a high chance of losing the invested budget even if the null hypothesis being considered is indeed false. On the other hand, if the α_j assigned is too high, the entire α wealth might be quickly exhausted and the testing procedure has

to halt. A policy is most likely to be successful if it can exploit some knowledge of the actual data distribution and the data exploration setting.

Another complication involves the statistical analysis of the individual tests that provide the p-values. To show that a testing procedure controls $mFDR$, we require that conditionally on the prior j - 1 outcomes (denoted as R_i), the level of the test of H_j must not exceed α_j:

$$P(R_j = 1 | R_{j-1}, R_{j-2}, ..., R_1) \leq \alpha_j. \tag{5.6}$$

Note that the tests do not need to be independent, though independence is the easiest setting that satisfies (5.6). Furthermore, the condition is only on the outcome of the previous tests, not on the values of their statistic (see discussion in Section 5.8).

In [80] a specialized strategy, *Best Foot Forward*, is designed for a specific situation where true discoveries are highly clustered. However for interactive data exploration we need to consider wider situations where the structure of true discoveries may not always be clustered but have other properties. We further model these general exploration settings in Section 5.9.

In the remainder of this section we propose different α-investing policies particularly suited for interactive data exploration. Each policy captures a different exploration strategy and aims to exploit different possible properties of the data and the exploration settings.

All our policies assign to each hypothesis a strictly positive budget $\alpha_j > 0$ as long as any α-wealth is available. If $p_j \leq \alpha_j$, the null hypothesis H_j is rejected (i.e., it is considered a discovery). Vice versa, if $p_j > \alpha_j$ is *accepted*. The current α-wealth $W(j)$ is then updated according to the rule in (5.5) and because of it controls $mFDR$ at level α as shown in [80].

5.6.3 β-Farsighted Investing Rule

Like with real investment, the question is if one should invest short or long-term. β-farsighted policies are aimed at preserving wealth wealth over long exploration sessions, while effectively using the available α-wealth for the evaluation of new hypotheses. Given $\beta \in [0, 1)$, we say that a policy is β-farsighted if it ensures that regardless of the outcome of the j-th test at least a fraction β of the current α-wealth $W(j-1)$ is preserved for future tests, that is for $j = 1, 2, \ldots$:

$$\begin{gathered} W(j) \geq \beta W(j-1), \\ W(j) - W(j-1) \geq (\beta - 1) W(j-1) \end{gathered} \tag{5.7}$$

An example of β-farsighted is given in Investing Rule 4. Indeed, this procedure achieves control of the $mFDR_\eta$ at level α.

Different choices for the parameter $\beta \in [0, 1)$ characterize how conservative the investing policy is. If there is high confidence on the first observed hypotheses being true discoveries, small values of beta (i.e., 0.25) would be more effective. Vice versa, high values of β (i.e. 0.9) ensure that even if the first hypotheses are true null, a large part of the α-wealth is preserved.

We say that an α investing policy is "*thrifty*" if it never fully commits its available α-wealth. The described β-farsighted is indeed thrifty. While the procedure will never halt due to the available

ALGORITHM 4 β-farsighted

$W(0) = \eta\alpha$
for $j = 1, 2, ...$ **do**
  $\alpha_j = \min\left(\alpha, \frac{W(j-1)(1-\beta)}{1+W(j-1)(1-\beta)}\right)$
  if $p(H_j) < \alpha_j$ **then**
    $W(j) = W(j-1) + \omega$
  else
    $W(j) = W(j-1) - \frac{\alpha_j}{1-\alpha_j} = \beta W(j-1)$

ALGORITHM 5 γ-fixed

$W(0) = \eta\alpha$
$\alpha^* = \frac{W(0)}{\gamma + W(0)}$
while $W(j-1) - \frac{\alpha^*}{1-\alpha^*} \geq 0$, for $j = 1, 2, \ldots$ **do**
  if $p(H_j) < \alpha^*$ **then**
    $W(j) = W(j-1) + \omega$
  else
    $W(j) = W(j-1) - \frac{\alpha^*}{1-\alpha^*} = W(j-1) - \frac{W(0)}{\gamma}$

α-wealth reaching zero, after a long series of acceptance of null hypotheses the available budget may be reduced so much that it will be effectively impossible to reject any more null hypotheses.

Although these policies may appear wasteful as there is no reward for wealth which has not been invested, they are aimed to preserve some of their current budget for future tests in case the hypotheses considered in the beginning of the testing procedure are not particularly trustworthy.

This investing rule is therefore particular suited for scenarios were the total number of false discoveries in long exploration sessions, potentially across multiple users, should be controlled.

While β-farsighted, is a generalization of the "*Best-foot-forward policy*" in [80], different values of β allow for higher flexibility.

5.6.4 γ-Fixed Investing Rule

A different *non-thrifty* procedure assigns to each hypothesis the same budget α^*. We call γ-*fixed* a procedure that assigns to each null hypothesis a fixed budget α_j equal to a fraction of the initial α-wealth $W(0)$, that is $\alpha^* = W(0)/\left(W(0) + \gamma\right)$, as long as any α-wealth is available.

The details of the γ-fixed procedure controlling $mFDR_\eta$ at level α can be found in the procedure for Investing Rule 5. Note that we define α^* as $W(0)/(\gamma + W(0))$ to ensure that the subtraction of the wealth is constantly $W(0)/\gamma$. Different choices for the parameter γ characterize how conservative the investing policy is. If there is high confidence on the first observed hypotheses being actual discoveries small values of γ (i.e. 5,10,20) would make more sense. Vice versa a high value of γ, e.g., 50, 100, ensures that even if the first hypotheses are true null, a large part of the α wealth is preserved.

ALGORITHM 6 δ-hopeful

1: $W(0) = \eta\alpha$
2: $\alpha^* = \frac{W(0)}{\delta + W(0)}$
3: $k^* = 0$
4: **while** $W(j-1) - \frac{\alpha^*}{1-\alpha^*} \geq 0$, for $j = 1, 2, \ldots$ **do**
5: **if** $p(H_j) < \alpha^*$ **then**
6: $W(j) = W(j-1) + \omega$
7: $\alpha^* = \min\left(\alpha, \frac{W(j)}{\delta + W(j)}\right)$
8: $k^* = j$
9: **else**
10: $W(j) = W(j-1) - \frac{\alpha^*}{1-\alpha^*} = W(j-1) - \frac{W(k^*)}{\alpha^*}$

5.6.5 δ-Hopeful Investing Rule

In a slight variation of the γ-fixed investing rule, we say that a policy is δ-hopeful if the budget is assigned to each hypothesis "*hoping*" that at least one of the next δ hypotheses will be rejected. Each time a null hypothesis is rejected the budget obtained from the rejection is re-invested when assigning budget over the next δ null hypotheses. γ-fixed and δ-hopeful operate by spreading the amount of α-wealth over a fixed number of hypotheses (either γ or δ), δ-hopeful is however "*less conservative*" than γ-fixed as it always operates by investing *all* currently available α-wealth over the next δ hypotheses. The details of the δ-fixed procedure controlling $mFDR_\eta$ at level α can be found in the procedure for Investing Rule 6.

5.6.6 ϵ-Hybrid Investing Rule

Because α-investing allows contextual information to be incorporated, the power of the resulting procedure is related to how well the design heuristic fits the actual data exploration scenario. For example, when the data exhibits more randomness, the γ-fixed rule tends to have more power than the δ-hopeful rule. Intuitively, the α-wealth decreases when testing a true null hypothesis, because the expectation of the change of wealth is negative when the p-value is uniformly distributed on $[0, 1]$. Thus the initial α-wealth is on average larger than the α-wealth available at subsequent steps. Furthermore, since the γ-fixed rule invests a constant fraction of the initial wealth, the power tends to be larger than δ-hopeful. Vice versa, when the data is less random, we expect the power of the γ-fixed rule to become lower than that of the δ-hopeful rule. This is due to the fact that in this setting more significant discoveries tend to keep the subsequent α-wealth high, potentially even higher than the initial wealth. We further study this difference in Section 5.9.

In order to have a robust performance in terms of power and false discovery rate, we design ϵ-hybrid investing rule that adjust the α_j assigned to the various tests based on the estimated data randomness. Our estimation of the randomness of the data is based on the ratio of rejected null hypotheses over a sliding window H_d constituted by the last d null hypotheses observed on a stream. We then compare this ratio with a "*randomness threshold*" $\epsilon \in (0, 1)$ and we conclude whether the

ALGORITHM 7 ϵ-hybrid

$W(0) = \eta\alpha$
$k^* = 0$
$H_d = []$ // Sliding window of size d
while $W(j-1) > 0$, for $j = 1, 2, \ldots$ **do**
  if Rejected$(H_d) \leq |H_d|\epsilon$ **then**
    $\alpha_j = \frac{W(0)}{\gamma + W(0)}$
  else
    $\alpha_j = \min\left(\alpha, \frac{W(k^*)}{\delta + W(k^*)}\right)$
  if $W(j-1) - \frac{\alpha_j}{1-\alpha_j} \geq 0$ **then**
    if $p(H_j) < \alpha_j$ **then**
      $W(j) = W(j-1) + \omega$
      $k* = j$
      $H_d[j] = R_j = 1$
    else
      $W(j) = W(j-1) - \frac{\alpha_j}{1-\alpha_j}$
      $H_d[j] = R_j = 0$

data exhibits high randomness or not. The procedures is outlined in Investing Rule 7.

5.6.7 Investment based on Support Population

An important intuition relative to the computation of the p-valuesis that is most likely to observe high p-valuesfor hypotheses which rely on a small number of data points, that is for hypothesis with low "*support population.*" It is thus only natural to pursue a strategy which adjusts the budget of each hypothesis based on its support population, assigning lower α-wealth budget to hypotheses which are more likely to exhibit higher p-value. In this section we discuss how to *bias* the amount budget assigned to each hypothesis so that hypotheses with more support data receive more "*trust*" (in terms of budget) from the procedure.

Let us denote as $|n|$ the total amount of data being used and by $|j|$ the available data for testing the j-th null hypothesis H_t. A simple way of correcting the assignment of the budget α_j in any of the previously mentioned hypothesis is to assign to the test of the hypothesis $\alpha_j f(\frac{|j|}{|n|})$. Depending on the choice of $f(\cdot)$ the impact of the correction may be more or less severe. Some possible choices for $f(\cdot)$ would be $f(\frac{|t|}{|n|}) = \left(\frac{|t|}{|n|}\right)^{\psi}$ for possible values of $\psi = 1, 2/3, 1/2, 1/3, \ldots$. Note, that this support-size dependent *bias* idea can be used with any of the previously described strategies. In the following, we show the ψ-support policy applied to the γ-fixed rule in Investing Rule 8.

5.6.8 What Happens If the Wealth is 0?

Among all our proposed investing policies, only β-farsighted is "*thrifty*" in that it never fully commits all of the available α-wealth. Still, the available wealth for β-farsighted could eventually become extremely small, to the point that hypotheses can be harder to reject. All the remaining procedures

ALGORITHM 8 ψ-support

1: $W(0) = \eta\alpha$
2: α^* is set by an α-investing rule
3: **while** $W(j-1) > 0$, for $j = 1, 2, \ldots$ **do**
4: $\quad \alpha_j = \alpha^* \left(\frac{|t|}{|n|}\right)^{\frac{1}{2}}$
5: $\quad$ **if** $W(j-1) - \frac{\alpha_j}{1-\alpha_j} \geq 0$ **then**
6: $\quad\quad$ **if** $p(H_j) < \alpha_j$ **then**
7: $\quad\quad\quad W(j) = W(j-1) + \omega$
8: $\quad\quad$ **else**
9: $\quad\quad\quad W(j) = W(j-1) - \frac{\alpha_j}{1-\alpha_j}$

are "*non-thrifty*" and can thus reach zero α-wealth, in which case the user should stop exploring because no more hypotheses can be rejected.

Theoretically, a vanishing α-wealth indicates higher uncertainty from the current data and hypotheses, and thus it is reasonable to restrain further exploration. On the other hand, it is only natural to wonder if it would be possible for the user to "*recover*" some of the lost α-wealth and thus continuing the testing procedure. One possible way would require the user to reconsider and possibly overturn some of the previous decisions on whether to reject or accept some null hypotheses using alternative testing procedures (i.e., the Benjamini-Hochberg procedure). There are however several challenges to be faced when pursuing this strategy: 1) great care has to be put on haw to combine results from different testing procedures (i.e., control of FDR for a subsets of hypotheses and control of *mFDR* for a distinct subset of hypotheses) and 2) testing hypotheses for a second time given the outcomes of other test implies a clear (and strong) dependence between the outcome of the tests and the p-valueassociated with the null hypotheses being considered. Therefore, depending on the context such control could only be achieved given additional assumptions about the level of control or would require adding additional data or the use of a hold-out dataset. We aim to study this problem in detail as part of future work.

5.7 Most Important Discoveries

In Section 5.4 we argued that the user should be able to "*mark* the important hypotheses (e.g., the ones she wants to include in a publication). This is particularly important as QUDE uses default hypotheses, which the user might consider as less important. In the following we show that if these "*important discoveries*" are selected from all the discoveries given by a testing procedure that controls FDR at level α independently of their p-values, then the FDR for the set of important discoveries is controlled at level α as well.

Theorem 5.1. *Assume that we executed a collection of hypothesis tests with a rejection rule that controls the FDR at α. Assume that the procedure rejected a nonempty set of null hypotheses R, and let $V \subseteq R$ be the set of false discoveries. If the hypothesis tests are independent then for any subset*

$R' \subseteq R$ *we have* $E[|V \cap R'|/|R'|] \leq \alpha$.

Proof. Let $p_1, \ldots, p_{|R|}$ be the p-valuesof the rejected hypotheses. Since the rejection rule controls the FDR at α we have

$$\sum_{i=1}^{|R|} \frac{i}{|R|} P(|V| = i \mid P_1 = p_1, \ldots, P_r = p_r) \leq \alpha \tag{5.8}$$

Assume that $|V| = i$. The p-valuesof true null hypotheses in V are i.i.d. uniformly distributed in $[0, 1]$. The set V is uniformly distributed among all the i-subsets of R. Let $p_i^V, \ldots p_{|V|}^V$ be the p-valuesof V, and let $p'_1, \ldots, p'_{|R'|}$ be the p-valuesof a subset of rejected hypotheses $R' \subseteq R$, then:

$$E[|V \cap R'| \mid |V| = i] = E[|\{p'_1, \ldots, p'_{|R'|}\} \cap \{p_1^V \ldots P_{|V|}^V\}| \mid |V| = i] = i\frac{|R'|}{|R|}. \tag{5.9}$$

Combining equations (5.8) and (5.9) we get:

$$\begin{aligned} E\left[\frac{|V \cap R'|}{|R'|}\right] &= \sum_{i=1}^{|R|} E\left[\frac{|V \cap R'|}{|R'|} \;\middle|\; |V| = i\right] P(|V| = i \mid P_1 = p_1, \ldots, P_r = p_r) \\ &= \sum_{i=1}^{|R|} \frac{1}{|R'|} i \frac{|R'|}{|R|} P(|V| = i \mid P_1 = p_1, \ldots, P_r = p_r) \leq \alpha \end{aligned} \tag{5.10}$$

□

Consider a set R' of important discoveries selected independently of the p-valuesof the corresponding tests from a larger set of discoveries R for which then $mFDR$ is controlled at level α. Using a proof similar to the one discussed in Theorem 5.1 it is possible to show that the $mFDR$ of R' is controlled at level α as well. This is an important result, as it implies that the user can select the important discoveries from a larger pool of discoveries while maintaining the control of FDR (or mFDR) at level α.

5.8 Limitations and Opportunities

QUDE is the first system that automatically controls the multiple hypothesis error during visual data exploration and as such, it is important to understand the assumptions and limitations of our current approach and the opportunities for future work.

Visualizations: Currently we only automatically derive a null-hypothesis for histograms. While we believe that many other visualizations (e.g., line charts, heatmaps, etc.) have natural null-hypothesis associated to them, exploring them remains future work.

Sequential Dependencies: As formulated in Section 5.6.2, our α-investing procedures assume that the p-value of each test is computed in the sample space conditioned on the outcome of previous tests. How restrictive is this assumption? Significant part of any exploration process is done in a sequential process, in which features, or variables, are selected one at a time. Once we selected the first $k-1$ features, we test for the significance of adding the k feature to the current model. This

process, which corresponds to testing nested hypothesis, trivially satisfies the condition formulated by (5.6). When not testing for nested hypothesis we need to be more careful with sequential dependency. Ideally, we want to test mutually independent features, or correct for the dependencies, but this may not be feasible. In practice, features have to be highly correlated to significantly distort the outcome of the process, and we can identify highly correlated features by computing the correlation coefficient in the data (with no testing). While modern statistics does not provide a fully analytical solution for the problem, our experiments show that independence assumption is a reasonable approximation for non-adversarial users and provides a best-effort attempt considering that often the only alternative is to leave the user in the dark.

Is it Possible to "Game" the System? For example, could a user boost her α-wealth for risky test by testing trivial hypothesis first ? While the short answer is "yes", it is not really gaming the system. mFDR controls the ratio of false over all discoveries and adding more trivial true hypotheses simply increases the denominator.

5.9 Experimental Evaluation

In this section, we evaluate QUDE in different data exploration settings to answer the following questions: (1) how the different α-investing rules perform in different exploration scenarios, (2) how different parameters change the performance of the rules, and (3) how to select the parameters.

5.9.1 Exploration Settings

The data exploration process can vary significantly depending on how the hypotheses are structured. For example, in some settings the explorer may not start with any particular set of questions as target, but gradually develops interests in certain aspect of the data. Such settings are predominant in interactive data exploration. On the other hand, the exploration may be structured around a clear subject, such as understanding gravity, and the exploration tends to progress from easier questions towards harder questions. Such cases arise frequently in data-driven scientific studies. In other cases the mining of insights might usually less structured.

To account for varying structures of the data exploration, we formulate three different data exploration scenarios, namely, the *targeted exploration*, the *free-form exploration*, and the *uniform exploration*. We use Markov processes to simulate the data exploration process under these models as a stream of hypotheses, and identify the suitability of different α-investing rules.

We model the data exploration as divided into two phases of different distributions of random noise. Concretely, we construct two two-state Markov chains, X_a and X_b, where the states correspond to ground truth labels, namely true or false null hypotheses. The two Markov chains have stationary distributions π_a and π_b. We start the process at one chain and then switch to the other chain at a preset time. If the Markov chain generates a significant hypothesis, it draws data points from two normal random variables with different means (the difference is sampled from the intervals

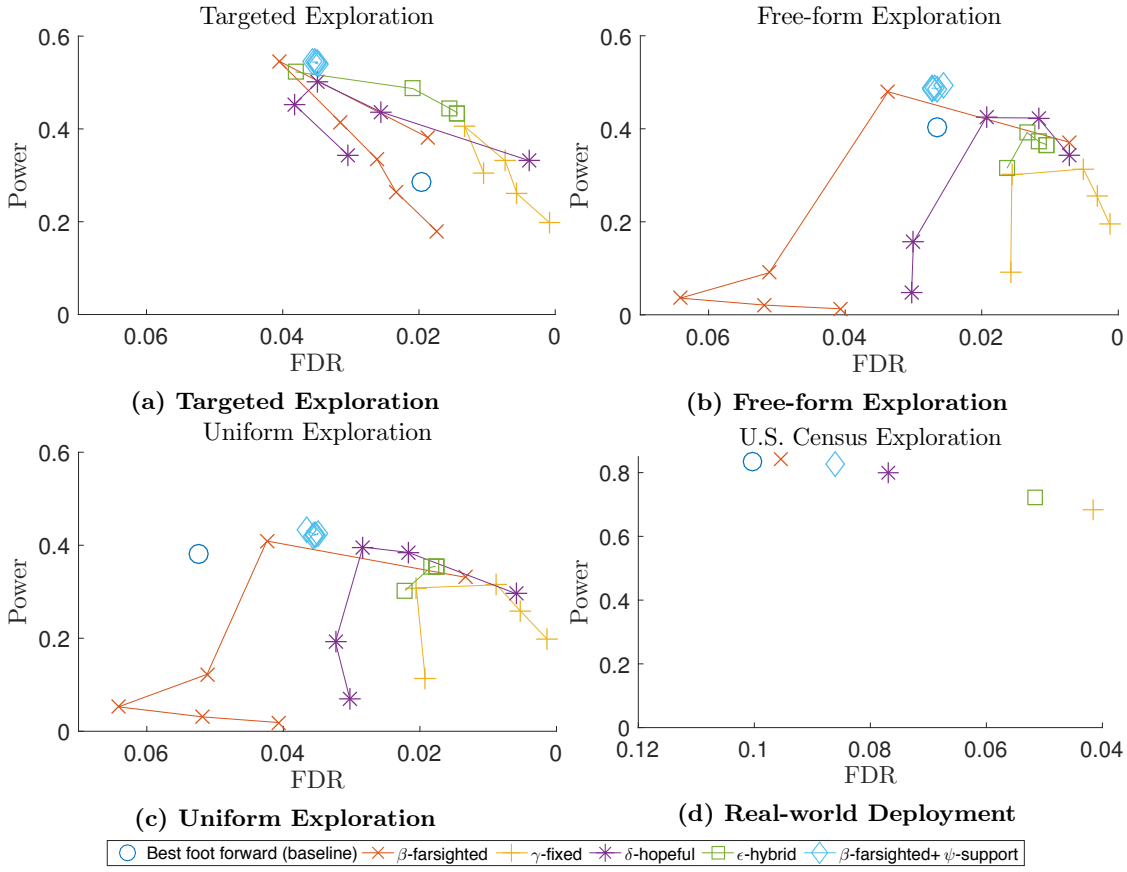


Figure 5.3: The Power-FDR curves with varying parametrization (Table 5.2) in different data exploration scenarios.

with boundaries 5/4, 5/2, 15/4, 5), whereas for an insignificant hypothesis (i.e. true null), it samples from two zero-mean random variables. The hypothesis tests are all t-tests. Afterwards, another uniform random variable is used to determine the number of records used per test, which simulates the selectivity of a histogram filter chain such as in Figure 5.1.

Finally, to demonstrate real-world applicability, we deploy the α-investing rules with the parameters obtained from the models to U.S. Census dataset [143] using an authentic user workflow [242].

5.9.2 Targeted Exploration

The user in the target exploration may have a well-defined set of questions to ask, and thus explores the data in a more structured manner. She starts with simpler questions such as "is there a salary difference between men and women," perhaps for confirmatory purposes. As data conforms to her initial questions, she dives deeper into the more complicated questions such as "does the state impact the salary between men and women." Note that the structure of the hypotheses progresses in such a way that the significant insights tend to cluster more from the beginning than towards the end. Thus

Rule	Parameter	Aggressive $\longleftrightarrow$ Conservative
β-farsighted	β	0.1, 0.3, 0.5, 0.7, 0.9, 0.99
γ-fixed	γ	8, 16, 32, 64, 128
δ-hopeful	δ	8, 16, 32, 64, 128
ϵ-hybrid	ϵ	0.1, 0.3, 0.5, 0.7, 0.9
ψ-support	ψ	1/2, 1/4, 1/6, 1/8, 1/10

Table 5.2: Parameters for Power-FDR curves in Figure 5.3

in the corresponding Markov process, we set the first phase of exploration with lower distribution of random noise than the second phase. Concretely, the stationary distributions of the true nulls are set to $\pi_a(0) = 0.25$ and $\pi_b(0) = 0.75$. The switching time is set to 30% of the process.

Figure 5.3 shows the performance power vs. *realized* FDR of the different α-investing rules in this setting. Notice, that we plot FDR in decreasing order. Thus, in general the top-right corner means better: more true discoveries (i.e., power) with fewer false discoveries (i.e., realized FDR). For each rule we vary the parameters according to Table 5.2 from more aggressive to more conservative settings, resulting in the Power-FDR curve. For the ϵ-hybrid and ψ-support strategies we keep $\delta = \gamma = 64$ and $\beta = 0.9$. We implement the Best foot forward [80] as a baseline; it does not allow for parametrization and is thus a single point.

Figure 5.3a shows that all proposed α-investing rules can outperform the original Best foot forward rule by achieving more power. Which policy to use depends on the context information from the user. For the thrifty policies, i.e., policies which allow an unbounded number of exploration steps, β-farsighted at $\beta = 0.9$ achieves 1.9x higher power than the baseline Best foot forward, while having slightly higher but still bounded error rate at 0.05. For a more conservative approach, increasing β to 0.99 reduces error by more than 50% while still having 1.3x (second to the top data point) higher power than the Best foot forward. Varying ψ-support shows very little impact and its augmented version of β-farsighted achieves the highest power at lower realized FDR than other thrifty strategies.

If the exploration steps are known or can be approximated a priori, non-thrifty policies can be used to further reduce the error rates. Among the non-thrifty policies, δ-hopeful is ideal if the user has an understanding on how many exploration steps she expects in the targeted exploration. It achieves up to 2x the power of Best foot forward and never drops below it. The δ-hopeful also nicely visualizes the impact of being too optimistic versus too pessimistic. For example, $\delta = 8$ and $\delta = 128$ achieve similar power but $\delta = 128$ has a much lower FDR.

ϵ-hybrid offers the best combination of power and error rate with sharp drop of FDR while maintaining the power within 10% of its peak. It offers similar error rate as the baseline Best foot forward, but has 1.7x higher power. This shows the appealing aspect of ϵ-hybrid that it combines the high power of δ-hopeful and the low error of γ-fixed.

5.9.3 Free-form Exploration

In free-form exploration, the explorer does *not* start with a specific set of questions in mind, but rather starting with creating an initial understanding by exploring different aspects of the dataset before narrowing down to certain aspect that is interesting. In terms of hypothesis testing, the fraction of significance insights tends to be lower at the beginning, but increases towards the end because hypotheses become more based upon the previously discovered as "interesting" findings. To simulate this effect, we set the first phase of the Markov process with higher distribution of random noise ($\pi_a(0) = 0.75$) than the second phase ($\pi_b(0) = 0.25$) with a total of 128 hypotheses and switching time at 70% of the process.

In this scenario the non-thrifty policies do not perform as well if the parameters are set too lower than the number of expected exploration steps. However, if set correctly, e.g., at about half of the number of expected steps as $\delta = 64$, δ-hopeful has a similar power as Best foot forward but with a much lower error rate.

For unbounded exploration, β-farsighted with $\beta = 0.9$ achieves 1.18x power over Best foot forward, while adding ψ-support reduces its error rate by 20%, making it the overall best strategy.

5.9.4 Uniform Exploration

The last data exploration model does not build upon how the hypotheses are structured or ordered, and therefore represents an average case. Specifically, the significant insights during the data exploration process are observed uniformly at random.

Interestingly, the Power-FDR curves of the policies in Figure 5.3(c) look similar to the ones in the free-form exploration in (b). One difference is that the baseline Best foot forward performs worse, having similar power but almost 2x the error rate at close to 0.05. This demonstrates the downside of having a specialized policy as Best foot forward, which optimizes for a special case where the significant insights are highly clustered.

For an exploration with or without known expected length, our α-investing rules have parametrization that outperform the baseline Best foot forward in error rate. As for power, the β-farsighted with ψ-support offers the highest power. With information of expected exploration length, the ϵ-hybrid achieves almost the same power but has 60% lower error rate than the baseline Best foot forward, making it the most balanced strategy.

5.9.5 Real-world Deployment

To test the applicability of the optimal parameters in the models, we use the U.S. Census dataset [143] with 10,000 records and derived a real exploration workflow from the user study in [242]. The workflow contains 117 hypotheses, similar to the configuration of our models. To generate ground truth labels, we run Bonferroni procedure [28] with family-wise error rate set to 10^{-20} on the Census dataset as the population. We then sample 10% of the Census dataset to run the user workflow with $\alpha = 0.05$.

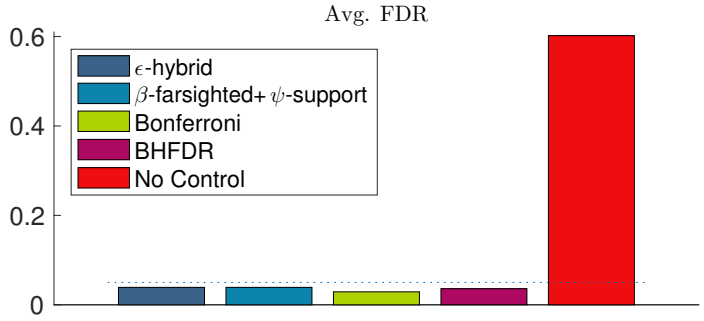


Figure 5.4: Avg. False Discovery Rate on Random Data

We do not vary the parameters for this experiment , but instead pick the best overall settings from the Markov simulation with $\beta = 0.9$, $\delta = 64$, $\gamma = 64$, $\epsilon = 0.5$, $\psi = 1/4$.

Figure 5.3(d) shows a similar trend: our proposed α-investing rules achieves lower error rate than the baseline Best foot forward. If the expected exploration length is unknown, the β-farsighted with ψ-support is better. Otherwise with known expected exploration length, the non-thrifty policies such as γ-fixed, δ-hopeful and ϵ-hybrid achieve significantly less error than the thrifty policy β-farsighted. The ϵ-hybrid combines the higher power of δ-hopeful and lower error rate of γ-fixed and is the best-balanced strategy.

Finally, note that the Best foot forward overachieves on the real-world dataset than the simulation because of the way we generate the ground truth; the Bonferroni procedure creates a more benign setting for this strategy (note that statisticians usually only use simulations to eliminate this bias).

5.9.6 Discussion

To conclude, if the expected number of hypotheses can be estimated a priori, ϵ-hybrid provides high power with significantly lower error rate than the baseline Best Foot Forward. Otherwise if the exploration is unbounded, β-farsighted is the best policy. Overall the β-farsighted with ψ-support achieves the highest power often at a lower realized FDR and presents a good choice independent of the exploration scenario.

Finally, it should be noted that while some of the power/FDR improvements appear to be minor, they can have profound statistical impact in practice, such as determining which discoveries can be deemed scientific, how much more data has to be collected, or what exploration path the user takes.

In Section 5.10 we further compare QUDE's dynamic scheme against the static FDR method [15], the FWER control procedure [28], and the scheme without multiple hypotheses control to illustrate QUDE's overall safety and efficiency.

5.10 Supplemental Experiments

We provide additional experimental results to illustrate the improved safety QUDE provides over the scheme without any multiple comparisons control, and the higher power over the static control

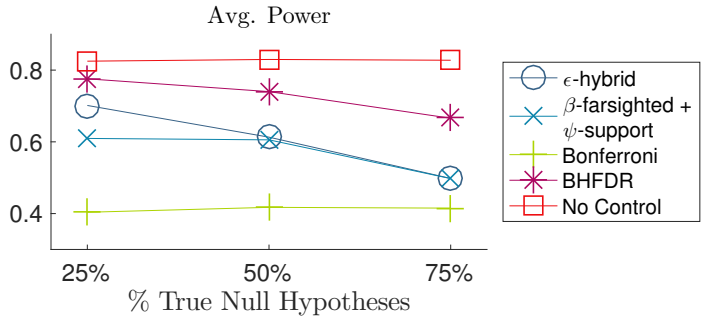


Figure 5.5: Avg. Power on Data with Varying Uncertainty

methods, including the false discovery rate (FDR) control procedure such as Benjamini-Hochberg (BHFDR) [15], and the family-wise error rate (FWER) control procedure such as Bonferroni [28]. In the following experiments we use the exploration settings as in 5.9.1 with 64 user observations, but with varying ratio of randomness.

5.10.1 Safety against Uncertainty

We further discuss the impact of false discovery on completely random data to show that in this worst case QUDE guarantees the bounded error rate, providing the same guarantee as the static FDR control procedure BHFDR and the FWER control procedure Bonferroni. On the other hand, the scheme without multiple comparison control is highly error-prone.

We use t-tests on all true null hypotheses generated from normal random variables with the same mean. The *average* FDR is based on 1000 repetitions, where for each repetition the *realized* FDR is either 1 or 0. As shown in Figure 5.4, the scheme without control results in as high as 60% false insights for a moderate number of 64 user observations. By contrast, methods with multiple hypotheses control achieve the guaranteed error rate as low as 5%.

5.10.2 FDR versus FWER

We compare the two multiple hypothesis control targets, FDR and FWER, to show that FDR (and its variant mFDR) provides the best trade-off in that it has much higher statistical power while having the same worst-case error bound than the FWER. This comparison motivates our choice of FDR as the control target for interactive data exploration.

On completely random data, controlling FDR also implies controlling FWER [15]. This can be seen in Figure 5.4 that both of our representative thrifty and non-thrifty α-investing rules achieve the same error rate as the Bonferroni at 0.05.

With varying degree of uncertainty as in Figure 5.5, the power of the Bonferroni remains only as low as 40%, whereas the power of the (m)FDR control procedures all achieve significantly higher power. With less randomness, i.e. lower fraction of true null hypotheses, both of QUDE's representative thrifty and non-thrifty α-investing rules achieve close to the static FDR control method

BHFDR and the scheme without control.

As the uncertainty increases, the power of our α-investing rules maintain higher than the FWER control procedure. The static method BHFDR has slightly better power than our α-investing rules because it operates offline where it collects and sorts all hypotheses based on their final p-values, which however limits its usability on dynamic settings; whereas our α-investing rules removes this limitation to support interactive data exploration without significant loss of power while with the similar guarantee on the error rate.

Finally, although the scheme without control achieves slightly higher power, it is at the expense of much higher error rates, such as about 100X on random data. By contrast, QUDE represents the optimal trade-off between the power and the risk.

5.11 Conclusion and Future Work

In this Chapter, we presented the first automatic approach to controlling the multiple hypothesis problem during data exploration. We showed how the QUDE systems integrates user feedback and presented several multiple hypothesis control techniques based on α-investing, which control *mFDR*, and are especially suited for controlling the error for interactive data exploration sessions. Finally, our evaluation showed that the techniques are indeed capable of controlling the number of false discoveries using synthetic and real world datasets. We consider this work as an important first step towards more sustainable discoveries in a time where the importance of data analysis is more pervasive than ever. We strongly believe that our work constitutes a departing point for a wealth of important research topics such as creating and evaluating other types of default hypothesis, developing new testing procedures (e.g., for interactive Bayesian tests) and investigating techniques to recover from cases where the testing procedure runs out of α-wealth.

Chapter 6

VizRec: a framework for secure data exploration via visual representation[1]

Visual representations of data (visualizations) are tools of great importance and widespread use in data analytic as they provide users visual insight to patterns in the observed data in a simple and effective way. However, since visualizations tools are applied to sample data, there is a a risk of visualizing random fluctuations in the sample rather than a true pattern in the data. This problem is even more significant when visualization is used to identify *interesting* patterns among many possible possibilities, or to identify an interesting deviation in pair of observations among many possible pairs, as commonly done in *visual recommendation systems*.

In this Chapter, we present *VizRec*, a framework for improving the performance of visual recommendation systems by quantifying the statistical significance of recommended visualizations. The proposed methodology allows to control the probability of misleading visual recommendations using both classical statistical testing procedures and a novel application of the Vapnik Chervonenkis (VC) dimension method which is a fundamental concept in statistical learning theory.

6.1 Introduction

Visual recommendation engines, such as SeeDB [232], Voyager2 [238], Rank-By-Feature [206], Show Me [152], MuVE [66], VizDeck [127], DeepEye [187], or Draco-Learn [166], aim to help users to more quickly explore a dataset and find interesting insights. To achieve that goal they use widely different approaches and techniques. For example, SeeDB [232] makes recommendations based on a reference view; it tries to find a visualization which is very different from the one the user has currently on the screen. In contrast, DeepEye [187] tries to generally recommend a good visualization for a given dataset based on previously generated visualizations.

[1]The results presented in this chapter are currently submitted to the 40th ACM SIGMOD International Conference on Management of Data (SIGMOD 2019). This is joint work with Leonhard Spielberg, Professor Tim Kraska, and Professor Eli Upfal.

However, all these systems do have in common that they can significantly increase the risk of finding false insights. This happens in the moment a visualization is not just a pretty picture but a tool presenting facts about the data to the user. For example, consider a user exploring a dataset containing information about different wines. After browsing the data for a bit, she creates a visualization of ranking by origin showing that wines from France are higher rated. If her only takeaway from the visual is, that in **this particular dataset** wines from France have a higher rating, there is no risk of false insight. Essentially, in this case no inference happens as she is completely aware that the next dataset could look entirely different. However, it is neither in the nature of users to constraint themselves to such thinking [243], nor would in many cases such a visualization be very insightful. Rather, based on the visualization she most likely would infer that French wines are *generally* rated higher; *generalizing* her visualization insight to general datasets and thus creating an actually interesting insight. Statistically savvy users will now test this insight on whether this generalization is actually statistically valid using the appropriate test. Even more technically savvy users will also consider other hypothesis they tried and adjust the statistical testing procedure to account for the multiple comparisons problem. This is important as every additional hypothesis, explicitly expressed as a test or implicitly observed through a visualization, increases the risk of finding insights which are just spurious effects.

However, what happens when the visualization recommendations are generated by one of the above systems? First and most importantly, the user does not know if the effect shown by the visualization is actually significant or not. Even worse, she can not use a standard statistical method and "simply" test the effect shown in the visualization for significance. Visual recommendation engines are potentially checking thousands of visualizations for their interesting-factor (e.g., in case of SeeDB how different the visualization is to the current one) in just a few seconds. As a result, by testing thousands of visualizations it is almost guaranteed that the system will find something "*interesting*" regardless of whether the observed phenomenon is actually statistically valid or not. A test for significance for the recommended visualization should therefore consider the whole history of tests done by the recommendation engine.

Advocates of visual recommendation engines usually argue that visual recommendations systems are meant to be as hypothesis generation engines, which should always be validated on a separate hold-out dataset. While this is a valid method to control false discoveries, it is also important to understand its implications: (1) None, really none, of the found insights from the exploration dataset should be regarded as an actual insight before they are validated. This is clearly problematic if one observation may steer towards another during the exploration. (2) Splitting a dataset into an exploration and a hold-out set can significantly reduce the power (i.e., the chance to find actual true phenomena). (3) The hold-out needs to be controlled for the multi-hypothesis problem unless the user only wants to use it exactly once for a single test.

In this Chapter, we present an alternative approach, called VizRec, a framework to make visual recommendation engines "safe". We focus on the visual recommendation technique proposed by SeeDB as it uses a clear semantic for what "interesting" means. However, our techniques can be

adjusted to other visualization frameworks or even hypothesis generation tools, like Data Polygamy, as long as the "interesting" criterion can be expressed as a statistical test. The core idea of VizRec is that it not only evaluates the interesting-factor using a statistical test, but also that it automatically adjusts the significance value based on the search space of the recommendation engine to avoid the multiple-hypothesis pitfall. We make the following contributions:

- We formalize the process of making visualization recommendations as statistical hypothesis testing.
- We discuss how different possible approaches to make visualization recommendation engines safe based on classical statistical testing, and on the use of Chernoff-type large deviation bounds. We further discuss how the performance of both these approaches decreases due to the necessity of accounting for a high number of potentially adaptively chosen tests.
- We propose a method based on the use of VC dimension, which allows controlling the probability of observing fake discoveries during the visualization recommendation process. VizRec allows control of the *Family Wise Error Rate* (FWER) at a given level δ.
- We evaluate the performance of our system, in comparison with SeeDB via extensive experimental analysis.

Related work: [190] introduced the VC approach to provide ϵ-approximations for the selectivity of queries. Whereas they also consider joins in addition to multi-attribute selection queries, by restricting to AND conjunctions over multiple attributes as used naturally in OLAP and visual recommender systems we were able to lower the required VC dimension.

When comparing continuous aggregates for visualizations using e.g. kernel-density estimation for distribution plots other statistical testing procedures like the Kolmogorov-Smirnov test can be used. Also a viable alternative to goodness-of-fit tests like Fisher's exact test, χ^2-test or the Kolmogorov-Smirnov test independence tests like Kendall's tau or Spearman's rank correlation test may be used. A combination of multiple tests also accounting for differences in shape as detailed in [185] can strengthen statistical guarantees on visualizations being different. Indeed, this is quite similar to hypothesis based feature extraction and selection approaches as in [45][219].

Recent work [199] introduced the problem of group-by queries leading to wrong interpretations, specifically in the case when `AVG` aggregates are used. To remedy this, the notion of a biased query is introduced. However, they do not account for the multiple comparison problem and also have no significant distance notion.

[22] introduced various control techniques for interactive data exploration scenarios. Whereas it accounts for the multiple comparison problem, it does not solve the problem of pointing out a statistical different enough distance between two visualizations.

[232] provides an approach to effectively compute visualizations over an exponential search space by using reuse of previous results and approximate queries. Visualizations are recommended by treating group-by results as normalized probability distributions and using various distance measures

between two probability distributions to yield a ranking in order to recommend top-k interesting visualizations. The authors found that the actual choice of the distance did not really alter results, which does not come at a great surprise given their relations as pointed out in [87].

As described in [210] zooming into particular interesting regions of the data is a key task performed by many users in the setting of data exploration. Our technique provides a simple and effective methodology which can be applied to a wide range of data. For example, an user may be given a geospatial dataset characterizing purchase power and want to have assistance in exploring interesting sub-regions. We believe our VC dimension-based approach can be easily extended to allow for more complicated query types such as these.

Chapter structure The remainder of this Chapter is organized as follows: In Section 6.2 we give a definition of the visualization recommendation problem in rigorous probabilistic terms. In Section 6.3 we discuss possible approaches for the visualization recommendation problems, and why highlight how they both suffer due to the necessity of accounting for a high number of statistical tests. In Section 6.4 we introduce our VizRec approach based on VC dimension and we argue how it allows to overcome the problems discussed in Section 6.3. In Section 6.5, we discuss some guidelines of practical interest for implementation, while in Section 6.6 we present an extensive experimental evaluation of the effectiveness of VizRec.

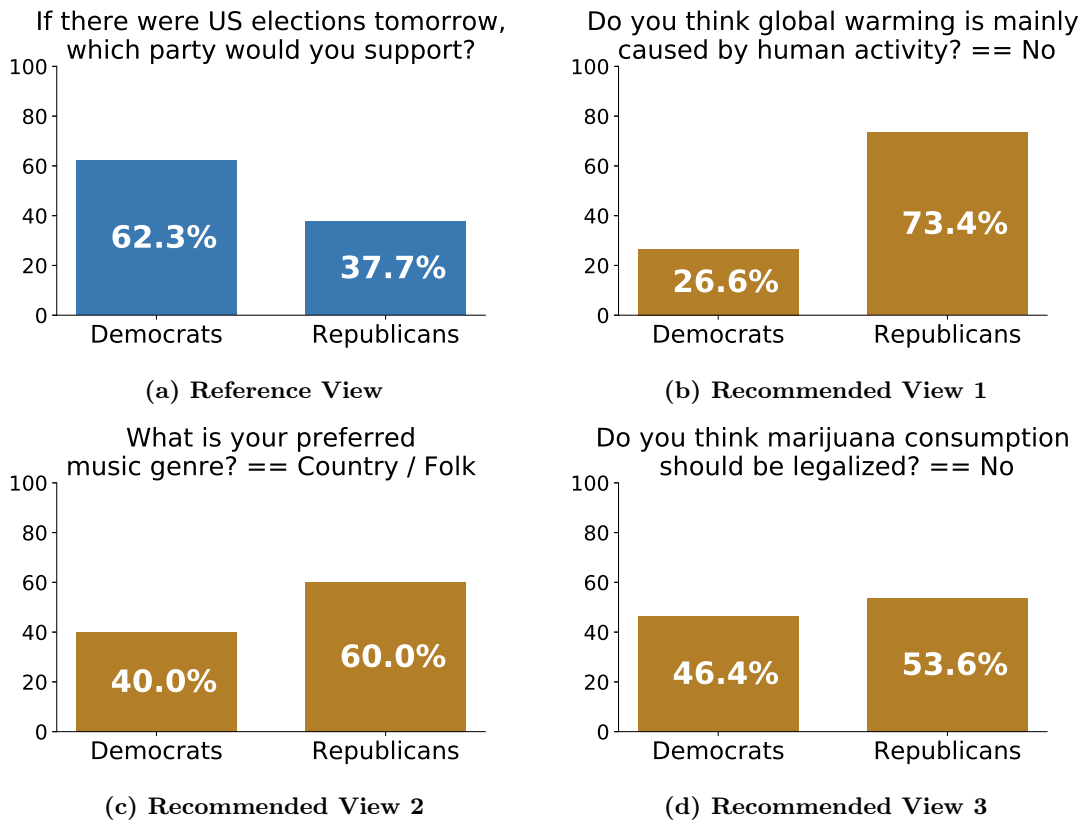
6.2 Problem Statement

In this section we first describe informally how SeeDB [232] makes visual recommendations, and then formalize the problem of providing statistically valid visualization recommendations.

6.2.1 SeeDB

SeeDB makes recommendations based on the currently shown visualization, and the corresponding *reference query*. SeeDB explores possible recommendations (*target queries*) by most commonly adding/changing the composition of the reference query. To rank the recommendations, SeeDB recommends to the user the most interesting target queries based on the deviations. Thus, SeeDB assumes a larger deviation indicates a more interesting target query. SeeDB can use different types of measures to quantify the difference between reference and target visualization. However, earth-mover distance or KL-divergence are prevalently used. Furthermore, SeeDB truncates uninteresting visualizations if the deviation value (e.g., KL-divergence) is below a certain threshold, which can be seen as a minimum visual distance.

To showcase how SeeDB can recommend spurious correlations, we used a survey conducted on Amazon Mechanical Turk [22] with $2,644$ answers for 35 (mostly unrelated) multiple-choice questions. Questions range from *Would you rather drive an electric car or a gas-powered car?* to *What is your highest achieved education level?* and *Do you play Pokemon Go?*.



(a) Reference View **(b) Recommended View 1**

(c) Recommended View 2 **(d) Recommended View 3**

Figure 6.1: An example of SeeDB [232] on survey data.

Suppose the user analyzes U.S. voting trends over this data set as shown in Figure 6.1a. Based on this reference view created by the user, the actual SeeDB algorithm over our data set would recommend Figures 6.1b-6.1d as visualizations (settings of SeeDB are similar to the ones in [232]'s Figure 1). All of the recommended visualizations by SeeDB seem first to be interesting as they clearly indicate a trend reversal and a correlation between certain beliefs and voting behavior. However, these trends could also be solely a random discovery in the way the visualization is just produced because for some reason the dataset by chance just produced such bar graphs for these queries. Indeed, in 6.6.1 we show that some of these recommendations are not *statistically safe recommendations*.

Having a larger dataset may help an analyst to be more certain that the recommendations are actually significant. However, considering the total number of samples used for these visualizations bears the fundamental question, how big the data actually needs to be in order to guarantee that there are no false discoveries. Further, when automatically exploring the dataset at some point with enough filtering as done by SeeDB every dataset becomes "too small" to guarantee anything. In the following, we now formalize the recommendation process of SeeDB, introduce the visual recommendation problem and its relation to hypothesis testing.

6.2.2 Problem set-up

Let Ω denote the *global sample space*, that is the set comprised of the records of the entire population of interest (e.g., the records of US citizens). We can imagine Ω in the form of a two-dimensional, relational table with N rows and m columns, where each column represents a *feature* or *attribute* of the records.

In many practical scenarios the analyst is in possession of a *sample* $\mathcal{D}$, $\mathcal{D} \subset \Omega$, composed by a much smaller number of records $n << N$, and aims to *estimate* the properties of Ω by analyzing the properties of the smaller dataset.

In this work we assume that the input to the recommendation engine is a dataset, $\mathcal{D}$, consists of n records chosen uniformly at random and independently from the universe Ω (e.g., our survey data). We refer to $\mathcal{D}$ as the *sample dataset* or the *training dataset* and denote by $\mathcal{F}_{\mathcal{D}}$ the probability distribution that generated the sample $\mathcal{D}$. Alternatively, one can consider $\mathcal{D}$ as a sample of size n from a distribution $\mathcal{F}_{\mathcal{D}}$ that corresponds to a possibly infinite domain.

As discussed in Section 6.1, rather than enabling users to only interpret visualizations for *some particular data set*, we want to make sure that when they try to generalize results a system provides them with statistical guarantees. Hence, we consider the input $\mathcal{D}$ not as a universal but as a *random sample* of the universe, and we focus on observation in $\mathcal{D}$ that apply to the entire universe. For example *prices in city centers are higher than in neighboring districts* instead of *prices in the city center for apartments A, B, C, ... in Manhattan are higher than for apartments X, Y, Z, ... in the Bronx, Queens and Brooklyn neighboring districts for the data collected in 2017 on July 21st.*

We divide the features (or attributes) of the records in Ω, and hence $\mathcal{D}$, into four groups:

1. **binary features**, taking values in $\{0, 1\}$ (e.g., unit has AC).
2. **discrete features with a total order**(e.g., #bedrooms, #bathrooms).
3. **continuous features with a total order**, (e.g., price of the unit).
4. **categorical features**, taking values in a finite unordered domain (e.g., city, zipcode).

We assume that each feature is equipped with a *natural metric*. This is achieved, by mapping the values of a feature to a real number (e.g., for a boolean feature by mapping the value True to 0, and False to 1).

6.2.3 Visualizations

A common form of visualization used in the dice-and-slice exploration setting of an OLAP (*On-Line Analytical Processing*) data cube is a bar graph. We formalize the type of visualizations we investigate in our approach in the following way:

Definition 6.0.1. *A visualization V is a tuple*

$$\Big(\mathcal{D}, F, X, Y, \mathrm{AGG}\Big)$$

which can be represented as a bar graph and which describes the result of a query *of the form*

SELECT X, COUNT(Y) FROM D WHERE F GROUP BY X

with the aggregate COUNT, represented on the y axis, being partitioned according to the values of a discrete feature X, after restricting the records of the input dataset $\mathcal{D}$ being considered to a subset for which the filter predicate F holds.

The *support* of a visualization V is the number of records of D which satisfy the predicate F, and is denoted as $|V|$. The *selectivity* of a visualization V, denoted as γ_V, is defined as the fraction of records which satisfy F, that is

$$\gamma_V := \frac{|\mathcal{D}|F|}{|\mathcal{D}|}. \tag{6.1}$$

Note that if the group-by attribute X being selected is not discrete, it will also be necessary to determinate a finite set ranges for its value, or "*buckets*", to be used in the visualization.
The aggregate `COUNT(Y)` *counts* of records which satisfy the query predicate grouped according to the values of the feature `X`, henceforth referred as the *group-by* feature.

While in this work we focus on the aggregate `COUNT(Y)`, our approach can be extended with minor modifications to the *average* `AVG(Y)` aggregate, which is given by the average of the values of the records which satisfy the query predicate grouped according to the values of the group-by feature.

Though other aggregates like `MIN(Y)`,`MAX(Y)`,`SUM(Y)` are used in systems like SeeDB [232] we believe that they do not add value over `COUNT(Y)` or `AVG(Y)` aggregates. Even worse, they may lead to misleading visualizations. `MIN(Y)`,`MAX(Y)` aggregates are *inherently* not suited to represent *statistically significant* behavior of distributions. Rather than using `MIN(Y)` or `MAX(Y)` aggregates, a user should consider *conditional expectations* (e.g., aggregates in the form $Y|Y > c$ for some constant $c \in \mathbb{R}$) to explore extreme values following the concepts of *extreme-value theory* as described in [88, 73]. While our results can be easily generalized to other types of visualizations (e.g., *heat maps*), for the sake of applicability, in this work we focus on the type captured by Definition 6.0.1.

Visualizations as distributions: When considering the `COUNT` aggregate, it is possible to interpret the data distribution represented by a histogram visualization V as representative of the *probability mass function* (pmf) of the *discrete* random variable X which takes the values of the group by feature `X`, each with probability corresponding count of the records for each column normalized according to the support of the visualization V_1. Such distribution, does indeed correspond to the distribution of the values of the group-by feature `X` with respect to the distribution $\mathcal{D}$ after *conditioning* (or *filtering*) with respect to the predicate F associated with V. Such correspondence between visualizations and distributions provides us a *natural criteria* to compare visualizations by evaluating their *statistical difference*.

6.2.4 Visualization recommendations

Following the paradigm of SeeDB [232], given a first "*starting* visualization V_1 we aim to identify other "*interesting*" visualizations to be recommended. In particular, this follows the widespread mantra of *Overview first, zoom and filter, then details-on-demand* [210] where a visualization system ideally lets the user first pick an interesting reference visualization and helps him then to automate the zoom, filter and details-on-demand tasks.

In this work we define a visualization V_2 to be interesting with respect to a starting visualization V_1 if V_2 and V_1 are *different*, that is, if they represent a different statistical behavior (i.e., different distribution) of the common group-by feature `X` under the predicates associated with V_1 and V_2, respectively. Consistently, the greater the difference between the reference V_1 and the candidate V_2, the higher the interest of V_2 as a *candidate recommendation* for V_1.

Note that the constraint according to which possible recommended visualization must share the same group-by feature as the starting visualization is a simple consequence of the fact that we are interested in the study of how different filter predicate conditions do influence the behaviors of the same feature `X`. In the absence of such constraint, the analyst may consider how *different* features behave, thus leading to observations of questionable interest.

Whereas the general problem of recommending interesting visualizations can come either in the way of anomaly detection or as according to the mantra of [210] we want to focus on the latter in this work. That is, we say that a candidate recommendation visualization V_2 is interesting with respect to a reference visualization V_1 if the two are "*different enough*". That is is their distance reaches a threshold ϵ according to some distance measure d.

$$\mathcal{V}_2 \text{ interesting w.r.t } \mathcal{V}_1 \Longleftrightarrow d(V_1, V_2) > \epsilon.$$

In its simplest form, ϵ may be zero. However it makes more sense to define ϵ in terms of a minimum visual distance ϵ_V required by a user to spot a difference[220] when shown both the reference and a visualization of interest.

While there is a high degree of generality in the selection of the notion of difference between visualizations to be used, in this work, we leverage the correspondence between visualizations which corresponds to Definition 6.0.1 and the pmf of the group-by feature condition on the filter of the visualization, and we measure the difference between visualizations based on the difference between the associated pmfs.

In this work, we use the "*Chebyschev distance*" to quantify the difference between visualizations. Given two pmfs $\mathcal{D}_1$ and $\mathcal{D}_2$ over the same support set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, the Chebyscev distance between $\mathcal{D}_1$ and $\mathcal{D}_2$ is given by:

$$d\left(D_1, D_2\right) = \max_{x \in \mathcal{X}} |\Pr_{\mathcal{D}_1}\left(x\right) - \Pr_{\mathcal{D}_2}\left(x\right)|, \tag{6.2}$$

where $\Pr_{\mathcal{D}_1}\left(x\right)$ (resp., $\Pr_{\mathcal{D}_2}\left(x\right)$) denotes the probability of a random variable taking value x according to the distribution $\Pr_{\mathcal{D}_1}\left(x\right)$ (resp., $\Pr_{\mathcal{D}_2}\left(x\right)$).

This choice of difference metric is particularly appropriate when comparing visualizations as it highlights the maximum difference between the *relative frequency* of a certain value of the group-by

feature in according to the conditional distribution given by the filter predicate of the two visualizations being considered. In other words, when comparing two histogram visualizations the Chebyschev distance captures the maximum difference between pairs of corresponding columns of the histograms.

6.3 Statistically safe visualizations and recommendations

While the statistical pitfalls of exploratory data analysis are well understood and documented (in scholarly papers [196, 224], "surprising statistical" discoveries [217, 94], and even a famous cartoon [239]), the connection to visualizations, and specifically visual recommendations engines, have only recently begun to be rigorously studied [244, 129, 40]. In this section, we formulate the statistical problem in the visual recommendation context and explore simple probabilistic techniques to solve it. A more powerful method, based on statistical learning theory is presented in the Section 6.4. While, for concreteness, in our presentation we focus on the SeeDB paradigms, the proposed model and techniques can be used for other recommendation systems too (e.g., Data Polygamy [43]).

A first crucial observation is that a system that provides visual representation of data and aims to highlight an *interesting relationship* between visualizations should provide tools to allow the analyst to ascertain that the phenomena being observed are actually *statistically relevant*. A recommender system should ensure that visualized result displays characteristics that are *non-random* and *visually intelligible*. That is a user looking at two visualizations should both be able to understand that they are different and why they are different without worrying whether visual features are due to missing support or random noise.

Recall that a dataset $\mathcal{D}$ available to our visualization decision algorithm is a sample from an underlying distribution $\mathcal{F}_{\mathcal{D}}$. Assume that a particular visualization is interesting with respect to $\mathcal{D}$. The question we are trying to answer is, how likely it is that the visualization is also interesting with respect to the underlying distribution $\mathcal{F}_{\mathcal{D}}$.

In probabilistic terms, each query with a filter predicate `F` corresponds to an event E over Ω. For concreteness consider a visualization of a histogram of a (discrete and finite) variable X conditioned on an event $E \neq \emptyset$, denoted $X|E$.

The true values (in $\mathcal{F}_{\mathcal{D}}$) for $k \in \operatorname{dom} X$ are given by

$$p_k := \mathbb{P}(X = k|E) = \frac{\mathbb{P}\left(X = k, E\right)}{\mathbb{P}(E)}.$$

We estimate these values in a dataset $\{X_1, \ldots, X_n\} \subseteq \mathcal{D}$ of size n by

$$\hat{p}_k := \frac{\sum_{i=1}^{n} \mathbb{1}_{\{X_i = k,\ X_i \in E\}}}{\sum_{i=1}^{n} \mathbb{1}_{\{X_i \in E\}}}. \tag{6.3}$$

If in $\mathcal{D}$ the histogram of $X|E$ is visually different from the histogram of X, what can we rigorously predict about the difference between the histograms in $\mathcal{F}_{\mathcal{D}}$?

Therefore, we say that the difference between two visualizations $\mathcal{V}_1$ and $\mathcal{V}_2$ is *statistically significant* if and only if the difference observed between the two in the finite sample $\mathcal{D}$ is due to an

actual difference between the two histograms with respect to the distribution $\mathcal{F}_{\mathcal{D}}$.The recommendation problem thus becomes in its general form to recommend a candidate visualization $\mathcal{V}_2$ for a reference $\mathcal{V}_1$ only if their corresponding histograms are statistically different with respect to the *true* underlying distribution $\mathcal{F}_{\mathcal{D}}$.

Our goal is to verify that interesting visualization flagged by our algorithm with respect to $\mathcal{D}$ *generalize* to interesting visualizations with respect to $\mathcal{F}_{\mathcal{D}}$.

6.3.1 Classical statistical testing

In the *classical* statistical testing setting, our problem could be framed in two ways: Either it could be formulated as a *goodness-of-fit* test or as a *homogeneity* test. In a *goodness-of-fit*-test for a given starting reference visualization $\mathcal{V}_1$ and a candidate visualization recommendation $\mathcal{V}_2$ a *hypothesis* is considered in the form of whether certain statistical attributes of the reference visualization (i.e. the expected attributes) fit the corresponding observed attributes from the candidate query. Classical tests include the single test χ^2-test for discrete distributions or the Kolmogorov-Smirnov test for continuous random variables. In the visualization context thus a candidate query would be selected as interesting when the null hypothesis of the attributes being similar is rejected. A *homogeneity*-test on the other hand tests the hypothesis that two samples were generated by the same underlying distribution, which may be unknown. This is done, for example, by a two samples χ^2-test, or a t-test comparing one column in two histograms. A system based on *homogeneity*-tests would then select a candidate query as interesting iff the sample corresponding to $\mathcal{V}_2$ is not homogeneous with the underlying data distribution of the reference query $\mathcal{V}_1$.

However, there are major difficulties in applying standard statistical tests to the visualization problem. First, depending on the input data the correct test needs to be selected. For example, when using a χ^2-test over discrete attributes, each bucket must not be empty. A general rule of thumb to make sure estimates are reliable is to have at least 5 samples per bucket. Further, there should be enough samples to actually use the χ^2-test. Else, Fisher's exact test should be used for small sample sizes. In addition to each test being only applicable to certain input data, they all do guarantee a *different* notion of interest. A user that is presented with the test results of one or multiple tests usually will not be able to immediately connect the results to the notion of a *significant visual difference* as described in Section 6.2.4. This brings up the problem on how comparable results of e.g a t-test against a χ^2-test actually are in terms of *visual difference*.

Second, when blindly throwing statistical tests at the visualization problem to deal with different types of input data and different sample sizes queries return, the question is whether the hypothesis being tested are not too *simple* for recognizing visual difference in a meaningful way. Consider for this a t-test that essentially compares whether the observed mean resembled the expected mean. Naturally, a consequence is that if they differ the candidate query should get recommended. This may however lead to many wrong recommendations merely because the null hypothesis used is too *simple* and gets rejected too often. The solution could be to use a test better suited for the problem, e.g. in the form of a χ^2-test. However, as we show in Section 6.6.3 a χ^2-test is not suited best to

spot a notion of *statistical significant visual difference* and comes with its own problems as pointed out in [52]. There is no free lunch and thus no universal single test that solves the visualization recommendation in general.

Third, most tests only offer merely *asymptotic guarantees* because of the test statistic they use. Especially for skewed distributions or queries that return only a small number of rows this is problematic. Consider once again a χ^2-test and a heavily skewed distribution over e.g. 20 buckets. It is then very unlikely that the test statistic for the number of samples used in a visualization setting is already χ^2-distributed.

Fourth, recommendations based on tests are not necessarily *symmetric* in the sense that if the candidate query was used as reference query, the old reference query would not get necessarily recommended at all. This is especially true for the χ^2-test.

Lastly, one might be tempted to simply combine statistical testing with a magical cutoff or subsequent selection of visualizations based on the distance measure introduced in Section 6.2. I.e., an algorithm could be to first apply statistical testing to get a candidate set of potentially interesting visualizations, rank them after the distance measure and then select all visualizations as interesting that have a distance higher than ϵ_S with respect to the reference visualization. However, this approach merely delays the problem of potentially making false discoveries: Though according to the tests the visualizations may be indeed *different* according to the difference guarantees the employed tests offer, they do not necessarily need to be *significantly different* enough. This again can be shown using the example in Section 6.6.3.

6.3.2 Recommendation validation via estimation

The goal of `VizRec` is to provide an efficient and rigorous way to verify that two visualizations are indeed *statistically different* with "*finite-sample*" guarantees.

In VizRec we use the sample dataset $\mathcal{D}$, and the visualizations obtained from it, in order to obtain *approximations* of the visualization according to the entire global sample space Ω using the Chernoff bound and later VC dimensions.

Consider a single histogram visualization $\mathcal{V}_1$, and assume it is comprised of K bars, one for each of the K possible values of the chosen group-by feature X. Let $p_{\mathcal{V}_1}(x_1), \ldots, p_{\mathcal{V}_1}(x_k)$, denote the normalized bars corresponding to $\mathcal{V}_1$. Note that such bars denote the probability of a randomly chosen record from Ω being such that $X = x_i$ conditioned on the fact that such record satisfies the predicate associated with $\mathcal{V}_1$.

Using the sample set $\mathcal{D}$ we compute an approximation for the $p_{\mathcal{V}_1}(x_i)$, which we denote as $\hat{p}_{\mathcal{V}_1}(x_i)$, following (6.3). As only these approximations can be computed from the available data, any choice regarding which visualizations should be recommended may depend only on such approximations.

In order to argue guarantees regarding the reliability of such decisions, it is necessary to bound the maximum difference between the correct and estimated sizes of bars in the normalized histograms.

In particular for a given δ (i.e., our level of control for *false positive recommendations*) we want

to compute the minimum value $\epsilon \in (0, 1)$

$$\Pr_{\mathcal{D}}\left(|p_{\mathcal{V}_1}(x_i) - \hat{p}_{\mathcal{V}_1}(x_i)| > \epsilon\right) < \delta.$$

Such value ϵ would in turn quantify the accuracy of the estimation of the $p_{\mathcal{V}_1}(x_i)$'s obtained by means of their empirical counterpart $\hat{p}_{\mathcal{V}_1}(x_i)$'s.

Let F denote the predicate associated with our visualization $\mathcal{V}_1$. We denote as $\Omega|F$ (resp., $\mathcal{D}|F$) the subset of Ω (resp., $\mathcal{D}$) which is composed by those records that satisfy the predicate F. Given a choice of group-by attribute X the value $p_{\mathcal{V}_1}(x_i)$ (resp., $\hat{p}_{\mathcal{V}_1}(x_i)$) corresponds to (resp., is computed as) the *relative frequency* of records such that $X = x_i$ in $\Omega|F$ (resp., $\mathcal{D}|F$).

Fact 6.1. *Let $\mathcal{D}$ be an uniform random sample of Ω composed by m records. For any choice of predicate, as specified in Definition 6.0.1, the subset $\mathcal{D}|F$ is a uniform random sample of $\Omega|F$ of size $|\mathcal{D}|F|$.*

Fact 6.1 is a straightforward consequence of the fact that $\mathcal{D}$ is an uniform sample of Ω.

From Fact 6.1 and from the definitions of the $p_{\mathcal{V}_1}(x_i)$'s and of the $\hat{p}_{\mathcal{V}_1}(x_i)$'s, clearly follows that the $\hat{p}_{\mathcal{V}_1}(x_i)$'s are *unbiased estimators* for the $p_{\mathcal{V}_1}(x_i)$'s. That is for, every value of the group-by feature we have:

$$\mathrm{E}_{\mathcal{D}}\left[\hat{p}_{\mathcal{V}_1}(x_i)\right] = p_{\mathcal{V}_1}(x_i). \tag{6.4}$$

In order to bound the estimation error $|p_{\mathcal{V}_1}(x_i) - \hat{p}_{\mathcal{V}_1}(x_i)|$ is therefore sufficient to bound the *deviation from expectation* (i.e., $p_{\mathcal{V}_1}(x_i)$) of the empirical estimate (i.e., $\hat{p}_{\mathcal{V}_1}(x_i)|$).

Chernoff-Bounds [163], allows to obtain such bounds as:

$$\Pr_{\mathcal{D}}\left(|p_{\mathcal{V}_1}(x_i) - \hat{p}_{\mathcal{V}_1}(x_i)| > \epsilon\right) \leq e^{-2|\mathcal{D}|F|\epsilon^2} \tag{6.5}$$

Recall our definition of *selectivity of a visualization* as $\gamma_{\mathcal{V}_1} = \frac{|\mathcal{D}|F|}{\mathcal{D}}$, we can then rewrite (6.5) as:

$$\Pr_{\mathcal{D}}\left(|p_{\mathcal{V}_1}(x_i) - \hat{p}_{\mathcal{V}_1}(x_i)| > \epsilon\right) \leq e^{-2\gamma_{\mathcal{V}_1} n\epsilon^2}, \tag{6.6}$$

where $n = |\mathcal{D}|$. A clear consequence of (6.6), is that the higher (resp., the lower) the selectivity of a visualization, the higher (resp., the lower) the quality of the estimate.

In [129], the authors use the same kind of bounds to develop a sampling algorithm which ensures the visual property of *relative ordering*. That is, for any pairs of vars corresponding to two possible values x_1 and x_2 of the same group-by feature X, if $\hat{p}_{\mathcal{V}_1}(x_1) > \hat{p}_{\mathcal{V}_1}(x_2)$, then, with high probability, $p_{\mathcal{V}_1}(x_1) > p_{\mathcal{V}_1}(x_2)$ holds as well.

While the method previously described based on an application of the Chernoff bound appears to be very useful and practical, it is important to remark that in a single application it may only offer guarantees on the quality of the approximation of *one* bar from a *single visualization*.

While it is in general possible to combine multiple applications of the Chernoff bound, the required correction leads to a quick and marked decrease of the quality of the bound. As an example if our visualization $\mathcal{V}_1$ is composed by K bars, a bound on the quality of the approximation of all of the bars will result in:

$$\Pr_{\mathcal{D}}\left(\max_{i=1,\ldots,K} |p_{\mathcal{V}_1}(x_i) - \hat{p}_{\mathcal{V}_1}(x_i)| > \epsilon\right) \leq Ke^{-2\gamma_{\mathcal{V}_1} n\epsilon^2}.$$

This bound is obtained using the *union bound* [163]. While potentially tolerable for a small value of K, for large values the performance decrease can possibly lead to a complete loss of significance of the bound itself.

6.3.3 Correcting for Adaptive Multi-Comparisons

The previous section showed how we can evaluate the interest of a single visualization as a statistical test. However, this does not yet control the multiple comparisons problem which is inherent in the recommendation system process that explores many possible visualizations. Clearly, if we let a recommendation system explore an unlimited number of possible visualizations, it will eventually find an "*interesting*" one, even in random data. How do distinguish real interesting visualization from spurious ones that are the results of random fluctuation in the sample?

The simplest and safest method to avoid the problem is to test every visualization recommendation on an independent sample not used during the exploration process that led to this recommendation. While easy and safe, this method is clearly not practical for a process that explores many possible visualizations. Data is limited, and we cannot set aside a holdout sample for each possible test. The process needs access to as much data as possible in the exploration process in order to discover all interesting insights. Can we control the generalization error when computing a number of visualizations based on one set of data?

Assume that in our exploration of possible interesting visualizations we tried ℓ different visualization patterns, and we computed for each of these patterns a bound h_i, $i = 1, \ldots, \ell$, on the probability that the corresponding observation in the sample $\mathcal{D}$ does not generalize to the distribution with respect to the entire global sample space Ω. It is tempting to conclude that the probability that any of the ℓ visualizations does not generalized is bounded by $\sum_{i=1}^{\ell} h_i$. Unfortunately, this probability is actually much larger when the choice of the tested visualization depends of the outcome of prior tests.

This phenomenon is often referred to as Freedman's paradox [] and the only known practical approach to correct for it is to sum the error probability of all possible tests, not only the tests actually executed [2]. Note that standard statistical techniques for controlling the Family-Wise-Error-Rate (FWER) or the False Discovery Rate (FDR) require that the collection of tests is fixed independent of the data and therefore do not apply to the adaptive exploration scenario.

In the visualization setting, we could decide a-priory that we only consider visualizations of a particular set of patterns, say conditioning on no more than k features. Such restriction defines a bound on the total size of the search space, say M. If we now explore the search space and recommend visualization that pass the individual visualization test with confidence level $\leq \alpha/M$ we are guaranteed that the probability that any of our recommendations does not generalize is bounded by α. As we show in the experiments section this method is only effective for relatively small search space. Next, we present a novel technique that is significantly more powerful in large and more complex search spaces.

[2] Theoretical methods, such as differential privacy [59] claim to offer an alternative method to address this issue. In practice however, the signal is lost in the added randomization before it becomes practical.

6.4 Statistical Guarantees Via Uniform Convergence Bounds

We now present a powerful alternative approach, providing strong and practical statistical guarantees through a novel application of VC-dimension theory.

VC-dimension is usually considered a highly theoretical concept, with limited practical applications, mostly because of the difficulty in estimating the value of the VC-dimension of interesting learning concept class.

Surprisingly, we develop a simple and effective method to compute the VC-dimension of a class of predicate queries which are used to generate the visualizations according to Definition 6.0.1, which leads to a practical and efficient solution to our problem. We start with a brief overview of the VC theory and its application to sample complexity. We then discuss its specific application to our visualization problem, emphasizing a simple, efficient, and easy to compute reduction of the theory to practical applications.

6.4.1 VC dimension

The Vapnik-Chernovenkis (VC) dimension is a measure of the complexity or expressiveness of a family of indicator functions (or equivalently a family of subsets) [230]. Formally, VC-dimension is defined on *range spaces*:

Definition 6.1.1. *A* range space *is a pair (X, R) where X is a (finite or infinite) set and R is a (finite or infinite) family of subsets of X. The members of X are called* points *and those of R are called* ranges*.*

Note that both X and R can be infinite. Consider now a projection of the ranges into a finite set of points A:

Definition 6.1.2. *Let (X, R) be a range space and let $A \subset X$ be a finite set of points in X.*

1. *The* projection *of R on A is defined as*
$$P_R(A) = \{r \cap A \ : \ r \in R\}.$$
2. *If $P_R(A) = 2^{|A|}$, then A is said to be* shattered by *R.*

The VC-dimension of a range space is the cardinality of the largest set shattered by the space:

Definition 6.1.3. *Let (X, R) be a range space. The* VC-dimension *of (X, R), denoted $\mathsf{VC}(X, R)$ is the maximum cardinality of a shattered subset of X. If there are arbitrary large shattered subsets, then $\mathsf{VC}(X, R) = \infty$.*

Note that a range space (X, R) with an arbitrarily large (or infinite) set of points X and an arbitrary large family of ranges R can have bounded VC-dimension (see section 6.4.2).

A simple example is the family of closed intervals in $\mathbb{R}$, where $X = \mathbb{R}$, and R corresponds to the (infinite) set of all possible closed intervals intervals $[a, b]$, such that $\forall a, b \in \mathbb{R}$ we have $0 \leq a \leq b \leq 1$). Let $A = \{x, y, z\}$ be *any* subset of $\mathbb{R}$ such that $x \leq y \leq z$. No interval in R can define the subset

$\{x, z\}$ so the VC-dimension of this range space is < 3. This observation is generalized in the well known result:

Lemma 6.2. *The VC-Dimension of the union of d closed intervals in $\mathbb{R}$ equals $2d$.*

VC-dimension, allows to characterize the sample complexity of a learning problem, that is it allows to obtain a tradeoff between the number of sample points being observed by a learning algorithm and the performances achievable by the algorithm itself.

Consider a range space (X, R), and a fixed range $r \in R$. If we sample uniformly at random a set $S \subset X$ of size $m := |S|$ we know that the fraction $\frac{|S \cap r|}{|S|}$ rapidly converges to the frequency of elements of r in X. Furthermore, there are standard bounds (Chernoff, Hoeffding []) for evaluating the quality of this approximation. The question becomes much harder when we want to estimate simultaneously the sizes or frequencies of all ranges in R using one sample of m elements. A finite VC-dimension implies an explicit upper bound on the number of random samples needed to achieve that within pre-defined error bounds (the *uniform convergence property*).

For a formal definition we need to distinguish between finite X, where we case estimate the sizes r, and infinite X, where we estimate $Pr(r)$, the frequency of r in a uniform distribution over X.

Definition 6.2.1 (Absolute approximation)**.** *Let (X, R) be a range space and let $0 \leq \epsilon \leq 1$. A subset $S \subset X$ is an absolute ϵ-approximation for X iff for all $r \in R$ we have that for finite $S \subseteq X$,*

$$\left| \frac{|r|}{|X|} - \frac{|S \cap r|}{|S|} \right| \leq \epsilon. \tag{6.7}$$

In [99] show an interesting connection between the VC dimension of a range space (X, R) and the number of samples which are necessaries in order to obtain absolute ϵ-approximations of X itself

Theorem 6.3 (Sample complexity [99])**.** *Let (X, R) be a range-space of VC-dimension at most d, and let and $0 < \epsilon, \delta < 1$. Then, there exists an absolute positive constant c such that any random subset $S \subseteq X$ of cardinality*

$$|S| \geq \frac{c}{\epsilon^2} \left(d + \log_2 \delta^{-1} \right) \tag{6.8}$$

is an ϵ-approximation for X with probability at least $1 - \delta$.

The constant c was shown experimentally [150] to be at most 0.5[3].

6.4.2 Statistically Valid Visualization through VC dimension

To apply the uniform convergence method via VC dimension to the visualization setup, we consider a range space (Ω, R), where Ω is global domain, and R consists of all the possible subsets of X that can be selected by visualizations predicates. That is, R includes all the subsets that correspond to *any* bar for *any* visualization which can be selected using the appropriate predicate filter. Given a choice of possible allowed predicates, we refer to the associate set of ranges as the "*query range space*" and we denote it Q.

[3]Indeed, we use $c = 0.5$ in our experimental evaluation.

The VC dimension of a query range class is a function of the type of select operators (i.e., $>, <, \geq, \leq, =, \neq$) and the number of (non-redundant) operators allowed on each feature in the construction of the allowed predicates. Note that depending on the *domain* of the selected features and the complexity according to which the predicate filters can be constructed, the number of possible predicates may be infinite. In order to use the VC-approach it is however sufficient to efficiently compute a finite *upper bound* of the VC-dimension of the set of *allowed predicates*. We discuss an efficient method for bounding the VC-dimension of a query range space in the next subsection (See Subsection 6.4.2).

In order to deploy the general results from the previous section, we have to verify that the sample $\mathcal{D}$ provides an ϵ-approximation for the values $p_{\mathcal{V}}$ for all the visualizations considered in the query range space Q.

To this end, it is useful to introduce the following, well known, property of VC dimension:

Fact 6.4. *Let (X, R) be a range space of VC dimension d. For any $X' \subseteq X$, the VC-dimension of (X', R) is bounded by d.*

Using this fact in conjunction with Theorem 6.3 we have:

Lemma 6.5. *Let (Ω, Q) denote the range space of the queries being considered with VC dimension bounded by d, and let $\delta \in (0, 1)$. Let $\mathcal{D}$ be a random subset of Ω. Then there exists a constant c, such that with probability at least $1 - \delta$ for any filter F defined in Q we have that the subset $\mathcal{D}|F$ is an ϵ_F-approximation of $\Omega|F$ with:*

$$\epsilon_F \geq \sqrt{\frac{c}{|D|F|}\,(d + \log_2 \delta^{-1})}.$$

Proof. Fact 6.1 ensures that given the dataset $\mathcal{D}$, for any choice of a predicate F we have that $\mathcal{D}|F$ is a random sample of $\Omega|F$. Therefore regardless of the specific choice of the predicate, we have that the VC dimension of the reduced range $(\Omega|F, Q)$ is bounded by d. From Theorem 6.3 we have that if:

$$|\mathcal{D}| \geq \frac{c}{\bar{\epsilon}}\left(d + \log_2 \delta^{-1}\right) \tag{6.9}$$

then $\mathcal{D}|F$ is an $\bar{\epsilon}$ approximation for the respective set $\Omega|F$. □

Lemma 6.5 provides us an efficient tool to evaluate the quality of our estimations $\hat{p}_{\mathcal{V}}$ of the actual ground truth values $p_{\mathcal{V}}$ for any choice of predicate associated with the visualization. In particular, Lemma 6.5 verifies that the quality decreases gradually the more *selective* the predicate associated with a visualization is. That is, the smaller the cardinality of $|\mathcal{D}|F|$, the higher the *uncertainty* $eps\bar{i}lon$ of the estimate is.

Corollary 6.6. *Let $\mathcal{D}$ be a random sample from Ω, and let Q be a query range space with VC dimension bounded from above by d. For any visualization with $\mathcal{V} \in Q$ and for any value $\delta \in (0, 1)$ we have that*

$$\Pr\{\Big|p_{\mathcal{V}}(X = x_i) - \hat{p}_{\mathcal{V}(X=X_i)}\Big| \geq \bar{\epsilon}\} < \delta, \tag{6.10}$$

where

$$\bar{\epsilon} \geq \frac{c}{|D|F|} \left(d + \log_2 \delta^{-}1\right), \tag{6.11}$$

F denotes the predicate associated with the visualization $\mathcal{V}$ and and X denotes the group-by feature being considered.

6.4.3 The VizRec recommendation validation criteria

Consider now a given reference visualization $\mathcal{V}_1$ and a candidate recommendation $\mathcal{V}_2$, both using X as the group-by feature, were the the domain of X has K values (i.e., $dom(X) = \{x_1, \ldots, x_K\}$). From Lemma 6.5, we have that with probability $1 - \delta$ the empirical estimates of the normalized columns are accurate within $\bar{\epsilon}$, which depends on the size of the subset of the sample dataset $\mathcal{D}$ used for the reconstruction of $\mathcal{V}_2$.

As argued in Section 6.3, we consider a candidate visualization worth of being recommended if it represents a different statistical behavior of the group-by feature with respect to the reference query.

Our VizRec strategy operated by comparing the values $\hat{p}_{\mathcal{V}_1}(x_i)$ and $\hat{p}_{\mathcal{V}_2}x_i$ for all values x_i in the domain of the chosen group-by feature. Let $\bar{\epsilon_1}$ (resp., $\bar{\epsilon_2}$) denote the uncertainty such that with probability at least $1 - \delta$ we have $\|p_{\mathcal{V}_1}(x_i) - \hat{p}_{\mathcal{V}_1}(x_i)\| \leq \bar{\epsilon_1}$ and $\|p_{\mathcal{V}_2}(x_i) - \hat{p}_{\mathcal{V}_2}(x_i)\| \leq \bar{\epsilon_2}$ accoding to Lemma 6.5. If it is the case that $|\hat{p}_{\mathcal{V}_1}(x_i) - \hat{p}_{\mathcal{V}_2}(x_i)| > \bar{\epsilon_1} + \bar{\epsilon_2}$ then we can conclude that with probability at least $1 - \delta$ we have $p_{\mathcal{V}_1}(x_i) \neq p_{\mathcal{V}_2}(x_i)$.

That is VizRec recognizes as *statistically different* (and hence, interesting) only pairs of visualizations for which the most different pair of corresponding columns differs by more than the error in the estimations from the sample. If that is the case, it is possible to guarantee that $\mathcal{V}_1$ and $\mathcal{V}_2$ are indeed according to the Chebyschev measure. Due to the uniform convergence bound ensured by the application of VC dimension, we can have that the probabilistic guarantees of this control hold *simultaneously* for all possible pairs of reference and candidate recommendation visualizations. The advantage of this approach compared to the use of multiple Chernoff bounds is discussed in Appendix 6.6.4. Further, our VC dimension approach is *agnostic* to the adaptive nature of the testing as it accounts *preventively* for *all* possible evaluations of pairs of visualizations. Threrefore, we have:

Theorem 6.7. *For any given $\delta \in (0, 1)$, VizRec ensures FWER control at level δ while offering visual recommendations.*

This criteria can be strengthened by imposing a higher threshold of difference between two visualization in order for a candidate visualization to be considered interesting. As an example, in Section 3.2, we discuss the possible use of a threshold ϵ_V denoting visual discernability. When using this, more restrictive constraint, VizRec would accept a candidate visualization as interesting only if

$$\max_{x_i \in dom(X)} |\hat{p}_{\mathcal{V}_1}(x_i) - \hat{p}_{\mathcal{V}_2}(x_i)| > \max\{\bar{\epsilon_1} + \bar{\epsilon_2}, \epsilon_{\mathcal{V}}\}.$$

We present a simplified psudocode of our VizRec procedure in Algorithm 9.

ALGORITHM 9 VizRec: Visual Recommendations with VC dimension

procedure VizRec
Input: Starting visualization $\mathcal{V}_1$, query space Q, sample dataset $\mathcal{D}$, FWER target control level $\delta \in (0, 1)$.
Output: A set of Y statistically safe recommendations sorted according to decreasing interest.
  $Y \leftarrow []$ ▷ Empty list of recommendations
  $X \leftarrow$ the group-by feature being considered.
  $F_{\mathcal{V}_1} \leftarrow$ the predicate associated with $\mathcal{V}_1$.
  $\bar{\epsilon}_1 \leftarrow \frac{d+\log_2 \delta^{-1}}{2|D|F_{\mathcal{V}_1}|}$ ▷ Uncertainty in $\mathcal{V}_1$ approx.
  for all $\mathcal{V}' \in Q$ **do**
    $F_{\mathcal{V}'} \leftarrow$ the predicate associated with $\mathcal{V}'$.
    $\bar{\epsilon}' \leftarrow \frac{d+\log_2 \delta^{-1}}{2|D|F_{\mathcal{V}'}|}$
    $dist \leftarrow \max_{x_i \in dom(X)} |\hat{p}_{\mathcal{V}_1}(x_i) - \hat{p}_{\mathcal{V}_2}(x_i)|$
    $interest \leftarrow dist - (\bar{\epsilon}_1 + \bar{\epsilon}_2)$
    if $dist \geq \bar{\epsilon}_1 + \bar{\epsilon}_2$ **then**
      $Y.append([\mathcal{V}', interest]$
    or if stricter criteria with ϵ_V
    if $dist \geq \max\{\bar{\epsilon}_1 + \bar{\epsilon}_2, \epsilon_V\}$ **then**
      $Y.append([\mathcal{V}', interest]$
  return sort Y according to the interest value.

Our VizRec approach operates as the equivalent of a two-sample test, in the sense that we assume that in general there is uncertainty in the reconstruction of both the reference visualization $\mathcal{V}_1$ and of the candidate $\mathcal{V}_2$. In some scenarios, it may be possible to assume that the reference visualization is given *as exact*. In such case in order for the candidate $\mathcal{V}_2$ to be recommended it would be sufficient that

$$\max_{x_i \in dom(X)} |\hat{p}_{\mathcal{V}_1}(x_i) - \hat{p}_{\mathcal{V}_2}(x_i)| > \bar{\epsilon_2}.$$

After identifying as set of recommendations whose interest is guaranteed with probability at least $1 - \delta$, VizRec ranks them according to the difference between their "*empirical interest*" (i.e., $\max_{x_i \in dom(X)} |\hat{p}_{\mathcal{V}_1}(x_i) - \hat{p}_{\mathcal{V}_2}(x_i)|$) and the uncertainty of the evaluation of such measure (i.e., $\bar{\epsilon_1} + \bar{\epsilon_2}$). While somewhat arbitrary, we chose this heuristic as it allows to emphasize the intrinsic value of visualizations with large support over those with small support.

6.4.4 The VC dimension of the Range Space

In order to actually deploy the VC dimension bounds previously discussed it is necessary to bound the VC dimension of the class of queries being considered. While challenging in general, we develop here a simple and effective bound on the VC dimension of the class of queries being considered based on the complexity of the constraints defining the predicates.

As discussed in Section 6.2.2, we assume that the values of the features can be mapped to real numbers. Hence constraint of values of a certain feature formalized using the operators $\geq, \leq, =$ and $\neq$ correspond to selecting intervals (either open or close) of the possible values of a feature. For each feature, the various clauses are *connected* by means of "`or`" operators. We characterize the

complexity of such connection by the minimum number of *non-redundant* open and close intervals of the value. In particular we say that a connection of intervals is *non-redundant* is there is no connection of fewer intervals that selects the same values.

The VC dimension of a class of queries can then be characterized according to the number of non-redundant constraints applied to the various features.

Lemma 6.8. *Let $\mathcal{Q}$ denote the class of query functions such that each query is a conjunction of connections of clauses on the value of distinct features. The VC dimension of Q is:*

$$VC\left(Q\right) = \sum_{i=1}^{m} 2\alpha_i + \beta_i, \tag{6.12}$$

where α_i (resp., β_i) denotes the maximum number of non-redundant closed (resp., open) intervals of values corresponding to the connection of constraints regarding the value of the i-th feature, for $1 \leq i \leq m$.

Proof. The proof is by induction on i: in the base case we have $i = 1$. In this case, the VC dimension of $\mathcal{Q}$ corresponds to the VC dimension of the union of α_1 closed intervals and β_1 open intervals on the line. By a simple modification of the result of Lemma 6.2, we have that it has VC dimension at most $2\alpha_1 + \beta_1$. Let us now inductively assume that the statement holds for $i > 1$. In order to conclude the proof we shall verify that it holds for $i + 1$ as well.

Assume towards contradiction that there exists a set X of $\sum_{j=1}^{i+1} 2\alpha_j + \beta_j$ points that can be shattered by $\mathcal{Q}$. From the inductive hypothesis, we have that for any subset of X with more than $\sum_{j=1}^{i} 2\alpha_j + \beta_j$ cannot be shattered by the family of query functions which can express constraints only on the features $1, 2, \ldots, i$. Without loss of generality let X' denote one of the maximal subsets of X which can be shattered using only the constraints on the features $1, 2, \ldots, i$. The following fact is important for our argument

Fact 1: Recall that the queries in $\mathcal{Q}$ are constituted by logical conjunctions (i.e., "`and`") of connections (i.e.,"`or`" statement) of constraints on a feature. Hence, for any function in $\mathcal{Q}$ if any of the $i + 1$ connections are such that they assume value "`false`", then the query will not select such point *regardless* of the value of the remaining i connections being conjuncted.

Consider any assignment π of $\{0, 1\}$ to the points in X' and let r_π the range which realizes such shattering.

If r_π would assign to any point in $X \setminus X'$ value "0", then, according to the structure of the queries, no constraint on the $(i+1)$-th feature would allow to assign to it value "1", and, hence, it would not be possible to shatter X.

Note that for any assignment π of $\{0, 1\}$ to the points in X', there may may not exists two ranges r_1 and r_2 such that based solely on constraints on the first i features, on would assign "0" to a point in $X \setminus X'$ and the other would assign "1" to the same point. If that would be the case, then i would be possible to shatter $\sum_{j=1}^{i} 2\alpha_j + \beta_j$ points using just constraints on the first i features and this would violate the inductive hypothesis.

Without loss of generality, in the following we can therefore assume that for any assignment π of $\{0, 1\}$ to the points in X' the ranges that realize such assignment just based on the first i features

would assign "1" to all the points in $X \setminus X'$. This implies that the shattering of the points in $X \setminus X'$ relies *solely* on the constraints on the values of the $i+1$-th feature.

Consider now the points in $X \setminus X'$, according to our assumption $|X \setminus X'| = 2\alpha_{i+1} + \beta_{i+1}$. As discusses in the base of the induction, it is not possible to shatter $2\alpha_{i+1} + \beta_{i+1}$ points using just α_{i+1} (resp., β_{i+1}) closed (resp., open) intervals on the $(i+1)$-th dimension.

Hence it is not possible to shatter X and we have a contradiction. □

The proof for Lemma 6.8, proceeds by induction of the number of features which can be used in constructing the queries, and builds on known results on the VC dimension of a class of functions constituted by the union of a finite number of closed intervals on $\mathbb{R}$.

ALGORITHM 10 Reduction to non-redundant intervals

procedure ReduceIntervals
Input: A conjunction of k clauses expressed as open or closed intervals
Output: An equivalent non-redundant conjunction of clauses
$C \leftarrow False$ ▷ Initialization non redundant intervals
if Any constraints of the kind "$c \leq a_i$" given as input **then**
$l_{\leq} \rightarrow \max a_i$ such that $c \leq a_i$ is clause;
else
$r_{\geq} \rightarrow \min a_i$ such that $c \geq a_i$ is clause;
$l_{<} \rightarrow \max a_i$ such that $c \neq a_i$ is clause;
$r_{>} \rightarrow \min a_i$ such that $c \neq a_i$ is clause;
$I \leftarrow$ list of closed intervals $(a_i \leq c \leq b_i)$ sorted according to the values of the a_i increasingly.
$op_l = \leq$
if $\exists (a_i, b_i) \in I | a_i \leq \max\{l_{\leq}, l_{<}\}$ **then**
$l' \leftarrow \max\{b_i | (a_i, b_i) \in I \text{ and } a_i \leq \max\{l_{\leq}, l_{<}\}\}$;
else
$l' \leftarrow \max\{l_{\leq}, l_{<}\}$;
if $l' \neq l_{\leq}$ **then**
$op_l \leftarrow <$
$op_r = \geq$
if $\exists (a_i, b_i) \in I | b_i \geq \min\{r_{\geq}, r_{>}\}$ **then**
$r' \leftarrow \min\{a_i | (a_i, b_i) \in I \text{ and } b_i \geq \min\{r_{\geq}, r_{>}\}\}$;
else
$r' \leftarrow \min\{r_{\geq}, r_{>}\}$;
if $r' \neq r_{\geq}$ **then**
$op_r \leftarrow >$
if $l' > r'$ **then**
return $C \leftarrow$ `True`
else if $l' = r'$ and $\big((op_l = \leq) \text{ or } (op_r = \geq)\big)$ **then**
return $C \leftarrow$ `True`
else
$C \leftarrow (c\ op_l\ l') \vee (c\ op_r\ r')$
Update I by removing all intervals (a_i, b_i) such that $b_i \leq l'$ or $a_i \geq r'$;
$I' \leftarrow \emptyset$
while $I \neq \emptyset$ **do** ▷ Remove intervals contained in larger intervals
$a \leftarrow \min\{a_i | (a_i, b_i) \in I\}$
$b \leftarrow \max\{b_i | (a, b_i) \in I\}$
$I' \leftarrow I' \cup \{(a, b)\}$
$I \leftarrow I \setminus \{(a_i, b_i) | a_i \geq a \text{ and } b_i \leq b\}$
while $I' \neq \emptyset$ **do** ▷ Merge intervals with superpositions
$a \leftarrow \min\{a_i | (a_i, b_i) \in I'\}$
$b \leftarrow b | (a, b) \in I')$
if $\exists b_i | (a_i, b_i) \in I' \text{ and } a_i \leq b$ **then**
$b' \leftarrow \max\{b_i | (a_i, b_i) \in I' \text{ and } a_i \leq b\}$
$I \leftarrow I \setminus \{(a_i, b_i) | a_i \geq a \text{ and } b_i \leq b'\}$
$I' \leftarrow I' \cup \{(a, b')\}$
else
$I' \leftarrow I' \cup \{(a, b)\}$
$C \leftarrow C \vee (a \leq c \leq b)$
return C ▷ Output non-redundant constraints

Bounding the VC dimension of a given query class

The following Algorithm 10 outlines a procedure which reduces an input query class Q to an equivalent non-redundant version and computed a bound on its VC dimension. Algorithm 10 operates "*consolidating*" redundant clauses in and equivalent minimal set. Such consolidation is achieved by first (lines 2-10) determining the minimal open intervals for each feature, and then by merging overlapping closed intervals whenever possible. Note that given a specific choice of a query class, Algorithm 10, needs to be run just once to bound its VC dimension.

6.4.5 Trade-off between query complexity and minimum allowable selectivity

Consider exploring the space of possible recommendations by constructing the filter condition one clause at a time. Note that as the filter condition grows in its complexity (i.e., multiple non-trivial clauses are added) the number of records selected by the predicate, and hence, it selectivity, will decrease. It appears therefore reasonable to start evaluating *simpler* predicate filters and then proceed *depth-first* by adding more and more clauses. While reasonable, such procedure will possibly lead to explore large number of queries. However, most of the filters obtained by composing a high number of filters will likely lead to visualizations supported by very little sample points, and, hence, intrinsically *unreliable*.

Our VC dimension approach allows the system to recognize this fact and to use it in order to limit the search space. As discussed in Section 6.4.5, the lower the selectivity γ of the filter condition F of a given visualization, the higher the uncertainty $\bar{\epsilon}$,

$$\bar{\epsilon} \geq \sqrt{\frac{d + \log_2 \delta^{-1}}{2n} \gamma^{-1}}$$

In order for the difference between a candidate and the starting visualization to be deemed *statistically relevant* their Chebyshev distance has to be higher than $\bar{\epsilon}$. The Chebyshev norm distance, and, hence, the maximum *interest* of a candidate visualization is one. This clearly implies that all visualizations whose selectivity γ is such that

$$\gamma \leq \frac{d + \log_2 \delta^{-1}}{2n} \tag{6.13}$$

are not going to be interesting according to our procedure, and, hence, when exploring the space of possible recommendations, we can stop refining the queries once the selectivity of the candidate visualization drops below the threshold given by (6.13). This allows to *prune* the search space by eliminating from the exploration queries which are "*not worth to be considered*" as possible recommendations.

While this may appear as a weakness of our approach, it is instead consistent with the basic principle of distinguishing statistically relevant phenomena from effects of the noise introduced by the sampling process. While visualizations which involve just a very limited number of sample points may appear to represent a very interesting distribution of a subset of the data, they are more

likely to represent random fluctuation in the selection of the input sample than a true phenomenon in the global domain.

By taking into consideration the selectivity of candidate visualizations, our method *automatically adjusts* the threshold of interest for candidate visualization.

6.5 Discussion

In this section, we discuss and motivate some guidelines to help the analyst determine which of the discussed tools are better suited for her actual setting.

6.5.1 Number of hypotheses being tested

Consider a scenario for which the system is limited to the analysis of a small number of possible visualizations. In this case, it is possible to evaluate which of these candidate visualizations are actually interesting with respect to the starting visualization V_1 by applying the χ^2 testing with opportune correction for the number of hypotheses being tested, which in this case would correspond to the number of candidate visualizations.

Further, if in the same exploration session the analyst wants also to evaluate which of the visualizations are interesting with respect to a different starting visualization V_2, this would require to treat all the additional candidate recommendations as additional hypotheses. In order to obtain FWER it will be necessary to correct for the number of hypotheses being tested, thus resulting in a loss of statistical power. Even though some FWER corrections such as the Holm or the Höchberg procedure allows to reduce the effect of the multiple hypotheses correction, compared to the the simple Bonferroni procedure, there is still a considerable decrease in the statistical power, in particular for settings for which only a low fraction of the hypotheses being tested have low p-values. Therefore the χ^2 testing approach appears to be more suitable for settings with a limited number of candidate recommendations being evaluated.

6.5.2 Bounding the complexity of the query class

The properties of our method can also be used to determine a bound the VC-dimension of the query range space being considered when looking to ensure that any candidate recommendation who differs from the reference visualization by at least θ is actually marked as a safe recommendation.

Let $\mathcal{V}_1$ (resp., $\mathcal{V}_2$) denote the reference (resp., a candidate) visualization, and let γ_1 (resp., γ_2) is selectivity.

Given the size of the available dataset $|\mathcal{D}|$, and the desired FWER control level $\delta \in (0, 1)$, the maximum VC dimension which guarantees to meet these requirements can be obtained from (6.13) as:

$$d \leq \theta^2 \min\{\gamma_1, \gamma_2\} n - \log_2(\delta^{-1}).$$

This bound can be used as a guideline to limit the structure of the queries being considered. That is, it offers indications on the number of different features which can be considered when building

the queries, and indications on their *complexity*, intended as the number of clauses being used in their construction (as discussed in Section 6.4.4).

6.5.3 Preprocessing heuristics

In this section we outline some preprocessing heuristics which allow to improve the effectiveness of our control procedures.

I. Removal of constant features: Features which assume the same value in all the records of the sample can be safely ignored.

II. Removal of identifier-type features: Features which assign a different *unique* value to each of the record can be removed. This is generally the case for *identifier* features (e.g.,"Street address" for real estate dataset). This appears justified as, due to the uniqueness of their values, they are not useful in constructing predicate conditions, nor they should be used as the "*group-by*" attribute (i.e., the attribute in the x-axis). All these heuristics share the fact that they allow to ignore some of the features (or columns) of the records. This will in turn impact both the search space and will allow to reduce the VC dimension of the query class being considered.

6.5.4 Modified χ^2-test

As described in [90] by modifying the χ^2-test to test a more complex null hypothesis

$$H_0 : \sum_{k=1}^{K} \frac{(p_k - \hat{p}_k)^2}{p_k} < \epsilon_{\chi^2} \tag{6.14}$$

together with the corresponding test statistic

$$T = n\gamma \sum_{k=1}^{K} \frac{(p_k - \hat{p}_k)^2}{p_k}$$

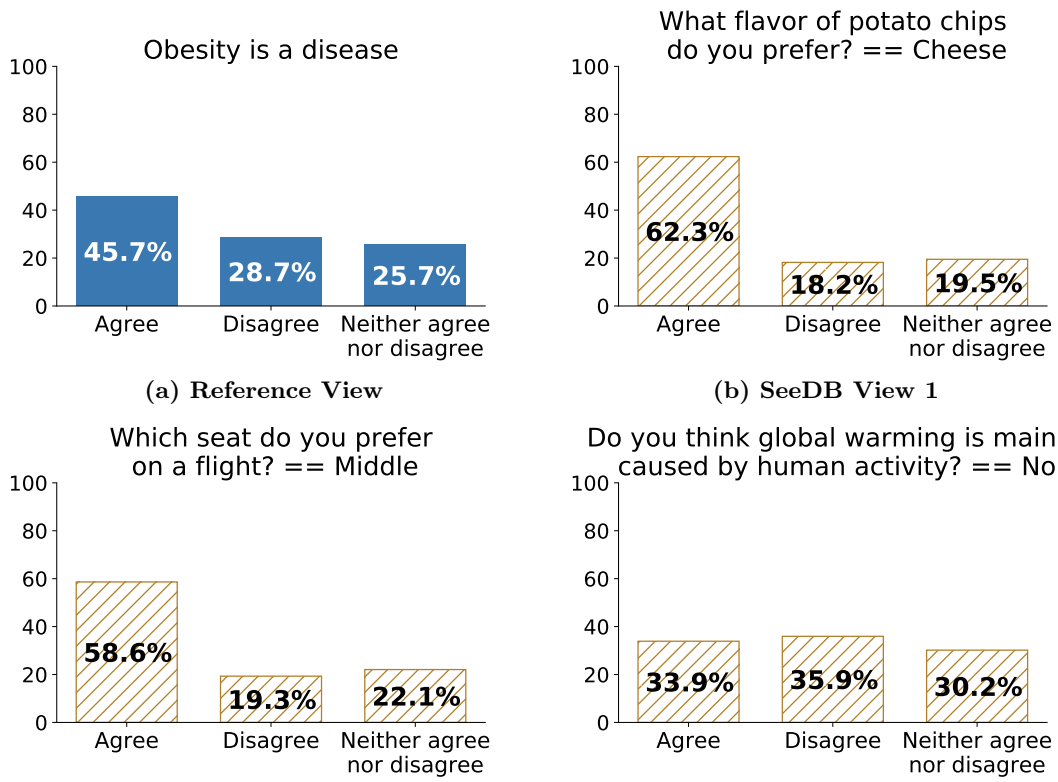
following a non-central χ^2 distribution with $K-1$ degrees of freedom and non-locality parameter

$$\lambda_{\text{nc}} = n\gamma\epsilon_{\chi^2} = m\epsilon_{\chi^2}.$$

The distance guaranteed then is the χ^2-distance d_{χ^2}. In fact, this testing procedure can be further generalized to use other distances that belong to the family of f-divergences like the total variation distance or Kullback-Leibler divergence via an suitable transformation or by using non-parametric models to estimate the distance measures before comparing them via statistical testing as in described in more detail in [123].

6.6 Experiments

In this section, we want to show how our framework can be applied towards both real data (i.e. the collected survey data) and synthetic data. We start by demonstrating different, problematic scenarios with our real dataset.



(a) Reference View **(b) SeeDB View 1**

(c) SeeDB View 2 **(d) SeeDB View 3**

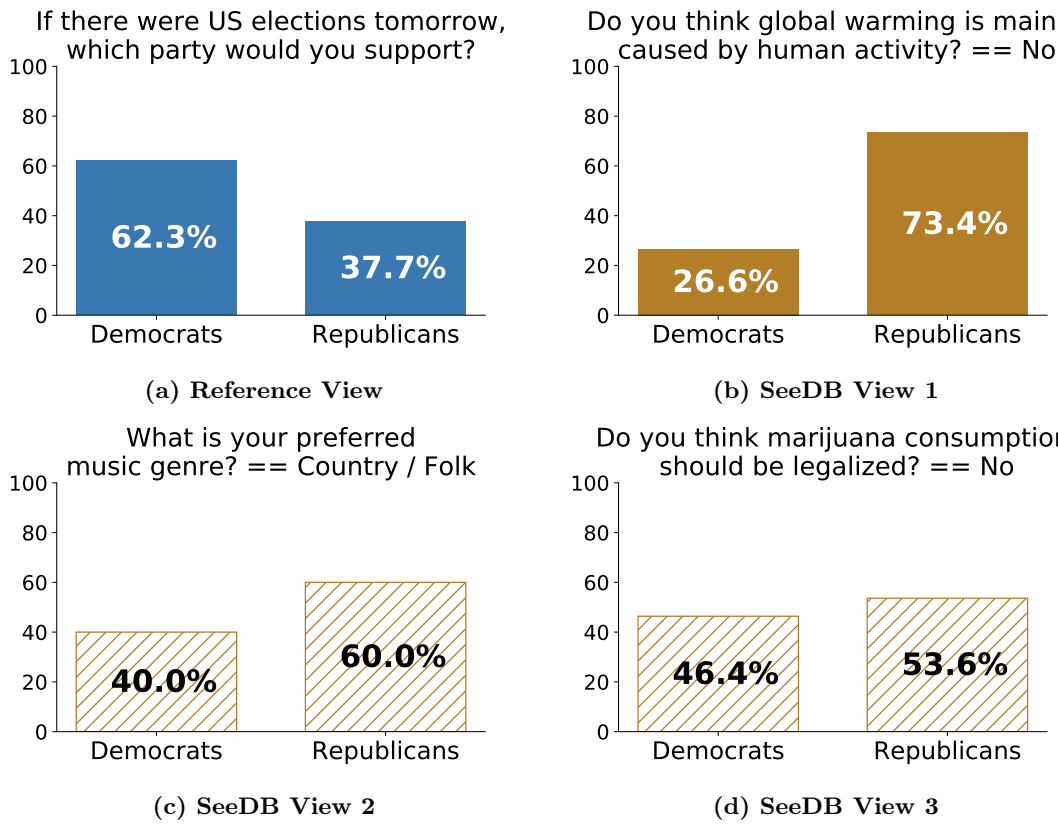
Figure 6.2: VizRec would not mark any of these visualizations as statistically significant as the difference computed with respect to the reference is not larger than the uncertainty of estimating the bars correctly.

6.6.1 Anecdotal examples

Our first example shows that a system without statistical control may trick the user to believe in insights that are actually not valid and merely random. For this, assume the user wants to explore whether there is a subpopulation that believes differently from the overall population when it comes to whether obesity is a disease or not.

As depicted in Section 6.2 a user may falsely believe that people who prefer potato chips with Cheese flavor are more likely to believe that obesity is a disease. Though people that consume potato chips may lean more towards a view that obesity is a disease, the insight that in particular people who prefer the Cheese flavor are the most interesting subpopulation is questionable. Since for all the other flavors in our study[4] no visualization is within the top results, picking particularly the Cheese flavour as a belief changer looks like a potential false discovery. The next recommended result seems even more random: An automatic recommender system would imply to the user that persons who prefer the middle seat are more likely to believe that obesity is a

[4]BBQ, Sea Salt & Vinegar, Sour Cream and Onion, Jalapeno, Cheddar and Sour Cream, Original/Plain, I don't eat chips



(a) Reference View **(b) SeeDB View 1**

(c) SeeDB View 2 **(d) SeeDB View 3**

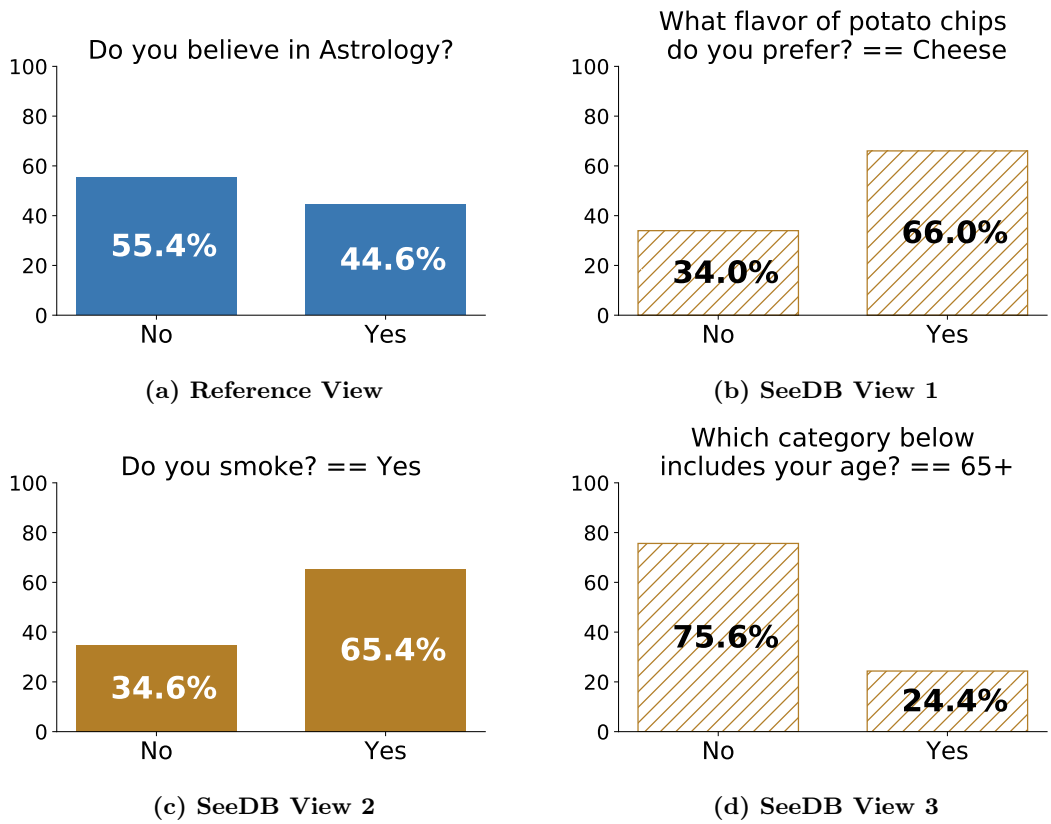
Figure 6.3: VizRec would also recommend the top visualization, but declares the other visualizations not being statistically significant enough.

disease. This seems hard to understand and more like some random result that the system produced.

In our second example, we want to show that VizRec may identify correctly the top SeeDB recommendation(s) as being statistically valid.

Here, the user was interested in finding out whether there is a subpopulation that has a different voting behavior for the two main parties in the United States (cf., Figure 6.3). The recommended top visualization coincides with the top SeeDB result and seems sound. However, VizRec prevents the user to attribute a preference towards the Republican party for persons who prefer to listen to Country and Folk music.

Finally, it can also be the case that SeeDB recommends something as the top result which the VizRec framework would rule out as being not significant at all. As depicted in Figure 6.4 again a questionable relation between people who prefer Cheese flavored potato chips and those who belief in Astrology would get recommended. However, this is actually a false discovery and not backed statistically. On the contrary, VizRec deemed a relation between smokers and a belief in Astrology statistically sound. In this example another interesting problem is demonstrated: Though the



(a) Reference View **(b) SeeDB View 1**

(c) SeeDB View 2 **(d) SeeDB View 3**

Figure 6.4: VizRec would not recommend the top visualization, but the second ranked one.

interestingness scores of the top results are pretty close to each other (e.g. here $d_\infty \approx 0.21$), the score d_∞ itself is not sufficient to determine statistical relevance. Particularly, employing a simple cutoff may lead to false discoveries. Besides the complexity of the exploration space, the number of samples used to estimate both the reference query and the candidate query need to be accounted for too.

These anecdotal examples demonstrate that without any statistical control the user is likely to run into making false discoveries. Our VC approach does not only account for the number of samples but also for the complexity of the data exploration and is thus a well-suited tool for avoiding false discoveries in a visual recommendation system.

6.6.2 Random data leads to no discoveries

A meaningful baseline for any safe visual recommendation system is to make sure that random data does not lead to any recommendations. To demonstrate that the VC approach will not recommend any false positives, we generated a synthetic dataset with uniformly distributed data. $100,000$ samples were generated in total with the first column being selected as aggregate and the other

3 columns as features. The aggregate is uniformly distributed over $\{1, 2, 3, 4\}$ and each of the 3 features are uniformly distributed over $\{1, ..., 9\}$.

With simple predicates (i.e. a queries formed from $\leq$ clauses solely) there are 1331 visualizations to be explored (a dummy value of $+\infty$ was used in the queries to make a feature active or not. E.g. consider a query of the form $(X_1 \leq 8) \wedge (X_2 \leq +\infty) \wedge (X_3 \leq 3)$. In this query, feature X_2 has no effect on the rows returned since $(X_1 \leq 8) \wedge (X_2 \leq +\infty) \wedge (X_3 \leq 3) \equiv (X_1 \leq 8) \wedge (X_3 \leq 3)$. Note that using $+\infty$-values in the clauses does not change the VC dimension.). As a reference, a uniform distribution over $\{1, 2, 3, 4\}$ was chosen. This means, that the expected support of any visualization is at least $10^5/9^3$ samples which is a fair amount to estimate 4 bars. When not accounting for the

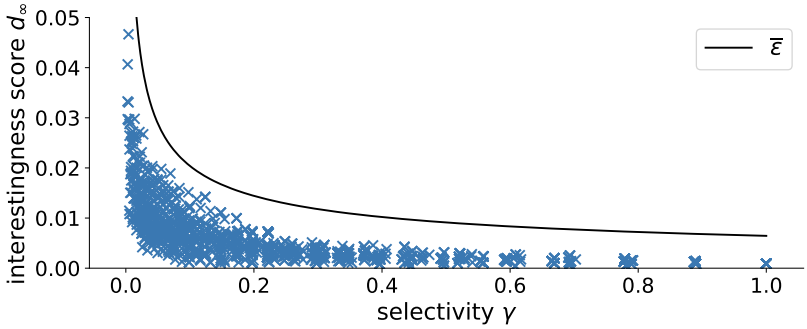


Figure 6.5: Blue dots represent interest scores all evaluated visualizations. The $\bar{\epsilon}$ curve denotes the threshold for recommendation using VC dimension = 4 to achieve control at level $\delta = 0.05$. The lower the VC dimension the more the $\bar{\epsilon}$ curve takes the form of an "L". Since there are no visualizations with scores higher than the $\bar{\epsilon}$-curve, no visualizations get recommended from the generated random data.

multiple comparison problem p-values below the threshold of $\alpha = 0.05$ occur inevitably. A system without FWER guarantees would classify them thus as false positives. Using Bonferroni (or other comparable corrections) remedies this while, however, incurring a noticeable loss in statistical power.

In comparison, the lowest ϵ the VC approach guarantees is $\epsilon_{\text{min}} = 0.0059$. As discussed in 6.4.2 the required threshold $\bar{\epsilon}$ to be met by the Chebychev norm induced distance measure d_∞ depends on the selectivity γ of the query. The necessity of this can be observed in Figure 6.5 too. With the interestingness scores(distances) being lower than the curve defined by $\bar{\epsilon}$ for all queries in Figure 6.5 the VC approach does not recommend any false positives in this experiment. Using different distributions instead of the uniform one showed comparable results.

6.6.3 Statistical Testing vs. VC approach

We now show that while statistical testing in the form of a χ^2-test is in general not a wrong ingredient for building a VRS, in some situations it is unable to spot meaningful visual differences which would however be opportunely recognized by our VizRec approach.

Assume we had a query that yielded $m = 1,200$ out of $n = 10,000$ samples and a perfect estimator

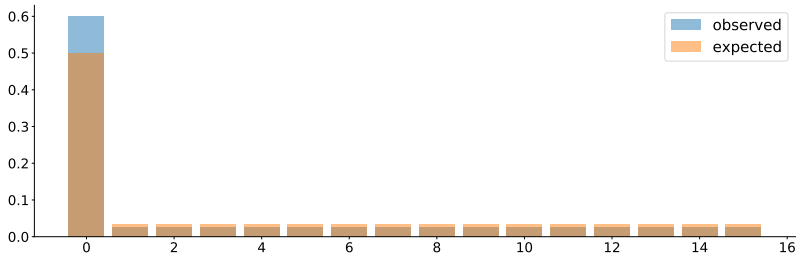


Figure 6.6: Comparing two close distributions that however should not be recommended since the visual difference criterion according to the VC dimension approach is not met.

for the true distribution function of the reference and the query distribution. These distributions shall be distributed as in Figure 6.6. Thus, the χ^2-test would yield a p-value of of $2.54 \cdot 10^{-5}$ implying that they are different when no more than 1967 visualizations under Bonferroni's correction are tested. However, at a VC dimension of 10 ($\delta = 0.05$) the required $\bar{\epsilon}$ must be at least 0.22 which is nearly twice as high as the 0.1 difference at the first bar as shown in Figure 6.6. Thus, the VC approach would not select this visualization as being significantly different enough given the modest sample size. Using the χ^2 test would recommend this visualization since it only spots that there is a difference but not whether the difference is significant enough.

In practice, scenarios like the one presented here occur especially due to outliers in the data. I.e. it could happen that for one feature value there are only 1-5 samples that would lead without any correction to a positive recommendation. Though a heuristic may ignore visualizations with less than 5 samples, this would come at the cost of ignoring rare phenomena and by using some magic number to define the threshold. I.e. when ignoring visualizations with less than b samples per bin, easily an example with $b+1$ samples for some bin could be found where χ^2 would pick up a non-visually significant visualization.

This underscores that a VRS using the χ^2-test would correctly identify two visualizations being different but can not guarantee a meaningful difference in terms of a distance which is crucial to build usable systems without luring the user into a false sense of security. One may argue that filtering out visualizations after having performed statistical testing would remedy this(which may work in practice when the interestingness score is high enough), but then there was no guarantee that the distances observed is statistically guaranteed. Whereas the question is dependent on the scenario (i.e. which query and which guarantees a system needs to fulfill) we want to point out, that there is a simple way to use the χ^2-test to control significant distance too.

Furthermore we want to underscore the point that the Chi-squared test is indeed a very powerful test but that the correct estimation of the distribution dominates the selectivity. I.e., when we guarantee that the estimates for the probability mass function are close enough to the true values, a testing procedure like χ^2-test will even under a million possible hypothesis only need a small number of samples to spot a difference between two distributions. We thereby define the

required number of point estimates to be in the range of $2 \leq K \leq 100$ bars as meaningful. In

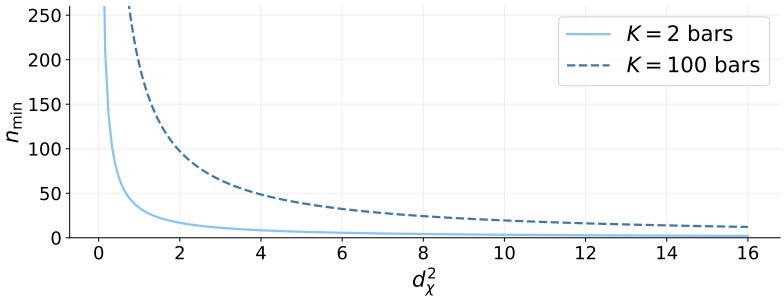


Figure 6.7: Chi-square distance d_{χ^2} and minimum number of samples $n_{\min}$ required for the χ^2-test to reject the null hypothesis assuming Bonferroni correction with $\alpha' = 0.05$ and 10^6 queries.

Figure 6.7 it is shown that even low values for the χ^2-distance d_χ^2 only require queries with hundreds of samples to be identified correctly.

6.6.4 VC bounds and Chernoff-Hoeffding bounds

When only a single visualization needs to be recommended, using Chernoff-Hoeffding bounds is both easier and yields a better guarantee than a VC-based bound. However, when testing multiple visualizations on the same data, the VC approach dominates Chernoff-Hoeffding bounds.

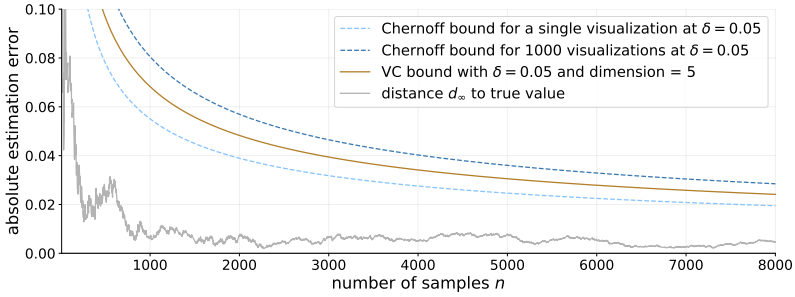


Figure 6.8: Observed estimation error for a single random experiment and bounds obtainable at a significance level of $\delta = 0.05$. For a VC dimension of $d = 1$, the bound is only slightly worse than the Chernoff bound of a single visualization. However, with increasing number of visualizations Chernoff bounds are more conservative than bounds obtained via VC.

To demonstrate this, random data was generated to estimate the probability mass function of a biased Binomial distributed according to $\sim \text{Bin}(10, 0.3)$. In Figure 6.8 a sample path is shown together with the theoretical bounds for a setup with a VC dimension of 5. Comparing to a Chernoff-Hoeffding bound of a single visualization, the VC approach outperforms Chernoff-Hoeffding bounds when learning multiple visualizations at once.

6.6.5 Restricting the search space

In this experiment we show that without restriction of the search space, it is likely that only few visualizations get recommended. For this, we took again the survey dataset and removed constants and identifier alike columns. To make it visually more distinct, we also added some artificial columns like a running identifier.

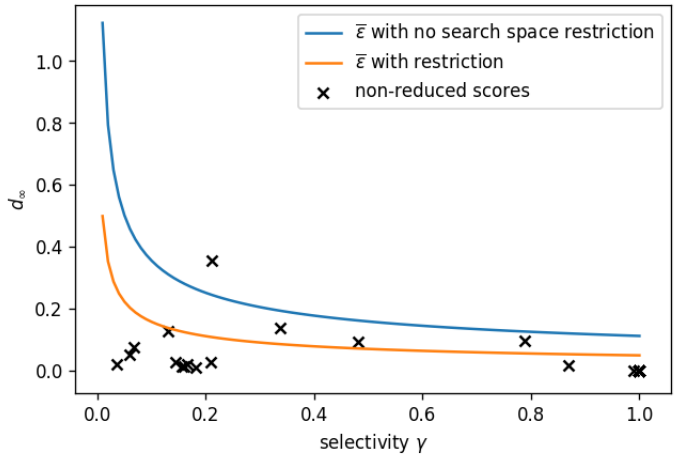


Figure 6.9: $\bar{\epsilon}$-curves before and after restricting the search space. The distance d_∞ needs to be higher than the uncertainty quantified via $\bar{\epsilon}$.

As shown in Figure 6.9, restricting the search space leads to more discoveries. I.e. the goal is to not be over-conservative and apply meaningful preprocessing or feature selection by the user first.

6.7 Conclusion

In this work, we demonstrated why users should build visualization recommendation systems with mechanisms to ensure statistical guarantees in order to prevent users from making false discoveries or regarding noisy data as relevant. As a novel way which supplements classical statistical testing as in [244] we introduced a technique based on statistical learning comparable in its power to [190]. We demonstrated various trade-offs and problems to consider when using either technique and provided a simple heuristic on how to explore the vast search space more efficiently by pruning it through different preprocessing steps.

Part III

Counting sub-graphs in massive dynamic graph streams

Chapter 7

TRÍEST: Counting Triangles in Massive Graph Streams[1]

In this Chapter, we present TRIÈST, a suite of one-pass streaming algorithms to compute unbiased, low-variance, high-quality approximations of the global and local (i.e., incident to each vertex) number of triangles in a fully-dynamic graph represented as an adversarial stream of edge insertions and deletions.

Our algorithms use reservoir sampling and its variants to exploit the user-specified memory space at all times. This is in contrast with previous approaches, which require hard-to-choose parameters (e.g., a fixed sampling probability) and offer no guarantees on the amount of memory they use. We analyze the variance of the estimations and show novel concentration bounds for these quantities.

Our experimental results on very large graphs demonstrate that TRIÈST outperforms state-of-the-art approaches in accuracy and exhibits a small update time.

7.1 Introduction

Exact computation of characteristic quantities of Web-scale networks is often impractical or even infeasible due to the humongous size of these graphs. It is natural in these cases to resort to *efficient-to-compute approximations* of these quantities that, when of sufficiently high quality, can be used as proxies for the exact values.

In addition to being huge, many interesting networks are *fully-dynamic* and can be represented as a *stream* whose elements are edges/nodes insertions and deletions which occur in an *arbitrary* (even adversarial) order. Characteristic quantities in these graphs are *intrinsically volatile*, hence

[1]The work presented in this chapter appeared in the ACM Transactions on Knowledge Discovery from Data - (TKDD) in the August 2017. A preliminary version of these results appeared in the technical program of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016). The paper received the Best Student Paper Award for the Research Track. This is joint work with Alessandro Epasto, Professor Matteo Riondato and Professor Eli Upfal.

there is limited added value in maintaining them exactly. The goal is rather to keep track, *at all times*, of a high-quality approximation of these quantities. For efficiency, the algorithms should *aim at exploiting the available memory space as much as possible* and they should *require only one pass over the stream.*

We introduce TRIÈST, a suite of *sampling-based, one-pass algorithms for adversarial fully-dynamic streams to approximate the global number of triangles and the local number of triangles incident to each vertex.* Mining local and global triangles is a fundamental primitive with many applications (e.g., community detection [20], topic mining [64], spam/anomaly detection [14, 144], ego-networks mining [67] and protein interaction networks analysis [160].)

Many previous works on triangle estimation in streams also employ sampling (see Sect. 7.3), but they usually require the user to specify *in advance* an *edge sampling probability* p that is fixed for the entire stream. This approach presents several significant drawbacks. First, choosing a p that allows to obtain the desired approximation quality requires to know or guess a number of properties of the input (e.g., the size of the stream). Second, a fixed p implies that the sample size grows with the size of the stream, which is problematic when the stream size is not known in advance: if the user specifies a large p, the algorithm may run out of memory, while for a smaller p it will provide a suboptimal estimation. Third, even assuming to be able to compute a p that ensures (in expectation) full use of the available space, the memory would be fully utilized only at the end of the stream, and the estimations computed throughout the execution would be suboptimal.

Contributions We address all the above issues by taking a significant departure from the fixed-probability, independent edge sampling approach taken even by state-of-the-art methods [144]. Specifically:

- We introduce TRIÈST (*TRI*angle *E*stimation from *ST*reams), a suite of *one-pass streaming algorithms* to approximate, at each time instant, the global and local number of triangles in a *fully-dynamic* graph stream (i.e., a sequence of edges additions and deletions in arbitrary order) using a *fixed amount of memory.* This is the first contribution that enjoys all these properties. TRIÈST only requires the user to specify *the amount of available memory*, an interpretable parameter that is definitively known to the user.

- Our algorithms maintain a sample of edges: they use the *reservoir sampling* [234] and *random pairing* [86] sampling schemes to exploit the available memory as much as possible. To the best of our knowledge, ours is the first application of these techniques to subgraph counting in fully-dynamic, arbitrarily long, adversarially ordered streams. We present an analysis of the unbiasedness and of the variance of our estimators, and establish strong concentration results for them. The use of reservoir sampling and random pairing requires additional sophistication in the analysis, as the presence of an edge in the sample is *not independent* from the concurrent presence of another edge. Hence, in our proofs we must consider the complex dependencies in events involving sets of edges. The gain is worth the effort: we prove that the variance of our

algorithms is smaller than that of state-of-the-art methods [144], and this is confirmed by our experiments.

- We conduct an extensive experimental evaluation of TRIÈST on very large graphs, some with billions of edges, comparing the performances of our algorithms to those of existing state-of-the-art contributions [144, 116, 180]. *Our algorithms significantly and consistently reduce the average estimation error by up to* 90% w.r.t. the state of the art, both in the global and local estimation problems, while using the same amount of memory. Our algorithms are also extremely scalable, showing update times in the order of hundreds of microseconds for graphs with billions of edges.

Chapter organization We formally introduce the settings and the problem in Sect. 7.2. In Sect. 7.3 we discuss related works. We present and analyze TRIÈST and discuss our design choices in Sect. 7.4. The results of our experimental evaluation are presented in Sect. 7.5. We draw our conclusions in Sect. 7.6. Towards improving readability, we defer some of the longer proofs to the end of the chapter.

7.2 Preliminaries

We study the problem of counting global and local triangles in a fully-dynamic undirected graph as an arbitrary (adversarial) stream of edge insertions and deletions.

Formally, for any (discrete) time instant $t \geq 0$, let $G^{(t)} = (V^{(t)}, E^{(t)})$ be the graph observed up to and including time t. At time $t = 0$ we have $V^{(t)} = E^{(t)} = \emptyset$. For any $t > 0$, at time $t + 1$ we receive an element $e_{t+1} = (\bullet, (u, v))$ from a stream, where $\bullet \in \{+, -\}$ and u, v are two distinct vertices. The graph $G^{(t+1)} = (V^{(t+1)}, E^{(t+1)})$ is obtained by *inserting a new edge or deleting an existing edge* as follows:

$$E^{(t+1)} = \begin{cases} E^{(t)} \cup \{(u, v)\} \text{ if } \bullet = \text{“}+\text{”} \\ E^{(t)} \setminus \{(u, v)\} \text{ if } \bullet = \text{“}-\text{”} \end{cases}.$$

If u or v do not belong to $V^{(t)}$, they are added to $V^{(t+1)}$. Nodes are deleted from $V^{(t)}$ when they have degree zero.

Edges can be added and deleted in the graph in an arbitrary adversarial order, i.e., as to cause the worst outcome for the algorithm, but we assume that the adversary has *no access to the random bits* used by the algorithm. We assume that *all operations have effect*: if $e \in E^{(t)}$ (resp. $e \notin E^{(t)}$), $(+, e)$ (resp. $(-, e)$) can not be on the stream at time $t + 1$.

Given a graph $G^{(t)} = (V^{(t)}, E^{(t)})$, a *triangle* in $G^{(t)}$ is a *set* of three edges $\{(u, v), (v, w), (w, u)\} \subseteq E^{(t)}$, with u, v, and w being three distinct vertices. We refer to $\{u, v, w\} \subseteq V^{(t)}$ as the *corners* of the triangle. We denote with $\Delta^{(t)}$ the set of *all* triangles in $G^{(t)}$, and, for any vertex $u \in V^{(t)}$, with $\Delta_u^{(t)}$ the subset of $\Delta^{(t)}$ containing all and only the triangles that have u as a corner.

Problem definition. We study the *Global* (resp. *Local*) Triangle Counting Problem in Fully-dynamic Streams, which requires to compute, at each time $t \geq 0$ an estimation of $|\Delta^{(t)}|$ (resp. for

each $u \in V$ an estimation of $|\Delta_u^{(t)}|$).

Multigraphs Our approach can be further extended to count the number of global and local triangles on a *multigraph* represented as a stream of edges. Using a formalization analogous to that discussed for graphs, for any (discrete) time instant $t \geq 0$, let $G^{(t)} = (V^{(t)}, \mathcal{E}^{(t)})$ be the multigraph observed up to and including time t, where $\mathcal{E}^{(t)}$ is now a *bag* of edges between vertices of $V^{(t)}$. The multigraph evolves through a series of edges additions and deletions according to almost the same process described for graphs, with the exception that now all operations must have effect on the *bag* of edges $\mathcal{E}^{(t)}$. Thus, for example, we may have $(+, e)$ on the stream at time t and again $(+, e)$ at time $t+1$. Some additional modifications to the model are needed to handle deletions appropriately and we outline them in Sect. 7.4.4. The definition of triangle in a multigraph is the same as in a graph. As before we denote with $\Delta^{(t)}$ the set of *all* triangles in $G^{(t)}$, but now this set may contain multiple triangles with the same set of vertices, although each of these triangles will be a different set of three edges among those vertices, i.e., a different subset of the bag $\mathcal{E}^{(t)}$. For any vertex $u \in V^{(t)}$, we still denote with $\Delta_u^{(t)}$ the subset of $\Delta^{(t)}$ containing all and only the triangles that have u as a corner, with a similar caveat as $\Delta^{(t)}$. The problems of global and local triangle counting in multigraph edge streams are defined exactly in the same way as for graph edge streams.

7.3 Related work

The literature on exact and approximate triangle counting is extremely rich, including exact algorithms, graph sparsifiers [226, 227], complex-valued sketches [154, 124], and MapReduce algorithms [223, 171, 175, 178, 177]. Here we restrict the discussion to the works most related to ours, i.e., to those presenting algorithms for counting or approximating the number of triangles from data streams. We refer to the survey by [140] for an in-depth discussion of other works. Table 7.1 presents a summary of the comparison, in terms of desirable properties, between this work and relevant previous contributions.

Table 7.1: Comparison with previous contributions

Work	Single pass	Fixed space	Local counts	Global counts	Fully-dynamic streams
[14]	✗	✓/✗[a]	✓	✗	✗
[132]	✗	✗	✗	✓	✗
[180]	✓	✓	✗	✓	✗
[116]	✓	✓	✗	✓	✗
[2]	✓	✗	✗	✓	✗
[144]	✓	✗	✓	✗/✓[b]	✗
This work	✓	✓	✓	✓	✓

[a] The required space is $O(|V^{(t)}|)$, which, although not dependent on the number of triangles or on the number of edges, is not fixed, in the sense that it cannot be fixed a-priori.

[b] Global triangle counting is not mentioned in the article, but the extension is straightforward.

Many authors presented algorithms for more restricted (i.e., less generic) settings than ours, or for which the constraints on the computation are more lax [10, 37, 121, 137]. For example, [14] and [132] present algorithms for approximate triangle counting from *static* graphs by performing multiple passes over the data. [180] and [116] propose algorithms for approximating only the global number of triangles from *edge-insertion-only* streams. [35] present a one-pass algorithm for fully-dynamic graphs, but the triangle count estimation is (expensively) computed only at the end of the stream and the algorithm requires, in the worst case, more memory than what is needed to store the entire graph. [2] apply the sampling-and-hold approach to insertion-only graph stream mining to obtain, only at the end of the stream and using non-constant space, an estimation of many network measures including triangles.

None of these works has *all* the features offered by TRIÈST: performs a single pass over the data, handles fully-dynamic streams, uses a fixed amount of memory space, requires a single interpretable parameter, and returns an estimation at each time instant. Furthermore, our experimental results show that we outperform the algorithms from [180, 116] on insertion-only streams.

[144] present an algorithm for insertion-only streams that is based on independent edge sampling with a fixed probability: for each edge on the stream, a coin with a user-specified fixed tails probability p is flipped, and, if the outcome is tails, the edge is added to the stored sample and the estimation of local triangles is updated. Since the memory is not fully utilized during most of the stream, the variance of the estimate is large. Our approach handles fully-dynamic streams and makes better use of the available memory space at each time instant, resulting in a better estimation, as shown by our analytical and experimental results.

[234] presents a detailed analysis of the reservoir sampling scheme and discusses methods to speed up the algorithm by reducing the number of calls to the random number generator. Random Pairing [86] is an extension of reservoir sampling to handle fully-dynamic streams with insertions and deletions. [47] generalize and extend the Random Pairing approach to the case where the elements on the stream are key-value pairs, where the value may be negative (and less than -1). In our settings, where the value is not less than -1 (for an edge deletion), these generalizations do not apply and the algorithm presented by [47] reduces essentially to Random Pairing.

7.4 Algorithms

We present TRIÈST, a suite of three novel algorithms for approximate global and local triangle counting from edge streams. The first two work on insertion-only streams, while the third can handle fully-dynamic streams where edge deletions are allowed. We defer the discussion of the multigraph case to Sect. 7.4.4.

Parameters Our algorithms keep an edge sample $\mathcal{S}$ containing up to M edges from the stream, where M is a positive integer parameter. For ease of presentation, we realistically assume $M \geq 6$. In Sect. 7.1 we motivated the design choice of only requiring M as a parameter and remarked on

its advantages over using a fixed sampling probability p. Our algorithms are designed to use the available space as much as possible.

Counters TRIÈST algorithms keep *counters* to compute the estimations of the global and local number of triangles. They *always* keep one global counter τ for the estimation of the global number of triangles. Only the global counter is needed to estimate the total triangle count. To estimate the local triangle counts, the algorithms keep a set of local counters τ_u for a subset of the nodes $u \in V^{(t)}$. The local counters are created on the fly as needed, and *always* destroyed as soon as they have a value of 0. Hence our algorithms use $O(M)$ space (with one exception, see Sect. 7.4.2).

Notation For any $t \geq 0$, let $G^{\mathcal{S}} = (V^{\mathcal{S}}, E^{\mathcal{S}})$ be the subgraph of $G^{(t)}$ containing all and only the edges in the current sample $\mathcal{S}$. We denote with $\mathcal{N}_u^{\mathcal{S}}$ the *neighborhood* of u in $G^{\mathcal{S}}$: $\mathcal{N}_u^{\mathcal{S}} = \{v \in V^{(t)} : (u,v) \in \mathcal{S}\}$ and with $\mathcal{N}_{u,v}^{\mathcal{S}} = \mathcal{N}_u^{\mathcal{S}} \cap \mathcal{N}_v^{\mathcal{S}}$ the *shared neighborhood* of u and v in $G^{\mathcal{S}}$.

Presentation We only present the analysis of our algorithms for the problem of *global* triangle counting. For each presented result involving the estimation of the global triangle count (e.g., unbiasedness, bound on variance, concentration bound) and potentially using other global quantities (e.g., the number of pairs of triangles in $\Delta^{(t)}$ sharing an edge), it is straightforward to derive the correspondent variant for the estimation of the local triangle count, using similarly defined local quantities (e.g., the number of pairs of triangles in $\Delta_u^{(t)}$ sharing an edge.)

7.4.1 A first algorithm – TRIÈST-BASE

We first present TRIÈST-BASE, which works on insertion-only streams and uses standard reservoir sampling [234] to maintain the edge sample $\mathcal{S}$:

- If $t \leq M$, then the edge $e_t = (u,v)$ on the stream at time t is deterministically inserted in $\mathcal{S}$.
- If $t > M$, TRIÈST-BASE flips a biased coin with heads probability M/t. If the outcome is heads, it chooses an edge $(w,z) \in \mathcal{S}$ uniformly at random, removes (w,z) from $\mathcal{S}$, and inserts (u,v) in $\mathcal{S}$. Otherwise, $\mathcal{S}$ is not modified.

After each insertion (resp. removal) of an edge (u,v) from $\mathcal{S}$, TRIÈST-BASE calls the procedure UPDATECOUNTERS that increments (resp. decrements) τ, τ_u and τ_v by $|\mathcal{N}_{u,v}^{\mathcal{S}}|$, and τ_c by one, for each $c \in \mathcal{N}_{u,v}^{\mathcal{S}}$.

The pseudocode for TRIÈST-BASE is presented in Alg. 11.

Estimation

For any pair of positive integers a and b such that $a \leq \min\{M, b\}$ let

$$\xi_{a,b} = \begin{cases} 1 & \text{if } b \leq M \\ \binom{b}{M} \Big/ \binom{b-a}{M-a} = \prod_{i=0}^{a-1} \frac{b-i}{M-i} & \text{otherwise} \end{cases} .$$

ALGORITHM 11 TRIÈST-BASE

Input: Insertion-only edge stream Σ, integer $M \geq 6$

$\mathcal{S} \leftarrow \emptyset$, $t \leftarrow 0$, $\tau \leftarrow 0$
$t \leftarrow 0$ ▷ number of edges so far
$W \leftarrow 0$ ▷ total weight of edges so far
for each element $(+, (u, v), w)$ from Σ **do**
    $t \leftarrow t + 1$
    $W \leftarrow W + w$
    if SAMPLEEDGE$((u, v), t)$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{(u, v)\}$
        UPDATECOUNTERS$(+, (u, v))$

function SAMPLEEDGE$((u, v), t)$
    if $t \leq M$ **then**
        return True
    else if FLIPBIASEDCOIN$(\frac{M}{t})$ = heads **then**
        $(u', v') \leftarrow$ random edge from $\mathcal{S}$
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(u', v')\}$
        UPDATECOUNTERS$(-, (u', v'))$
        return True
    return False

function UPDATECOUNTERS$((\bullet, (u, v)))$
    $\mathcal{N}^{\mathcal{S}}_{u,v} \leftarrow \mathcal{N}^{\mathcal{S}}_{u} \cap \mathcal{N}^{\mathcal{S}}_{v}$
    for all $c \in \mathcal{N}^{\mathcal{S}}_{u,v}$ **do**
        $\tau \leftarrow \tau \bullet 1$
        $\tau_c \leftarrow \tau_c \bullet 1$
        $\tau_u \leftarrow \tau_u \bullet 1$
        $\tau_v \leftarrow \tau_v \bullet 1$

As shown in the following lemma, $\xi^{-1}_{k,t}$ is the probability that k edges of $G^{(t)}$ are all in $\mathcal{S}$ at time t, i.e., the k-th order inclusion probability of the reservoir sampling scheme.

Lemma 7.1. *For any time step t and any positive integer $k \leq t$, let B be any subset of $E^{(t)}$ of size $|B| = k \leq t$. Then, at the end of time step t,*

$$\Pr(B \subseteq \mathcal{S}) = \begin{cases} 0 & \text{if } k > M \\ \xi^{-1}_{k,t} & \text{otherwise} \end{cases} .$$

Before proving Lemma 7.1, we need to introduce the following lemma, which states a well known property of the reservoir sampling scheme.

Lemma 7.2 ([234, Sect. 2])**.** *For any $t > M$, let A be any subset of $E^{(t)}$ of size $|A| = M$. Then, at the end of time step t,*

$$\Pr(\mathcal{S} = A) = \frac{1}{\binom{|E^{(t)}|}{M}} = \frac{1}{\binom{t}{M}},$$

i.e., the set of edges in $\mathcal{S}$ at the end of time t is a subset of $E^{(t)}$ of size M chosen uniformly *at random from all subsets of $E^{(t)}$ of the same size.*

Proof of Lemma 7.1. If $k > \min\{M, t\}$, we have $\Pr(B \subseteq \mathcal{S}) = 0$ because it is impossible for B to be equal to $\mathcal{S}$ in these cases. From now on we then assume $k \leq \min\{M, t\}$.

If $t \leq M$, then $E^{(t)} \subseteq \mathcal{S}$ and $\Pr(B \subseteq \mathcal{S}) = 1 = \xi_{k,t}^{-1}$.

Assume instead that $t > M$, and let $\mathcal{B}$ be the family of subsets of $E^{(t)}$ that *1.* have size M, and *2.* contain B:

$$\mathcal{B} = \{C \subset E^{(t)} \ : \ |C| = M, B \subseteq C\} \ .$$

We have

$$|\mathcal{B}| = \binom{|E^{(t)}| - k}{M - k} = \binom{t-k}{M-k} \ . \tag{7.1}$$

From this and and Lemma 7.2 we then have

$$\begin{aligned}\Pr(B \subseteq \mathcal{S}) = \Pr(\mathcal{S} \in \mathcal{B}) &= \sum_{C \in \mathcal{B}} \Pr(\mathcal{S} = C) \\ &= \frac{\binom{t-k}{M-k}}{\binom{t}{M}} = \frac{\binom{t-k}{M-k}}{\binom{t-k}{M-k}\prod_{i=0}^{k-1}\frac{t-i}{M-i}} = \prod_{i=0}^{k-1}\frac{M-i}{t-i} = \xi_{k,t}^{-1} \ .\end{aligned}$$

□

We make use of this lemma in the analysis of TRIÈST-BASE.

Let, for any $t \geq 0$, $\xi^{(t)} = \xi_{3,t}$ and let $\tau^{(t)}$ (resp. $\tau_u^{(t)}$) be the value of the counter τ at the end of time step t (i.e., after the edge on the stream at time t has been processed by TRIÈST-BASE) (resp. the value of the counter τ_u at the end of time step t if there is such a counter, 0 otherwise). When queried at the end of time t, TRIÈST-BASE returns $\xi^{(t)}\tau^{(t)}$ (resp. $\xi^{(t)}\tau_u^{(t)}$) as the estimation for the global (resp. local for $u \in V^{(t)}$) triangle count.

Analysis

We now present the analysis of the estimations computed by TRIÈST-BASE. Specifically, we prove their unbiasedness (and their exactness for $t \leq M$) and then show an exact derivation of their variance and a concentration result. We show the results for the global counts, but results analogous to those in Thms. 7.3, 7.5, and 7.6 hold for the local triangle count for any $u \in V^{(t)}$, replacing the global quantities with the corresponding local ones. We also compare, theoretically, the variance of TRIÈST-BASE with that of a fixed-probability edge sampling approach [144], showing that TRIÈST-BASE has smaller variance for the vast majority of the stream.

Expectation:

We have the following result about the estimations computed by TRIÈST-BASE.

Theorem 7.3. *We have*

$$\begin{aligned}\xi^{(t)}\tau^{(t)} = \tau^{(t)} &= |\Delta^{(t)}| \ \textit{if } t \leq M \\ \mathbb{E}\left[\xi^{(t)}\tau^{(t)}\right] &= |\Delta^{(t)}| \ \textit{if } t > M \ .\end{aligned}$$

The TRIÈST-BASE estimations are not only *unbiased in all cases*, but actually *exact for* $t \leq M$, i.e., for $t \leq M$, they are the true global/local number of triangles in $G^{(t)}$.

We denote with $\Delta^{\mathcal{S}}$ the set of triangles in $G^{\mathcal{S}}$. To prove Thm. 7.3, we need to first introduce the folowing technical lemma:

Lemma 7.4. *After each call to* UpdateCounters*, we have* $\tau = |\Delta^{\mathcal{S}}|$ *and* $\tau_v = |\Delta^{\mathcal{S}}_v|$ *for any* $v \in V_{\mathcal{S}}$ *s.t.* $|\Delta^{\mathcal{S}}_v| \geq 1$.

Proof. We only show the proof for τ, as the proof for the local counters follows the same steps.

The proof proceeds by induction. The thesis is true after the first call to UpdateCounters at time $t = 1$. Since only one edge is in $\mathcal{S}$ at this point, we have $\Delta^{\mathcal{S}} = 0$, and $\mathcal{N}^{\mathcal{S}}_{u,v} = \emptyset$, so UpdateCounters does not modify τ, which was initialized to 0. Hence $\tau = 0 = \Delta^{\mathcal{S}}$.

Assume now that the thesis is true for any subsequent call to UpdateCounters up to some point in the execution of the algorithm where an edge (u, v) is inserted or removed from $\mathcal{S}$. We now show that the thesis is still true after the call to UpdateCounters that follows this change in $\mathcal{S}$. Assume that (u, v) was *inserted* in $\mathcal{S}$ (the proof for the case of an edge being removed from $\mathcal{S}$ follows the same steps). Let $\mathcal{S}^{\text{b}} = \mathcal{S} \setminus \{(u, v)\}$ and τ^{b} be the value of τ before the call to UpdateCounters and, for any $w \in V_{\mathcal{S}^{\text{b}}}$, let τ^{b}_w be the value of τ_w before the call to UpdateCounters. Let $\Delta^{\mathcal{S}}_{u,v}$ be the set of triangles in $G_{\mathcal{S}}$ that have u and v as corners. We need to show that, after the call, $\tau = |\Delta^{\mathcal{S}}|$. Clearly we have $\Delta^{\mathcal{S}} = \Delta^{\mathcal{S}^{\text{b}}} \cup \Delta^{\mathcal{S}}_{u,v}$ and $\Delta^{\mathcal{S}^{\text{b}}} \cap \Delta^{\mathcal{S}}_{u,v} = \emptyset$, so

$$|\Delta^{\mathcal{S}}| = |\Delta^{\mathcal{S}^{\text{b}}}| + |\Delta^{\mathcal{S}}_{u,v}|$$

We have $|\Delta^{\mathcal{S}}_{u,v}| = |\mathcal{N}^{\mathcal{S}}_{u,v,}|$ and, by the inductive hypothesis, we have that $\tau^{\text{b}} = |\Delta^{\mathcal{S}^{\text{b}}}|$. Since UpdateCounters increments τ by $|\mathcal{N}^{\mathcal{S}}_{u,v,}|$, the value of τ after UpdateCounters has completed is exactly $|\Delta^{\mathcal{S}}|$. □

From here, the proof of Thm. 7.3 is a straightforward application of Lemma 7.4 for the case $t \leq M$ and of that lemma, the definition of expectation, and Lemma 7.1 otherwise.

Proof of Thm. 7.3. We prove the statement for the estimation of global triangle count. The proof for the local triangle counts follows the same steps.

If $t \leq M$, we have $G_{\mathcal{S}} = G^{(t)}$ and from Lemma 7.4 we have $\tau^{(t)} = |\Delta^{\mathcal{S}}| = |\Delta^{(t)}|$, hence the thesis holds.

Assume now that $t > M$, and assume that $|\Delta^{(t)}| > 0$, otherwise, from Lemma 7.4, we have $\tau^{(t)} = |\Delta^{\mathcal{S}}| = 0$ and TRIÈST-BASE estimation is deterministically correct. Let $\lambda = (a, b, c) \in \Delta^{(t)}$, (where a, b, c are edges in $E^{(t)}$) and let $\delta^{(t)}_\lambda$ be a random variable that takes value $\xi^{(t)}$ if $\lambda \in \Delta_{\mathcal{S}}$ (i.e., $\{a, b, c\} \subseteq \mathcal{S}$) at the end of the step instant t, and 0 otherwise. From Lemma 7.1, we have that

$$\mathbb{E}\left[\delta^{(t)}_\lambda\right] = \xi^{(t)} \Pr(\{a, b, c\} \subseteq \mathcal{S}) = \xi^{(t)} \frac{1}{\xi_{3,t}} = \xi^{(t)} \frac{1}{\xi^{(t)}} = 1 \ . \tag{7.2}$$

We can write

$$\xi^{(t)} \tau^{(t)} = \sum_{\lambda \in \Delta^{(t)}} \delta^{(t)}_\lambda$$

and from this, (7.2), and linearity of expectation, we have

$$\mathbb{E}\left[\xi^{(t)} \tau^{(t)}\right] = \sum_{\lambda \in \Delta^{(t)}} \mathbb{E}\left[\delta^{(t)}_\lambda\right] = |\Delta^{(t)}| \ .$$

□

Variance

We now analyze the variance of the estimation returned by TRIÈST-BASE for $t > M$ (the variance is 0 for $t \leq M$.)

Let $r^{(t)}$ be the *total* number of unordered pairs of distinct triangles from $\Delta^{(t)}$ sharing an edge,[2] and $w^{(t)} = \binom{|\Delta^{(t)}|}{2} - r^{(t)}$ be the number of unordered pairs of distinct triangles that do not share any edge.

Theorem 7.5. *For any $t > M$, let $f(t) = \xi^{(t)} - 1$,*

$$g(t) = \xi^{(t)} \frac{(M-3)(M-4)}{(t-3)(t-4)} - 1$$

and

$$h(t) = \xi^{(t)} \frac{(M-3)(M-4)(M-5)}{(t-3)(t-4)(t-5)} - 1 \ \ (\leq 0).$$

We have:

$$\mathrm{Var}\left[\xi(t)\tau^{(t)}\right] = |\Delta^{(t)}| f(t) + r^{(t)} g(t) + w^{(t)} h(t). \tag{7.3}$$

In our proofs, we carefully account for the fact that, as we use reservoir sampling [234], the presence of an edge a in $\mathcal{S}$ is *not independent* from the concurrent presence of another edge b in $\mathcal{S}$. This is not the case for samples built using fixed-probability independent edge sampling, such as MASCOT [144]. When computing the variance, we must consider not only pairs of triangles that share an edge, as in the case for independent edge sampling approaches, but also pairs of triangles sharing no edge, since their respective presences in the sample are not independent events. The gain is worth the additional sophistication needed in the analysis, as the contribution to the variance by triangles no sharing edges is *non-positive* ($h(t) \leq 0$), i.e., it reduces the variance. A comparison of the variance of our estimator with that obtained with a fixed-probability independent edge sampling approach, is discussed in Sect. 7.4.1.

Proof of Thm. 7.5. Assume $|\Delta^{(t)}| > 0$, otherwise the estimation is deterministically correct and has variance 0 and the thesis holds. Let $\lambda \in \Delta^{(t)}$ and $\delta_\lambda^{(t)}$ be as in the proof of Thm. 7.3. We have

[2] Two distinct triangles can share at most one edge.

$\mathrm{Var}[\delta_\lambda^{(t)}] = \xi^{(t)} - 1$ and from this and the definition of variance and covariance we obtain

$$\begin{aligned}
\mathrm{Var}\left[\xi^{(t)}\tau^{(t)}\right] &= \mathrm{Var}\left[\sum_{\lambda\in\Delta^{(t)}} \delta_\lambda^{(t)}\right] = \sum_{\lambda\in\Delta^{(t)}}\sum_{\gamma\in\Delta^{(t)}} \mathrm{Cov}\left[\delta_\lambda^{(t)}, \delta_\gamma^{(t)}\right] \\
&= \sum_{\lambda\in\Delta^{(t)}} \mathrm{Var}\left[\delta_\lambda^{(t)}\right] + \sum_{\substack{\lambda,\gamma\in\Delta^{(t)}\\ \lambda\neq\gamma}} \mathrm{Cov}\left[\delta_\lambda^{(t)}, \delta_\gamma^{(t)}\right] \\
&= |\Delta^{(t)}|(\xi^{(t)} - 1) + \sum_{\substack{\lambda,\gamma\in\Delta^{(t)}\\ \lambda\neq\gamma}} \mathrm{Cov}\left[\delta_\lambda^{(t)}, \delta_\gamma^{(t)}\right] \\
&= |\Delta^{(t)}|(\xi^{(t)} - 1) + \sum_{\substack{\lambda,\gamma\in\Delta^{(t)}\\ \lambda\neq\gamma}} \left(\mathbb{E}\left[\delta_\lambda^{(t)}\delta_\gamma^{(t)}\right] - \mathbb{E}\left[\delta_\lambda^{(t)}\right]\mathbb{E}\left[\delta_\gamma^{(t)}\right]\right) \\
&= |\Delta^{(t)}|(\xi^{(t)} - 1) + \sum_{\substack{\lambda,\gamma\in\Delta^{(t)}\\ \lambda\neq\gamma}} \left(\mathbb{E}\left[\delta_\lambda^{(t)}\delta_\gamma^{(t)}\right] - 1\right) \quad . \qquad (7.4)
\end{aligned}$$

Assume now $|\Delta^{(t)}| \geq 2$, otherwise we have $r^{(t)} = w^{(t)} = 0$ and the thesis holds as the second term on the r.h.s. of (7.4) is 0. Let λ and γ be two distinct triangles in $\Delta^{(t)}$. If λ and γ do not share an edge, we have $\delta_\lambda^{(t)}\delta_\gamma^{(t)} = \xi^{(t)}\xi^{(t)} = \xi_{3,t}^2$ if all *six* edges composing λ and γ are in $\mathcal{S}$ at the end of time step t, and $\delta_\lambda^{(t)}\delta_\gamma^{(t)} = 0$ otherwise. From Lemma 7.1 we then have that

$$\begin{aligned}
\mathbb{E}\left[\delta_\lambda^{(t)}\delta_\gamma^{(t)}\right] &= \xi_{3,t}^2 \Pr\left(\delta_\lambda^{(t)}\delta_\gamma^{(t)} = \xi_{3,t}^2\right) = \xi_{3,t}^2 \frac{1}{\xi_{6,t}} = \xi_{3,t} \prod_{j=3}^{5} \frac{M-j}{t-j} \\
&= \xi^{(t)} \frac{(M-3)(M-4)(M-5)}{(t-3)(t-4)(t-5)} \quad . \qquad (7.5)
\end{aligned}$$

If instead λ and γ share exactly an edge we have $\delta_\lambda^{(t)}\delta_\gamma^{(t)} = \xi_{3,t}^2$ if all *five* edges composing λ and γ are in $\mathcal{S}$ at the end of time step t, and $\delta_\lambda^{(t)}\delta_\gamma^{(t)} = 0$ otherwise. From Lemma 7.1 we then have that

$$\begin{aligned}
\mathbb{E}\left[\delta_\lambda^{(t)}\delta_\gamma^{(t)}\right] &= \xi_{3,t}^2 \Pr\left(\delta_\lambda^{(t)}\delta_\gamma^{(t)} = \xi_{3,t}^2\right) = \xi_{3,t}^2 \frac{1}{\xi_{5,t}} = \xi_{3,t} \prod_{j=3}^{4} \frac{M-j}{t-j} \\
&= \xi^{(t)} \frac{(M-3)(M-4)}{(t-3)(t-4)} \quad . \qquad (7.6)
\end{aligned}$$

The thesis follows by combining (7.4), (7.5), (7.6), recalling the definitions of $r^{(t)}$ and $w^{(t)}$, and slightly reorganizing the terms. □

Concentration

We have the following concentration result on the estimation returned by TRIÈST-BASE. Let $h^{(t)}$ denote the maximum number of triangles sharing a single edge in $G^{(t)}$.

Theorem 7.6. *Let $t \geq 0$ and assume $|\Delta^{(t)}| > 0$.*[3] *For any $\varepsilon, \delta \in (0,1)$, let*

$$\Phi = \sqrt[3]{8\varepsilon^{-2}\frac{3h^{(t)}+1}{|\Delta^{(t)}|}\ln\left(\frac{(3h^{(t)}+1)e}{\delta}\right)} \quad .$$

[3] If $|\Delta^{(t)}| = 0$, our algorithms correctly estimate 0 triangles.

If

$$M \geq \max\left\{t\Phi\left(1+\frac{1}{2}\ln^{2/3}\left(t\Phi\right)\right), 12\varepsilon^{-1}+e^2, 25\right\},$$

then $|\xi^{(t)}\tau^{(t)} - |\Delta^{(t)}|| < \varepsilon|\Delta^{(t)}|$ *with probability* $> 1-\delta$.

The roadmap to proving Thm. 7.6 is the following:

1. we first define two simpler algorithms, named INDEP and MIX. The algorithms use, respectively, fixed-probability independent sampling of edges and reservoir sampling (but with a different estimator than the one used by TRIÈST-BASE);
2. we then prove concentration results on the estimators of INDEP and MIX. Specifically, the concentration result for INDEP uses a result by [95] on graph coloring, while the one for MIX will depend on the concentration result for INDEP and on a Poisson-approximation-like technical result stating that probabilities of events when using reservoir sampling are close to the probabilities of those events when using fixed-probability independent sampling;
3. we then show that the estimates returned by TRIÈST-BASE are close to the estimates returned by MIX;
4. finally, we combine the above results and show that, if M is large enough, then the estimation provided by MIX is likely to be close to $|\Delta^{(t)}|$ and since the estimation computed by TRIÈST-BASE is close to that of MIX, then it must also be close to $|\Delta^{(t)}|$.

Note: for ease of presentation, in the following we use $\phi^{(t)}$ to denote the estimation returned by TRIÈST-BASE, i.e., $\phi^{(t)} = \xi^{(t)}\tau^{(t)}$.

The INDEP algorithm The INDEP algorithm works as follows: it creates a sample $\mathcal{S}_{\text{IN}}$ by sampling edges in $E^{(t)}$ independently with a fixed probability p. It estimates the global number of triangles in $G^{(t)}$ as

$$\phi_{\text{IN}}^{(t)} = \frac{\tau_{\text{IN}}^{(t)}}{p^3},$$

where $\tau_{\text{IN}}^{(t)}$ is the number of triangles in $\mathcal{S}_{\text{IN}}$. This is for example the approach taken by MASCOT-C [144].

The MIX algorithm The MIX algorithm works as follows: it uses reservoir sampling (like TRIÈST-BASE) to create a sample $\mathcal{S}_{\text{MIX}}$ of M edges from $E^{(t)}$, but uses a different estimator than the one used by TRIÈST-BASE. Specifically, MIX uses

$$\phi_{\text{MIX}}^{(t)} = \left(\frac{t}{M}\right)^3 \tau^{(t)}$$

as an estimator for $|\Delta^{(t)}|$, where $\tau^{(t)}$ is, as in TRIÈST-BASE, the number of triangles in $G^{\mathcal{S}}$ (TRIÈST-BASE uses $\phi^{(t)} = \frac{t(t-1)(t-2)}{M(M-1)(M-2)}\tau^{(t)}$ as an estimator.)

We call this algorithm MIX because it uses reservoir sampling to create the sample, but computes the estimate as if it used fixed-probability independent sampling, hence in some sense it "mixes" the two approaches.

Concentration results for INDEP and MIX We now show a concentration result for INDEP. Then we show a technical lemma (Lemma 7.8) relating the probabilities of events when using reservoir sampling to the probabilities of those events when using fixed-probability independent sampling. Finally, we use these results to show that the estimator used by MIX is also concentrated (Lemma 7.10).

Lemma 7.7. *Let $t \geq 0$ and assume $|\Delta^{(t)}| > 0$.[4] For any $\varepsilon, \delta \in (0,1)$, if*

$$p \geq \sqrt[3]{2\varepsilon^{-2} \ln\left(\frac{3h^{(t)}+1}{\delta}\right) \frac{3h^{(t)}+1}{|\Delta^{(t)}|}} \tag{7.7}$$

then

$$\Pr\left(|\phi_{\text{IN}}^{(t)} - \Delta^{(t)}|| < \varepsilon|\Delta^{(t)}|\right) > 1-\delta \ .$$

Proof. Let H be a graph built as follows: H has one node for each triangle in $G^{(t)}$ and there is an edge between two nodes in H if the corresponding triangles in $G^{(t)}$ share an edge. By this construction, the maximum degree in H is $3h^{(t)}$. Hence by the Hajanal-Szeméredi's theorem [95] there is a proper coloring of H with at most $3h^{(t)}+1$ colors such that for each color there are at least $L = \frac{|\Delta^{(t)}|}{3h^{(t)}+1}$ nodes with that color.

Assign an arbitrary numbering to the triangles of $G^{(t)}$ (and, therefore, to the nodes of H) and let X_i be a Bernoulli random variable, indicating whether the triangle i in $G^{(t)}$ is in the sample at time t. From the properties of independent sampling of edges we have $\Pr(X_i = 1) = p^3$ for any triangle i. For any color c of the coloring of H, let $\mathcal{X}_c$ be the set of r.v.'s X_i such that the node i in H has color c. Since the coloring of H which we are considering is proper, the r.v.'s in $\mathcal{X}_c$ are independent, as they correspond to triangles which do not share any edge and edges are sampled independent of each other. Let Y_c be the sum of the r.v.'s in $\mathcal{X}_c$. The r.v. Y_c has a binomial distribution with parameters $|\mathcal{X}_c|$ and p_t^3. By the Chernoff bound for binomial r.v.'s, we have that

$$\begin{aligned}\Pr\left(|p^{-3}Y_c - |\mathcal{X}_c|| > \varepsilon|\mathcal{X}_c|\right) &< 2\exp\left(-\varepsilon^2 p^3 |\mathcal{X}_c|/2\right) \\ &< 2\exp\left(-\varepsilon^2 p^3 L/2\right) \\ &\leq \frac{\delta}{3h^{(t)}+1},\end{aligned}$$

where the last step comes from the requirement in (7.7).Then by applying the union bound over all the (at most) $3h^{(t)}+1$ colors we get

$$\Pr(\exists \text{ color } c \text{ s.t. } |p^{-3}Y_c - |\mathcal{X}_c|| > \varepsilon|\mathcal{X}_c|) < \delta \ .$$

Since $\phi_{\text{IN}}(t) = p^{-3}\sum_{\text{color } c} Y_c$, from the above equation we have that, with probability at least $1-\delta$,

$$\begin{aligned}|\phi_{\text{IN}}^{(t)} - |\Delta^{(t)}|| &\leq \left|\sum_{\text{color } c} p^{-3}Y_c - \sum_{\text{color } c}|\mathcal{X}_c|\right| \\ &\leq \sum_{\text{color } c}|p^{-3}Y_c - |\mathcal{X}_c|| \leq \sum_{\text{color } c}\varepsilon|\mathcal{X}_c| \leq \varepsilon|\Delta^{(t)}| \ .\end{aligned}$$

□

[4] For $|\Delta^{(t)}| = 0$, INDEP correctly and deterministically returns 0 as the estimation.

The above result is of independent interest and can be used, for example, to give concentration bounds to the estimation computed by MASCOT-C [144].

We remark that we can not use the same approach from Lemma 7.7 to show a concentration result for TRIÈST-BASE because it uses reservoir sampling, hence the event of having a triangle a in $\mathcal{S}$ and the event of having another triangle b in $\mathcal{S}$ are not independent.

We can however show the following general result, similar in spirit to the well-know Poisson approximation of balls-and-bins processes [164].

Fix the parameter M and a time $t > M$. Let $\mathcal{S}_{\text{MIX}}$ be a sample of M edges from $E^{(t)}$ obtained through reservoir sampling (as MIX would do), and let $\mathcal{S}_{\text{IN}}$ be a sample of the edges in $E^{(t)}$ obtained by sampling edges independently with probability M/t (as INDEP would do). We remark that the size of $\mathcal{S}_{\text{IN}}$ is in $[0, t]$ but not necessarily M.

Lemma 7.8. *Let $f : 2^{E^{(t)}} \to \{0, 1\}$ be an arbitrary binary function from the powerset of $E^{(t)}$ to $\{0, 1\}$. We have*

$$\Pr\left(f(\mathcal{S}_{\text{MIX}}) = 1\right) \leq e\sqrt{M}\Pr\left(f(\mathcal{S}_{\text{IN}}) = 1\right) \ .$$

Proof. Using the law of total probability, we have

$$\begin{aligned}
\Pr\left(f(\mathcal{S}_{\text{IN}}) = 1\right) &= \sum_{k=0}^{t} \Pr\left(f(\mathcal{S}_{\text{IN}}) = 1 \ \middle|\ |\mathcal{S}_{\text{IN}}| = k\right) \Pr(|\mathcal{S}_{\text{IN}}| = k) \\
&\geq \Pr\left(f(\mathcal{S}_{\text{IN}}) = 1 \ \middle|\ |\mathcal{S}_{\text{IN}}| = M\right) \Pr(|\mathcal{S}_{\text{IN}}| = M) \\
&\geq \Pr\left(f(\mathcal{S}_{\text{MIX}}) = 1\right) \Pr\left(|\mathcal{S}_{\text{IN}}| = M\right), \qquad (7.8)
\end{aligned}$$

where the last inequality comes from Lemma 7.2: the set of edges included in $\mathcal{S}_{\text{MIX}}$ is a uniformly-at-random subset of M edges from $E^{(t)}$, and the same holds for $\mathcal{S}_{\text{IN}}$ when conditioning its size being M.

Using the Stirling approximation $\sqrt{2\pi n}(\frac{n}{e})^n \leq n! \leq e\sqrt{n}(\frac{n}{e})^n$ for any positive integer n, we have

$$\begin{aligned}
\Pr\left(|\mathcal{S}_{\text{IN}}| = M\right) &= \binom{t}{M}\left(\frac{M}{t}\right)^M\left(\frac{t-M}{t}\right)^{t-M} \\
&\geq \frac{t^t\sqrt{t}\sqrt{2\pi}e^{-t}}{e^2\sqrt{M}\sqrt{t-M}e^{-t}M^M(t-M)^{t-M}}\frac{M^M(t-M)^{t-M}}{t^t} \\
&\geq \frac{1}{e\sqrt{M}} \ .
\end{aligned}$$

Plugging this into (7.8) concludes the proof. □

We now use the above two lemmas to show that the estimator $\phi^{(t)}_{\text{MIX}}$ computed by MIX is concentrated. We will first need the following technical fact.

Fact 7.9. *For any $x \geq 5$, we have*

$$\ln\left(x(1 + \ln^{2/3} x)\right) \leq \ln^2 x \ .$$

Lemma 7.10. *Let $t \geq 0$ and assume $|\Delta^{(t)}| < 0$. For any $\varepsilon, \delta \in (0, 1)$, let*

$$\Psi = 2\varepsilon^{-2}\frac{3h^{(t)} + 1}{|\Delta^{(t)}|}\ln\left(e\frac{3h^{(t)} + 1}{\delta}\right) \ .$$

If

$$M \geq \max\left\{ t\sqrt[3]{\Psi}\left(1+\frac{1}{2}\ln^{2/3}\left(t\sqrt[3]{\Psi}\right)\right), 25\right\}$$

then

$$\Pr\left(|\phi^{(t)}_{\text{MIX}} - |\Delta^{(t)}|| < \varepsilon|\Delta^{(t)}|\right) \geq 1-\delta \ .$$

Proof. For any $S \subseteq E^{(t)}$ let $\tau(S)$ be the number of triangles in S, i.e., the number of triplets of edges in S that compose a triangle in $G^{(t)}$. Define the function $g : 2^{E^{(t)}} \to \mathbb{R}$ as

$$g(S) = \left(\frac{t}{M}\right)^3 \tau(S) \ .$$

Assume that we run INDEP with $p = M/t$, and let $\mathcal{S}_{\text{IN}} \subseteq E^{(t)}$ be the sample built by INDEP (through independent sampling with fixed probability p). Assume also that we run MIX with parameter M, and let $\mathcal{S}_{\text{MIX}}$ be the sample built by MIX (through reservoir sampling with a reservoir of size M). We have that $\phi^{(t)}_{\text{IN}} = g(\mathcal{S}_{\text{IN}})$ and $\phi^{(t)}_{\text{MIX}} = g(\mathcal{S}_{\text{MIX}})$. Define now the binary function $f : 2^{E^{(t)}} \to \{0,1\}$ as

$$f(S) = \begin{cases} 1 & \text{if } |g(S) - |\Delta^{(t)}|| > \varepsilon|\Delta^{(t)}| \\ 0 & \text{otherwise} \end{cases} \ .$$

We now show that, for M as in the hypothesis, we have

$$p \geq \sqrt[3]{2\varepsilon^{-2}\frac{3h^{(t)}+1}{|\Delta^{(t)}|}\ln\left(e\sqrt{M}\frac{3h^{(t)}+1}{\delta}\right)} \ . \tag{7.9}$$

Assume for now that the above is true. From this, using Lemma 7.7 and the above fact about g we get that

$$\Pr\left(|\phi^{(t)}_{\text{IN}} - |\Delta^{(t)}|| > \varepsilon|\Delta^{(t)}|\right) = \Pr\left(f(\mathcal{S}_{\text{IN}}) = 1\right) < \frac{\delta}{e\sqrt{M}} \ .$$

From this and Lemma 7.8, we get that

$$\Pr\left(f(\mathcal{S}_{\text{MIX}}) = 1\right) \leq \delta$$

which, from the definition of f and the properties of g, is equivalent to

$$\Pr\left(|\phi^{(t)}_{\text{MIX}} - |\Delta^{(t)}|| > \varepsilon|\Delta^{(t)}|\right) \leq \delta$$

and the proof is complete. All that is left is to show that (7.9) holds for M as in the hypothesis.

Since $p = M/t$, we have that (7.9) holds for

$$\begin{aligned} M^3 &\geq t^3 2\varepsilon^{-2}\frac{3h^{(t)}+1}{|\Delta^{(t)}|}\ln\left(\sqrt{M}e\frac{3h^{(t)}+1}{\delta}\right) \\ &= t^3 2\varepsilon^{-2}\frac{3h^{(t)}+1}{|\Delta^{(t)}|}\left(\ln\left(e\frac{3h^{(t)}+1}{\delta}\right)+\frac{1}{2}\ln M\right) \ . \end{aligned} \tag{7.10}$$

We now show that (7.10) holds.

Let $A = t\sqrt[3]{\Psi}$ and let $B = t\sqrt[3]{\Psi}\ln^{2/3}\left(t\sqrt[3]{\Psi}\right)$. We now show that $A^3 + B^3$ is greater or equal to the r.h.s. of (7.10), hence $M^3 = (A+B)^3 > A^3 + B^3$ must also be greater or equal to the r.h.s. of (7.10), i.e., (7.10) holds. This really reduces to show that

$$B^3 \geq t^3 2\varepsilon^{-2}\frac{3h^{(t)}+1}{|\Delta^{(t)}|}\frac{1}{2}\ln M \tag{7.11}$$

as the r.h.s.of (7.10) can be written as

$$A^3 + t^3 2\varepsilon^{-2}\frac{3h^{(t)}+1}{|\Delta^{(t)}|}\frac{1}{2}\ln M \ .$$

We actually show that

$$B^3 \geq t^3 \Psi \frac{1}{2} \ln M \tag{7.12}$$

which implies (7.11) which, as discussed, in turn implies (7.10). Consider the ratio

$$\frac{B^3}{t^3 \Psi \frac{1}{2} \ln M} = \frac{\frac{1}{2} t^3 \Psi \ln^2(t\sqrt[3]{\Psi})}{t^3 \Psi \frac{1}{2} \ln M} = \frac{\ln^2(t\sqrt[3]{\Psi})}{\ln M} \geq \frac{\ln^2(t\sqrt[3]{\Psi})}{\ln\left(t\sqrt[3]{\Psi}\left(1+\ln^{2/3}\left(t\sqrt[3]{\Psi}\right)\right)\right)} \ . \tag{7.13}$$

We now show that $t\sqrt[3]{\Psi} \geq 5$. By the assumptions $t > M \geq 25$ and by

$$t\sqrt[3]{\Psi} \geq \frac{t}{\sqrt[3]{|\Delta^{(t)}|}} \geq \sqrt{t}$$

which holds because $|\Delta^{(t)}| \leq t^{3/2}$ (in a graph with t edges there can not be more than $t^{3/2}$ triangles) we have that $t\sqrt[3]{\Psi} \geq 5$. Hence Fact 7.9 holds and we can write, from (7.13):

$$\frac{\ln^2(t\sqrt[3]{\Psi})}{\ln\left(t\sqrt[3]{\Psi}\left(1+\ln^{2/3}\left(t\sqrt[3]{\Psi}\right)\right)\right)} \geq \frac{\ln^2(t\sqrt[3]{\Psi})}{\ln^2\left(t\sqrt[3]{\Psi}\right)} \geq 1,$$

which proves (7.12), and in cascade (7.11), (7.10), (7.9), and the thesis. □

Relationship between TRIÈST-BASE and MIX When both TRIÈST-BASE and MIX use a sample of size M, their respective estimators $\phi^{(t)}$ and $\phi^{(t}_{\text{MIX}}$ are related as discussed in the following resul:

Lemma 7.11. *For any $t > M$ we have*

$$\left|\phi^{(t)} - \phi^{(t)}_{\text{MIX}}\right| \leq \phi^{(t)}_{\text{MIX}} \frac{4}{M-2} \ .$$

Proof. We start by stating the following fact:

Fact 7.12. *For any $x > 2$, we have*

$$\frac{x^2}{(x-1)(x-2)} \leq 1 + \frac{4}{x-2} \ .$$

Let us now consider the ratio between $\frac{t(t-1)(t-2)}{M(M-1)(M-2)}$ and $(t/M)^3$. We have:

$$\begin{aligned} 1 \leq \frac{t(t-1)(t-2)}{M(M-1)(M-2)} \left(\frac{M}{t}\right)^3 &= \frac{M^2}{(M-1)(M-2)} \frac{(t-1)(t-2)}{t^2} \\ &\leq \frac{M^2}{(M-1)(M-2)} \\ &\leq 1 + \frac{4}{M-2} \end{aligned}$$

where the last step follows from Fact 7.12. Using this, we obtain

$$\begin{aligned} \left|\phi^{(t)} - \phi^{(t)}_{\text{MIX}}\right| &= \left|\tau^{(t)} \frac{t(t-1)(t-2)}{M(M-1)(M-2)} - \tau^{(t)} \left(\frac{t}{M}\right)^3\right| \\ &= \left|\tau^{(t)} \left(\frac{t}{M}\right)^3 \left(\frac{t(t-1)(t-2)}{M(M-1)(M-2)} \left(\frac{M}{t}\right)^3 - 1\right)\right| \\ &\leq \tau^{(t)} \left(\frac{t}{M}\right)^3 \frac{4}{M-2} \\ &= \phi^{(t)}_{\text{MIX}} \frac{4}{M-2} \ . \end{aligned}$$

□

Tying everything together Finally we can use the previous lemmas to prove our concentration result for TRIÈST-BASE.

Proof of Thm. 7.6. For M as in the hypothesis we have, from Lemma 7.10, that

$$\Pr\left(\phi^{(t)}_{\text{MIX}} \leq (1+\varepsilon/2)|\Delta^{(t)}|\right) \geq 1-\delta \ .$$

Suppose the event $\phi^{(t)}_{\text{MIX}} \leq (1+\varepsilon/2)|\Delta^{(t)}|$ (i.e., $|\phi^{(t)}_{\text{MIX}} - |\Delta^{(t)}|| \leq \varepsilon|\Delta^{(t)}|/2$) is indeed verified. From this and Lemma 7.11 we have

$$|\phi^{(t)} - \phi^{(t)}_{\text{MIX}}| \leq \left(1+\frac{\varepsilon}{2}\right)|\Delta^{(t)}|\frac{4}{M-2} \leq |\Delta^{(t)}|\frac{6}{M-2},$$

where the last inequality follows from the fact that $\varepsilon < 1$. Hence, given that $M \geq 12\varepsilon^{-1} + e^2 \geq 12\varepsilon^{-1} + 2$, we have

$$|\phi^{(t)} - \phi^{(t)}_{\text{MIX}}| \leq |\Delta^{(t)}|\frac{\varepsilon}{2} \ .$$

Using the above, we can then write:

$$\begin{aligned}
|\phi^{(t)} - |\Delta^{(t)}|| &= |\phi^{(t)} - \phi^{(t)}_{\text{MIX}} + \phi^{(t)}_{\text{MIX}} - |\Delta^{(t)}|| \\
&\leq |\phi^{(t)} - \phi^{(t)}_{\text{MIX}}| + |\phi^{(t)}_{\text{MIX}} - |\Delta^{(t)}|| \\
&\leq \frac{\varepsilon}{2}|\Delta^{(t)}| + \frac{\varepsilon}{2}|\Delta^{(t)}| = \varepsilon|\Delta^{(t)}|
\end{aligned}$$

which completes the proof. □

Comparison with fixed-probability approaches

We now compare the variance of TRIÈST-BASE to the variance of the fixed probability sampling approach MASCOT-C [144], which samples edges *independently* with a fixed probability p and uses $p^{-3}|\Delta_{\mathcal{S}}|$ as the estimate for the global number of triangles at time t. As shown by [144, Lemma 2], the variance of this estimator is

$$\text{Var}[p^{-3}|\Delta_{\mathcal{S}}|] = |\Delta^{(t)}|\bar{f}(p) + r^{(t)}\bar{g}(p),$$

where $\bar{f}(p) = p^{-3} - 1$ and $\bar{g}(p) = p^{-1} - 1$.

Assume that we give MASCOT-C the additional information that the stream has finite length T, and assume we run MASCOT-C with $p = M/T$ so that the expected sample size at the end of the stream is M.[5] Let $\mathbb{V}^{(t)}_{\text{fix}}$ be the resulting variance of the MASCOT-C estimator at time t, and let $\mathbb{V}^{(t)}$ be the variance of our estimator at time t (see (7.3)). For $t \leq M$, $\mathbb{V}^{(t)} = 0$, hence $\mathbb{V}^{(t)} \leq \mathbb{V}^{(t)}_{\text{fix}}$.

For $M < t < T$, we can show the following result:

Lemma 7.13. *Let $0 < \alpha < 1$ be a constant. For any constant $M > \max(\frac{8\alpha}{1-\alpha}, 42)$ and any $t \leq \alpha T$ we have $\mathbb{V}^{(t)} < \mathbb{V}^{(t)}_{\text{fix}}$.*

Proof. We start by stating the following fact:

Fact 7.14. *For any $x > 42$, we have*

$$\frac{x^2}{(x-3)(x-4)} \leq 1 + \frac{8}{x} \ .$$

[5] We are giving MASCOT-C a significant advantage: if only space M were available, we should run MASCOT-C with a sufficiently smaller $p' < p$, otherwise there would be a constant probability that MASCOT-C would run out of memory.

We focus on $t > M > 42$ otherwise the theorem is immediate. We show that for such conditions $f(M,t) < \bar{f}(M/T)$ and $g(M,t) < \bar{g}(M/T)$. Using the fact that $t \leq \alpha T$ and Fact 7.12, we have

$$\begin{aligned} f(M,t) - \bar{f}(M/T) &= \frac{t(t-1)(t-2)}{M(M-1)(M-2)} - \frac{T^3}{M^3} \\ &< \frac{\alpha^3 T^3}{M^3} \frac{M^2}{(M-1)(M-2)} - \frac{T^3}{M^3} \\ &\leq \frac{\alpha^3 T^3}{M^3} \left(1 + \frac{4}{M-2}\right) - \frac{T^3}{M^3} \\ &\leq \frac{T^3}{M^3} \left(\alpha^3 + \frac{4\alpha^3}{M-2} - 1\right) \quad . \end{aligned} \tag{7.14}$$

Given that T and M are ≥ 42, the r.h.s. of (7.14) is non-positive iff

$$\alpha^3 + \frac{4\alpha^3}{M-2} - 1 \leq 0 \ .$$

Solving for M we have that the above is verified when $M \geq \frac{4\alpha^3}{1-\alpha^3} + 2$. This is always true given our assumption that $M > \max(\frac{8\alpha}{1-\alpha}, 42)$: for any $0 < \alpha < 0.6$, we have $\frac{4\alpha^3}{1-\alpha^3} + 2 < 42 \leq M$ and for any $0.6 \leq \alpha < 1$ we have $\frac{4\alpha^3}{1-\alpha^3} + 2 < \frac{8\alpha}{1-\alpha} \leq M$. Hence the r.h.s. of (7.14) is ≤ 0 and $f(M,t) < \bar{f}(M/T)$.

We also have:

$$\begin{aligned} g(M,t) - \bar{g}(M/T) &= \frac{t(t-1)(t-2)(M-3)(M-4)}{(t-3)(t-4)M(M-1)(M-2)} - \frac{T}{M} \\ &< \frac{t}{M} \frac{t^2}{(t-3)(t-4)} - \frac{T}{M} \\ &\leq \frac{t}{M} \left(1 + \frac{8}{t}\right) - \frac{T}{M}, \end{aligned} \tag{7.15}$$

where the last inequality follow from Fact 7.14, since $t > M > 42$. Now, from (7.15) since $t \leq \alpha T$ and $t > M$, we can write:

$$g(M,t) - \bar{g}(M/T) < \frac{T}{M} \left(\alpha + \frac{8\alpha}{M} - 1\right) \quad .$$

The r.h.s. of this equation is non-positive given the assumption $M > \frac{8\alpha}{1-\alpha}$, hence $g(M,t) < \bar{g}(M/T)$. □

For example, if we set $\alpha = 0.99$ and run TRIÈST-BASE with $M \geq 400$ and MASCOT-C with $p = M/T$, we have that TRIÈST-BASE has strictly smaller variance than MASCOT-C for 99% of the stream.

What about $t = T$? The difference between the definitions of $\mathbb{V}^{(t)}_{\text{fix}}$ and $\mathbb{V}^{(t)}$ is in the presence of $\bar{f}(M/T)$ instead of $f(t)$ (resp. $\bar{g}(M/T)$ instead of $g(t)$) as well as the additional term $w^{(t)}h(M,t) \leq 0$ in our $\mathbb{V}^{(t)}$. Let $M(T)$ be an arbitrary slowly increasing function of T. For $T \to \infty$ we can show that $\lim_{T\to\infty} \frac{\bar{f}(M(T)/T)}{f(T)} = \lim_{T\to\infty} \frac{\bar{g}(M(T)/T)}{g(T)} = 1$, hence, *informally*, $\mathbb{V}^{(T)} \to \mathbb{V}^{(T)}_{\text{fix}}$, for $T \to \infty$.

A similar discussion also holds for the method we present in Sect. 7.4.2, and explains the results of our experimental evaluations, which shows that our algorithms have strictly lower (empirical) variance than fixed probability approaches for most of the stream.

Update time

The time to process an element of the stream is dominated by the computation of the shared neighborhood $\mathcal{N}_{u,v}$ in UpdateCounters. A Mergesort-based algorithm for the intersection requires $O\left(\deg(u) + \deg(v)\right)$ time, where the degrees are w.r.t. the graph $G_{\mathcal{S}}$. By storing the neighborhood of each vertex in a Hash Table (resp. an AVL tree), the update time can be reduced to $O(\min\{\deg(v), \deg(u)\})$ (resp. amortized time $O(\min\{\deg(v), \deg(u)\} + \log \max\{\deg(v), \deg(u)\})$).

7.4.2 Improved insertion algorithm – triès-impr

triès-impr triès-impr is a variant of triès-base with small modifications that result in higher-quality (i.e., lower variance) estimations. The changes are:

1. UpdateCounters is called *unconditionally for each element on the stream*, before the algorithm decides whether or not to insert the edge into $\mathcal{S}$. W.r.t. the pseudocode in Alg. 11, this change corresponds to moving the call to UpdateCounters on line 6 to *before* the **if** block. mascot [144] uses a similar idea, but triès-impr is significantly different as triès-impr allows edges to be removed from the sample, since it uses reservoir sampling.

2. triès-impr *never* decrements the counters when an edge is removed from $\mathcal{S}$. W.r.t. the pseudocode in Alg. 11, we remove the call to UpdateCounters on line 13.

3. UpdateCounters performs a *weighted* increase of the counters using $\eta^{(t)} = \max\{1, (t-1)(t-2)/(M(M-1))\}$ as weight. W.r.t. the pseudocode in Alg. 11, we replace "1" with $\eta^{(t)}$ on lines 19–22 (given change 2 above, all the calls to UpdateCounters have $\bullet = +$).

The resulting pseudocode for triès-impr is presented in Alg. 12.

Counters If we are interested only in estimating the global number of triangles in $G^{(t)}$, triès-impr needs to maintain only the counter τ and the edge sample $\mathcal{S}$ of size M, so it still requires space $O(M)$. If instead we are interested in estimating the local triangle counts, at any time t triès-impr maintains (non-zero) local counters *only* for the nodes u such that at least one triangle with a corner u has been detected by the algorithm up until time t. The number of such nodes may be greater than $O(M)$, but this is the price to pay to obtain estimations with lower variance (Thm. 7.16).

Estimation

When queried for an estimation, triès-impr returns the value of the corresponding counter, unmodified.

Analysis

We now present the analysis of the estimations computed by triès-impr, showing results involving their unbiasedness, their variance, and their concentration around their expectation. Results

ALGORITHM 12 TRIÈST-IMPR

Input: Insertion-only edge stream Σ, integer $M \geq 6$

$\mathcal{S} \leftarrow \emptyset$, $t \leftarrow 0$, $\tau \leftarrow 0$
for each element $(+, (u, v))$ from Σ **do**
  $t \leftarrow t + 1$
  UPDATECOUNTERS(u, v)
  if SAMPLEEDGE$((u, v), t)$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(u, v)\}$

function SAMPLEEDGE$((u, v), t)$
  if $t \leq M$ **then**
    return True
  else if FLIPBIASEDCOIN$(\frac{M}{t})$ = heads **then**
    $(u', v') \leftarrow$ random edge from $\mathcal{S}$
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(u', v')\}$
    return True
  return False

function UPDATECOUNTERS(u, v)
  $\mathcal{N}^{\mathcal{S}}_{u,v} \leftarrow \mathcal{N}^{\mathcal{S}}_{u} \cap \mathcal{N}^{\mathcal{S}}_{v}$
  $\eta = \max\{1, (t-1)(t-2)/(M(M-1))\}$
  for all $c \in \mathcal{N}^{\mathcal{S}}_{u,v}$ **do**
    $\tau \leftarrow \tau + \eta$
    $\tau_c \leftarrow \tau_c + \eta$
    $\tau_u \leftarrow \tau_u + \eta$
    $\tau_v \leftarrow \tau_v + \eta$

analogous to those in Thms. 7.15, 7.16, and 7.18 hold for the local triangle count for any $u \in V^{(t)}$, replacing the global quantities with the corresponding local ones.

Expectation

As in TRIÈST-BASE, the estimations by TRIÈST-IMPR are *exact* at time $t \leq M$ and unbiased for $t > M$. The proof of the following theorem follows steps similar to those in the proof of Thm 7.3:

Theorem 7.15. *We have* $\tau^{(t)} = |\Delta^{(t)}|$ *if* $t \leq M$ *and* $\mathbb{E}\left[\tau^{(t)}\right] = |\Delta^{(t)}|$ *if* $t > M$.

Proof. If $t \leq M$ TRIÈST-IMPR behaves exactly like TRIÈST-BASE, and the statement follows from Lemma 7.3.

Assume now $t > M$ and assume that $|\Delta^{(t)}| > 0$, otherwise, the algorithm deterministically returns 0 as an estimation and the thesis follows. Let $\lambda \in \Delta^{(t)}$ and denote with a, b, and c the edges of λ and assume, w.l.o.g., that they appear in this order (not necessarily consecutively) on the stream. Let t_λ be the time step at which c is on the stream. Let δ_λ be a random variable that takes value $\xi_{2,t_\lambda - 1}$ if a and b are in $\mathcal{S}$ at the end of time step $t_\lambda - 1$, and 0 otherwise. Since it must be $t_\lambda - 1 \geq 2$, from Lemma 7.1 we have that

$$\Pr\left(\delta_\lambda = \xi_{2,t_\lambda - 1}\right) = \frac{1}{\xi_{2,t_\lambda - 1}} \quad . \tag{7.16}$$

When $c = (u, v)$ is on the stream, i.e., at time t_λ, TRIÈST-IMPR calls UPDATECOUNTERS and increments the counter τ by $|\mathcal{N}^{\mathcal{S}}_{u,v}|\xi_{2,t_\lambda - 1}$, where $|\mathcal{N}^{\mathcal{S}}_{u,v}|$ is the number of triangles with (u, v) as an

edge in $\Delta^{\mathcal{S}\cup\{c\}}$. All these triangles have the corresponding random variables taking the same value $\xi_{2,t_\lambda-1}$. This means that the random variable $\tau^{(t)}$ can be expressed as

$$\tau^{(t)} = \sum_{\lambda\in\Delta^{(t)}} \delta_\lambda \ .$$

From this, linearity of expectation, and (7.16), we get

$$\mathbb{E}\left[\tau^{(t)}\right] = \sum_{\lambda\in\Delta^{(t)}} \mathbb{E}[\delta_\lambda] = \sum_{\lambda\in\Delta^{(t)}} \xi_{2,t_\lambda-1}\Pr\left(\delta_\lambda = \xi_{2,t_\lambda-1}\right) = \sum_{\lambda\in\Delta^{(t)}} \xi_{2,t_\lambda-1}\frac{1}{\xi_{2,t_\lambda-1}} = |\Delta^{(t)}| \ .$$

□

Variance

We now show an *upper bound* to the variance of the TRIÈST-IMPR estimations for $t > M$. The proof relies on a very careful analysis of the covariance of two triangles which depends on the order of arrival of the edges in the stream (which we assume to be adversarial). For any $\lambda \in \Delta^{(t)}$ we denote as t_λ the time at which the last edge of λ is observed on the stream. Let $z^{(t)}$ be the number of unordered pairs (λ, γ) of distinct triangles in $G^{(t)}$ that share an edge g and are such that:

1. g is neither the last edge of λ nor γ on the stream; and
2. $\min\{t_\lambda, t_\gamma\} > M + 1$.

Theorem 7.16. *Then, for any time $t > M$, we have*

$$\mathrm{Var}\left[\tau^{(t)}\right] \leq |\Delta^{(t)}|(\eta^{(t)} - 1) + z^{(t)}\frac{t-1-M}{M} \ . \tag{7.17}$$

The bound to the variance presented in (7.17) is rather pessimistic and loose. Specifically, it does not contain the *negative* contribution to the variance given by the $\binom{|\Delta^{(t)}|}{2} - z^{(t)}$ triangles that do not satisfy the requirements in the definition of $z^{(t)}$. Among these pairs there are, for example, all pairs of triangles not sharing any edge, but also many pairs of triangles that share an edge, as the definition of $z^{(t)}$ consider only a subsets of these. All these pairs would give a negative contribution to the variance, i.e., decrease the r.h.s. of (7.17), whose more correct form would be

$$|\Delta^{(t)}|(\eta^{(t)} - 1) + z^{(t)}\frac{t-1-M}{M} + \left(\binom{|\Delta^{(t)}|}{2} - z^{(t)}\right)\omega_{M,t}$$

where $\omega_{M,t}$ is (an upper bound to) the minimum *negative* contribution of a pair of triangles that do not satisfy the requirements in the definition of $z^{(t)}$. Sadly, computing informative upper bounds to $\omega_{M,t}$ is not straightforward, even in the restricted setting where only pairs of triangles not sharing any edge are considered.

To prove Thm. 7.16 we first need Lemma 7.17: For any time step t and any edge $e \in E^{(t)}$, we denote with t_e the time step at which e is on the stream. For any $\lambda \in \Delta^{(t)}$, let $\lambda = (\ell_1, \ell_2, \ell_3)$, where the edges are numbered in order of appearance on the stream. We define the event D_λ as the event that ℓ_1 and ℓ_2 are both in the edge sample $\mathcal{S}$ at the end of time step $t_\lambda - 1$.

Lemma 7.17. *Let $\lambda = (\ell_1, \ell_2, \ell_3)$ and $\gamma = (g_1, g_2, g_3)$ be two* disjoint *triangles, where the edges are numbered in order of appearance on the stream, and assume, w.l.o.g., that the last edge of λ is on the stream* before *the last edge of γ. Then*

$$\Pr(D_\gamma \mid D_\lambda) \leq \Pr(D_\gamma) \ .$$

Proof. Consider first the case where *all* edges of λ appear on the stream before *any* edge of γ, i.e.,

$$t_{\ell_1} < t_{\ell_2} < t_{\ell_3} < t_{g_1} < t_{g_2} < t_{g_3} \ .$$

The presence or absence of either or both ℓ_1 and ℓ_2 in $\mathcal{S}$ at the beginning of time step t_{ℓ_3} (i.e., whether D_λ happens or not) has no effect whatsoever on the probability that g_1 and g_2 are in the sample $\mathcal{S}$ at the beginning of time step t_{g_3}. Hence in this case,

$$\Pr(D_\gamma \mid D_\lambda) = \Pr(D_\gamma) \ .$$

Consider now the case where, for any $i \in \{1, 2\}$, the edges $g_1, \ldots, g_i$ appear on the stream before ℓ_3 does. Define now the events

- A_i: "the edges $g_1, \ldots, g_i$ are in the sample $\mathcal{S}$ at the beginning of time step t_{ℓ_3}."
- B_i: if $i = 1$, this is the event "the edge g_2 is inserted in the sample $\mathcal{S}$ during time step t_{g_2}." If $i = 2$, this event is the whole event space, i.e., the event that happens with probability 1.
- C: "neither g_1 nor g_2 were among the edges removed from $\mathcal{S}$ between the beginning of time step t_{ℓ_3} and the beginning of time step t_{g_3}."

We can rewrite D_γ as

$$D_\gamma = A_i \cap B_i \cap C \ .$$

Hence

$$\begin{aligned} \Pr(D_\gamma \mid D_\lambda) &= \Pr\left(A_i \cap B_i \cap C \mid D_\lambda\right) \\ &= \Pr\left(A_i \mid D_\lambda\right) \Pr\left(B_i \cap C \mid A_i \cap D_\lambda\right) \ . \end{aligned} \tag{7.18}$$

We now show that

$$\Pr\left(A_i \mid D_\lambda\right) \leq \Pr\left(A_i\right) \ .$$

If we assume that $t_{\ell_3} \leq M+1$, then all the edges that appeared on the stream up until the beginning of t_{ℓ_3} are in $\mathcal{S}$. Therefore,

$$\Pr\left(A_i \mid D_\lambda\right) = \Pr(A_i) = 1 \ .$$

Assume instead that $t_{\ell_3} > M+1$. Among the $\binom{t_{\ell_3}-1}{M}$ subsets of $E^{(t_{\ell_3}-1)}$ of size M, there are $\binom{t_{\ell_3}-3}{M-2}$ that contain ℓ_1 and ℓ_2. From Lemma 7.2, we have that at the beginning of time t_{ℓ_3}, $\mathcal{S}$ is a subset of size M of $E^{(t_{\ell_3}-1)}$ chosen uniformly at random. Hence, if we condition on the fact that $\{\ell_1, \ell_2\} \subset \mathcal{S}$, we have that $\mathcal{S}$ is chosen uniformly at random from the $\binom{t_{\ell_3}-3}{M-2}$ subsets of $E^{(t_{\ell_3}-1)}$ of size M that contain ℓ_1 and ℓ_2. Among these, there are $\binom{t_{\ell_3}-3-i}{M-2-i}$ that also contain $g_1, \ldots, g_i$. Therefore,

$$\Pr(A_i \mid D_\lambda) = \frac{\binom{t_{\ell_3}-3-i}{M-2-i}}{\binom{t_{\ell_3}-3}{M-2}} = \prod_{j=0}^{i-1} \frac{M-2-j}{t_{\ell_3}-3-j} \ .$$

From Lemma 7.1 we have

$$\Pr(A_i) = \frac{1}{\xi_{i,t_{\ell_3}-1}} = \prod_{j=0}^{i-1} \frac{M-j}{t_{\ell_3}-1-j},$$

where the last equality comes from the assumption $t_{\ell_3} > M+1$. From the same assumption and from the fact that for any $j \geq 0$ and any $y \geq x > j$ it holds $\frac{x-j}{y-j} \leq \frac{x}{y}$, then we have

$$\Pr(A_i \mid D_\lambda) \leq \Pr(A_i) \ .$$

This implies, from (7.18), that

$$\Pr(D_\gamma \mid D_\lambda) \leq \Pr(A_i) \Pr(B_i \cap C \mid A_i \cap D_\lambda) \ . \tag{7.19}$$

Consider now the events B_i and C. When conditioned on A_i, these event are *both independent* from D_λ: if the edges $g_1, \ldots, g_i$ are in $\mathcal{S}$ at the beginning of time t_{ℓ_3}, the fact that the edges ℓ_1 and ℓ_2 were also in $\mathcal{S}$ at the beginning of time t_{ℓ_3} has no influence whatsoever on the actions of the algorithm (i.e., whether an edge is inserted in or removed from $\mathcal{S}$). Thus,

$$\Pr(A_i)\Pr(B_i \cap C \mid A_i \cap D_\lambda) = \Pr(A_i)\Pr(B_i \cap C \mid A_i) \ .$$

Putting this together with (7.19), we obtain

$$\Pr(D_\gamma \mid D_\lambda) \leq \Pr(A_i)\Pr(B_i \cap C \mid A_i) \leq \Pr(A_i \cap B_i \cap C) \leq \Pr(D_\gamma) \ ,$$

where the last inequality follows from the fact that $D_\gamma = A_i \cap B_i \cap C$ by definition. □

We can now prove Thm. 7.16.

Proof of Thm. 7.16. Assume $|\Delta^{(t)}| > 0$, otherwise TRIÈST-IMPR estimation is deterministically correct and has variance 0 and the thesis holds. Let $\lambda \in \Delta^{(t)}$ and let δ_λ be a random variable that takes value $\xi_{2,t_\lambda-1}$ if both ℓ_1 and ℓ_2 are in $\mathcal{S}$ at the end of time step $t_\lambda - 1$, and 0 otherwise. Since

$$\mathrm{Var}\left[\delta_\lambda\right] = \xi_{2,t_\lambda-1} - 1 \leq \xi_{2,t-1},$$

we have:

$$\begin{aligned}
\mathrm{Var}\left[\tau^{(t)}\right] &= \mathrm{Var}\left[\sum_{\lambda\in\Delta^{(t)}} \delta_\lambda\right] = \sum_{\lambda\in\Delta^{(t)}}\sum_{\gamma\in\Delta^{(t)}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] \\
&= \sum_{\lambda\in\Delta^{(t)}} \mathrm{Var}\left[\delta_\lambda\right] + \sum_{\substack{\lambda,\gamma\in\Delta^{(t)}\\ \lambda\neq\gamma}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] \\
&\leq |\Delta^{(t)}|(\xi_{2,t-1} - 1) + \sum_{\substack{\lambda,\gamma\in\Delta^{(t)}\\ \lambda\neq\gamma}} \left(\mathbb{E}[\delta_\lambda\delta_\gamma] - \mathbb{E}[\delta_\lambda]\mathbb{E}[\delta_\gamma]\right) \\
&\leq |\Delta^{(t)}|(\xi_{2,t-1} - 1) + \sum_{\substack{\lambda,\gamma\in\Delta^{(t)}\\ \lambda\neq\gamma}} \left(\mathbb{E}[\delta_\lambda\delta_\gamma] - 1\right) \ . \qquad (7.20)
\end{aligned}$$

For any $\lambda \in \Delta^{(t)}$ define $q_\lambda = \xi_{2,t_\lambda-1}$. Assume now $|\Delta^{(t)}| \geq 2$, otherwise we have $r^{(t)} = w^{(t)} = 0$ and the thesis holds as the second term on the r.h.s. of (7.20) is 0. Let now λ and γ be two distinct triangles in $\Delta^{(t)}$ (hence $t \geq 5$). We have

$$\mathbb{E}\left[\delta_\lambda\delta_\gamma\right] = q_\lambda q_\gamma \Pr\left(\delta_\lambda\delta_\gamma = q_\lambda q_\gamma\right)$$

The event "$\delta_\lambda\delta_\gamma = q_\lambda q_\gamma$" is the intersection of events $D_\lambda \cap D_\gamma$, where D_λ is the event that the first two edges of λ are in $\mathcal{S}$ at the end of time step $t_\lambda - 1$, and similarly for D_γ. We now look at $\Pr(D_\lambda \cap D_\gamma)$ in the various possible cases.

Assume that λ and γ do not share any edge, and, w.l.o.g., that the third (and last) edge of λ appears on the stream before the third (and last) edge of γ, i.e., $t_\lambda < t_\gamma$. From Lemma 7.17 and Lemma 7.1 we then have

$$\Pr(D_\lambda \cap D_\gamma) = \Pr(D_\gamma|D_\lambda)\Pr(D_\lambda) \leq \Pr(D_\gamma)\Pr(D_\lambda) \leq \frac{1}{q_\lambda q_\gamma} \ .$$

Consider now the case where λ and γ share an edge g. W.l.o.g., let us assume that $t_\lambda \leq t_\gamma$ (since the shared edge may be the last on the stream both for λ and for γ, we may have $t_\lambda = t_\gamma$). There are the following possible sub-cases :

g is the last on the stream among all the edges of λ and γ In this case we have $t_\lambda = t_\gamma$. The event "$D_\lambda \cap D_\gamma$" happens if and only if the *four* edges that, together with g, compose λ and γ are all in $\mathcal{S}$ at the end of time step $t_\lambda - 1$. Then, from Lemma 7.1 we have

$$\Pr(D_\lambda \cap D_\gamma) = \frac{1}{\xi_{4,t_\lambda-1}} \leq \frac{1}{q_\lambda}\frac{(M-2)(M-3)}{(t_\lambda-3)(t_\lambda-4)} \leq \frac{1}{q_\lambda}\frac{M(M-1)}{(t_\lambda-1)(t_\lambda-2)} \leq \frac{1}{q_\lambda q_\gamma} \ .$$

g is the last on the stream among all the edges of λ and the first among all the edges of γ In this case, we have that D_λ and D_γ are independent. Indeed the fact that the first two edges of λ (neither of which is g) are in $\mathcal{S}$ when g arrives on the stream has no influence on the probability that g and the second edge of γ are inserted in $\mathcal{S}$ and are not evicted until the third edge of γ is on the stream. Hence we have

$$\Pr(D_\lambda \cap D_\gamma) = \Pr(D_\gamma)\Pr(D_\lambda) = \frac{1}{q_\lambda q_\gamma} \ .$$

g is the last on the stream among all the edges of λ and the second among all the edges of γ In this case we can follow an approach similar to the one in the proof for Lemma 7.17 and have that

$$\Pr(D_\lambda \cap D_\gamma) \leq \Pr(D_\gamma)\Pr(D_\lambda) \leq \frac{1}{q_\lambda q_\gamma} \ .$$

The intuition behind this is that if the first two edges of λ are in $\mathcal{S}$ when g is on the stream, their presence lowers the probability that the first edge of γ is in $\mathcal{S}$ at the same time, and hence that the first edge of γ and g are in $\mathcal{S}$ when the last edge of γ is on the stream.

g is not the last on the stream among all the edges of λ There are two situations to consider, or better, one situation and all other possibilities. The situation we consider is that

1. g is the first edge of γ on the stream; and
2. the second edge of γ to be on the stream is on the stream at time $t_2 > t_\lambda$.

Suppose this is the case. Recall that if D_λ is verified, than we know that g is in $\mathcal{S}$ at the beginning of time step t_λ. Define the following events:

- E_1: "the set of edges evicted from $\mathcal{S}$ between the beginning of time step t_λ and the beginning of time step t_2 does not contain g."
- E_2: "the second edge of γ, which is on the stream at time t_2, is inserted in $\mathcal{S}$ and the edge that is evicted is not g."
- E_3: "the set of edges evicted from $\mathcal{S}$ between the beginning of time step $t_2 + 1$ and the beginning of time step t_γ does not contain either g or the second edge of γ."

We can then write

$$\Pr(D_\gamma \mid D_\lambda) = \Pr(E_1 \mid D_\lambda)\Pr(E_2 \mid E_1 \cap D_\lambda)\Pr(E_3 \mid E_2 \cap E_1 \cap D_\lambda) \ .$$

We now compute the probabilities on the r.h.s., where we denote with $\mathbb{1}_{t_2>M}(1)$ the function that has value 1 if $t_2 > M$, and value 0 otherwise:

$$\begin{aligned}
\Pr(E_1 \mid D_\lambda) &= \prod_{j=\max\{t_\lambda, M+1\}}^{t_2-1} \left(\left(1-\frac{M}{j}\right) + \frac{M}{j}\left(\frac{M-1}{M}\right)\right) \\
&= \prod_{j=\max\{t_\lambda, M+1\}}^{t_2-1} \frac{j-1}{j} = \frac{\max\{t_\lambda - 1, M\}}{\max\{M, t_2-1\}} \ ; \\
\Pr(E_2 \mid E_1 \cap D_\lambda) &= \frac{M}{\max\{t_2, M\}} \frac{M - \mathbb{1}_{t_2>M}(1)}{M} = \frac{M - \mathbb{1}_{t_2>M}(1)}{\max\{t_2, M\}} \ ; \\
\Pr(E_3 \mid E_2 \cap E_1 \cap D_\lambda) &= \prod_{j=\max\{t_2+1, M+1\}}^{t_\gamma-1} \left(\left(1-\frac{M}{j}\right) + \frac{M}{j}\left(\frac{M-2}{M}\right)\right) \\
&= \prod_{j=\max\{t_2+1, M+1\}}^{t_\gamma-1} \frac{j-2}{j} = \frac{\max\{t_2, M\}\max\{t_2-1, M-1\}}{\max\{t_\gamma-2, M-1\}\max\{t_\gamma-1, M\}} \ .
\end{aligned}$$

Hence, we have

$$\Pr(D_\gamma \mid D_\lambda) = \frac{\max\{t_\lambda - 1, M\}(M - \mathbb{1}_{t_2>M}(1))\max\{t_2-1, M-1\}}{\max\{M, t_2-1\}\max\{(t_\gamma-2)(t_\gamma-1), M(M-1)\}} \ .$$

With a (somewhat tedious) case analysis we can verify that

$$\Pr(D_\gamma \mid D_\lambda) \leq \frac{1}{q_\gamma}\frac{\max\{M, t_\lambda - 1\}}{M} \ .$$

Consider now the complement of the situation we just analyzed. In this case, two edges of γ, that is, g and another edge h are on the stream before time t_λ, in some non-relevant order (i.e., g could be the first or the second edge of γ on the stream). Define now the following events:

- E_1: "h and g are both in $\mathcal{S}$ at the beginning of time step t_λ."
- E_2: "the set of edges evicted from $\mathcal{S}$ between the beginning of time step t_λ and the beginning of time step t_γ does not contain either g or h."

We can then write

$$\Pr(D_\gamma \mid D_\lambda) = \Pr(E_1 \mid D_\lambda)\Pr(E_2 \mid E_1 \cap D_\lambda) \ .$$

If $t_\lambda \leq M+1$, we have that $\Pr(E_1 \mid D_\lambda) = 1$. Consider instead the case $t_\lambda > M+1$. If D_λ is verified, then both g and the other edge of λ are in $\mathcal{S}$ at the beginning of time step t_λ. At this time, all subsets of $E^{(t_\lambda-1)}$ of size M and containing both g and the other edge of λ have an equal probability of being $\mathcal{S}$, from Lemma 7.2. There are $\binom{t_\lambda-3}{M-2}$ such sets. Among these, there are $\binom{t_\lambda-4}{M-3}$ sets that also contain h. Therefore, if $t_\lambda > M+1$, we have

$$\Pr(E_1 \mid D_\lambda) = \frac{\binom{t_\lambda-4}{M-3}}{\binom{t_\lambda-3}{M-2}} = \frac{M-2}{t_\lambda-3} \ .$$

Considering what we said before for the case $t_\lambda \leq M+1$, we then have

$$\Pr(E_1 \mid D_\lambda) = \min\left\{1, \frac{M-2}{t_\lambda-3}\right\} \ .$$

We also have

$$\Pr(E_2 \mid E_1 \cap D_\lambda) = \prod_{j=\max\{t_\lambda, M+1\}}^{t_\gamma-1} \frac{j-2}{j} = \frac{\max\{(t_\lambda-2)(t_\lambda-1), M(M-1)\}}{\max\{(t_\gamma-2)(t_\gamma-1), M(M-1)\}} \ .$$

Therefore,

$$\Pr(D_\gamma \mid D_\lambda) = \min\left\{1, \frac{M-2}{t_\lambda - 3}\right\} \frac{\max\{(t_\lambda-2)(t_\lambda-1), M(M-1)\}}{\max\{(t_\gamma-2)(t_\gamma-1), M(M-1)\}} \ .$$

With a case analysis, one can show that

$$\Pr(D_\gamma \mid D_\lambda) \leq \frac{1}{q_\gamma} \frac{\max\{M, t_\lambda - 1\}}{M} \ .$$

To recap we have the following two scenarios when considering two distinct triangles γ and λ:

1. if λ and γ share an edge and, assuming w.l.o.g. that the third edge of λ is on the stream no later than the third edge of γ, and the shared edge is neither the last among all edges of λ to appear on the stream nor the last among all edges of γ to appear on the stream, then we have

$$\begin{aligned}\mathrm{Cov}[\delta_\lambda, \delta_\gamma] &\leq \mathbb{E}[\delta_\lambda \delta_\gamma] - 1 = q_\lambda q_\gamma \Pr(\delta_\lambda \delta_\gamma = q_\lambda q_\gamma) - 1 \\ &\leq q_\lambda q_\gamma \frac{1}{q_\lambda q_\gamma} \frac{\max\{M, t_\lambda - 1\}}{M} - 1 \leq \frac{\max\{M, t_\lambda - 1\}}{M} - 1 \leq \frac{t - 1 - M}{M};\end{aligned}$$

   where the last inequality follows from the fact that $t_\lambda \leq t$ and $t - 1 \geq M$.

   For the pairs (λ, γ) such that $t_\lambda \leq M + 1$, we have $\max\{M, t_\lambda - 1\}/M = 1$ and therefore $\mathrm{Cov}[\delta_\lambda, \delta_\gamma] \leq 0$. We should therefore only consider the pairs for which $t_\lambda > M + 1$. Their number is given by $z^{(t)}$.

2. in all other cases, including when γ and λ do not share an edge, we have $\mathbb{E}[\delta_\lambda \delta_\gamma] \leq 1$, and since $\mathbb{E}[\delta_\lambda]\mathbb{E}[\delta_\gamma] = 1$, we have

$$\mathrm{Cov}[\delta_\lambda, \delta_\gamma] \leq 0 \ .$$

Hence, we can bound

$$\sum_{\substack{\lambda, \gamma \in \Delta^{(t)} \\ \lambda \neq \gamma}} \mathrm{Cov}[\delta_\lambda, \delta_\gamma] \leq z^{(t)} \frac{t - 1 - M}{M}$$

and the thesis follows by combining this into (7.20). □

Concentration

We now show a concentration result on the estimation of TRIÈST-IMPR, which relies on Chebyshev's inequality [164, Thm. 3.6] and Thm. 7.16.

Theorem 7.18. *Let $t \geq 0$ and assume $|\Delta^{(t)}| > 0$. For any $\varepsilon, \delta \in (0, 1)$, if*

$$M > \max\left\{\sqrt{\frac{2(t-1)(t-2)}{\delta\varepsilon^2|\Delta^{(t)}| + 2} + \frac{1}{4}} + \frac{1}{2}, \frac{2z^{(t)}(t-1)}{\delta\varepsilon^2|\Delta^{(t)}|^2 + 2z^{(t)}}\right\}$$

then $|\tau^{(t)} - |\Delta^{(t)}|| < \varepsilon|\Delta^{(t)}|$ with probability $> 1 - \delta$.

Proof. By Chebyshev's inequality it is sufficient to prove that

$$\frac{\mathrm{Var}[\tau^{(t)}]}{\varepsilon^2|\Delta^{(t)}|^2} < \delta \ .$$

We can write

$$\frac{\mathrm{Var}[\tau^{(t)}]}{\varepsilon^2|\Delta^{(t)}|^2} \leq \frac{1}{\varepsilon^2|\Delta^{(t)}|}\left((\eta(t) - 1) + z^{(t)}\frac{t - 1 - M}{M|\Delta^{(t)}|}\right) \ .$$

Hence it is sufficient to impose the following two conditions:

Condition 1

$$\frac{\delta}{2} > \frac{\eta(t)-1}{\varepsilon^2|\Delta^{(t)}|} > \frac{1}{\varepsilon^2|\Delta^{(t)}|}\frac{(t-1)(t-2)-M(M-1)}{M(M-1)}, \tag{7.21}$$

which is verified for:

$$M(M-1) > \frac{2(t-1)(t-2)}{\delta\varepsilon^2|\Delta^{(t)}|+2}.$$

As $t > M$, we have $\frac{2(t-1)(t-2)}{\delta\varepsilon^2|\Delta^{(t)}|+2} > 0$. The condition in (7.21) is thus verified for:

$$M > \frac{1}{2}\left(\sqrt{4\frac{2(t-1)(t-2)}{\delta\varepsilon^2|\Delta^{(t)}|+2}+1}+1\right)$$

Condition 2

$$\frac{\delta}{2} > z^{(t)}\frac{t-1-M}{M\varepsilon^2|\Delta^{(t)}|^2},$$

which is verified for:

$$M > \frac{2z^{(t)}(t-1)}{\delta\varepsilon^2|\Delta^{(t)}|^2+2z^{(t)}}.$$

The theorem follows. □

In Thms. 7.16 and 7.18, it is possible to replace the value $z^{(t)}$ with the more interpretable $r^{(t)}$, which is agnostic to the order of the edges on the stream but gives a looser upper bound to the variance.

7.4.3 Fully-dynamic algorithm – TRIÈST-FD

TRIÈST-FD TRIÈST-FD computes unbiased estimates of the global and local triangle counts in a *fully-dynamic stream where edges are inserted/deleted in any arbitrary, adversarial order*. It is based on *random pairing* (RP) [86], a sampling scheme that extends reservoir sampling and can handle deletions. The idea behind the RP scheme is that edge deletions seen on the stream will be "compensated" by future edge insertions. Following RP, TRIÈST-FD keeps a counter d_i (resp. d_o) to keep track of the number of uncompensated edge deletions involving an edge e that was (resp. was *not*) in $\mathcal{S}$ at the time the deletion for e was on the stream.

When an edge deletion for an edge $e \in E^{(t-1)}$ is on the stream at the beginning of time step t, then, if $e \in \mathcal{S}$ at this time, TRIÈST-FD removes e from $\mathcal{S}$ (effectively decreasing the number of edges stored in the sample by one) and increases d_i by one. Otherwise, it just increases d_o by one. When an edge insertion for an edge $e \notin E^{(t-1)}$ is on the stream at the beginning of time step t, if $d_\mathrm{i} + d_\mathrm{o} = 0$, then TRIÈST-FD follows the standard reservoir sampling scheme. If $|\mathcal{S}| < M$, then e is deterministically inserted in $\mathcal{S}$ without removing any edge from $\mathcal{S}$ already in $\mathcal{S}$, otherwise it is inserted in $\mathcal{S}$ with probability M/t, replacing an uniformly-chosen edge already in $\mathcal{S}$. If instead $d_\mathrm{i} + d_\mathrm{o} > 0$, then e is inserted in $\mathcal{S}$ with probability $d_\mathrm{i}/(d_\mathrm{i} + d_\mathrm{o})$; since it must be $d_\mathrm{i} > 0$, then it must be $|\mathcal{S}| < M$ and no edge already in $\mathcal{S}$ needs to be removed. In any case, after having handled the eventual insertion of e into $\mathcal{S}$, the algorithm decreases d_i by 1 if e was inserted in $\mathcal{S}$, otherwise it decreases d_o by 1. TRIÈST-FD also keeps track of $s^{(t)} = |E^{(t)}|$ by appropriately incrementing or

ALGORITHM 13 TRIÈST-FD

Input: Fully-dynamic edge stream Σ, integer $M \geq 6$

$\mathcal{S} \leftarrow \emptyset$, $d_\text{i} \leftarrow 0$, $d_\text{o} \leftarrow 0$, $t \leftarrow 0$, $s \leftarrow 0$
for each element $\big(\bullet, (u,v)\big)$ from Σ **do**
  $t \leftarrow t + 1$
  $s \leftarrow s \bullet 1$
  if $\bullet = +$ **then**
    if SAMPLEEDGE (u,v) **then**
      UPDATECOUNTERS$\big(+, (u,v)\big)$ ▷ UPDATECOUNTERS is defined as in Alg. 11.
  else if $(u,v) \in \mathcal{S}$ **then**
    UPDATECOUNTERS$\big(-, (u,v)\big)$
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(u,v)\}$
    $d_\text{i} \leftarrow d_\text{i} + 1$
  else $d_\text{o} \leftarrow d_\text{o} + 1$

function SAMPLEEDGE(u,v)
  if $d_\text{o} + d_\text{i} = 0$ **then**
    if $|\mathcal{S}| < M$ **then**
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{(u,v)\}$
      return True
    else if FLIPBIASEDCOIN$(\frac{M}{t})$ = heads **then**
      Select (z,w) uniformly at random from $\mathcal{S}$
      UPDATECOUNTERS$\big(-, (z,w)\big)$
      $\mathcal{S} \leftarrow \big(\mathcal{S} \setminus \{(z,w)\}\big) \cup \{(u,v)\}$
      return True
  else if FLIPBIASEDCOIN$\Big(\frac{d_\text{i}}{d_\text{i}+d_\text{o}}\Big)$ = heads **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(u,v)\}$
    $d_\text{i} \leftarrow d_\text{i} - 1$
    return True
  else
    $d_\text{o} \leftarrow d_\text{o} - 1$
    return False

decrementing a counter by 1 depending on whether the element on the stream is an edge insertion or deletion. The pseudocode for TRIÈST-FD is presented in Alg. 13 where the UPDATECOUNTERS procedure is the one from Alg. 11.

Estimation

We denote as $M^{(t)}$ the size of $\mathcal{S}$ at the end of time t (we always have $M^{(t)} \leq M$). For any time t, let $d_\text{i}^{(t)}$ and $d_\text{o}^{(t)}$ be the value of the counters d_i and d_o at the end of time t respectively, and let $\omega^{(t)} = \min\{M, s^{(t)} + d_\text{i}^{(t)} + d_\text{o}^{(t)}\}$. Define

$$\kappa^{(t)} = 1 - \sum_{j=0}^{2} \binom{s^{(t)}}{j} \binom{d_\text{i}^{(t)} + d_\text{o}^{(t)}}{\omega^{(t)} - j} \Big/ \binom{s^{(t)} + d_\text{i}^{(t)} + d_\text{o}^{(t)}}{\omega^{(t)}} \ . \tag{7.22}$$

For any three positive integers a, b, c s.t. $a \leq b \leq c$, define[6]

$$\psi_{a,b,c} = \binom{c}{b} \Big/ \binom{c-a}{b-a} = \prod_{i=0}^{a-1} \frac{c-i}{b-i} \ .$$

[6] We follow the convention that $\binom{0}{0} = 1$.

When queried at the end of time t, for an estimation of the global number of triangles, TRIÈST-FD returns

$$\rho^{(t)} = \begin{cases} 0 \text{ if } M^{(t)} < 3 \\ \frac{\tau^{(t)}}{\kappa^{(t)}} \psi_{3,M^{(t)},s^{(t)}} = \frac{\tau^{(t)}}{\kappa^{(t)}} \frac{s^{(t)}(s^{(t)}-1)(s^{(t)}-2)}{M^{(t)}(M^{(t)}-1)(M^{(t)}-2)} \text{ othw.} \end{cases}$$

TRIÈST-FD can keep track of $\kappa^{(t)}$ during the execution, each update of $\kappa^{(t)}$ taking time $O(1)$. Hence the time to return the estimations is still $O(1)$.

Analysis

We now present the analysis of the estimations computed by TRIÈST-IMPR, showing results involving their unbiasedness, their variance, and their concentration around their expectation. Results analogous to those in Thms. 7.19, 7.25, and 7.30 hold for the local triangle count for any $u \in V^{(t)}$, replacing the global quantities with the corresponding local ones.

Expectation

Let t^* be the first $t \geq M+1$ such that $|E^{(t)}| = M+1$, if such a time step exists (otherwise $t^* = +\infty$).

Theorem 7.19. *We have $\rho^{(t)} = |\Delta^{(t)}|$ for all $t < t^*$, and $\mathbb{E}\left[\rho^{(t)}\right] = |\Delta^{(t)}|$ for $t \geq t^*$.*

The proof, relies on properties of RP and on the definitions of $\kappa^{(t)}$ and $\rho^{(t)}$. Specifically, it uses Lemma 7.22, which is the correspondent of Lemma 7.1 but for RP, and some additional technical lemmas (including an equivalent of Lemma 7.4 but for RP) and combine them using the law of total expectation by conditioning on the value of $M^(t)$.

Before proving Thm. 7.19 we need the following technical lemmas.

The following is a corollary of [86, Thm. 1].

Lemma 7.20. *For any $t > 0$, and any j, $0 \leq j \leq s^{(t)}$, let $\mathcal{B}^{(t)}$ be the collection of subsets of $E^{(t)}$ of size j. For any $B \in \mathcal{B}^{(t)}$ it holds*

$$\Pr\left(\mathcal{S} = B \mid M^{(t)} = j\right) = \frac{1}{\binom{|E^{(t)}|}{j}} \ .$$

That is, conditioned on its size at the end of time step t, $\mathcal{S}$ is equally likely to be, at the end of time step t, any of the subsets of $E^{(t)}$ of that size.

The next lemma is an immediate corollary of [86, Thm. 2].

Lemma 7.21. *Recall the definition of $\kappa^{(t)}$ from (7.22). We have*

$$\kappa^{(t)} = \Pr(M^{(t)} \geq 3) \ .$$

The next lemma follows from Lemma 7.20 in the same way as Lemma 7.1 follows from Lemma 7.2.

Lemma 7.22. *For any time step t and any j, $0 \leq j \leq s^{(t)}$, let B be any subset of $E^{(t)}$ of size $|B| = k \leq s^{(t)}$. Then, at the end of time step t,*

$$\Pr\left(B \subseteq \mathcal{S} \mid M^{(t)} = j\right) = \begin{cases} 0 & \text{if } k > j \\ \frac{1}{\psi_{k,j,s^{(t)}}} & \text{otherwise} \end{cases} \ .$$

The next two lemmas discuss properties of TRIÈST-FD for $t < t^*$, where t^* is the first time that $|E^{(t)}|$ had size $M + 1$ ($t^* \geq M + 1$).

Lemma 7.23. *For all $t < t^*$, we have:*

1. $d_{\mathrm{o}}^{(t)} = 0$; *and*
2. $\mathcal{S} = E^{(t)}$; *and*
3. $M^{(t)} = s^{(t)}$.

Proof. Since the third point in the thesis follows immediately from the second, we focus on the first two points.

The proof is by induction on t. In the base base for $t = 1$: the element on the stream must be an insertion, and the algorithm deterministically inserts the edge in $\mathcal{S}$. Assume now that it is true for all time steps up to (but excluding) some $t \leq t^* - 1$. We now show that it is also true for t.

Assume the element on the stream at time t is a deletion. The corresponding edge must be in $\mathcal{S}$, from the inductive hypothesis. Hence TRIÈST-FD removes it from $\mathcal{S}$ and increments the counter d_{i} by 1. Thus it is still true that $\mathcal{S} = E^{(t)}$ and $d_{\mathrm{o}}^{(t)} = 0$, and the thesis holds.

Assume now that the element on the stream at time t is an insertion. From the inductive hypothesis we have that the current value of the counter d_{o} is 0.

If the counter d_{i} has currently value 0 as well, then, because of the hypothesis that $t < t^*$, it must be that $|\mathcal{S}| = M^{(t-1)} = s^{(t-1)} < M$. Therefore TRIÈST-FD always inserts the edge in $\mathcal{S}$. Thus it is still true that $\mathcal{S} = E^{(t)}$ and $d_{\mathrm{o}}^{(t)} = 0$, and the thesis holds.

If otherwise $d_{\mathrm{i}} > 0$, then TRIÈST-FD flips a biased coin with probability of heads equal to

$$\frac{d_{\mathrm{i}}}{d_{\mathrm{i}} + d_{\mathrm{o}}} = \frac{d_{\mathrm{i}}}{d_{\mathrm{i}}} = 1,$$

therefore TRIÈST-FD always inserts the edge in $\mathcal{S}$ and decrements d_{i} by one. Thus it is still true that $\mathcal{S} = E^{(t)}$ and $d_{\mathrm{o}}^{(t)} = 0$, and the thesis holds. □

The following result is an immediate consequence of Lemma 7.21 and Lemma 7.23.

Lemma 7.24. *For all $t < t^*$ such that $s^{(t)} \geq 3$, we have $\kappa^{(t)} = 1$.*

We can now prove Thm. 7.19.

Proof of Thm. 7.19. Assume for now that $t < t^*$. From Lemma 7.23, we have that $s^{(t)} = M^{(t)}$. If $M^{(t)} < 3$, then it must be $s^{(t)} < 3$, hence $|\Delta^{(t)}| = 0$ and indeed the algorithm returns $\rho^{(t)} = 0$ in this case. If instead $M^{(t)} = s^{(t)} \geq 3$, then we have

$$\rho^{(t)} = \frac{\tau^{(t)}}{\kappa^{(t)}} \ .$$

From Lemma 7.24 we have that $\kappa^{(t)} = 1$ for all $t < t^*$, hence $\rho^{(t)} = \tau^{(t)}$ in these cases. Since (an identical version of) Lemma 7.4 also holds for TRIÈST-FD, we have $\tau^{(t)} = |\Delta^{\mathcal{S}}| = |\Delta^{(t)}|$, where the last equality comes from the fact that $\mathcal{S} = E^{(t)}$ (Lemma 7.23). Hence $\rho^{(t)} = |\Delta^{(t)}|$ for any $t \leq t^*$, as in the thesis.

Assume now that $t \geq t^*$. Using the law of total expectation, we can write

$$\mathbb{E}\left[\rho^{(t)}\right] = \sum_{j=0}^{\min\{s^{(t)},M\}} \mathbb{E}\left[\rho^{(t)} \mid M^{(t)} = j\right] \Pr\left(M^{(t)} = j\right) \ . \tag{7.23}$$

Assume that $|\Delta^{(t)}| > 0$, otherwise, the algorithm deterministically returns 0 as an estimation and the thesis follows. Let λ be a triangle in $\Delta^{(t)}$, and let $\delta_\lambda^{(t)}$ be a random variable that takes value

$$\frac{\psi_{3,M^{(t)},s^{(t)}}}{\kappa^{(t)}} = \frac{s^{(t)}(s^{(t)}-2)(s^{(t)}-2)}{M^{(t)}(M^{(t)}-1)(M^{(t)}-2)} \frac{1}{\kappa^{(t)}}$$

if all edges of λ are in $\mathcal{S}$ at the end of the time instant t, and 0 otherwise. Since (an identical version of) Lemma 7.4 also holds for TRIÈST-FD, we can write

$$\rho^{(t)} = \sum_{\lambda \in \Delta^{(t)}} \delta_\lambda^{(t)} \ .$$

Then, using Lemma 7.21 and Lemma 7.22, we have, for $3 \leq j \leq \min\{M, s^{(t)}\}$,

$$\begin{aligned}\mathbb{E}\left[\rho^{(t)} \mid M^{(t)} = j\right] &= \sum_{\lambda \in \Delta^{(t)}} \frac{\psi_{3,j,s^{(t)}}}{\kappa^{(t)}} \Pr\left(\delta_\lambda^{(t)} = \frac{\psi_{3,j,s^{(t)}}}{\kappa^{(t)}} \mid M^{(t)} = j\right) \\ &= |\Delta^{(t)}| \frac{\psi_{3,j,s^{(t)}}}{\kappa^{(t)}} \frac{1}{\psi_{3,j,s^{(t)}}} = \frac{1}{\kappa^{(t)}} |\Delta^{(t)}|, \end{aligned} \tag{7.24}$$

and

$$\mathbb{E}\left[\rho^{(t)} \mid M^{(t)} = j\right] = 0, \text{if } 0 \leq j \leq 2. \tag{7.25}$$

Plugging this into (7.23), and using Lemma 7.21, we have

$$\mathbb{E}\left[\rho^{(t)}\right] = |\Delta^{(t)}| \frac{1}{\kappa^{(t)}} \sum_{j=3}^{\min\{s^{(t)},M\}} \Pr(M^{(t)} = j) = |\Delta^{(t)}| \ .$$

□

Variance

Theorem 7.25. *Let $t > t^*$ s.t. $|\Delta^{(t)}| > 0$ and $s^{(t)} \geq M$. Suppose we have $d^{(t)} = d_o^{(t)} + d_i^{(t)} \leq \alpha s^{(t)}$ total unpaired deletions at time t, with $0 \leq \alpha < 1$. If $M \geq \frac{1}{2\sqrt{\alpha'-\alpha}} 7 \ln s^{(t)}$ for some $\alpha < \alpha' < 1$, we have:*

$$\begin{aligned}\mathrm{Var}\left[\rho^{(t)}\right] \leq\ & (\kappa^{(t)})^{-2} |\Delta^{(t)}| \left(\psi_{3,M(1-\alpha'),s^{(t)}} - 1\right) + (\kappa^{(t)})^{-2} 2 \\ & + (\kappa^{(t)})^{-2} r^{(t)} \left(\psi^2_{3,M(1-\alpha'),s^{(t)}} \psi^{-1}_{5,M(1-\alpha'),s^{(t)}} - 1\right)\end{aligned}$$

The proof of Thm. 7.25 uses two results on the variance of $\rho^{(t)}$ conditioned on a specific value of $M^{(t)}$ (Lemmas 7.26 and 7.27), and an analysis of the probability distribution of $M^{(t)}$ (Lemma 7.28 and Corollary 7.29). These results are then combined using the law of total variance.

Lemma 7.26. *For any time $t \geq t^*$, and any j, $3 \leq j \leq \min\{s^{(t)}, M\}$, we have:*

$$\begin{aligned}\mathrm{Var}\left[\rho^{(t)} | M^{(t)} = j\right] = (\kappa^{(t)})^{-2} \Bigg(& |\Delta^{(t)}| \left(\psi_{3,j,s^{(t)}} - 1\right) + r^{(t)} \left(\psi^2_{3,j,s^{(t)}} \psi^{-1}_{5,j,s^{(t)}} - 1\right) \\ & + w^{(t)} \left(\psi^2_{3,j,s^{(t)}} \psi^{-1}_{6,j,s^{(t)}} - 1\right)\Bigg)\end{aligned} \tag{7.26}$$

An analogous result holds for any $u \in V^{(t)}$, replacing the global quantities with the corresponding local ones.

Proof. The proof is analogous to that of Theorem 7.5, using j in place of M, $s^{(t)}$ in place of t, $\psi_{a,j,s^{(t)}}$ in place of $\xi_{a,t}$, and using Lemma 7.22 instead of Lemma 7.1. The additional $(k^{(t)})^{-2}$ multiplicative term comes from the $(k^{(t)})^{-1}$ term used in the definition of $\rho^{(t)}$. □

The term $w^{(t)}\left(\psi^2_{3,j,s^{(t)}}\psi^{-1}_{6,j,s^{(t)}}-1\right)$ is non-positive.

Lemma 7.27. *For any time $t \geq t^*$, and any j, $6 < j \leq \min\{s^{(t)}, M\}$, if $s^{(t)} \geq M$ we have:*

$$\mathrm{Var}\left[\rho^{(t)}|M^{(t)}=i\right] \leq (\kappa^{(t)})^{-2}\left(|\Delta^{(t)}|\left(\psi_{3,j,s^{(t)}}-1\right)+r^{(t)}\left(\psi^2_{3,j,s^{(t)}}\psi^{-1}_{5,j,s^{(t)}}-1\right)\right),\ \textit{for } i \geq j$$

$$\mathrm{Var}\left[\rho^{(t)}|M^{(t)}=i\right] \leq (\kappa^{(t)})^{-2}\left(|\Delta^{(t)}|\left(\psi_{3,3,s^{(t)}}-1\right)+r^{(t)}\left(\psi^2_{3,5,s^{(t)}}\psi^{-1}_{5,5,s^{(t)}}-1\right)\right),\ \textit{for } i < j$$

An analogous result holds for any $u \in V^{(t)}$, replacing the global quantities with the corresponding local ones.

Proof. The proof follows by observing that the term $w^{(t)}\left(\psi^2_{3,j,s^{(t)}}\psi^{-1}_{6,j,s^{(t)}}-1\right)$ is non-positive, and that (7.26) is a non-increasing function of the sample size. □

The following lemma deals with properties of the r.v. $M^{(t)}$.

Lemma 7.28. *Let $t > t^*$, with $s^{(t)} \geq M$. Let $d^{(t)} = d_o^{(t)} + d_i^{(t)}$ denote the total number of unpaired deletions at time t.*[7] *The sample size $M^{(t)}$ follows the hypergeometric distribution:*[8]

$$\Pr\left(M^{(t)}=j\right) = \begin{cases} \binom{s^{(t)}}{j}\binom{d^{(t)}}{M-j}\Big/\binom{s^{(t)}+d^{(t)}}{M} & \textit{for } \max\{M-d^{(t)},0\} \leq j \leq M \\ 0 & \textit{otherwise} \end{cases}. \tag{7.27}$$

We have

$$\mathbb{E}\left[M^{(t)}\right] = M\frac{s^{(t)}}{s^{(t)}+d^{(t)}}, \tag{7.28}$$

and for any $0 < c < 1$

$$\Pr\left(M^{(t)} > \mathbb{E}\left[M^{(t)}\right] - cM\right) \geq 1 - \frac{1}{e^{2c^2M}}\ . \tag{7.29}$$

Proof. Since $t > t*$, from the definition of t^* we have that the $M^{(t)}$ has reached size M at least once (at t^*). From this and the definition of $d^{(t)}$ (number of uncompensated deletion), we have that $M^{(t)}$ can not be less than $M - d^{(t)}$. The rest of the proof for (7.27) and for (7.28) follows from [86, Thm. 2].

The concentration bound in (7.29) follows from the properties of the hypergeometric distribution discussed by [214]. □

The following is an immediate corollary from Lemma 7.28.

Corollary 7.29. *Consider the execution of* TRIÈST-FD *at time $t > t^*$. Suppose we have $d^{(t)} \leq \alpha s^{(t)}$, with $0 \leq \alpha < 1$ and $s^{(t)} \geq M$. If $M \geq \frac{1}{2\sqrt{\alpha'-\alpha}}c' \ln s^{(t)}$ for $\alpha < \alpha' < 1$, we have:*

$$\Pr\left(M^{(t)} \geq M(1-\alpha')\right) > 1 - \frac{1}{\left(s^{(t)}\right)^{c'}}.$$

[7] While both $d_o^{(t)}$ and $d_i^{(t)}$ are r.v.s, their sum is not.

[8] We use here the convention that $\binom{0}{0} = 1$, and $\binom{k}{0} = 1$.

We can now prove Thm. 7.25.

Proof of Thm. 7.25. From the law of total variance we have:

$$\begin{aligned}\mathrm{Var}\left[\rho^{(t)}\right] &= \sum_{j=0}^{M}\mathrm{Var}\left[\rho^{(t)}|M^{(t)}=j\right]\Pr\left(M^{(t)}=j\right)\\ &+\sum_{j=0}^{M}\mathbb{E}\left[\rho^{(t)}|M^{(t)}=j\right]^2\left(1-\Pr\left(M^{(t)}=j\right)\right)\Pr\left(M^{(t)}=j\right)\\ &-2\sum_{j=1}^{M}\sum_{i=0}^{j-1}\mathbb{E}\left[\rho^{(t)}|M^{(t)}=j\right]\Pr\left(M^{(t)}=j\right)\mathbb{E}\left[\rho^{(t)}|M^{(t)}=i\right]\Pr\left(M^{(t)}=i\right).\end{aligned}$$

As shown in (7.24) and (7.25), for any $j=0,1,\ldots,M$ we have $\mathbb{E}\left[\rho^{(t)}|M^{(t)}=j\right]\geq 0$. This in turn implies:

$$\begin{aligned}\mathrm{Var}\left[\rho^{(t)}\right] &\leq \sum_{j=0}^{M}\mathrm{Var}\left[\rho^{(t)}|M^{(t)}=j\right]\Pr\left(M^{(t)}=j\right)\\ &+\sum_{j=0}^{M}\mathbb{E}\left[\rho^{(t)}|M^{(t)}=j\right]^2\left(1-\Pr\left(M^{(t)}=j\right)\right)\Pr\left(M^{(t)}=j\right).\end{aligned}\tag{7.30}$$

Let us consider separately the two main components of (7.30). From Lemma 7.27 we have:

$$\sum_{j=0}^{M}\mathrm{Var}\left[\rho^{(t)}|M^{(t)}=j\right]\Pr\left(M^{(t)}=j\right)=\tag{7.31}$$

$$\begin{aligned}&\sum_{j\geq M(1-\alpha')}^{M}\mathrm{Var}\left[\rho^{(t)}|M^{(t)}=j\right]\Pr\left(M^{(t)}=j)\right)+\sum_{j=0}^{M(1-\alpha')}\mathrm{Var}\left[\rho^{(t)}|M^{(t)}=j\right]\Pr\left(M^{(t)}=j\right)\\ &\leq(\kappa^{(t)})^{-2}\left(|\Delta^{(t)}|\left(\psi_{3,j,s^{(t)}}-1\right)+r^{(t)}\left(\psi^2_{3,j,s^{(t)}}\psi^{-1}_{5,j,s^{(t)}}-1\right)\right)\times\Pr\left(M^{(t)}>M(1-\alpha')\right)\\ &\leq(\kappa^{(t)})^{-2}\left(|\Delta^{(t)}|\frac{\left(s^{(t)}\right)^3}{6}+r^{(t)}\frac{s^{(t)}}{6}\right)\Pr\left(M^{(t)}\leq M(1-\alpha')\right)\end{aligned}\tag{7.32}$$

According to our hypothesis $M\geq\frac{1}{2\sqrt{\alpha'-\alpha}}7\ln s^{(t)}$, thus we have, from Corollary 7.29:

$$\Pr\left(M^{(t)}\leq M(1-\alpha'))\right)\leq\frac{1}{(s^{(t)})^7}.$$

As $r^{(t)}<|\Delta^{(t)}|^2$ and $|\Delta^{(t)}|\leq(s^{(t)})^3$ we have:

$$(\kappa^{(t)})^{-2}\left(|\Delta^{(t)}|\frac{\left(s^{(t)}\right)^3}{6}+r^{(t)}\frac{s^{(t)}}{6}\right)\Pr\left(M^{(t)}\leq M(1-\alpha')\right)\leq(\kappa^{(t)})^{-2}$$

We can therefore rewrite (7.32) as:

$$\begin{aligned}\sum_{j=0}^{M}\mathrm{Var}\left[\rho^{(t)}|M^{(t)}=j\right]\Pr\left(M^{(t)}=j\right)&\leq(\kappa^{(t)})^{-2}\left(|\Delta^{(t)}|\left(\psi_{3,M(1-\alpha'),s^{(t)}}-1\right)\right)\\ &+(\kappa^{(t)})^{-2}\left(r^{(t)}\left(\psi^2_{3,M(1-\alpha'),s^{(t)}}\psi^{-1}_{5,M(1-\alpha'),s^{(t)}}-1\right)+1\right).\end{aligned}\tag{7.33}$$

Let us now consider the term $\sum_{j=0}^{M} \mathbb{E}\left[\rho^{(t)}|M^{(t)}=j\right]^2 (1-\Pr\left(M^{(t)}=j\right))\Pr\left(M^{(t)}=j\right)$. Recall that, from (7.24) and (7.25), we have $\mathbb{E}\left[\rho^{(t)}|M^{(t)}=j\right] = |\Delta^{(t)}|(\kappa^{(t)})^{-1}$ for $j=3,\ldots,M$, and $\mathbb{E}\left[\rho^{(t)}|M^{(t)}=j\right]=0$ for $j=0,1,2$. From Corollary 7.29 we have that for $j \leq (1-\alpha')M$ and $M \geq \frac{1}{2\sqrt{\alpha'-\alpha}} 7 \ln s^{(t)}$

$$\Pr\left(M^{(t)}=j\right) \leq \Pr\left(M^{(t)} \leq (1-\alpha')M\right) \leq \frac{1}{\left(s^{(t)}\right)^7},$$

and thus:

$$\sum_{j=0}^{(1-\alpha')M} \mathbb{E}\left[\rho^{(t)}|M^{(t)}=j\right]^2 (1-\Pr\left(M^{(t)}=j\right))\Pr\left(M^{(t)}=j\right) \leq \frac{(1-\alpha')M|\Delta^{(t)}|^2(\kappa^{(t)})^{-2}}{\left(s^{(t)}\right)^7}$$
$$\leq (1-\alpha')(\kappa^{(t)})^{-2}, \tag{7.34}$$

where the last passage follows since, by hypothesis, $M \leq s^{(t)}$.

Let us now consider the values $j > (1-\alpha')M$, we have:

$$\begin{aligned}
&\sum_{j>(1-\alpha')M}^{M} \mathbb{E}\left[\rho^{(t)}|M^{(t)}=j\right]^2 (1-\Pr\left(M^{(t)}=j\right))\Pr\left(M^{(t)}=j\right) \\
&\leq \alpha' M|\Delta^{(t)}|^2(\kappa^{(t)})^{-2}\left(1-\sum_{j>(1-\alpha')M}^{M}\Pr\left(M^{(t)}=j\right)\right) \\
&\leq \alpha' M|\Delta^{(t)}|^2(\kappa^{(t)})^{-2}\left(1-\Pr\left(M^{(t)}>(1-\alpha')M\right)\right) \\
&\leq \frac{\alpha' M|\Delta^{(t)}|^2(\kappa^{(t)})^{-2}}{\left(s^{(t)}\right)^7} \\
&\leq \alpha'(\kappa^{(t)})^{-2},
\end{aligned} \tag{7.35}$$

where the last passage follows since, by hypothesis, $M \leq s^{(t)}$.

The theorem follows from composing the upper bounds obtained in (7.33), (7.34) and (7.35) according to (7.30). □

Concentration

The following result relies on Chebyshev's inequality and on Thm. 7.25, and the proof follows the steps similar to those in the proof for Thm. 7.16.

Theorem 7.30. *Let $t \geq t^*$ s.t. $|\Delta^{(t)}| > 0$ and $s^{(t)} \geq M$. Let $d^{(t)} = d_o^{(t)} + d_i^{(t)} \leq \alpha s^{(t)}$ for some $0 \leq \alpha < 1$. For any $\varepsilon, \delta \in (0,1)$, if for some $\alpha < \alpha' < 1$*

$$M > \max\left\{\frac{1}{\sqrt{a'-\alpha}} 7 \ln s^{(t)},\right.$$
$$(1-\alpha')^{-1}\left(\sqrt[3]{\frac{2s^{(t)}(s^{(t)}-1)(s^{(t)}-2)}{\delta\varepsilon^2|\Delta^{(t)}|(\kappa^{(t)})^2 + 2\frac{|\Delta^{(t)}|-2}{|\Delta^{(t)}|}}} + 2\right),$$
$$\left.\frac{(1-\alpha')^{-1}}{3}\left(\frac{r^{(t)}s^{(t)}}{\delta\varepsilon^2|\Delta^{(t)}|^2(\kappa^{(t)})^{-2} + 2r^{(t)}}\right)\right\}$$

then $|\rho^{(t)} - |\Delta^{(t)}|| < \varepsilon|\Delta^{(t)}|$ with probability $> 1-\delta$.

Proof. By Chebyshev's inequality it is sufficient to prove that

$$\frac{\mathrm{Var}[\rho^{(t)}]}{\varepsilon^2|\Delta^{(t)}|^2} < \delta \ .$$

From Lemma 7.25, for $M \geq \frac{1}{\sqrt{a'-\alpha}} 7 \ln s^{(t)}$ we have:

$$\begin{aligned}\mathrm{Var}\left[\rho^{(t)}\right] &\leq (\kappa^{(t)})^{-2}|\Delta^{(t)}|\left(\psi_{3,M(1-\alpha'),s^{(t)}} - 1\right) \\ &+ (\kappa^{(t)})^{-2} r^{(t)} \left(\psi^2_{3,M(1-\alpha'),s^{(t)}} \psi^{-1}_{5,M(1-\alpha'),s^{(t)}} - 1\right) \\ &+ (\kappa^{(t)})^{-2} 2\end{aligned}$$

Let $M' = (1-\alpha')M$. In order to verify that the lemma holds, it is sufficient to impose the following two conditions:

Condition (1)

$$\frac{\delta}{2} > \frac{(\kappa^{(t)})^{-2}\left(|\Delta^{(t)}|\left(\psi_{3,M(1-\alpha'),s^{(t)}} - 1\right) + 2\right)}{\varepsilon^2|\Delta^{(t)}|^2}.$$

As by hypothesis $|\Delta^{(t)}| > 0$, we can rewrite this condition as:

$$\frac{\delta}{2} > \frac{(\kappa^{(t)})^{-2}\left(\psi_{3,M(1-\alpha'),s^{(t)}} - (\frac{|\Delta^{(t)}|-2}{|\Delta^{(t)}|}\right)}{\varepsilon^2|\Delta^{(t)}|}$$

which is verified for:

$$\begin{aligned}M'(M'-1)(M'-2) &> \frac{2s^{(t)}(s^{(t)}-1)(s^{(t)}-2)}{\delta\varepsilon^2|\Delta^{(t)}|(\kappa^{(t)})^2 + 2\frac{|\Delta^{(t)}|-2}{|\Delta^{(t)}|}}, \\ M' &> \sqrt[3]{\frac{2s^{(t)}(s^{(t)}-1)(s^{(t)}-2)}{\delta\varepsilon^2|\Delta^{(t)}|(\kappa^{(t)})^2 + 2\frac{|\Delta^{(t)}|-2}{|\Delta^{(t)}|}}} + 2, \\ M &> (1-\alpha')^{-1}\left(\sqrt[3]{\frac{2s^{(t)}(s^{(t)}-1)(s^{(t)}-2)}{\delta\varepsilon^2|\Delta^{(t)}|(\kappa^{(t)})^2 + 2\frac{|\Delta^{(t)}|-2}{|\Delta^{(t)}|}}} + 2\right).\end{aligned}$$

Condition (2)

$$\frac{\delta}{2} > \frac{(\kappa^{(t)})^{-2}}{\varepsilon^2|\Delta^{(t)}|^2} r^{(t)} \left(\psi^2_{3,M(1-\alpha'),s^{(t)}} \psi^{-1}_{5,M(1-\alpha'),s^{(t)}} - 1\right). \tag{7.36}$$

As we have:

$$(\kappa^{(t)})^{-2} r^{(t)} \left(\psi^2_{3,M(1-\alpha'),s^{(t)}} \psi^{-1}_{5,M(1-\alpha'),s^{(t)}} - 1\right) \leq (\kappa^{(t)})^{-2} r^{(t)} \left(\frac{s^{(t)}}{6M(1-\alpha')} - 1\right)$$

The condition (7.36) is verified for:

$$M > \frac{(1-\alpha')^{-1}}{3}\left(\frac{r^{(t)}s^{(t)}}{\delta\varepsilon^2|\Delta^{(t)}|^2(\kappa^{(t)})^{-2} + 2r^{(t)}}\right).$$

The theorem follows. □

7.4.4 Counting global and local triangles in multigraphs

We now discuss how to extend TRIÈST to approximate the local and global triangle counts in multigraphs.

TRIÈST-BASE on multigraphs

TRIÈST-BASE can be adapted to work on multigraphs as follows. First of all, the sample $\mathcal{S}$ should be considered a *bag*, i.e., it may contain multiple copies of the same edge. Secondly, the function UPDATECOUNTERS must be changed as presented in Alg. 14, to take into account the fact that inserting or removing an edge (u, v) from $\mathcal{S}$ respectively increases or decreases the global number of triangles in $G^{\mathcal{S}}$ by a quantity that depends on the product of the number of edges $(c, u) \in \mathcal{S}$ and $(c, v) \in \mathcal{S}$, for c in the shared neighborhood (in $G^{\mathcal{S}}$) of u and v (and similarly for the local number of triangles incidents to c).

ALGORITHM 14 UPDATECOUNTERS function for TRIÈST-BASE on multigraphs

```
1:  function UpdateCounters((•, (u, v)))
2:      N^S_{u,v} ← N^S_u ∩ N^S_v
3:      for all c ∈ N^S_{u,v} do
4:          y_{c,u} ← number of edges between c and u in S
5:          y_{c,v} ← number of edges between c and v in S
6:          y_c ← y_{c,u} · y_{c,v}
7:          τ ← τ • y_c
8:          τ_c ← τ_c • y_c
9:          τ_u ← τ_u • y_c
10:         τ_v ← τ_v • y_c
```

For this modified version of TRIÈST-BASE, that we call TRIÈST-BASE-M, an equivalent version of Lemma 7.4 holds. Therefore, we can prove a result on the unbiasedness of TRIÈST-BASE-M equivalent (i.e., with the same statement) as Thm. 7.3. The proof of such result is also the same as the one for Thm. 7.3.

To analyze the variance of TRIÈST-BASE-M, we need to take into consideration the fact that, in a multigraph, a pair of triangles may share *two* edges, and the variance depends (also) on the number of such pairs. Let $r_1^{(t)}$ be the number of unordered pairs of distinct triangles from $\Delta^{(t)}$ sharing an edge and let $r_2^{(t)}$ be the number of unordered pairs of distinct triangles from $\Delta^{(t)}$ sharing *two* edges (such pairs may exist in a multigraph, but not in a simple graph). Let $q^{(t)} = \binom{|\Delta^{(t)}|}{2} - r_1^{(t)} - r_2^{(t)}$ be the number of unordered pairs of distinct triangles that do not share any edge.

Theorem 7.31. *For any $t > M$, let $f(t) = \xi^{(t)} - 1$,*

$$g(t) = \xi^{(t)} \frac{(M-3)(M-4)}{(t-3)(t-4)} - 1$$

and

$$h(t) = \xi^{(t)} \frac{(M-3)(M-4)(M-5)}{(t-3)(t-4)(t-5)} - 1 \ \ (\leq 0),$$

and

$$j(t) = \xi^{(t)} \frac{M-3}{t-3} - 1 \ .$$

We have:

$$\mathrm{Var}\left[\xi(t)\tau^{(t)}\right] = |\Delta^{(t)}| f(t) + r_1^{(t)} g(t) + r_2^{(t)} j(t) + q^{(t)} h(t).$$

The proof follows the same lines as the one for Thm. 7.5, with the additional steps needed to take into account the contribution of the $r_2^{(t)}$ pairs of triangles in $G^{(t)}$ sharing two edges.

TRIÈST-IMPR on multigraphs

A variant TRIÈST-IMPR-M of TRIÈST-IMPR for multigraphs can be obtained by using the function UPDATECOUNTERS defined in Alg. 14, modified to increment[9] the counters by $\eta^{(t)} y_c^{(t)}$, rather than $y_c^{(t)}$, where $\eta^{(t)} = \max\{1, (t-1)(t-2)/(M(M-1))\}$. The result stated in Thm. 7.15 holds also for the estimations computed by TRIÈST-IMPR-M. An upper bound to the variance of the estimations, similar to the one presented in Thm. 7.16 for TRIÈST-IMPR, could potentially be obtained, but its derivation would involve a high number of special cases, as we have to take into consideration the order of the edges in the stream.

TRIÈST-FD on multigraphs

TRIÈST-FD can be modified in order to provide an approximation of the number of global and local triangles on multigraphs observed as a stream of edge deletions and deletions. It is however necessary to clearly state the data model. We assume that for all pairs of vertices $u, v \in V^{(t)}$, each edge connecting u and v is assigned a label that is unique among the edges connecting u and v. An edge is therefore uniquely identified by its endpoints and its label as $((u, v), label)$. Elements of the stream are now in the form $(\bullet, (u, v), label)$, where $\bullet$ is either $+$ or $-$. This assumption, somewhat strong, is necessary in order to apply the *random pairing* sampling scheme [86] to fully-dynamic multigraph edge streams.

Within this model, we can obtain an algorithm TRIÈST-FD-M for multigraphs by adapting TRIÈST-FD as follows. The sample $\mathcal{S}$ is a *set* of elements $((u, v), label)$. When a deletion $(-, (u, v), label)$ is on the stream, the sample $\mathcal{S}$ is modified if and only if $((u, v), label)$ belongs to $\mathcal{S}$. This change can be implemented in the pseudocode from Alg. 13 by modifying line 8 to be

"**else if** $((u, v), label) \in \mathcal{S}$ **then**" .

Additionally, the function UPDATECOUNTERS to be used is the one presented in Alg. 14.

We can prove a result on the unbiasedness of TRIÈST-FD-M equivalent (i.e., with the same statement) as Thm. 7.19. The proof of such result is also the same as the one for Thm. 7.19. An upper bound to the variance of the estimations, similar to the one presented in Thm. 7.25 for TRIÈST-FD, could be obtained by considering the fact that in a multigraph two triangles can share two edges, in a fashion similar to what we discussed in Thm. 7.31.

7.4.5 Discussion

We now briefly discuss over the algorithms we just presented, the techniques they use, and the theoretical results we obtained for TRIÈST, in order to highlight advantages, disadvantages, and limitations of our approach.

On reservoir sampling Our approach of using reservoir sampling to keep a random sample of edges can be extended to many other graph mining problems, including approximate counting of

[9] As in TRIÈST-IMPR, all calls to UPDATECOUNTERS in TRIÈST-IMPR-M have $\bullet = +$. See also Alg. 12.

other subgraphs more or less complex than triangles (e.g., squares, trees with a specific structure, wedges, cliques, and so on). The estimations of such counts would still be unbiased, but as the number of edges composing the subgraph(s) of interest increases, the variance of the estimators also increases, because the probability that all edges composing a subgraph are in the sample (or all but the last one when the last one arrives, as in the case of TRIÈST-IMPR), decreases as their number increases. Other works in the triangle counting literature [180, 116] use samples of wedges, rather than edges. They perform worse than TRIÈST in both accuracy and runtime (see Sect. 7.5), but the idea of sampling and storing more complex structures rather than simple edges could be a potential direction for approximate counting of larger subgraphs.

On the analysis of the variance We showed an exact analysis of the variance of TRIÈST-BASE but for the other algorithms we presented *upper bounds* to the variance of the estimates. These bounds can still be improved as they are not currently tight. For example, we already commented on the fact that the bound in (7.17) does not include a number of negative terms that would tighten it (i.e., decrease the bound), and that could potentially be no smaller than the term depending on $z^{(t)}$. The absence of such terms is due to the fact that it seems very challenging to obtain non-trivial *upper bounds* to them that are valid for every $t > M$. Our proof for this bound uses a careful case-by-case analysis, considering the different situations for pair of triangles (e.g., sharing or not sharing an edge, and considering the order of edges on the stream). It may be possible to obtain tighter bounds to the variance by following a more holistic approach that takes into account the fact that the sizes of the different classes of triangle pairs are highly dependent on each other.

Another issue with the bound to the variance from (7.17) is that the quantity $z^{(t)}$ depends on the order of edges on the stream. As already discussed, the bound can be made independent of the order by loosening it even more. Very recent developments in the sampling theory literature [54] presented sampling schemes and estimators whose second-order sampling probabilities do not depend on the order of the stream, so it should be possible to obtain such bounds also for the triangle counting problem, but a sampling scheme different than reservoir sampling would have to be used, and a careful analysis is needed to establish its net advantages in terms of performances and scalability to billion-edges graphs.

On the trade-off between speed and accuracy We concluded both previous paragraphs in this subsection by mentioning techniques different than reservoir sampling of edges as potential directions to improve and extend our results. In both cases these techniques are more complex not only in their analysis but also *computationally*. Given that the main goal of algorithms like TRIÈST is to make it possible to analyze graphs with billions (and possibly more) nodes, the gain in accuracy need to be weighted against expected slowdowns in execution. As we show in our experimental evaluation in the next section, TRIÈST, especially in the TRIÈST-IMPR variant, actually seems to strike the right balance between accuracy and tradeoff, when compared with existing contributions.

7.5 Experimental evaluation

We evaluated TRIÈST on several real-world graphs with up to a billion edges. The algorithms were implemented in `C++`, and ran on the Brown University CS department cluster.[10] Each run employed a single core and used at most 4 GB of RAM. The code is available from `http://bigdata.cs.brown.edu/triangles.html`. Most of this section is related to experiments on graphs, while results for multigraphs are described in Sect 7.5.3.

Datasets We created the streams from the following publicly available graphs (properties in Table 7.2).

Patent (Co-Aut.) and Patent (Cit.) The *Patent (Co-Aut.)* and *Patent (Cit.)* graphs are obtained from a dataset of ≈ 2 million U.S. patents granted between '75 and '99 [96]. In *Patent (Co-Aut.)*, the nodes represent inventors and there is an edge with timestamp t between two co-inventors of a patent if the patent was granted in year t. In *Patent (Cit.)*, nodes are patents and there is an edge (a, b) with timestamp t if patent a cites b and a was granted in year t.

LastFm The LastFm graph is based on a dataset [38, 225] of ≈ 20 million `last.fm` song listenings, ≈ 1 million songs and ≈ 1000 users. There is a node for each song and an edge between two songs if ≥ 3 users listened to both on day t.

Yahoo!-Answers The Yahoo! Answers graph is obtained from a sample of ≈ 160 million answers to ≈ 25 millions questions posted on Yahoo! Answers [50]. An edge connects two users at time $max(t_1, t_2)$ if they both answered the same question at times t_1, t_2 respectively. We removed 6 outliers questions with more than 5000 answers.

Twitter This is a snapshot [139, 25] of the Twitter followers/following network with ≈ 41 million nodes and ≈ 1.5 billions edges. We do not have time information for the edges, hence we assign a random timestamp to the edges (of which we ignore the direction).

Ground truth To evaluate the accuracy of our algorithms, we computed the *ground truth* for our smaller graphs (i.e., the exact number of global and local triangles for each time step), using an exact algorithm. The entire current graph is stored in memory and when an edge u, v is inserted (or deleted) we update the current count of local and global triangles by checking how many triangles are completed (or broken). As exact algorithms are not scalable, computing the exact triangle count is feasible only for small graphs such as Patent (Co-Aut.), Patent (Cit.) and LastFm. Table 7.2 reports the exact total number of triangles at the end of the stream for those graphs (and an estimate for the larger ones using TRIÈST-IMPR with $M = 1000000$).

[10] `https://cs.brown.edu/about/system/services/hpc/grid/`

Graph	$\|V\|$	$\|E\|$	$\|E_u\|$	$\|\Delta\|$
Patent (Co-Aut.)	1,162,227	3,660,945	2,724,036	3.53×10^6
Patent (Cit.)	2,745,762	13,965,410	13,965,132	6.91×10^6
LastFm	681,387	43,518,693	30,311,117	1.13×10^9
Yahoo! Answers	2,432,573	1.21×10^9	1.08×10^9	7.86×10^{10}
Twitter	41,652,230	1.47×10^9	1.20×10^9	3.46×10^{10}

Table 7.2: Properties of the dynamic graph streams analyzed. $|V|$, $|E|$, $|E_u|$, $|\Delta|$ refer respectively to the number of nodes in the graph, the number of edge addition events, the number of distinct edges additions, and the maximum number of triangles in the graph (for Yahoo! Answers and Twitter estimated with TRIÈST-IMPR with $M = 1000000$, otherwise computed exactly with the naïve algorithm).

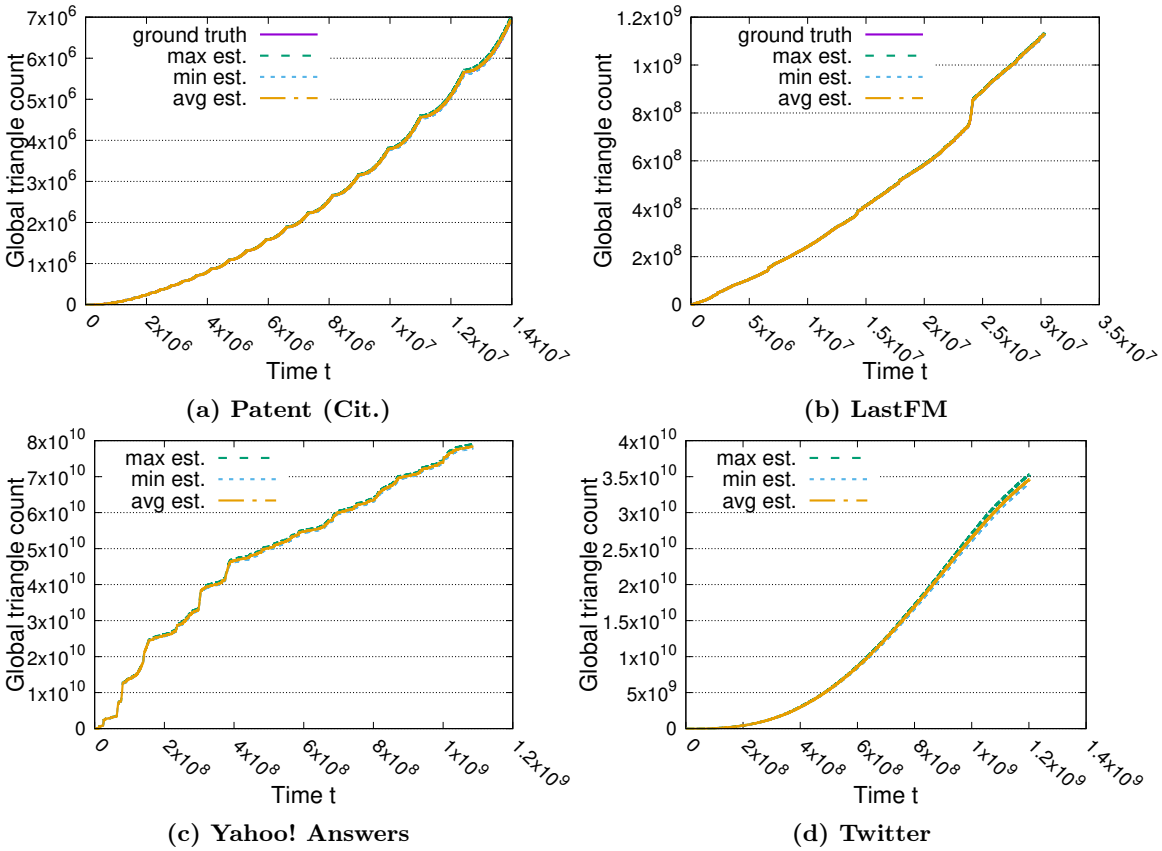
7.5.1 Insertion-only case

We now evaluate TRIÈST on insertion-only streams and compare its performances with those of state-of-the-art approaches [144, 116, 180], showing that TRIÈST has an average estimation error significantly smaller than these methods both for the global and local estimation problems, while using the same amount of memory.

Estimation of the global number of triangles Starting from an empty graph we add one edge at a time, in timestamp order. Figure 7.1 illustrates the evolution, over time, of the estimation computed by TRIÈST-IMPR with $M = 1{,}000{,}000$. For smaller graphs for which the ground truth can be computed exactly, the curve of the exact count is practically indistinguishable from TRIÈST-IMPR estimation, showing the precision of the method. The estimations have very small variance even on the very large Yahoo! Answers and Twitter graphs (point-wise max/min estimation over ten runs is almost coincident with the average estimation). These results show that TRIÈST-IMPR is very accurate even when storing less than a 0.001 fraction of the total edges of the graph.

Comparison with the state of the art We compare quantitatively with three state-of-the-art methods: MASCOT [144], JHA ET AL. [116] and PAVAN ET AL. [180]. MASCOT is a suite of local triangle counting methods (but provides also a global estimation). The other two are global triangle counting approaches. None of these can handle fully-dynamic streams, in contrast with TRIÈST-FD. We first compare the three methods to TRIÈST for the global triangle counting estimation. MASCOT comes in two memory efficient variants: the basic MASCOT-C variant and an improved MASCOT-I variant.[11] Both variants sample edges with fixed probability p, so there is no guarantee on the amount of memory used during the execution. To ensure fairness of comparison, we devised the following experiment. First, we run both MASCOT-C and MASCOT-I for $\ell = 10$ times with a fixed

[11] In the original work [144], this variant had no suffix and was simply called MASCOT. We add the -I suffix to avoid confusion. Another variant MASCOT-A can be forced to store the entire graph with probability 1 by appropriately selecting the edge order (which we assume to be adversarial) so we do not consider it here.



(a) Patent (Cit.) **(b) LastFM**

(c) Yahoo! Answers **(d) Twitter**

Figure 7.1: Estimation by TRIÈST-IMPR of the global number of triangles over time (intended as number of elements seen on the stream). The max, min, and avg are taken over 10 runs. The curves are *indistinguishable on purpose*, to highlight the fact that TRIÈST-IMPR estimations have very small error and variance. For example, the ground truth (for graphs for which it is available) is indistinguishable even from the max/min point-wise estimations over ten runs. For graphs for which the ground truth is not available, the small deviations from the avg suggest that the estimations are also close to the true value, given that our algorithms gives unbiased estimations.

p using the same random bits for the two algorithms run-by-run (i.e. the same coin tosses used to select the edges) measuring each time the number of edges M_i' stored in the sample at the end of the stream (by construction this the is same for the two variants run-by-run). Then, we run our algorithms using $M = M_i'$ (for $i \in [\ell]$). We do the same to fix the size of the edge memory for JHA ET AL. [116] and PAVAN ET AL. [180].[12] This way, all algorithms use the same amount of memory for storing edges (run-by-run).

[12]More precisely, we use $M_i'/2$ estimators in PAVAN ET AL. as each estimator stores two edges. For JHA ET AL. we set the two reservoirs in the algorithm to have each size $M_i'/2$. This way, all algorithms use M_i' cells for storing (w)edges.

We use the *MAPE* (Mean Average Percentage Error) to assess the accuracy of the global triangle estimators over time. The MAPE measures the average percentage of the prediction error with respect to the ground truth, and is widely used in the prediction literature [109]. For $t = 1, \ldots, T$, let $\overline{\Delta}^{(t)}$ be the estimator of the number of triangles at time t, the MAPE is defined as $\frac{1}{T}\sum_{t=1}^{T}\left|\frac{|\Delta^{(t)}|-\overline{\Delta}^{(t)}}{|\Delta^{(t)}|}\right|$.[13]

In Fig. 7.2a, we compare the average MAPE of TRIÈST-BASE and TRIÈST-IMPR as well as the two MASCOT variants and the other two streaming algorithms for the Patent (Co-Aut.) graph, fixing $p = 0.01$. TRIÈST-IMPR has the smallest error of all the algorithms compared.

We now turn our attention to the efficiency of the methods. Whenever we refer to one operation, we mean handling one element on the stream, either one edge addition or one edge deletion. The average update time per operation is obtained by dividing the total time required to process the entire stream by the number of operations (i.e., elements on the streams).

Figure 7.2b shows the average update time per operation in Patent (Co-Aut.) graph, fixing $p = 0.01$. Both JHA ET AL. [116] and PAVAN ET AL. [180] are up to ≈ 3 orders of magnitude slower than the MASCOT variants and TRIÈST. This is expected as both algorithms have an update complexity of $\Omega(M)$ (they have to go through the entire reservoir graph at each step), while both MASCOT algorithms and TRIÈST need only to access the neighborhood of the nodes involved in the edge addition.[14] This allows both algorithms to efficiently exploit larger memory sizes. We can use efficiently M up to 1 million edges in our experiments, which only requires few megabytes of RAM.[15] MASCOT is one order of magnitude faster than TRIÈST (which runs in ≈ 28 micros/op), because it does not have to handle edge removal from the sample, as it offers no guarantees on the used memory. As we will show, TRIÈST has much higher precision and scales well on billion-edges graphs.

Given the slow execution of the other algorithms on the larger datasets we compare in details TRIÈST only with MASCOT.[16] Table 7.3 shows the average MAPE of the two approaches. The results confirm the pattern observed in Figure 7.2a: TRIÈST-BASE and TRIÈST-IMPR both have an average error significantly smaller than that of the basic MASCOT-C and improved MASCOT variant respectively. We achieve up to a 91% (i.e., 9-fold) reduction in the MAPE while using the same amount of memory. This experiment confirms the theory: reservoir sampling has overall lower or equal variance in all steps for the same expected total number of sampled edges.

To further validate this observation we run TRIÈST-IMPR and the improved MASCOT-I variant using the same (expected memory) $M = 10000$. Figure 7.3 shows the max-min estimation over 10

[13] The MAPE is not defined for t s.t. $\Delta^{(t)} = 0$ so we compute it only for t s.t. $|\Delta^{(t)}| > 0$. All algorithms we consider are guaranteed to output the correct answer for t s.t. $|\Delta^{(t)}| = 0$.

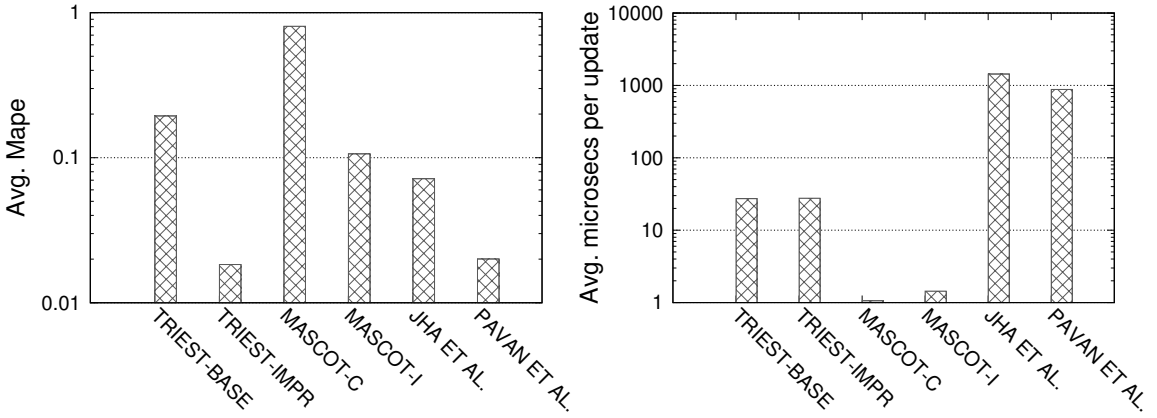
[14] We observe that PAVAN ET AL. [180] would be more efficient with batch updates. However, we want to estimate the triangles continuously at each update. In their experiments they use batch sizes of million of updates for efficiency.

[15] The experiments by [116] use M in the order of 10^3, and in those by [180], large M values require large batches for efficiency.

[16] We attempted to run the other two algorithms but they did not complete after 12 hours for the larger datasets in Table 7.3 with the prescribed p parameter setting.

Graph	Impr.	p	Max. MAPE		Avg. MAPE		
			MASCOT	TRIÈST	MASCOT	TRIÈST	Change
Patent (Cit.)	N	0.01	0.9231	0.2583	0.6517	0.1811	-72.2%
	Y	0.01	0.1907	0.0363	0.1149	0.0213	-81.4%
	N	0.1	0.0839	0.0124	0.0605	0.0070	-88.5%
	Y	0.1	0.0317	0.0037	0.0245	0.0022	-91.1%
Patent (Co-aut.)	N	0.01	2.3017	0.3029	0.8055	0.1820	-77.4%
	Y	0.01	0.1741	0.0261	0.1063	0.0177	-83.4%
	N	0.1	0.0648	0.0175	0.0390	0.0079	-79.8%
	Y	0.1	0.0225	0.0034	0.0174	0.0022	-87.2%
LastFm	N	0.01	0.1525	0.0185	0.0627	0.0118	-81.2%
	Y	0.01	0.0273	0.0046	0.0141	0.0034	-76.2%
	N	0.1	0.0075	0.0028	0.0047	0.0015	-68.1%
	Y	0.1	0.0048	0.0013	0.0031	0.0009	-72.1%

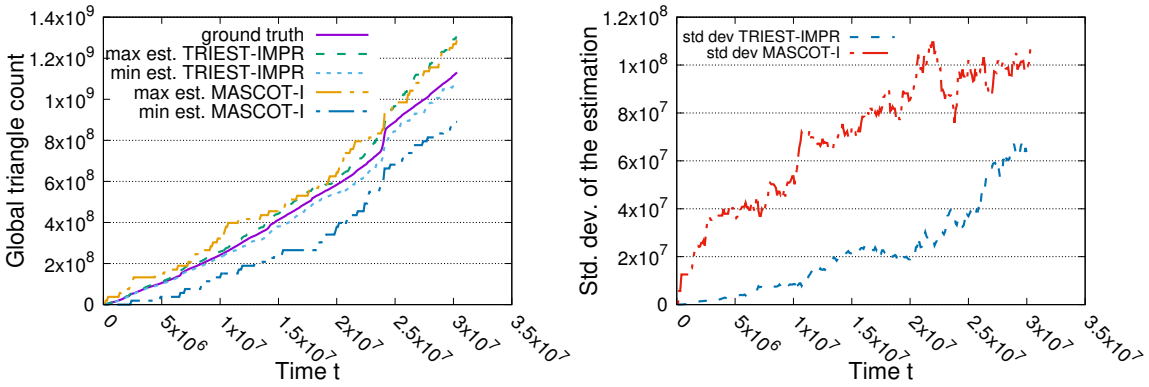
Table 7.3: Global triangle estimation MAPE for TRIÈST and MASCOT. The rightmost column shows the reduction in terms of the avg. MAPE obtained by using TRIÈST. Rows with Y in column "Impr." refer to improved algorithms (TRIÈST-IMPR and MASCOT-I) while those with N to basic algorithms (TRIÈST-BASE and MASCOT-C).



(a) MAPE **(b) Update Time**

Figure 7.2: Average MAPE and average update time of the various methods on the Patent (Co-Aut.) graph with $p = 0.01$ (for MASCOT, see the main text for how we computed the space used by the other algorithms) – insertion only. TRIÈST-IMPR has the lowest error. Both PAVAN ET AL. and JHA ET AL. have very high update times compared to our method and the two MASCOT variants.

runs and the standard deviation of the estimation over those runs. TRIÈST-IMPR shows significantly lower standard deviation (hence variance) over the evolution of the stream, and the max and min lines are also closer to the ground truth. This confirms our theoretical observations in the previous sections. Even with very low M (about 2/10000 of the size of the graph) TRIÈST gives high-quality estimations.



(a) Ground truth, max, and min **(b) Standard deviation**

Figure 7.3: Accuracy and stability of the estimation of TRIÈST-IMPR with $M = 10000$ and of MASCOT-I with same expected memory, on LastFM, over 10 runs. TRIÈST-IMPR has a smaller standard deviation and moreover the max/min estimation lines are closer to the ground truth. Average estimations not shown as they are qualitatively similar.

Local triangle counting We compare the precision in local triangle count estimation of TRIÈST with that of MASCOT [144] using the same approach of the previous experiment. We can not compare with JHA ET AL. and PAVAN ET AL. algorithms as they provide only global estimation. As in [144], we measure the Pearson coefficient and the average ε error (see [144] for definitions). In Table 7.4 we report the Pearson coefficient and average ε error over all timestamps for the smaller graphs.[17] TRIÈST (significantly) improves (i.e., has higher correlation and lower error) over the state-of-the-art MASCOT, using the same amount of memory.

Trade-offs between memory and accuracy We study the trade-offs between the sample size M, the running time, and the accuracy of the estimators. Figure 7.4a shows the trade-offs between the accuracy of the estimation (as MAPE) and the size M for the smaller graphs for which the ground truth number of triangles can be computed exactly using the naïve algorithm. Even with small M, TRIÈST-IMPR achieves very low MAPE value. As expected, larger M corresponds to higher accuracy and for the same M TRIÈST-IMPR outperforms TRIÈST-BASE.

Figure 7.4b shows the average time per update in microseconds (μs) for TRIÈST-IMPR as function of M. Some considerations on the running time are in order. First, a larger edge sample (larger M) generally requires longer average update times per operation. This is expected as a larger sample corresponds to a larger sample graph on which to count triangles. Second, on average a few hundreds microseconds are sufficient for handling any update even in very large graphs with billions of edges. Our algorithms can handle hundreds of thousands of edge updates (stream elements) per second, with very small error (Fig. 7.4a), and therefore TRIÈST can be used efficiently and effectively in high-velocity contexts. The larger average time per update for Patent (Co-Auth.) can be explained by the fact that the graph is relatively dense and has a small size (compared to the larger Yahoo!

[17] For efficiency, in this test we evaluate the local number of triangles of all nodes every 1000 edge updates.

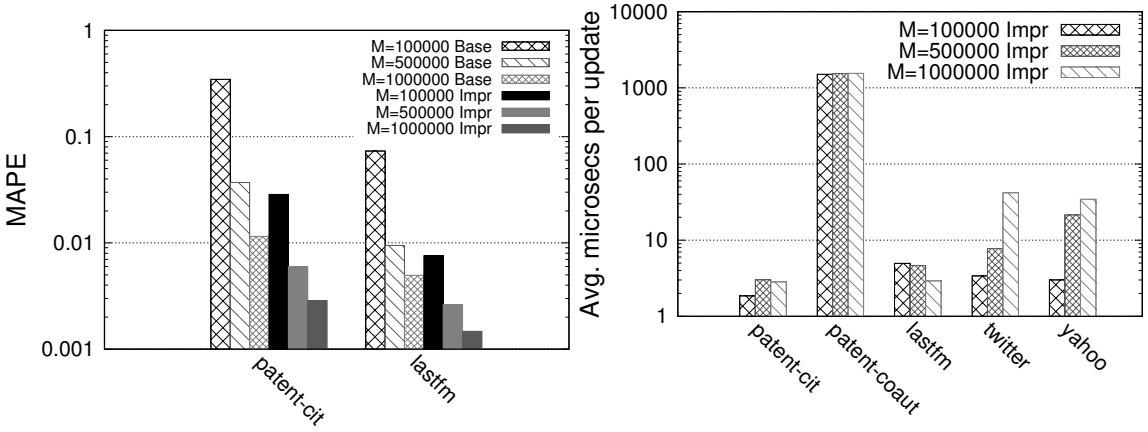
Graph	Impr.	p	Avg. Pearson			Avg. ε Err.		
			MASCOT	TRIÈST	Change	MASCOT	TRIÈST	Change
LastFm	Y	0.1	0.99	1.00	+1.18%	0.79	0.30	-62.02%
		0.05	0.97	1.00	+2.48%	0.99	0.47	-52.79%
		0.01	0.85	0.98	+14.28%	1.35	0.89	-34.24%
	N	0.1	0.97	0.99	+2.04%	1.08	0.70	-35.65%
		0.05	0.92	0.98	+6.61%	1.32	0.97	-26.53%
		0.01	0.32	0.70	+117.74%	1.48	1.34	-9.16%
Patent (Cit.)	Y	0.1	0.41	0.82	+99.09%	0.62	0.37	-39.15%
		0.05	0.24	0.61	+156.30%	0.65	0.51	-20.78%
		0.01	0.05	0.18	+233.05%	0.65	0.64	-1.68%
	N	0.1	0.16	0.48	+191.85%	0.66	0.60	-8.22%
		0.05	0.06	0.24	+300.46%	0.67	0.65	-3.21%
		0.01	0.00	0.003	+922.02%	0.86	0.68	-21.02%
Patent (Co-aut.)	Y	0.1	0.55	0.87	+58.40%	0.86	0.45	-47.91%
		0.05	0.34	0.71	+108.80%	0.91	0.63	-31.12%
		0.01	0.08	0.26	+222.84%	0.96	0.88	-8.31%
	N	0.1	0.25	0.52	+112.40%	0.92	0.83	-10.18%
		0.05	0.09	0.28	+204.98%	0.92	0.92	0.10%
		0.01	0.01	0.03	+191.46%	0.70	0.84	20.06%

Table 7.4: Comparison of the quality of the local triangle estimations between our algorithms and the state-of-the-art approach in [144]. Rows with Y in column "Impr." refer to improved algorithms (TRIÈST-IMPR and MASCOT-I) while those with N to basic algorithms (TRIÈST-BASE and MASCOT-C). In virtually all cases we significantly outperform MASCOT using the same amount of memory.

and Twitter graphs). More precisely, the average time per update (for a fixed M) depends on two main factors: the average degree and the length of the stream. The denser the graph is, the higher the update time as more operations are needed to update the triangle count every time the sample is modified. On the other hand, the longer the stream, for a fixed M, the lower is the frequency of updates to the reservoir (it can be show that the expected number of updates to the reservoir is $O(M(1 + \log(\frac{t}{M})))$ which grows sub-linearly in the size of the stream t). This explains why the average update time for the large and dense Yahoo! and Twitter graphs is so small, allowing the algorithm to scale to billions of updates.

Alternative edge orders In all previous experiments the edges are added in their natural order (i.e., in order of their appearance).[18] While the natural order is the most important use case, we have assessed the impact of other ordering on the accuracy of the algorithms. We experiment with both the uniform-at-random (u.a.r.) order of the edges and the random BFS order: until all the graph is explored a BFS is started from a u.a.r. unvisited node and edges are added in order of their

[18]Excluding Twitter for which we used the random order, given the lack of timestamps.

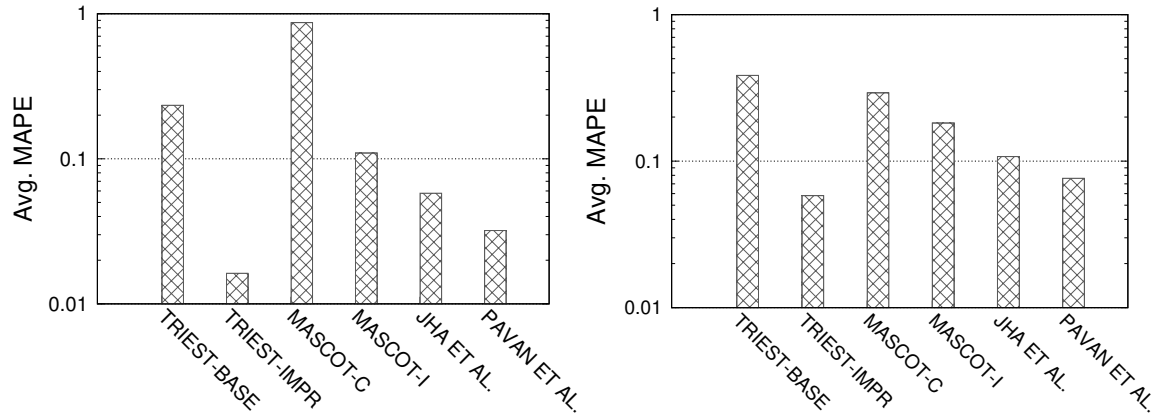


(a) Trade-off between M and MAPE

(b) Trade-off bewtween M and average time per update (TRIÈST-IMPR)

Figure 7.4: Trade-offs between M and MAPE and average time per update in μs – edge insertion only. Larger M implies lower errors but generally higher update times.

visit (neighbors are explored in u.a.r. order). The results for the random BFS order and u.a.r. order (Fig. 7.5) confirm that TRIÈST has the lowest error and is very scalable in every tested ordering.



(a) BFS order

(b) Uniform-at-random order

Figure 7.5: Average MAPE on Patent (Co-Aut.), with $p = 0.01$ (for MASCOT, see the main text for how we computed the space used by the other algorithms) – insertion only in Random BFS order and in uniform-at-random order. TRIÈST-IMPR has the lowest error.

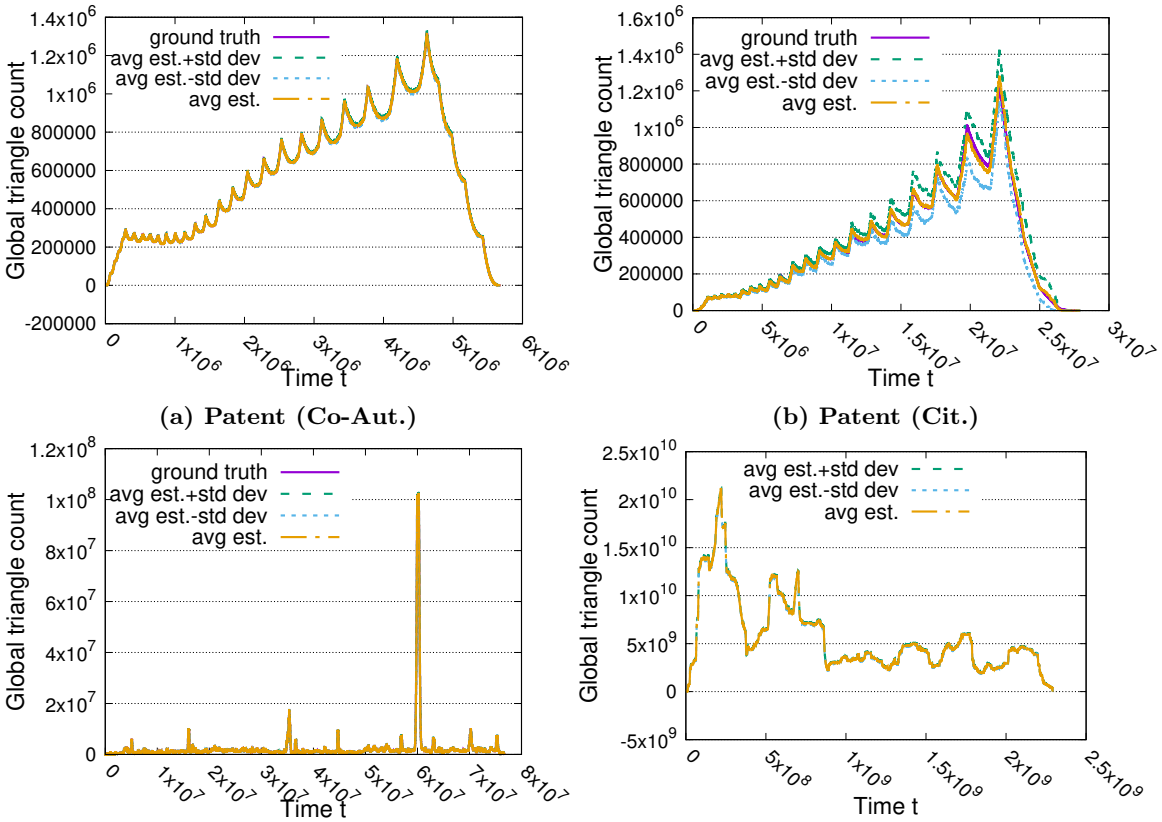
7.5.2 Fully-dynamic case

We evaluate TRIÈST-FD on fully-dynamic streams. We cannot compare TRIÈST-FD with the algorithms previously used [116, 180, 144] as they only handle insertion-only streams.

In the first set of experiments we model deletions using the widely used *sliding window model*, where a sliding window of the most recent edges defines the current graph. The sliding window

model is of practical interest as it allows to observe recent trends in the stream. For Patent (Co-Aut.) & (Cit.) we keep in the sliding window the edges generated in the last 5 years, while for LastFm we keep the edges generated in the last 30 days. For Yahoo! Answers we keep the last 100 millions edges in the window[19].

Figure 7.6 shows the evolution of the global number of triangles in the sliding window model using TRIÈST-FD using M = 200,000 (M = 1,000,000 for Yahoo! Answers). The sliding window scenario is significantly more challenging than the addition-only case (very often the entire sample of edges is flushed away) but TRIÈST-FD maintains good variance and scalability even when, as for LastFm and Yahoo! Answers, the global number of triangles varies quickly.



(a) Patent (Co-Aut.) **(b) Patent (Cit.)**

(c) LastFM **(d) Yahoo! Answers**

Figure 7.6: Evolution of the global number of triangles in the fully-dynamic case (sliding window model for edge deletion). The curves are *indistinguishable on purpose*, to remark the fact that TRIÈST-FD estimations are extremely accurate and consistent. We comment on the observed patterns in the text.

Continuous monitoring of triangle counts with TRIÈST-FD allows to detect patterns that would otherwise be difficult to notice. For LastFm (Fig. 7.6(c)) we observe a sudden spike of several order of magnitudes. The dataset is anonymized so we cannot establish which songs are responsible for

[19]The sliding window model is not interesting for the Twitter dataset as edges have random timestamps. We omit the results for Twitter but TRIÈST-FD is fast and has low variance.

this spike. In Yahoo! Answers (Fig. 7.6(d)) a popular topic can create a sudden (and shortly lived) increase in the number of triangles, while the evolution of the Patent co-authorship and co-citation networks is slower, as the creation of an edge requires filing a patent (Fig. 7.6(a) and (b)). The almost constant increase over time[20] of the number of triangles in Patent graphs is consistent with previous observations of *densification* in collaboration networks as in the case of nodes' degrees [141] and the observations on the density of the densest subgraph [68].

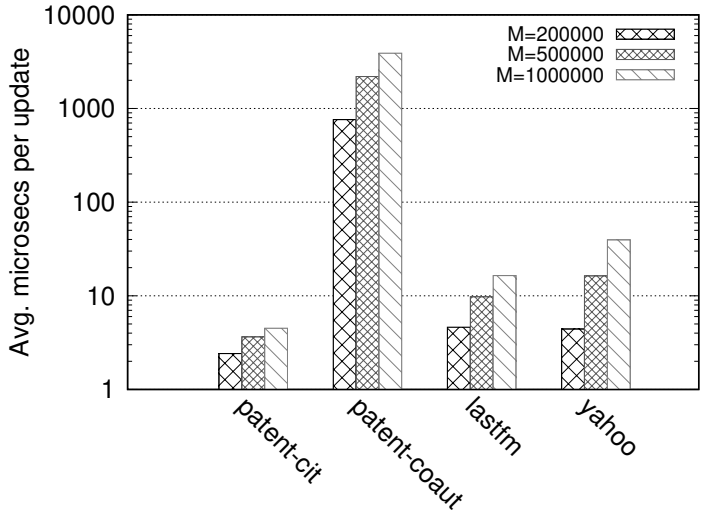


Figure 7.7: Trade-offs between the avg. update time (μs) and M for TRIÈST-FD.

Table 7.5 shows the results for both the local and global triangle counting estimation provided by TRIÈST-FD. In this case we can not compare with previous works, as they only handle insertions. It is evident that precision improves with M values, and even relatively small M values result in a low MAPE (global estimation), high Pearson correlation and low ε error (local estimation). Figure 7.7 shows the tradeoffs between memory (i.e., accuracy) and time. In all cases our algorithm is very fast and it presents update times in the order of hundreds of microseconds for datasets with billions of updates (Yahoo! Answers).

Alternative models for deletion We evaluate TRIÈST-FD using other models for deletions than the sliding window model. To assess the resilience of the algorithm to massive deletions we run the following experiments. We added edges in their natural order but each edge addition is followed with probability q by a mass deletion event where each edge currently in the the graph is deleted with probability d independently. We run experiments with $q = 3{,}000{,}000^{-1}$ (i.e., a mass deletion expected every 3 millions edges) and $d = 0.80$ (in expectation 80% of edges are deleted). The results are shown in Table 7.6.

We observe that TRIÈST-FD maintains a good accuracy and scalability even in face of a massive

[20]The decline at the end is due to the removal of the last edges from the sliding window after there are no more edge additions.

Graph	M	Avg. Global MAPE	Avg. Local Pearson	Avg. Local ε Err.
LastFM	200000	0.005	0.980	0.020
	1000000	0.002	0.999	0.001
Patent (Co-Aut.)	200000	0.010	0.660	0.300
	1000000	0.001	0.990	0.006
Patent (Cit.)	200000	0.170	0.090	0.160
	1000000	0.040	0.600	0.130

Table 7.5: Estimation errors for TRIÈST-FD.

Graph	M	Avg. Global MAPE	Avg. Local Pearson	Avg. Local ε Err.
LastFM	200000	0.040	0.620	0.53
	1000000	0.006	0.950	0.33
Patent (Co-Aut.)	200000	0.060	0.278	0.50
	1000000	0.006	0.790	0.21
Patent (Cit.)	200000	0.280	0.068	0.06
	1000000	0.026	0.510	0.04

Table 7.6: Estimation errors for TRIÈST-FD mass deletion experiment, $q = 3{,}000{,}000^{-1}$ and $d = 0.80$.

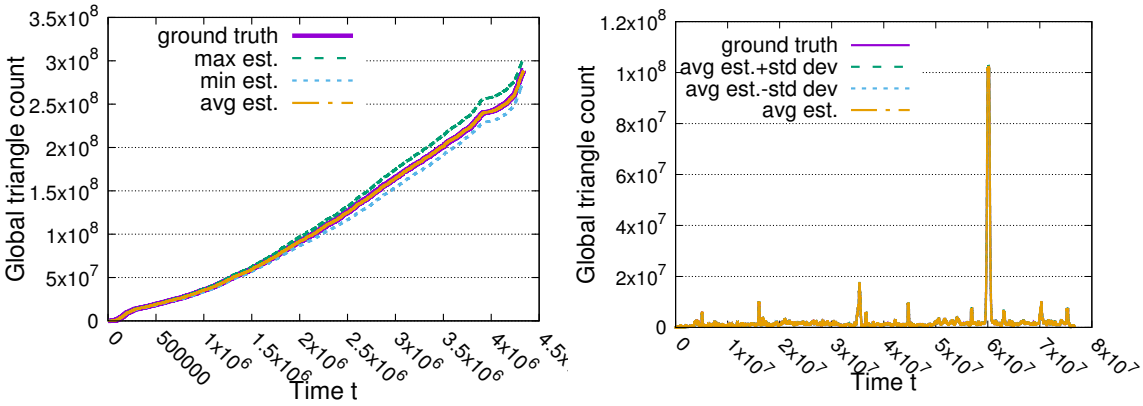
(and unlikely) deletions of the vast majority of the edges: e.g., for LastFM with $M = 200000$ (resp. $M = 1{,}000{,}000$) we observe 0.04 (resp. 0.006) Avg. MAPE.

7.5.3 Multigraphs

We now evaluate our algorithms designed for multigraphs. We obtained multigraph versions of Patent (Co-Auth.) (resp. LastFM) by allowing multiple edges to be placed between pairs of authors (resp. songs) at multiple time steps (i.e., edges with different timestamps) if the two authors co-author multiple papers (resp. the songs are co-listened in different dates). We ran our insertion-only algorithms on these multigraphs and report the results in the next paragraphs.

Figure 7.8 shows the evolution of the number of triangles in the two datasets as estimated by our TRIÈST-IMPR-M algorithm using $M = 100{,}000$. For these smaller datasets we are able to compute the exact number of triangles. Our algorithm is very precise with average, min and max estimations close to the ground truth. The overall observations made for the simple graph case also hold for the multigraph case: our suite of algorithms allows precise and efficient estimation of the number of triangles with limited memory.

Figure 7.9 shows the average update time in microseconds using TRIÈST-IMPR-M algorithm in our multigraph datasets: few microseconds are sufficient on average to update the triangle estimation, consistently with the results of the previous section.



(a) Patent (Co-Aut.)

(b) LastFM

Figure 7.8: Evolution of the global number of triangles in the insertion-only case on multigraphs using TRIÈST-IMPR-M and $M = 100{,}000$. The algorithm estimations are consistently very accurate, and the curves are shown as almost undistinguishable on purpose to highlight this fact.

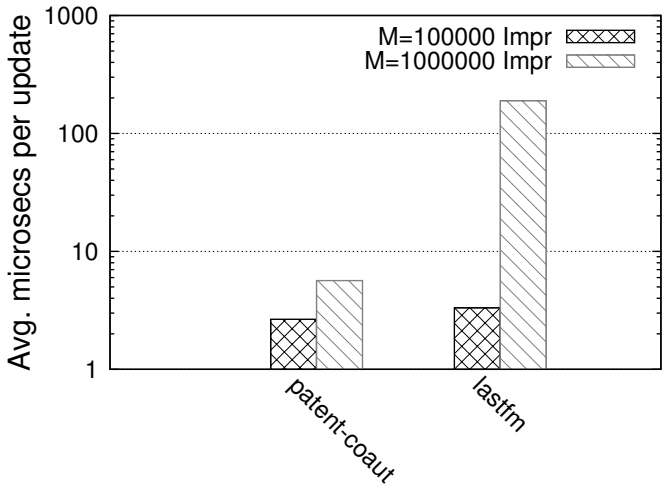


Figure 7.9: Trade-offs between the avg. update time (μs) and M for TRIÈST-IMPR-M – multigraphs.

Finally we evaluate the accuracy of the estimation using our TRIÈST-BASE-M and TRIÈST-IMPR-M algorithms. The results are shown in Table 7.7. We observe that TRIÈST-BASE-M and TRIÈST-IMPR-M maintain a good accuracy with performance comparable to the one observed for the insertion-only stream.

7.6 Conclusions

We presented TRIÈST, the first suite of algorithms that use reservoir sampling and its variants to continuously maintain unbiased, low-variance estimates of the local and global number of triangles

Graph	M	Global Error TRIÈST-BASE-M		Global Error TRIÈST-IMPR-M	
		Avg. MAPE	Max MAPE	Avg. MAPE	Max MAPE
LastFM	100000	0.015	0.024	0.008	0.015
	1000000	0.006	0.012	0.003	0.008
Paten (Co-Aut.)	100000	0.068	0.141	0.023	0.049
	1000000	0.011	0.017	0.003	0.006

Table 7.7: Estimation errors for TRIÈST-BASE-M and TRIÈST-IMPR-M – multigraphs.

in fully-dynamic graphs streams of arbitrary edge/vertex insertions and deletions using a fixed, user-specified amount of space. Our experimental evaluation shows that TRIÈST outperforms state-of-the-art approaches and achieves high accuracy on real-world datasets with more than one billion of edges, with update times of hundreds of microseconds.

Chapter 8

Tiered Sampling: An Efficient Method for Approximate Counting Sparse Motifs in Massive Graph Streams[1]

In this Chapter, we introduce Tiered Sampling, a further development of the techniques discussed for TRIÉST for estimating the count of sparse motifs in massive graphs whose edges are observed in a stream. Our technique requires only a single pass on the data and uses a memory of fixed size M, which can be magnitudes smaller than the number of edges.

Our methods address the challenging task of counting sparse motifs - sub-graph patterns that have low probability of appearing in a sample of M edges in the graph, which is the maximum amount of data available to the algorithms in each step. To obtain an unbiased and low variance estimate of the count we partition the available memory to tiers (layers) of reservoir samples. While the base layer is a standard reservoir sample of edges, other layers are reservoir samples of sub-structures of the desired motif. By storing more frequent sub-structures of the motif, we increase the probability of detecting an occurrence of the sparse motif we are counting, thus decreasing the variance and error of the estimate.

While we focus on the designing and analysis of algorithms for counting 4-cliques, we present a method which allows generalizing Tiered Sampling to obtain high-quality estimates for the number of occurrence of any sub-graph of interest, while reducing the analysis effort due to specific properties of the pattern of interest.

We demonstrate the advantage of our method in the specific applications of counting sparse

[1] A preliminary version of the results presented in this chapter appeared in the proceedings of the 5-th IEEE International Conference on Big Data (BigData 17). This is joint work with Erisa Terolli and Professor Eli Upfal.

4-cliques and 5-cliques in massive graphs. We present a complete analytical analysis and extensive experimental results using both synthetic and real-world data. Our results demonstrate the advantage of our method in obtaining high-quality approximations for the number of 4 and 5-cliques for large graphs using a very limited amount of memory, significantly outperforming the single edge sample approach for counting sparse motifs in large scale graphs.

8.1 Introduction

Counting motifs (sub-graphs with a given pattern) in large graphs is a fundamental primitive in graph mining with numerous practical applications including link prediction and recommendation [142], community detection [20], topic mining [64], spam and anomaly detection [14, 145, 69], protein interaction networks analysis [161], and analysis of temporal patterns [146].

Computing the exact count of motifs in massive, Web-scale networks is often impractical or even infeasible. Furthermore, many interesting networks, such as social networks, are continuously growing. Hence, there is limited value in maintaining an exact count. Rather, the goal is to have, *at any time*, a high-quality approximation of the quantity of interest. To obtain a scalable and efficient solution for massive size graphs we focus on the well-studied model of one-pass stream computing. Our algorithms use a memory of fixed size M, where M is significantly smaller than the size of the input graph. The input graph is observed as a *stream* of edges in an *arbitrary* order, and the algorithm has only one pass on the input. The goal of the algorithm is to compute, at any given time, an unbiased and low-variance estimate of the count of motif occurrences in the graph observed up to that time.

Given its theoretical and practical importance, the problem of counting motifs in graph streams has received a lot of attention in the literature, with particular emphasis on the approximation of the number of 3-cliques (triangles) [51, 179, 138]. A standard approach to this problem is to sample up to M edges uniformly at random, using a fixed sampling probability or, more efficiently, reservoir sampling. A count of the number of motifs in the sample, extrapolated (normalized) appropriately, gives an unbiased estimate for the number of occurrences in the entire graph. The variance (and error) of this method depends on the expected number of occurrences in the sample. In particular, for sparse motifs that are unlikely to appear many times in the sample, this method exhibits high variance (higher than the actual count), which makes it useless for counting. Note that when the input graph is significantly larger than the memory size M, a motif that is unlikely to appear in a random sample of M edges may still have a large count in the graph. Also, as we attempt to count larger structures than triangles, these structures are more likely to be sparse in the graph. It is therefore important to obtain efficient methods for counting *sparser* motifs in massive-scale graph streams.

In this Chapter, we introduce the concept of Tiered Sampling in stream computing. To obtain an unbiased and low variance estimate for the amount of sparse motif in massive-scale graphs we partition the available memory to tiers (layers) of reservoir samples. The base tier is a standard

reservoir sample of individual edges, while other tiers are reservoir samples of sub-structures of the desired motif. This strategy significantly improves the probability of detecting occurrences of the motif.

Assume that we count motifs with k edges. If all the available memory is used to store a sample of the edges, we would need $k-1$ of the motif's edges to be in the sample when the last edge of the motif is observed on the stream. The probability of this event decreases exponentially in k. Assume now that we use part of the available memory to store a sample of the observed occurrences of a fixed "*prototype*" sub-motif with $k/2$ edges. We are more likely to observe such motifs (we only need $k/2-1$ of their edges to be in the edge sample when the last edge is observed in the stream), and they are more likely to stay in the second reservoir sample since they are still relatively sparse. We now observe a full motif when the current edge in the stream completes an occurrence of the motif with edges and with a prototype sub-motifs in the two reservoirs. For an appropriate choice of parameters, and with fixed total memory size, this event has a significantly higher probability than observing the $k-1$ edges in the edge sample. To obtain an unbiased estimate of the count, the number of observed occurrences needs to be carefully normalized by the probabilities of observing each of the components.

In this work, we make the following main contributions:

- We introduce the*Tiered Sampling* framework for counting sparse motifs in large scale graph streams using multi-layer reservoir samples.
- We develop and fully analyze two algorithms for counting the number of 4-cliques in a graph using two-tier reservoir sampling.
- For the purpose of comparison, we analyze a standard (single-tier) reservoir sample algorithm for 4-cliques counting problem.
- We verify the advantage of our multi-layer algorithms by analytically comparing their performance to a standard (single-tier) reservoir sample algorithm for 4-cliques counting problem on random Barábasi-Albert graphs.
- We develop the ATS4C technique, which allows to adaptively adjust the sub-division of memory space among the two tiers according to the properties of the graph being considered.
- We conduct an extensive experimental evaluation of our algorithms for counting 4-cliques on massive graphs with up to hundreds of millions of edges. We show the quality of the achieved estimations by comparing them with the actual ground truth value. Our algorithms are also extremely scalable, showing update times in the order of hundreds of microseconds for graphs with billions of edges. Further, we show that our methods consistently outperform alternative approaches based on edge sampling.
- We present a sequence of steps which allows to generalize the TieredSampling approach to count any arbitrary sub-graph of interest. We show how to greatly simplify the analysis

while retaining high-quality estimates. As an example, we obtain TS5C, which estimates the number of 5-cliques in a graph stream using an edge reservoir sample and a second reservoir sample of the 4-cliques observed on the stream.

Chapter organization: In Section 8.2 we introduce the notation used in the presentation and some fundamental concepts used in this work. In Section 5.2 we discuss methods and result related to our work from the literature. In Section 8.4 we introduce the TieredSampling approach and its application to the problem of counting 4-cliques in a graph steam. In Section 8.4.1 (resp., Section 8.4.2) we present algorithm TS4C_1 (resp., TS4C_2) and we study its statistical properties. In Section 8.6 we delve into the comparison of the methods based on the TieredSampling approach with methods for counting 4-cliques using a single reservoir sample of edges observed on the stream, such as algorithm FourEst which we present and analyze in Section 8.6.1. We compare analytically the variance of the proposed methods (Section 8.6.2), and their performance for random Barábasi-Albert graphs [8] (Section 8.6.3). In Section 8.7, we introduce a variation of the previous TieredSampling algorithms, in which the partition of the memory among the two samples (*tiers*) being used can be adjusted dynamically to adapt to the properties of the graph of interest. We showcase the quality of the estimators provide by TieredSampling algorithms and their benefit with respect to edge-sampling approaches in Section 7.5 via extensive experimental evaluation on many real-world graphs of size ranging from millions to several hundred million edges. Finally, in Section 8.9 we discuss how to generalize the TieredSampling to any sub-graph of interest. In particular, we show how an opportune simplified analysis may ease such generalization by reducing the effort of the analysis. As an example, we present and evaluate algorithm TS5C which yields estimates the count of 5-cliques in a graph stream.

In this work we make the following main contributions:

- We introduce the concept of *Tiered Sampling* for counting sparse concepts in large scale graph stream using multi-layer reservoir samples.
- We develop and fully analyze two algorithms for counting the number of 4-cliques in a graph using two tier reservoir sampling.
- For comparison purpose we analyze a standard (one tier) reservoir sample algorithm for 4-cliques counting problem.
- We verify the advantage of our multi-layer algorithms by analytically comparing their performance to a standard (one tier) reservoir sample algorithm for 4-cliques counting problem on random Barábasi-Albert graphs.
- We develop the ATS4C technique, which allows to adaptively adjust the sub-division of memory space among the two tiers according to the properties of the graph being considered.
- We conduct an extensive experimental evaluation of the 4-clique algorithms on massive graphs with up to hundreds of millions of edges. We show the quality of the achieved estimations

by comparing them with the actual ground truth value. Our algorithms are also extremely scalable, showing update times in the order of hundreds of microseconds for graphs with billions of edges.

- We demonstrate the generality of our approach through a second application of the two-tier method, estimating the number of 5-cliques in a graph stream using a second reservoir sample of the 4-cliques observed on the stream.

To the best of our knowledge these are the first fully analyzed, one pass stream algorithms for the 4 and 5-cliques counting problem.

8.2 Preliminaries

The notation used in this chapter is similar to that used in Chapter 7, albeit with some minor differences. For any (discrete) time step $t \geq 0$, we denote the graph observed up to and including time t as $G^{(t)} = (V^{(t)}, E^{(t)})$, where $V^{(t)}$ (resp., $E^{(t)}$) denotes the set of vertices (resp., edges) of $G^{(t)}$. At time $t = 0$ we have $V^{(t)} = E^{(t)} = \emptyset$. For any $t > 0$, at time $t + 1$ we receive one single edge $e_{t+1} = (u, v)$ from a stream, where u, v are two distinct vertices. $G^{(t+1)}$ is thus obtained by *inserting the new edge*: $E^{(t+1)} = E^{(t)} \cup \{(u, v)\}$; if either u or v do not belong to $V^{(t)}$, they are added to $V^{(t+1)}$. Edges can be added just once (we discuss a generalization for *multigraphs* at the end of Section 8.2) in an arbitrary adversarial order, i.e., as to cause the worst outcome for the algorithm. We, however, assume that the adversary has *no access to the random bits* used by the algorithm.

This work explores the idea of storing a sample of *prototype* sub-motifs in order to enhance the count of a sparse motif. For concreteness we focus on estimating the counts of 4-cliques and 5-cliques. Given a graph $G^{(t)} = (V^{(t)}, E^{(t)})$, a *k-clique* in $G^{(t)}$ is a *set* of $\binom{k}{2}$ (distinct) edges connecting a set of k (distinct) vertices.

Problem definition. We study the 4-Clique Counting Problem in Graph Edges Streams, which requires to compute, at each time $t \geq 0$ an estimation of $|\mathcal{C}_k^{((t))}|$. We denote by $\mathcal{C}_k^{(t)}$ the set of *all* k-cliques in $G^{(t)}$.

Reservoir Sampling: Our work makes use of the *reservoir sampling scheme* [234]. Consider a stream of elements e_i observed in discretized time steps. Given a fixed sample size $M > 0$, for any time step t the reservoir sampling scheme allows to maintain a uniform sample $\mathcal{S}$ of size $\min\{M, t\}$ of the t elements observed on the stream:

- If $t \leq M$, then the element $e_t = (u, v)$ on the stream at time t is deterministically inserted in $\mathcal{S}$.

- If $t > M$, then the sampling mechanism flips a biased coin with heads probability M/t. If the outcome is "*heads*", it chooses an element e_i uniformly at random from those currently in $\mathcal{S}$ to be replaced by e_t. Otherwise, $\mathcal{S}$ is not modified.

Symbol	Explanation
Σ	Edge stream
t	Time Step, i.e., number of edges observed up to step t (included)
t_Δ	Number of triangles observed up to now
$G^{(t)}$	Graph observed up to time t (included)
e_t	Edge observed in the stream at time t.
$\mathcal{C}_k^{(t)}$	Set of k-cliques in $G^{(t)}$
$\varkappa^{(t)}$	Estimation of number of 4-cliques at time t
$\mathcal{S}_e$	Uniform sample of edges
$\mathcal{S}_e^{(t)}$	Content of $\mathcal{S}_e$ at the beginning of time t
$\mathcal{S}_\Delta$	Sample of observed triangles
$\mathcal{S}_\Delta^{(t)}$	Content of $\mathcal{S}_\Delta$ at the beginning of time t
M	Memory Size
M_e	Edge sample memory size
M_Δ	Triangle sample memory size
$\tau^{(t)}$	Number of triangles seen by TS4C_1 (TS4C_2) up to time t
$\mathcal{N}_u^{\mathcal{S}_e^{(t)}}$	Neighborhood of u with respect to edges in $\mathcal{S}_e^{(t)}$
$\mathcal{N}_{u,v}^{\mathcal{S}}$	Common neighbors of u and v
α	Splitting coefficient

Table 8.1: Notation Table

When using reservoir sampling for estimating the number of occurrences of a sub-graph of interest it is necessary to compute the probability of multiple edges elements being in $\mathcal{S}$ at the same time.

Lemma 8.1 (Lemma 4.1 [51])**.** *For any time step t and any positive integer $k \leq t$, let B be any subset of size $|B| = k \leq \min\{M, t\}$ of the element observed on the stream. Then,* at the end *of time step t (i.e., after updating the sample at time t), we have* $\Pr(B \subseteq \mathcal{S}) = 1$ *if $t \leq M$, and* $\Pr(B \subseteq \mathcal{S}) = \prod_{i=0}^{k-1} \frac{M-i}{t-i}$ *otherwise.*

Evaluation: In our experimental analysis (Sections 8.6.3 and 7.5) we measure the accuracy of the obtained estimator *through the evolution of the graph* in terms of their *Mean Average Percentage Error* (MAPE) [110]. The MAPE measures the relative error of an estimator (in this case, $\varkappa^{(t)}$) with respect to the ground truth (in this case, $|\mathcal{C}_k^{(t)}|$) averaged over t time steps, that is:

$$MAPE = \frac{1}{t} \sum_{i=1}^{t} \frac{|\varkappa^{(t)} - |\mathcal{C}_k^{(t)}||}{|\mathcal{C}_k^{(t)}|}.$$

Multigraphs: Our approach can be extended to count the number of subgraphs on a *multigraph* represented as a stream of edges. Using a formalization analogous to that discussed for graphs, for any (discrete) time instant $t \geq 0$, let $G^{(t)} = (V^{(t)}, \mathcal{E}^{(t)})$ be the multigraph observed up to and

including time t, where $\mathcal{E}^{(t)}$ is now a *bag* of edges between vertices of $V^{(t)}$. The multigraph evolves through a series of edges additions according to a process similar to the one described for graphs. The definition of the occurrence of a sub-graph (e.g., a 3,4 or 5-clique) in a multigraph is the same as in a graph. As before we denote with $\mathcal{C}_k^{(t)}$ the set of *all* k-cliques in $G^{(t)}$, but now this set may contain multiple k-cliques with the same set of vertices, although each of these triangles will be a different set of edges among those vertices, i.e., subset of the bag $\mathcal{E}^{(t)}$ which differ by at least one element. The problem of 4-clique counting in multigraph edge streams is defined exactly in the same way as for graph edge streams. For the sake of simplicity, in the remainder of the presentation we focus on the analysis of graph edges streams.

8.3 Related Work

Counting subgraphs in large networks is a well-studied problem in data mining which was originally brought to attention in the seminal work of Milo et al. [161] on the analysis of protein interaction networks. In particular, many contributions in the literature have focused on the *triangle counting problem*, that is, the counting of the number of 3-cliques, including exact algorithms, MapReduce algorithms [172, 176], and streaming algorithms [51, 3, 117, 179].

Previous works in the literature on counting graph motifs [188, 6] can also be used to estimate the number of cliques in large graphs. Other recent works on *graphlets* (i.e., small subgraphs) counting introduced randomized [114, 173] and MapReduce [76] algorithms. These require however prior information on the graph such as its degeneracy (for [114]) or the vertex degree ordering (for [76]) or the vertex degrees [173]. In [32] Bressan et al. present and compare *Monte Carlo* and *Color Coding* approaches for obtaining estimates of the count of graphlets with five or more vertices in the static setting. These approaches are not, however, used in the streaming setting.

The idea of using sub-structures of a graph motif in order to improve the estimation of its frequency in a massive graph has been previously explored in literature. In [29], Bordino et al. proposed a data stream algorithm that estimates the number of occurrences of a given subgraph by sampling its "*prototypes*" (i.e., sub-structures). While this approach is shown to be effective in estimating the number of occurrences of motifs with three and four edges, it requires multiple passes through the graph stream and further knowledge on the properties of the graph. In [118], Jha et al. proposed an algorithm that effectively and efficiently approximates the frequencies of all 4-vertex subgraphs by sampling paths of length three. However, this algorithm requires prior knowledge of the degrees of all the vertices in the graph and cannot be used in the streaming setting. In [4], Ahmed et al. use reservoir sampling to develop the "*graph priority sampling*" framework for counting subgraphs. In a recent work [114], Jain and Seshadri propose a clique counting method based on Túran's Theorem that requires knowledge of the degeneracy of the observed graph. Bressan et al. show an application of the color coding scheme [33] for the static setting which which use *colorful trees* as *prototypes* to guide the sampling phase. Some alternative approaches focus on counting specific graphs such as *butterfly* sub-graphs in bipartite graph streams [200].

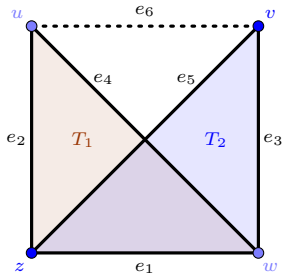


Figure 8.1: Detection of 4-clique using triangles

In this work we present a sampling-based, one pass algorithm for insertion only streams to approximate the global number of cliques found in large graphs. Furthermore, our algorithms do not require any further information on the properties of the graph being observed.

Using a strategy similar to our TieredSampling approach, in [117] Jha and Seshadri propose a one-pass streaming algorithm for triangle counting. This algorithm uses a first reservoir for edges which are then used to generate a stream of *wedges* (i.e., paths of length two) stored in a second reservoir. This approach appears to be not worthwhile for triangle counting as it is consistently outperformed by a simpler strategy based on a single reservoir presented in [51]. This is due to the fact that, as in most large graphs of interest wedges are much more frequent than edges, it is generally not worth devoting a large fraction of the available memory space to maintaining wedges over edges. In [5], Ahmed et al. used a similar approach based on the use of multiple reservoir sample levels while developing a sampling-based streaming algorithm for the for approximate bipartite projection in streaming bipartite networks: first, they maintain a reservoir of sampled bipartite edges with sampling weights that favor the selection of high similarity nodes. Second, arriving edges generate a stream of *similarity updates* based on their adjacency with the current sample. These updates are aggregated in a second reservoir sample-based stream aggregator to yield the final unbiased

8.4 TieredSampling application to 4-clique counting

In this section, we present $TS4C_1$ and $TS4C_2$, two applications of our TieredSampling approach for counting the number of 4-cliques in an undirected graph observed as an edge stream. Rather than counting a 4-cliques *only* when the currently observed edge completes a 4-clique with five other edges maintained in an edge sample (similarly as what successfully done for 3-cliques in [51]), we increase the probability of observing a clique by using a 3-cliques (i.e., triangles) reservoir sample. Our two TieredSampling algorithms partition the available memory into two samples, an edges reservoir sample and a triangles reservoir sample. $TS4C_1$ attempts in each step to construct a 4-cliques using the currently observed edge, two edges from the edge reservoir sample and one triangle from the triangle reservoir sample. $TS4C_2$ attempts at each step to construct a 4-clique from the currently observed edge and two triangles from the triangle reservoir sample. That is, both algorithms use

ALGORITHM 15 TS4C$_1$ - Tiered Sampling for 4-Clique counting

Input: Insertion-only edge stream Σ, integers M, M_Δ
$\mathcal{S}_e \leftarrow \emptyset$, $\mathcal{S}_\Delta \leftarrow \emptyset$, $t \leftarrow 0$, $t_\Delta \leftarrow 0$, $\sigma \leftarrow 0$
for each element (u, v) from Σ **do** ▷ Process each edge (u,v) coming from the stream in discretized timesteps
  $t \leftarrow t + 1$
  UPDATE4CLIQUES(u, v) ▷ Update the 4-clique estimator by considering the new 4-cliques closed by edge (u,v)
  UPDATETRIANGLES(u, v) ▷ Update the triangles reservoir with the new triangles formed by edge (u,v)
  SAMPLEEDGE$((u, v), t)$ ▷ Update the edges sample with the edge (u,v) according to RS scheme

function UPDATE4CLIQUES$((u, v), t)$
  for each triangle $(u, w, z) \in \mathcal{S}_\Delta$ **do** ▷ For each triangle with vertices u, w and z
    if $(v, w) \in \mathcal{S}_e \wedge (v, z) \in \mathcal{S}_e$ **then** ▷ If triangle (u, w, z) forms a 4-clique (u,v,w,z) with two edges in $\mathcal{S}_e$
      $p \leftarrow$ PROBCLIQUE$((u, w, z), (v, w), (v, z))$ ▷ Calculate probability of observing 4-clique (u,v,w,z)
      $\sigma \leftarrow \sigma + p^{-1}/2$ ▷ Update the 4-clique estimator

  for each triangle $(v, w, z) \in \mathcal{S}_\Delta$ **do** ▷ For each triangle with vertices v, w and z
    if $(u, w) \in \mathcal{S}_e \wedge (u, z) \in \mathcal{S}_e$ **then** ▷ In case triangle (v,w,z) forms a 4-clique (u,v,w,z) with two edges in $\mathcal{S}_e$
      $p \leftarrow$ PROBCLIQUE$((v, w, z), (u, w), (u, z))$ ▷ Calculate probability of observing 4-clique (u,v,w,z)
      $\sigma \leftarrow \sigma + p^{-1}/2$ ▷ Update the 4-clique estimator

function UPDATETRIANGLES$((u, v), t)$
  $\mathcal{N}^{\mathcal{S}}_{u,v} \leftarrow \mathcal{N}^{\mathcal{S}}_u \cap \mathcal{N}^{\mathcal{S}}_v$
  for each element w from $\mathcal{N}^{\mathcal{S}}_{u,v}$ **do** ▷ For each triangle (u,v,w)
    $t_\Delta \leftarrow t_\Delta + 1$ ▷ Increment the number of observed triangles
    SAMPLETRIANGLE(u, v, w) ▷ Update the triangles sample with the triangle (u,v,w) according to RS Scheme

triangle sub-patterns of a 4-clique as *prototype* sub-graphs used to aid in the detection of an entire 4-clique. At each time step t, both algorithms maintain a *running estimation* $\varkappa^{(t)}$ of $|\mathcal{C}_4^{(t)}|$. Clearly $\varkappa^{(0)} = 0$ and the estimator is increased every time a 4-clique is "*detected*" on the stream. The two algorithms also maintains a counter $\tau^{(t)}$ for the number of triangles observed in the stream up to time t. This value is used by the reservoir sampling scheme which manages the triangle reservoir.

8.4.1 Algorithm TS4C$_1$

Algorithm TS4C$_1$ maintains an *edges (resp., triangles) reservoir sample* $\mathcal{S}_e$ (resp., $\mathcal{S}_\Delta$) of *fixed* size M_e (resp., M_Δ). From Lemma 8.1, for any t the probability of any edge e (resp., triangle T) observed on the stream Σ (resp., observed by the TS4C$_1$) to be included in S_e (resp., S_Δ) is M_e/t (resp. $M_\Delta/\tau^{(t)}$). We denote as $\mathcal{S}_e^{(t)}$ (resp., $\mathcal{S}_\Delta^{(t)}$) the set of edges (resp., triangles) in $\mathcal{S}_e$ (resp., $\mathcal{S}_\Delta$) *before*

any update to the sample(s) occurring at step t. We denote with $\mathcal{N}_u^{\mathcal{S}_e^{(t)}}$ the *neighborhood* of u with respect to the edges in $\mathcal{S}_e^{(t)}$, that is $\mathcal{N}_u^{\mathcal{S}_e^{(t)}} = \{v \in V^{\mathcal{S}_e^{(t)}} \ : \ (u,v) \in \mathcal{S}_e^{(t)}\}$.

Let $e_t = (u,v)$ be the edge observed on the stream at time t. At each step TS4C$_1$ executes three main tasks:

- *Estimation update:* TS4C$_1$ invokes the function UPDATE 4-CLIQUES to detect any 4-clique completed by (u,v) (see Fig. 8.1 for an example). That is, the algorithm verifies whether in triangle reservoir $\mathcal{S}_\Delta$ there exists any triangle T which includes u (resp., v). Note that since each edge is observed just once and the edge (u,v) is being observed for the first time no triangle in $\mathcal{S}_\Delta$ can include both u and v. For any such triangle $T' = \{u,w,z\}$ (or $\{v,w,z\}$) the algorithm checks whether the edges (v,w) and (v,z) (resp., (u,w) and (u,z)) are currently in $\mathcal{S}_e$. When such conditions are meet, we say that a 4-clique is "*observed on the stream*". The algorithm then uses PROBCLIQUE to compute the *exact* probability p of the observation based on the timestamps of all its edges. The estimator $\varkappa$ is then increased by $p^{-1}/2$.

- *Triangle sample update:* using UPDATETRIANGLES the algorithm verifies whether the edge e_t completes any triangle with the edges in $\mathcal{S}_e^{(t)}$. If that is the case, we say that a new triangle T^* is *observed on the stream*. The counter τ is increased by one and the new triangle is a candidate for inclusion in $\mathcal{S}_\Delta$ with probability M_Δ/τ^t.

- *Edge sample update:* the algorithm updates the edge sample $\mathcal{S}_e$ according to the Reservoir Sampling scheme described in Section 8.2.

Each time a 4-clique is observed on the stream, TS4C$_1$ uses PROBCLIQUE to compute the *exact probability* of the observation. Such computation is all but trivial as it is influenced by both the order according to which the edges of the 4-clique were observed on the stream and by the number of triangles observed on the stream $\tau^{(t)}$. The analysis proceeds using a (somehow tedious) analysis of *all* the 5! possible orderings of the first five edges of the clique observed on the stream. Before presenting the analysis for computing the *exact probability* of observing a 4-clique on the stream in Lemma 8.3, we introduce Lemma 8.2

Lemma 8.2. *Let $\lambda \in \mathcal{C}_4^{(t)}$ with $\lambda = \{e_1,e_2,e_3,e_4,e_5,e_6\}$ using Figure 8.1 as reference. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. λ can be observed by* TS4C$_1$ *at time t_6 either as a combination of triangle $T_1 = \{e_1,e_2,$ or $e_4\}$ and edges $e_3 = (v,w)$ and $e_5 = (v,z)$, or as a combination of triangle $T_2 = \{e_1,e_3,e_5\}$ and edges $e_2 = (u,w)$ and $e_4 = (u,z)$.*

Proof. As presented in Algorithm 15, TS4C$_1$ can detect λ only when its last edge is observed on the stream (hence, t_6). When e_6 is observed, the algorithm first evaluates whether there is any triangle in $\mathcal{S}_\Delta^{(t_6)}$ that shares one of the two endpoint u or v from e_6. Since at this step (i.e., the execution of function UPDATE4CLIQUES) the triangle sample is yet to be updated based on the observation of e_6, the only triangle sub-structures of λ which may have been observed on the stream, and thus included in $\mathcal{S}_\Delta^{(t_6)}$ are $T_1 = \{e_1,e_2,e_4\}$ and $T_2 = \{e_1,e_3,e_5\}$. If any of these is indeed in $\mathcal{S}_\Delta^{(t_6)}$, TS4C$_1$

proceeds to check whether the remaining two edges required to complete λ (resp., e_3, e_5 for T_1, or e_2, e_4 for T_2) are in $\mathcal{S}_e$. λ is thus observed either once or twice depending on which just one or both of these conditions are verified. □

Lemma 8.3. *Let $\lambda \in \mathcal{C}_4^{(t)}$ with $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ using Figure 8.1 as reference. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. Let $t_{1,2,4} = \max\{t_1, t_2, t_4, M_e + 1\}$. The probability p_λ of λ being observed on the stream by* TS4C$_1$ *using the triangle $T_1 = \{e_1, e_2, e_4\}$ and the edges e_3, e_5, is computed by the* ProbClique *function as:*

$$p_\lambda = \frac{M_e}{t_{1,2,4}-1}\frac{M_e-1}{t_{1,2,4}-2}\min\{1, \frac{M_\Delta}{\tau^{(t_6)}}\}p'$$

where if $t_6 \leq M_e$

$$p' = 1,$$

if $\min\{t_3, t_5\} > t_{1,2,4}$

$$p' = \frac{M_e}{t_6-1}\frac{M_e-1}{t_6-2},$$

if $\max\{t_3, t_5\} > t_{1,2,4} > \min\{t_3, t_5\}$

$$p' = \frac{M_e-1}{t_6-2}\frac{M_e-2}{t_{1,2,4}-3}\frac{t_{1,2,4}-1}{t_6-1}$$

otherwise

$$p' = \frac{M_e-2}{t_{1,2,4}-3}\frac{M_e-3}{t_{1,2,4}-4}\frac{t_{1,2,4}-1}{t_6-1}\frac{t_{1,2,4}-2}{t_6-2}.$$

Proof. Let us define the event $E_\lambda = \lambda$ *is observed on the stream by* TS4C$_1$ *using triangle* $T_1 = \{e_1, e_2, e_4\}$ *and edges* e_3, e_5. Further let $E_{T_1} = T_1 \in \mathcal{S}_\Delta^{(t_6)}$, and $E_{3,5} = \{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)}$. Given the definition of TS4C$_1$ we have:

$$E_\lambda = E_{T_1} \wedge E_{3,5},$$

and hence:

$$p_\lambda = \Pr(E_\lambda) = \Pr\left(E_{T_1} \wedge E_{3,5}\right) = \Pr\left(E_{3,5}|E_{T_1}\right)\Pr(E_{T_1}).$$

In order to study $\Pr(E_{T_1})$ we shall introduce event $E_{S(T_1)}$ ="*triangle T_1 is observed on the stream by* TS4C$_1$". From the definition of TS4C$_1$ we know that T_1 is observed on the stream iff when the last edge of T_1 is observed on the stream at $\max\{t_1, t_2, t_4\}$ the remaining two edges are in the edge sample. Applying Bayes's rule of total probability we have:

$$\Pr(E_{T_1}) = \Pr\left(E_{T_1}|E_{S(T_1)}\right)\Pr\left(E_{S(T_1)}\right).$$

and thus:

$$p_\lambda = \Pr\left(E_{3,5}|E_{T_1}\right)\Pr\left(E_{T_1}|E_{S(T_1)}\right)\Pr\left(E_{S(T_1)}\right). \tag{8.1}$$

Let $t_{1,2,4} = \max\{t_1, t_2, t_4, M+1\}$, in order for T_1 to be observed by TS4C$_1$ it is required that when the last edge of T_1 is observed on the stream at $t_{1,2,4}$ its two remaining edges are kept in $\mathcal{S}_e$. From Lemma 7.1 we have:

$$\Pr\left(E_{S(T_1)}\right) = \frac{M_e}{t_{1,2,4}-1}\frac{M_e-1}{t_{1,2,4}-2}, \tag{8.2}$$

and:

$$\Pr\left(E_{T_1}|E_{S(T_1)}\right) = \frac{M_e}{\tau^{t_6}}. \tag{8.3}$$

Let us now consider $\Pr\left(E_{3,5}|E_{T_1}\right)$. While the content of $\mathcal{S}_\Delta$ itself does not influence the content of $\mathcal{S}_e$, the fact that T_1 is maintained in $\mathcal{S}_\Delta^{(t_6)}$ implies that is has been observed on the stream at a

previous time and hence, that two of its edges have been maintained in $\mathcal{S}_e$ *at least* until the last of its edges has been observed on the stream. We thus have $\Pr\left(E_{3,5}|E_{T_1}\right) = \Pr\left(E_{3,5}|E_{S(T_1)}\right)$. In order to study $p' = \Pr\left(E_{3,5}|E_{S(T_1)}\right)$ it is necessary to distinguish the possible (5!) different arrival orders for edges e_1, e_2, e_3, e_4 and e_5. However in an efficient analysis we reduce the number of cases to be considered to just four:

- $t_6 \leq M + e$: in this case *all* the edges observed on the stream up until t_6 are deterministically inserted in $\mathcal{S}_e$ and thus $p' = 1$.

- $\min\{t_3, t_5\} > t_{1,2,4}$: in this cases both edges e_3 and e_5 are observed after *all* the edges composing T_1 have already been observed on the stream. As for any $t > t_{1,2,4}$ the event $E_{S(T_1)}$ does not imply that any of the edges of T_1 is *still* in $\mathcal{S}_e$ we have:
$$p' = \Pr\left(E_{3,5}|E_{T_1}\right) = \Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)}\right),$$
and thus, from Lemma 7.1:
$$p' = \frac{M_e}{t_6 - 1}\frac{M_e - 1}{t_6 - 2}.$$

- $\max\{t_3, t_5\} > t_{1,2,4} > \min\{t_3, t_5\}$: in this case only one of the edges e_3, e_5 is observed after all the edges in T_1 are observed. We need to therefore take into consideration that the two edges of T_1 are kept in S_e until $t_{1,2,4}$. Let $e_{3,5}^M$ (resp., $e_{3,5}^m$) denote the last (resp., first) edge observed on the stream between e_3 and e_5.
$$\begin{aligned} p' &= \Pr\left(e_{3,5}^M \in \mathcal{S}_e^{(t_6)} | e_{3,5}^m \in \mathcal{S}_e^{(t_6)}\right) \Pr\left(e_{3,5}^m \in \mathcal{S}_e^{(t_6)}\right), \\ &= \frac{M_e - 1}{t_6 - 2} \Pr\left(e_{3,5}^m \in \mathcal{S}_e^{(t_6)} | e_{3,5}^m \in \mathcal{S}_e^{(t_{1,2,4})}\right) \Pr\left(e_{3,5}^m \in \mathcal{S}_e^{(t_{1,2,4})}\right), \\ &= \frac{M_e - 1}{t_6 - 2}\frac{t_{1,2,4} - 1}{t_6 - 1}\frac{M_e - 2}{t_{1,2,4} - 3}. \end{aligned}$$

- $t_{1,2,4} > \max\{t_2, t_4\}$: in all the remaining cases both e_3 and e_5 are observed before the last edge of T_1 has been observed. Hence:
$$\begin{aligned} p' &= \Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | \{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_{1,2,4})}\right) \Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_{1,2,4})}\right), \\ &= \frac{M_e - 2}{t_{1,2,4} - 3}\frac{M_e - 3}{t_{1,2,4} - 4}\frac{t_{1,2,4} - 1}{t_6 - 1}\frac{t_{1,2,4} - 2}{t_6 - 2}. \end{aligned}$$

The lemma follows combining the result for the values of p' with (8.2) and (8.3) in (8.1). □

In our proofs, we carefully account for the fact that, as we use reservoir sampling [234], the presence of an edge (resp., of a triangle) in $\mathcal{S}_e$ (resp., $\mathcal{S}_\Delta$) is *not independent* from the concurrent presence of another edge (resp., triangle) in $\mathcal{S}_\Delta$. Further, we account for the fact that whether a triangle can be in $\mathcal{S}_\Delta$ *only if* it was previously detected by TS4C$_1$ (using the UpdateTriangles function from Algorithm 15), which itself is dependent on which edges are held in the sample when the last edge of the triangle itself is observed on the stream. In order to account for such dependencies, it is necessary to break down the order of arrival of the edges themselves.

The following corollary provides a lower bound to the probability according to which a 4-clique is observed by TS4C$_1$. Although generally loose, this result will provide simplification and insight in the analysis of the variance of the estimate obtained using TS4C$_1$.

Corollary 8.4. *Let $\lambda \in \mathcal{C}_4^{(t)}$ defined as in the statement of Lemma 8.3. The probability p_λ of λ being observed on the stream by* TS4C$_1$ *using the triangle $T_1 = \{e_1, e_2, e_4\}$ and the edges e_3, e_5 is evaluated by the* ProbClique *function as:*

$$p_\lambda \geq \min\Big\{1, \left(\frac{M_e}{t-1}\right)^4 \frac{M_\Delta}{\tau^{(t)}}\Big\}$$

Analysis of the TS4C$_1$ estimator We now present the analysis of the estimations obtained using TS4C$_1$. First we show their *unbiasedness* and we then provide a bound on their variance. In the following we denote as t_Δ the first time step at for which the number of triangles seen by TS4C$_1$ exceeds M_Δ.

Theorem 8.5. *The estimator $\varkappa$ returned by* TS4C$_1$ *is unbiased, that is $\varkappa^{(t)} = |\mathcal{C}_k^{(t)}|$ if $t \leq \min\{M_e, t_\Delta\}$. Further, $\mathbb{E}\left[\varkappa^{(t)}\right] = |\mathcal{C}_k^{(t)}|$ if $t \leq \min\{M_e, t_\Delta\}$.*

Proof. From Lemma 8.2 we have that TS4C$_1$ can detect any 4-clique $\lambda \in \mathcal{C}_4^{(t)}$ in exactly two ways: either using triangle T_1 and edges e_3, e_5, or by using triangle T_2 and edges e_2, e_4 (use Figure 8.1 as a reference).

For each $\lambda \in \mathcal{C}_4^{(t)}$ let us consider the random variable δ_{λ_1} (resp. δ_{λ_2}) which takes value $p_{\lambda_1}^{-1}/2$ (resp., $p_{\lambda_2}^{-1}/2$) if the 4-clique λ is observed by TS4C$_1$ using triangle T_1 (resp., T_2) or zero otherwise. Let p_{λ_1} (resp., p_{λ_2}) denote the probability of such event: we then have $\mathrm{E}\left[\delta_{\lambda_1}\right] = \mathrm{E}\left[\delta_{\lambda_2}\right] = 1/2$.

From Lemma 8.3, we have that the estimator $\varkappa^{(t)}$ computed using TS4C$_1$ can be expressed as $\varkappa^{(t)} = \sum_{\lambda \in \mathcal{C}_4^{(t)}} (\delta_{\lambda_1} + \delta_{\lambda_2})$. From the previous discussion and by applying linearity of expectation we thus have:

$$\mathrm{E}\left[\varkappa^{(t)}\right] = \mathrm{E}\left[\sum_{\lambda \in \mathcal{C}_4^{(t)}} (\delta_{\lambda_1} + \delta_{\lambda_2})\right] = \sum_{\lambda \in \mathcal{C}_4^{(t)}} \big(\mathrm{E}\left[\delta_{\lambda_1}\right] + \mathrm{E}\left[\delta_{\lambda_2}\right]\big) = \sum_{\lambda \in \mathcal{C}_4^{(t)}} 1 = |\mathcal{C}_4^{(t)}|.$$

Finally, let t^* denote the first step for which the number of triangles detected by TS4C$_1$ exceeds M_Δ. For $t \leq \min\{M_e, t^*\}$, the entire graph $G^{(t)}$ is maintained in $\mathcal{S}_e$ and all the triangles in $G^{(t)}$ are stored in $\mathcal{S}_\Delta$. Hence all the cliques in $G^{(t)}$ are deterministically observed by TS4C$_1$ *in both ways* and we therefore have $\varkappa^{(t)} = |\mathcal{C}_4^{(t)}|$. □

We now introduce an *upper bound* to the variance of the TS4C$_1$ estimations. We present here the most general result for $t \geq M_e, t_\Delta$. Note that if $t \leq \min\{M_e, t_\Delta\}$, from Theorem 8.5, $\varkappa^{(t)} = |\mathcal{C}_k^{(t)}|$ and hence $\mathrm{Var}\left[\varkappa^{(t)}\right] = 0$. For $\min\{M_e, t_\Delta\} < t \leq \max\{M_e, t_\Delta\}$, $\mathrm{Var}\left[\varkappa^{(t)}\right]$ admits an upper bound similar to the one in Theorem 8.6.

Theorem 8.6. *For any time $t > \min\{M_e, t_\Delta\}$, he estimator $\varkappa$ returned by* TS4C$_1$ *satisfies*

$$\begin{aligned} Var\left[\varkappa^{(t)}\right] \leq |\mathcal{C}_4^{(t)}| &\left(c\left(\frac{t-1}{M_e}\right)^4 \left(\frac{\tau^{(t)}}{M_\Delta}\right) - 1\right) + 2a^{(t)}\left(c\frac{t-1}{M_e} - 1\right) \\ &+ 2b^{(t)}\left(c\left(\frac{t-1}{M_e}\right)^2 \left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta} + \frac{3}{4}\frac{t-1}{M_e}\right) - 1\right), \end{aligned} \tag{8.4}$$

where $a^{(t)}$ (resp., $b^{(t)}$) denotes the number of unordered pairs of 4-cliques which share one edge (resp., three edges) in $G^{(t)}$, and $c \geq \frac{M_e^3}{(M_e-1)(M_e-2)(M_e-3)}$.

Similarly to what done in the proof of Theorem 8.5, in the proof of our result on the bound of the variance of the estimate we associate two random variables to each of the 4-cliques of the graph (one for each of the two possible ways of detecting such 4-clique). The proof relies on a careful analysis of the order of arrival of the edges shared between a pair of two 4-cliques in the stream (which we assume to be adversarial), as shown in Lemma 8.7. When bounding the variance, we must consider not only pairs of 4-cliques that share edges, but also pairs of 4-cliques sharing no edges, since the respective presences of the parts used to detect them (i.e., edges and triangles) in the respective samples are not independent events.

In order to gain some insight on the variance bound in Theorem 8.6, note that the first, and dominant, term of (8.4) is given by the total number of 4-cliques in $G^{(t)}$ (i.e., $|\mathcal{C}_4^{(t)}|$) multiplied by $\left(c\left(\frac{t-1}{M_e}\right)^4\left(\frac{\tau^{(t)}}{M_\Delta}\right)-1\right)$ which corresponds *approximately* to $c(p_\lambda)^{-1}$, where p_λ denotes the *lower bound* to the probability of TS4C$_1$ detecting a 4-clique as provided by Corollary 8.4. This suggest a natural relation between these quantities: as the probability of detecting a 4-clique decreases (resp., the total number of 4-cliques increases) the variance increases accordingly.

Before presenting the proof of Theorem 8.6, we introduce some the following lemma which concerns the possible size of the edge intersection of two 4-cliques.

Lemma 8.7. *Any pair (λ, γ) of distinct 4-cliques in $G^{(t)}$ can share either one, three or no edges. If λ and γ share three edges, those three edges compose a triangle.*

Proof. Suppose that λ and γ share exactly two distinct edges. This implies that they share at least three distinct nodes, and thus must share the three edges connecting each pair out of said three nodes. This constitutes a contradiction. Suppose instead that λ and γ share four or five edges while being distinct. This implies that they must share four vertices, hence they cannot be distinct cliques. This leads to a contradiction. □

Lemma 8.7, allows us to point out the various cases to be considered in the main proof of Theorem 8.6.

Proof of Theorem 8.6. Assume $|\mathcal{C}_4^{(t)}| > 0$, otherwise TS4C$_1$ estimation is deterministically correct and has variance 0 and the thesis holds. For each $\lambda \in \mathcal{C}_4^{(t)}$ let $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, without loss of generality let us assume the edges are disposed as in Figure 8.1. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. Let $t_{1,2,4} = \max\{t_1, t_2, t_4, M_e + 1\}$. Let us consider the random variable δ_{λ_1} (resp. δ_{λ_2}) which takes value $p_{\lambda_1}^{-1}/2$ (resp., $p_{\lambda_2}^{-1}/2$) if the 4-clique λ is observed by TS4C$_1$ using triangle $T_1 = \{e_1, e_2, e_4\}$ (resp., T_2) and edges e_3, e_5 (resp., e_2, e_4) or zero otherwise. Let p_{λ_1} (resp., p_{λ_2}) denote the probability of such event.

Since, from Lemma 8.3 we know:

$$\mathrm{Var}\left[\delta_{\lambda_1}\right] = \frac{p_{\lambda_1}^{-1}}{4} - \frac{1}{4} \leq \frac{1}{4}\left(\frac{\tau^{(t)}}{M_\Delta}\prod_{i=0}^{3}\frac{t-1-i}{M_e-i} - 1\right)$$

$$\mathrm{Var}\left[\delta_{\lambda_2}\right] = \frac{p_{\lambda_2}{}^{-1}}{4} - \frac{1}{4} \leq \frac{1}{4}\left(\frac{\tau^{(t)}}{M_\Delta}\prod_{i=0}^{3}\frac{t-1-i}{M_e-i} - 1\right).$$

we have:

$$\begin{aligned}
\mathrm{Var}\left[\varkappa^{(t)}\right] &= \mathrm{Var}\left[\sum_{\lambda\in\mathcal{C}_4^{(t)}}\delta_{\lambda_1}+\delta_{\lambda_2}\right] = \sum_{\lambda\in\mathcal{C}_4^{(t)}}\sum_{\gamma\in\mathcal{C}_4^{(t)}}\sum_{i\in\{1,2\}}\sum_{j\in\{1,2\}}\mathrm{Cov}\left[\delta_{\lambda_i},\delta_{\gamma_j}\right] \\
&= \sum_{\lambda\in\mathcal{C}_4^{(t)}}\left(\mathrm{Var}\left[\delta_{\lambda_1}\right]+\mathrm{Var}\left[\delta_{\lambda_2}\right]\right) + \sum_{\lambda\in\mathcal{C}_4^{(t)}}\left(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\lambda_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\lambda_1}\right]\right) \\
&+ \sum_{\substack{\lambda,\gamma\in\mathcal{C}_4^{(t)}\\ \lambda\neq\gamma}}\Big(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_2}\right]\Big) \\
&= \frac{|\mathcal{C}_4^{(t)}|}{2}\left(\frac{\tau^{(t)}}{M_\Delta}\prod_{i=0}^{3}\frac{t-1-i}{M_e-i}-1\right) + \sum_{\lambda\in\mathcal{C}_4^{(t)}}\left(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\lambda_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\lambda_1}\right]\right) \\
&+ \sum_{\substack{\lambda,\gamma\in\mathcal{C}_4^{(t)}\\ \lambda\neq\gamma}}\Big(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_2}\right]\Big) \qquad (8.5) \\
&\leq \frac{|\mathcal{C}_4^{(t)}|}{2}\left(\frac{\tau^{(t)}}{M_\Delta}c\left(\frac{t-1}{M_e}\right)^4-1\right) + \sum_{\lambda\in\mathcal{C}_4^{(t)}}\left(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\lambda_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\lambda_1}\right]\right) \\
&+ \sum_{\substack{\lambda\in\mathcal{C}_4^{(t)}\\ \lambda\neq\gamma}}\Big(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\gamma_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_1}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\gamma_2}\right]\Big)\ . \qquad (8.6)
\end{aligned}$$

We now proceed to analyze the various covariance terms appearing in (8.5). In the following we refer to

- The second summation in (8.5), $\sum_{\lambda\in\mathcal{C}_4^{(t)}}\left(\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\lambda_2}\right]+\mathrm{Cov}\left[\delta_{\lambda_2},\delta_{\lambda_1}\right]\right)$, concerns the sum of the covariances of pairs of random variables each corresponding to one of the two possible ways of detecting a 4-clique using TS4C$_1$. Let us consider one single element of the summation:

$$\mathrm{Cov}\left[\delta_{\lambda_1},\delta_{\lambda_2}\right] = \mathrm{E}\left[\delta_{\lambda_1}\delta_{\lambda_2}\right] - \mathrm{E}\left[\delta_{\lambda_1}\right]\mathrm{E}\left[\delta_{\lambda_2}\right] = \mathrm{E}\left[\delta_{\lambda_1}\delta_{\lambda_2}\right] - 1/4.$$

Let us now focus on $\mathrm{E}\left[\delta_{\lambda_1}\delta_{\lambda_2}\right]$, according to the definition of δ_{λ_1} and δ_{λ_2} we have:

$$\begin{aligned}
\mathrm{E}\left[\delta_{\lambda_1}\delta_{\lambda_2}\right] &= \frac{p_{\lambda_1}^{-1}p_{\lambda_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} \wedge \delta_{\lambda_2} = p_{\lambda_2}^{-1}\right) \\
&= \frac{p_{\lambda_1}^{-1}p_{\lambda_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\lambda_2} = p_{\lambda_2}^{-1}\right)\Pr\left(\delta_{\lambda_2} = p_{\lambda_2}^{-1}\right) \\
&\leq \frac{p_{\lambda_1}^{-1}p_{\lambda_2}^{-1}}{4}\Pr\left(\delta_{\lambda_2} = p_{\lambda_2}^{-1}\right) \\
&\leq \frac{p_{\lambda_1}^{-1}p_{\lambda_2}^{-1}}{4}p_{\lambda_2} \\
&\leq \frac{p_{\lambda_1}^{-1}}{4}.
\end{aligned}$$

We can therefore conclude:

$$\begin{aligned}
\sum_{\lambda \in \mathcal{C}_4^{(t)}} \left(\mathrm{Cov}\left[\delta_{\lambda_1}, \delta_{\lambda_2}\right] + \mathrm{Cov}\left[\delta_{\lambda_2}, \delta_{\lambda_1}\right]\right) &\leq \frac{|\mathcal{C}_4^{(t)}|}{2}\left(\frac{\tau^{(t)}}{M_\Delta}\prod_{i=0}^{3}\frac{t-1-i}{M_e-i} - 1\right) \\
&\leq \frac{|\mathcal{C}_4^{(t)}|}{2}\left(\frac{\tau^{(t)}}{M_\Delta}c\left(\frac{t-1}{M_e}\right)^4 - 1\right)
\end{aligned} \tag{8.7}$$

- The third summation in (8.5), includes the covariances of all $|\mathcal{C}_4^{(t)}|\left(2|\mathcal{C}_4^{(t)}| - 1\right)$ unordered pairs of random variables corresponding each to one of the two possible ways of counting distinct 4-cliques in $\mathcal{C}_4^{(t)}$. In order to provide a significant bound it is necessary to divide the possible pairs of 4-cliques depending on how many edges they share (if any). From Lemma 8.7 we have that any pair of 4-cliques λ and γ can share either one, three or no edges. In the remainder of our analysis we shall distinguishing three group of pairs of 4-cliques based on how many edges they share. In the following, we present, without loss of generality, bounds for $\mathrm{Cov}\left[\delta_{\lambda_1}, \delta_{\gamma_2}\right]$. The results steadily holds for the other possible combinations $\mathrm{Cov}\left[\delta_{\lambda_1}, \delta_{\gamma_1}\right]$, $\mathrm{Cov}\left[\delta_{\lambda_2}, \delta_{\gamma_1}\right]$, $\mathrm{Cov}\left[\delta_{\lambda_2}, \delta_{\gamma_2}\right]$, $\mathrm{Cov}\left[\delta_{\gamma_1}, \delta_{\lambda_1}\right]$, $\mathrm{Cov}\left[\delta_{\gamma_1}, \delta_{\lambda_2}\right]$, $\mathrm{Cov}\left[\delta_{\gamma_2}, \delta_{\lambda_1}\right]$, and $\mathrm{Cov}\left[\delta_{\gamma_2}, \delta_{\lambda_2}\right]$

  1. λ and γ do not share any edge:

$$\begin{aligned}
\mathrm{E}\left[\delta_{\lambda_1}\delta_{\gamma_2}\right] &= \frac{p_{\lambda_1}^{-1}p_{\gamma_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} \wedge \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right) \\
&= \frac{p_{\lambda_1}^{-1}p_{\gamma_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right)\Pr\left(\delta_{\lambda_2} = p_{\lambda_2}^{-1}\right)
\end{aligned}$$

     The term $\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right)$ denotes the probability of TS4C$_1$ observing λ using T_1 and edges e_3 and e_5 conditioned of the fact that γ was observed by the algorithm using T_3 and edges g_3 and g_5. Note that as λ and γ do not share any edge, no edge of γ will be used by TS4C$_1$ to detect λ. Rather, if any edge of γ is included in $\S_e$ or if T_3 is included in $\mathcal{S}_\Delta$, this would lessen the probability of TS4C$_1$ detecting λ using $T1, e_3$ and e_5 as some of space in $\mathcal{S}_e$ or $\mathcal{S}_\Delta$ may be occupied by edges or triangle sub-structures for

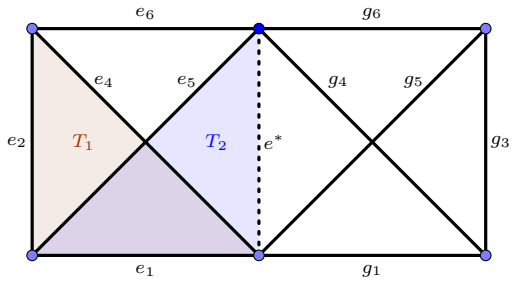


Figure 8.2: Cliques sharing one edge.

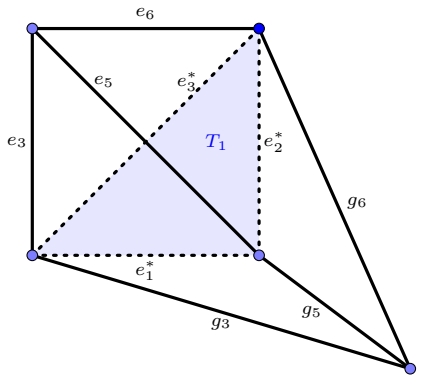


Figure 8.3: Cliques sharing three edges.

γ. Therefore we have $\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right) \leq \Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1}\right)$ and thus:

$$\begin{aligned}
\mathrm{E}\left[\delta_{\lambda_1}\delta_{\gamma_2}\right] &= \frac{p_{\lambda_1}^{-1}p_{\gamma_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right)\Pr\left(\delta_{\lambda_2} = p_{\lambda_2}^{-1}\right) \\
&\leq \frac{p_{\lambda_1}^{-1}p_{\gamma_2}^{-1}}{4}\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1}\right)\Pr\left(\delta_{\lambda_2} = p_{\lambda_2}^{-1}\right) \\
&\leq \frac{p_{\lambda_1}^{-1}p_{\gamma_2}^{-1}}{4}p_{\lambda_1}p_{\lambda_2} \\
&\leq \frac{1}{4}
\end{aligned}$$

We therefore have $\mathrm{Cov}\left[\delta_{\lambda_1}, \delta_{\gamma_2}\right] \leq 0$. Hence we can conclude that the contribution of the covariances of the pairs of random variables corresponding to 4-cliques that do not share any edge to the summation in (8.5) is less or equal to zero.

2. λ and γ share exactly one edge $e^* = \lambda \cap \gamma$ as shown in Figure 8.2 Let us consider the event $E* =$"$e^* \cap T_1 \cap E^{(t_{1,2,4}-1)} \in \mathcal{S}_e^{t_{1,2,4}}$, and $e^* \cap \{e_3, e_5\} \cap E^{(t_6-1)} \in S_e^{(t_6)}$". Clearly $\Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right) \leq \Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | E*\right)$. Recall from Lemma 8.3 that $\Pr\left(\delta_{\lambda_1} | E^*\right) = \Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*\right)\Pr\left(T_1 \in \mathcal{S}_\Delta^{(t_6)} | E*\right)\Pr\left(S(T_1) | E^*\right)$, where $S(T_1)$ denotes the event "T_1 is observed on the stream by TS4C$_1$". By applying the law of total probability e have that $\Pr\left(T_1 \in \mathcal{S}_\Delta^{(t_6)} | E*\right) \leq \Pr\left(T_1 \in \mathcal{S}_\Delta^{(t_6)}\right)$. The remaining two terms are influenced differently depending on whether $e^* \in T_1$ or $e^* \in \{e_3, e_5\}$

– if $e^* \in T_1$: we then have $\Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*\right) \leq \Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)}\right)$. This follows from the properties of the reservoir sampling scheme as the fact that the edge e^* is in $\mathcal{S}_e$ means that one unit of the available memory space required to hold e_3 or e_5 is occupied, at least for some time, by e^*. If e^* is the last edge of T_1 observed in the stream we then have:

$$\begin{aligned}\Pr\left(S(T_1)|E^*\right) &= \Pr\left(\{e_1, e_2, e_4\} \setminus \{e^*\} \in \mathcal{S}_e^{(}t_1, 2, 4)|E^*\right) \\ &= \Pr\left(\{e_1, e_2, e_4\} \setminus \{e^*\} \in \mathcal{S}_e^{(}t_1, 2, 4)\right) \\ &\leq \frac{M_e}{t_{1,2,4} - 1} \frac{M_e - 1}{t_{1,2,4} - 2}\end{aligned}$$

Suppose instead that $e*$ is *not* the last edge of T_1. Assume further , without loss of generality, that $t_2 > \max\{t_1, t_4\}$. TS4C$_1$ observes T_1 iff $\{e_1, e_4\} \in \mathcal{S}_e^{(}t_2)$. As in E^* we assume that once observed e^* is *always* maintained in $\mathcal{S}_e$ until $t_{1,2,4}$ we have:

$$\begin{aligned}\Pr\left(S(T_1)|E^*\right) &= \Pr\left(\{e_1, e_4\} \setminus \{e^*\} \in \mathcal{S}_e^{(}t_1, 2, 4)|E^*\right) \\ &\leq \frac{M_e - 1}{t_{1,2,4} - 2}\end{aligned}$$

– if $e^* \in \{e_3, e_5\}$: we then have $\Pr\left(S(T_1)|E^*\right) \leq \Pr\left(S(T_1)\right)$. This follows from the properties of the reservoir sampling scheme as the fact that the edge e^* is in $\mathcal{S}_e$ means that one unit of the available memory space required to hold the first two edges of T_1 until $t_{1,2,4}$ is occupied, at least for some time, by e^*. Further, using Lemma 8.3 we have:

  * if $t_6 \leq M_e$, then $\Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*\right) = 1$
  * if $\min\{t_3, t_5\} > t_{1,2,4}$, then $\Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*\right) \geq \frac{M_e}{t_6 - 1}$,
  * if $\max\{t_3, t_5\} > t_{1,2,4} > \min\{t_3, t_5\}$, then $\Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*\right) \geq \max\{\frac{M_e - 1}{t_6 - 2}, \frac{M_e - 2}{t_{1,2,4} - 3} \frac{t_{1,2,4} - 1}{t_6 - 1}\}$,
  * otherwise $\Pr\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*\right) \geq \frac{M_e - 2}{t_{1,2,4} - 3} \frac{t_{1,2,4} - 1}{t_6 - 1}$.

Putting together these various results we have that $p_\lambda^{(-1)} \Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right) \leq p_\lambda^{(-1)} \Pr\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | E*\right) \leq c \frac{t_6 - 1}{M_e}$ with $c = s$. Hence we have $\mathrm{E}\left[\delta_{\lambda_1} \delta_{\gamma_2}\right] \leq \frac{c}{4} \frac{t_6 - 1}{M_e} \leq \frac{c}{4} \frac{t - 1}{M_e}$. We can thus bound the contribution to the third component of (8.5) given by the pairs of random variables corresponding to 4-cliques that share one edge as:

$$2a^{(t)} \left(c \frac{t - 1}{M_e} - 1\right), \tag{8.8}$$

where $a^{(t)}$ denotes the number of unordered pairs of 4-cliques which share one edge in $G^{(t)}$.

3. λ and γ share three edges $\{e_1^*, e_2^*, e_3^*\}$ which form a triangle sub-structure for both λ and γ. Let us refer to Figure 8.3, without loss of generality let T_1 denote the triangle shared between the two cliques. We distinguish the kind of pairs for the random variables δ_{λ_i} and δ_{γ_j} cases:

– $\delta_{\lambda_i} = p_{\lambda_i}^{-1}$ if $T_1 \in \mathcal{S}_\Delta^{t_6-1} \wedge \{e_3, e_5\} \subseteq \mathcal{S}_e^{t_6-1}$ and $\delta_{\gamma_j} = p_{\gamma_j}^{-1}$ if $T_1 \in \mathcal{S}_\Delta^{t_\gamma-1} \wedge \{g_3, g_5\} \subseteq \mathcal{S}_e^{t_\gamma-1}$, where t_γ denotes the time step at which the last edge of γ is observed. This is the case for which the random variables δ_{λ_i} and δ_{γ_j} corresponds to TS4C$_1$ observing λ and γ using the *shared triangle* T_1. Let us consider the event $E* =$"$T_1 \in \mathcal{S}_\Delta^{(t_6)}$. Clearly $\Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}\right) \leq \Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*\right)$. In this case we have $\Pr\left(T_1 \in \mathcal{S}^{(t_6)} | E^*\right) \leq 1$, while $\Pr\left(\{e_3, e_5\} \in \mathcal{S}_e^{(t_6)} | E^*\right) \leq Prob\{e_3, e_5\} \in \mathcal{S}_e^{(t_6)}$. This second fact follows from the properties of the reservoir sampling scheme as the fact that the edges e_1^*, e_2^* and e_3^* are in $\mathcal{S}_e$ *at least* for the time required for T_1 to be observed, means that at least two unit of the available memory space required to hold the edges e_3, e_5 are occupied, at least for some time. Putting together these various results we have that $p_{\lambda_i}^{(-1)} Prob\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1} \leq p_{\lambda_i}^{(-1)} \Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*\right) \leq c\left(\frac{t_6-1}{M_e}\right)^2 \frac{\tau^{(t)}}{\mathcal{S}_\Delta}$. Hence we have $\mathrm{E}\left[\delta_{\lambda_i}\delta_{\gamma_j}\right] \leq \frac{c}{4}\left(\frac{t_6-1}{M_e}\right)^2 \frac{\tau^{(t)}}{\mathcal{S}_\Delta}$ and $\mathrm{Cov}\left[\delta_{\lambda_i}, \delta_{\gamma_j}\right] \leq \frac{c}{4}\left(\frac{t_6-1}{M_e}\right)^2 \frac{\tau^{(t)}}{\mathcal{S}_\Delta} - \frac{1}{4}$.

– in all the remaining case, then the random variables δ_{λ_i} and δ_{γ_j} corresponds to FourEstnot observing λ and γ using the shared triangle T_1 for both of them. Let T^* denote the triangle sub-structure used by TS4C$_1$ to count λ with respect to δ_{λ_i}. Let us consider the event $E^* =$"$\{e_1^*, e_2^*, e_3^*\} \cap T_1 \cap E^{(t_{1,2,4}-1)} \in \mathcal{S}_e^{t_{1,2,4}}$, $T_1 \in \mathcal{S}_\Delta^{(t_6)}$ unless one of its edges is the last edge of T^* observed on the stream, and $\{e_1^*, e_2^*, e_3^*\} \cap \{e_3, e_5\} \cap E^{(t_6-1)} \in S_e^{(t_6)}$ if $e^* \in \{e_3, e_5\}$", where $E^{(t)}$ denotes the set of edges observed up until time t included. Clearly $\Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}\right) \leq \Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*\right)$. Note that in this case $|\{e_1^*, e_2^*, e_3^*\} \cap T_1| + |\{e_1^*, e_2^*, e_3^*\} \cap \{e_3, e_5\}|$. By analyzing $\Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*\right)$ in this case using similar steps as the ones described for the other sub-cases we have $p_{\lambda_i}^{(-1)} \Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}\right) \leq p_{\lambda_i}^{(-1)} \Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*\right) \leq c\left(\frac{t_6-1}{M_e}\right)^3$. Hence we have $\mathrm{E}\left[\delta_{\lambda_i}\delta_{\gamma_j}\right] \leq \frac{c}{4}\left(\frac{t_6-1}{M_e}\right)^3$ and $\mathrm{Cov}\left[\delta_{\lambda_i}, \delta_{\gamma_j}\right] \leq \frac{c}{4}\left(\frac{t_6-1}{M_e}\right)^3 - \frac{1}{4}$.

We can thus bound the contribution to the third component of (8.5) given by the pairs of random variables corresponding to 4-cliques that share three edges as:

$$2b^{(t)}\left(c\left(\frac{t-1}{M_e}\right)^2\left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta} + \frac{3}{4}\frac{t-1}{M_e}\right) - 1\right), \tag{8.9}$$

where $b^{(t)}$ denotes the number of unordered pairs of 4-cliques which share three edges in $G^{(t)}$.

The Theorem follows by combining (8.7), (8.8) and (8.9) in (8.5). □

Memory partition across layers In most practical scenario we assume that a certain amount of total available memory M is available for algorithm TS4C$_1$. A natural question that arises is what is the best way of spitting the available memory between $\mathcal{S}_e$ and $\mathcal{S}_\Delta$. While different heuristics are possible, in our work we chose to assign the available space in such a way that the dominant first term of upper bound of TS4C$_1$ in Theorem 8.6 is minimized. For $0 < \alpha < 1$ let $M_e = \alpha M$ and

$M_\Delta = (1-\alpha)\,M$. Then we have that $|\mathcal{C}_4^{(t)}|\left(c\left(\frac{t-1}{\alpha M}\right)^4\left(\frac{\tau^{(t)}-1}{(1-\alpha)M}\right)-1\right)$ is minimized for $\alpha = 4/5$. This convenient splitting rule works well in most cases, and is used in most of the paper. In Section 8.7, we discuss a more sophisticated *adaptive* dynamic allocation of the available memory among the two sample tiers, named ATS4C. We present experimental results for this method in Section 8.8.2.

Concentration bound Based on the bound on the variance in Theorem 8.6, we now show a concentration bound for the estimator $\varkappa^{(t)}$ returned by $\mathrm{TS4C}_1$. The proof of Theorem 8.8 is based on the application of Chebyshev's inequality [164, Thm. 3.6].

Theorem 8.8. *Let $t > \min\{M_e, t_\Delta\}$ and assume $|\mathcal{C}_4^{(t)}| > 0$. Let $a^{(t)}$ (resp., $b^{(t)}$) denote the number of unordered pairs of 4-cliques which share one edge (resp., three edges) in $G^{(t)}$, and $c \geq \frac{M_e^3}{(M_e-1)(M_e-2)(M_e-3)}$. Further, let $M_e = \alpha M$ (resp., $M_e = (1-\alpha)\,M$), for $\alpha \in (0,1)$. For any $\varepsilon, \delta \in (0,1)$, if*

$$M > \alpha^{-1}\max\Bigg\{\sqrt[5]{\frac{\alpha}{1-\alpha}\frac{3c(t-1)^4(\tau^{(t)})}{\delta\varepsilon^2|\mathcal{C}_4^{(t)}|}}, \frac{6ca^{(t)}(t-1)}{\delta\varepsilon^2|\mathcal{C}_4^{(t)}|^2}, \sqrt[3]{\frac{3cb^{(t)}c\,(t-1)^2\left(\alpha\tau^{(t)}+3(1-\alpha)(t-1)\right)}{2(1-\alpha)\delta\varepsilon^2|\mathcal{C}_4^{(t)}|^2}}\Bigg\},$$

then the estimator $\varkappa^{(t)}$ returned by $\mathrm{TS4C}_1$ satisfies:

$$Pr\left(|\varkappa^{(t)} - |\mathcal{C}_4^{(t)}| < \varepsilon|\mathcal{C}_4^{(t)}|\right) > 1-\delta.$$

Proof. By Chebyshev's inequality it is sufficient to prove that

$$\frac{\mathrm{Var}\left[\varkappa^{(t)}\right]}{\varepsilon^2|\mathcal{C}_4^{(t)}|^2} < \delta\ .$$

From Theorem 8.6, we can write:

$$\begin{aligned}
\frac{\mathrm{Var}\left[\tau^{(t)}\right]}{\varepsilon^2|\mathcal{C}_4^{(t)}|^2} &\leq \frac{1}{\varepsilon^2|\mathcal{C}_4^{(t)}|}\Bigg(|\mathcal{C}_4^{(t)}|\left(c\left(\frac{t-1}{M_e}\right)^4\left(\frac{\tau^{(t)}}{M_\Delta}\right)-1\right)+2a^{(t)}\left(c\frac{t-1}{M_e}-1\right)\\
&\quad+2b^{(t)}\left(c\left(\frac{t-1}{M_e}\right)^2\left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta}+\frac{3}{4}\frac{t-1}{M_e}\right)-1\right)\Bigg)\\
&< \frac{1}{\varepsilon^2|\mathcal{C}_4^{(t)}|}\left(|\mathcal{C}_4^{(t)}|c\left(\frac{t-1}{M_e}\right)^4\left(\frac{\tau^{(t)}}{M_\Delta}\right)+2a^{(t)}\left(c\frac{t-1}{M_e}\right)+2b^{(t)}c\left(\frac{t-1}{M_e}\right)^2\left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta}+\frac{3}{4}\frac{t-1}{M_e}\right)\right)\\
&= \frac{1}{\varepsilon^2|\mathcal{C}_4^{(t)}|}\left(|\mathcal{C}_4^{(t)}|c\frac{(t-1)^4\tau^{(t)}}{\alpha^4(1-\alpha)M^5}+2a^{(t)}c\left(\frac{t-1}{\alpha M}\right)+2b^{(t)}c\left(\frac{t-1}{\alpha M}\right)^2\frac{\alpha\tau^{(t)}+3(1-\alpha)(t-1)}{4\alpha(1-\alpha)M}\right)\\
&= \frac{1}{\varepsilon^2|\mathcal{C}_4^{(t)}|}\left(|\mathcal{C}_4^{(t)}|c\frac{(t-1)^4\tau^{(t)}}{\alpha^4(1-\alpha)M^5}+2a^{(t)}c\left(\frac{t-1}{\alpha M}\right)+2b^{(t)}c\frac{(t-1)^2\left(\alpha\tau^{(t)}+3(1-\alpha)(t-1)\right)}{4\alpha^3(1-\alpha)M^3}\right)\ .
\end{aligned}$$

Hence it is sufficient to impose the following three conditions:

Condition 1

$$\begin{aligned}
\frac{\delta}{3} &> \frac{1}{\varepsilon^2|\mathcal{C}_4^{(t)}|^2}|\mathcal{C}_4^{(t)}|c\frac{(t-1)^4\tau^{(t)}}{\alpha^4(1-\alpha)M^5} \qquad (8.10)\\
&> \frac{\alpha}{1-\alpha}\frac{c\,(t-1)^4\tau^{(t)}}{\varepsilon^2|\mathcal{C}_4^{(t)}|\alpha^5M^5},
\end{aligned}$$

which is verified for:

$$M > \alpha^{-1} \sqrt[5]{\frac{3\alpha}{1-\alpha} \frac{c\,(t-1)^4\,\tau^{(t)}}{\delta\varepsilon^2|\mathcal{C}_4^{(t)}|}}.$$

Condition 2

$$\frac{\delta}{3} > \frac{1}{\varepsilon^2|\mathcal{C}_4^{(t)}|^2} 2a^{(t)} c \left(\frac{t-1}{\alpha M}\right),$$

which is verified for:

$$M > \alpha^{-1} \frac{6a^{(t)} c\,(t-1)}{\delta\varepsilon^2|\mathcal{C}_4^{(t)}|^2}.$$

Condition 3

$$\frac{\delta}{3} > \frac{1}{\varepsilon^2|\mathcal{C}_4^{(t)}|^2} 2b^{(t)} c \frac{(t-1)^2 \left(\alpha\tau^{(t)} + 3\,(1-\alpha)\,(t-1)\right)}{4\alpha^3(1-\alpha)M^3}, \tag{8.11}$$

which is verified for:

$$M > \alpha^{-1} \sqrt[3]{\frac{3b^{(t)} c\,(t-1)^2 \left(\alpha\tau^{(t)} + 3\,(1-\alpha)\,(t-1)\right)}{2\delta\varepsilon^2(1-\alpha)|\mathcal{C}_4^{(t)}|^2}}.$$

The theorem follows. □

Note that for $t \leq \min\{M_e, t_\Delta\}$ all the edges (resp., triangles) observed up to time t can be maintained in $\mathcal{S}_e$ (resp., $\mathcal{S}_\Delta$), and, hence, we have $\varkappa^{(t)} = |\mathcal{C}_k^{(t)}|$ and $\mathrm{Var}\left[\varkappa^{(t)}\right] = 0$.

Theorem 8.8, provides a bound on the relative ϵ-approximation of $|\mathcal{C}_4^{(t)}|$ provided by TS4C$_1$. In particular, Theorem 8.8 suggest that while the variance of the estimator is bound to increase as $|\mathcal{C}_4^{(t)}|$ increases (as stated in Theorem 8.6), for a given value δ, the size of available sample memory M required to achieve an ϵ-approximation for $|\mathcal{C}_4^{(t)}|$ with probability at least $1-\delta$ decreases for higher values of $|\mathcal{C}_4^{(t)}|$.

8.4.2 Algorithm TS4C$_2$

While the version of TS4C$_1$ presented in Algorithm 15 detects 4-cliques by using one triangle sub-structure form $\mathcal{S}_\Delta$ and two edges from $\mathcal{S}_e$, it is possible to use different sub-structures to achieve the same goal. We now present a variation of TS4C$_1$, called TS4C$_2$, which detects a 4-clique when the current observed edge completes a 4-clique using *two triangles* currently in $\mathcal{S}_\Delta$. Towards improving the presentation, the proofs of all the technical results discussed in this section are deferred to Section 8.5.

TS4C$_1$ and TS4C$_2$ differ *only* in the *Estimation update* step, implemented by the function UPDATE4CLIQUES which determinates how the occurrences of 4-cliques are detected and the estimators are correspondingly updated. The pseudocode for TS4C$_2$ is presented in Algorithm 16:

- Whenever a new edge $e_t = (u, v)$ is observed on the stream at time T, TS4C$_2$ invokes the function UPDATE 4-CLIQUES (Algorithm 16) which verifies whether in triangle reservoir $\mathcal{S}_\Delta$ there exists any (unordered) pair of triangles $\{T_1, T_2\}$ such that $T_1 = \{u, w, z\}$ and $T_2 = \{v, w, z\}$, for $w, z \in V^{(t)}$. When such conditions are meet, we say that a 4-clique $\{u, v, w, z\}$ is

ALGORITHM 16 TS4C$_2$ - Tiered Sampling for 4-Clique counting using 2 triangle sub-structures

function UPDATE4CLIQUES($(u, v), t$)
  for each $(u, w, z) \in \mathcal{S}_\Delta \wedge (v, w, z) \in \mathcal{S}_\Delta$ **do** ▷ For each 4-clique (u,v,w,z) formed by two triangles (u,w,z) and (v,w,z)
    $p \leftarrow$ PROBCLIQUE($(u, w, z), (v, w, z)$) ▷ Calculate the probability of observing 4-clique (u,v,w,z)
    $\sigma \leftarrow \sigma + p^{-1}$ ▷ Update the 4-clique estimator

"*observed on the stream*" by TS4C$_2$. The algorithm then uses PROBCLIQUE to compute the *exact* probability p of the observation based on the timestamps of all its edges according to the results of Lemma 8.10. The estimator $\varkappa$ is then increased by p^{-1}.

- TS4C$_2$ then proceeds in updating the triangle sample $\mathcal{S}_\Delta$ and the edge sample $\mathcal{S}_e$ follow-wing the same steps as TS4C$_1$ as described in Section 8.4.1.

The difference in the way 4-cliques are detected by TS4C$_2$ corresponds to a difference in the probability of such detections. Before introducing the lemma for calculating the probability of detecting a 4-clique using TS4C$_2$, we introduce the following technical Lemma 8.9 which states that each 4-clique can be observed just once using TS4C$_1$. In order to simplify the presentation, in the following we use the following notation:

$$t^M_{1,2,\ldots,i} \triangleq \max\{t_1, t_2, \ldots, t_i\}$$
$$t^m_{1,2,\ldots,i} \triangleq \min\{t_1, t_2, \ldots, t_i\}$$

Lemma 8.9. *Let $\lambda \in \mathcal{C}_4^{(t)}$ with $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ using Figure 8.1 as reference. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. λ can be observed by* TS4C$_2$ *at time t_6 only by a combination of two triangles $T_1 = \{e_1, e_2, e_4\}$ and $T_2 = \{e_1, e_3, e_5\}$.*

Proof. TS4C$_2$ can detect λ only when its last edge is observed on the stream (hence, t_6). When e_6 is observed, the algorithm first evaluates whether there are two triangles in $\mathcal{S}_\Delta^{(t_6)}$, that share two endpoints and the other endpoints are u and v respectively. Since at this step (i.e., the execution of function UPDATE4CLIQUES) the triangle sample is jet to be updated based on the observation of e_6, the only triangle sub-structures of λ which may have been observed on the stream, and thus included in $\mathcal{S}_\Delta^{(t_6)}$ are $T_1 = \{e_1, e_2, e_4\}$ and $T_2 = \{e_1, e_3, e_5\}$. λ is thus observed if and only if both of the triangles T_1 and T_2 are found in $\mathcal{S}_\Delta^{(t_6)}$. □

Lemma 8.9, marks an important difference between TS4C$_2$ and TS4C$_1$ as for the latter there are multiple ways of detecting the same 4-clique as discussed in Section 8.4. Such difference impacts the analysis of the probability of detecting a 4-clique using TS4C$_2$:

Lemma 8.10. *Let $\lambda \in \mathcal{C}_4^{(t)}$ with $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ using Figure 8.1 as reference. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively)*

and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. *The probability* p_λ *of* λ *being observed on the stream by* TS4C$_2$, *is computed by* ProbClique *as:*

$$p_\lambda = \min\{1, \frac{M_e}{t^M_{1,3,5}-1}\frac{M_e-1}{t^M_{1,3,5}-2}\}\min\{1, \frac{M_\Delta}{\tau^{(t_6)}}\frac{M_\Delta-1}{\tau^{(t_6)}-1}\}p'$$

where:

$$p' = \begin{cases} 1, & if\ t^M_{1,2,4} \leq M_e \\ \frac{M_e-2}{t_1-3}\frac{M_e-3}{t_1-4}, & if\ t_1 > t^M_{2,3,4,5} \\ \frac{M_e-2}{t_1-3}\frac{M_e-3}{t_1-4}, & if\ t^M_{3,5} > t_1 > \max\{t^m_{3,5}, t_2, t_4\} \\ \frac{M_e-2}{t^M_{2,4}-3}, & if\ t^M_{3,5} > t^M_{2,4} > \max\{t^m_{3,5}, t^m_{2,4}, t_1\} \\ \frac{M_e}{t_1-1}\frac{M_e-1}{t_1-2}, & if\ t^m_{3,5} > t_1 > t^M_{2,4} \\ \frac{M_e-1}{t^M_{2,4}-2} & if\ t^m_{3,5} > t^M_{2,4} > t_1 \\ \frac{M_e-1}{t^M_{2,4}-1}\frac{M_e-2}{t_1-3}\frac{t_1-1}{t_2-1}, & if\ t^M_{2,4} > t_1 > \max\{M_e, t^m_{2,4}, t^M_{3,5}\} \\ \frac{M_e-1}{t^M_{2,4}-1}\frac{M_e}{t_2-1} & if\ t^M_{2,4} > M_e > t_1 > \max\{t^m_{2,4}, t^M_{3,5}\} \\ \frac{t_3-1}{t^M_{2,4}-1}\frac{t_3-2}{t^M_{2,4}-2}\frac{M_e-2}{t^M_{3,5}-3}, & if\ t^M_{2,4} > t^M_{3,5} > \max\{M_e, t^m_{3,5}, t^m_{2,4}, t_1\} \\ \frac{M_e}{t^M_{2,4}-1}\frac{M_e-1}{t^M_{2,4}-2}, & if\ t^M_{2,4} > M_e > t^M_{3,5} > \max\{t^m_{3,5}, t^m_{2,4}, t_1\} \\ \frac{M_e}{t^M_{2,4}-1}\frac{M_e-1}{t^M_{2,4}-2}, & if\ t^m_{2,4} > t_1 > t^M_{3,5} \\ \frac{M_e-1}{t^M_{2,4}-2}\frac{t^M_{3,5}-1}{t^M_{2,4}-1}, & if\ t^m_{2,4} > t^M_{3,5} > \max\{M, t_1\} \\ \frac{M_e-1}{t^M_{2,4}-2}\frac{M_e}{t^M_{2,4}-1}, & if\ t^m_{2,4} > M_e > t^M_{3,5} > t_1 \end{cases}$$

The analysis presents several complications due to the interplay of the probabilities of observing each of the two triangles that share an edge. For a detailed proof of Lemma 8.11 and of the other results in this section (Theorems 8.12,8.13, 8.14), we refer the reader to Appendix 8.5. The following corollary presents a *lower bound* to the probability of a 4-clique being observed by TS4C$_2$. While rather loose, especially for 4-cliques that are detected towards the beginning of the edge stream, it provides a useful intuition for the analysis of the variance of the estimator presented in Theorem 8.13.

Corollary 8.11. *Let* $\lambda \in \mathcal{C}^{(t)}_4$ *and let* p_λ *denote the probability of* λ *being observed on the stream by* TS4C$_2$. *We have:*

$$p_\lambda \leq \left(\frac{M_e}{t-1}\right)^4 \left(\frac{M_\Delta}{\tau^{(t)}}\right)^2.$$

Analysis of TS4C$_2$ estimator

Theorem 8.12. *The estimator* $\varkappa$ *returned by* TS4C$_2$ *is unbiased, that is* $\varkappa^{(t)} = |\mathcal{C}^{(t)}_k|$ *if* $t \leq \min\{M_e, t_\Delta\}$. *Further,* $\mathbb{E}\left[\varkappa^{(t)}\right] = |\mathcal{C}^{(t)}_k|$ *if* $t \leq \min\{M_e, t_\Delta\}$.

The proof of 8.12 closely follows the steps of the proof of 8.5. The main differences are given by the fact that there is one unique possible way according to which 4-cliques are observed by TS4C$_2$ (Lemma 8.15), and the probability of such event is given by 8.9.

The analysis of the variance of TS4C$_2$ presents considerable differences with respect to the analysis of the variance of TS4C$_1$, which are due to the different strategies that the two algorithms use to detect 4-cliques. The key component of the analysis depends on the analysis of the covariance of all the possible pairs of the binary random variables that are each associated with the detection of one of the 4-cliques. As stated in Lemma 8.9, for TS4C$_2$ there is just a single possible way according to which a 4-clique can be detected. Thus, the number and the distribution of the random variables considered in the analysis of the variance of TS4C$_2$ are clearly different from those of the random variables considered in the previous section for the analysis of the variance of TS4C$_1$.

Theorem 8.13. *For any time $t > \min\{M_e, t_\Delta\}$, the estimator $\varkappa$ returned by* TS4C$_2$ *satisfies*

$$Var\left[\varkappa^{(t)}\right] \leq |\mathcal{C}_4^{(t)}| \left(c \left(\frac{t-1}{M_e}\right)^4 \left(\frac{\tau^{(t)}}{M_\Delta}\right)^2 - 1 \right) + 2a^{(t)} \left(c\frac{t-1}{M_e} - 1 \right) + 2b^{(t)} \left(c \left(\frac{t-1}{M_e}\right)^2 \left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta} + \frac{3}{4}\frac{t-1}{M_e}\right) - 1 \right), \tag{8.12}$$

where $a^{(t)}$ (resp., $b^{(t)}$) denotes the number of unordered pairs of 4-cliques which share one edge (resp., three edges) in $G^{(t)}$, and $c \geq \frac{M_e^3}{(M_e-1)(M_e-2)(M_e-3)}$.

Note that the first, and dominant, term of (8.12) is given by the total number of 4-cliques in $G^{(t)}$ (i.e., $|\mathcal{C}_4^{(t)}|$) multiplied by $\left(c \left(\frac{t-1}{M_e}\right)^4 \left(\frac{\tau^{(t)}}{M_\Delta}\right)^2 - 1 \right)$ which corresponds *approximately* to $c(p_\lambda)^{-1}$, where p_λ denotes the *lower bound* to the probability of TS4C$_2$ detecting a 4-clique as provided by Corollary 8.11. This suggests a natural relation between these quantities: as the probability of detecting a 4-clique decreases (resp., the total number of 4-cliques increases) the variance increases accordingly. The bound on the variance of TS4C$_2$ is similar to the corresponding result for TS4C$_1$ in Theorem 8.13. While it appears that the leading term of the variance for TS4C$_2$ is higher than that those of TS4C$_2$ by a $\frac{\tau^{(t)}}{M_\Delta}$ factor, it should be noted that this is mostly a result of the simplification of the analysis which is, however, required to achieve a closed form expression for the variance. In our experimental analysis on random (Section 8.6.3) and real-world graphs (Section 8.8) we observe that TS4C$_1$ and TS4C$_2$ generally exhibit comparable performance in terms of the quality of the produced estimated and their variance.

Memory partition: Following the same criterion discussed in Section 8.4.1, we use $|M_e| = 2M/3$ and $|M_\Delta| = M/3$ as a general rule for assigning the available memory space between the two sample levels. Given the fact that there is a much higher emphasis on the role and the use of the triangle sub-patterns in TS4C$_2$ compared to TS4C$_1$, it should not be surprising that the preferred memory split for the former assigned a higher fraction of the available memory space to the triangle sample compared to the latter. We use this assignment in the remainder of the paper and for the experimental evaluation of the performance of our algorithm.

Concentration bound: Based on the bound on the variance in Theorem 8.13, we now show a concentration bound for the estimator $\varkappa^{(t)}$ returned by TS4C$_2$ based on the application of Chebyshev's

inequality [164, Thm. 3.6].

Theorem 8.14. *Let $t > \min\{M_e, t_\Delta\}$ and assume $|\mathcal{C}_4^{(t)}| > 0$. Let $a^{(t)}$ (resp., $b^{(t)}$) denote the number of unordered pairs of 4-cliques which share one edge (resp., three edges) in $G^{(t)}$, and $c \geq \frac{M_e^3}{(M_e-1)(M_e-2)(M_e-3)}$. Further, let $M_e = \alpha M$ (resp., $M_e = (1-\alpha)M$), for $\alpha \in (0,1)$. For any $\varepsilon, \delta \in (0,1)$, if*

$$M > \alpha^{-1} \max\left\{ \sqrt[3]{\frac{\alpha}{1-\alpha}\frac{3c(t-1)^2(\tau^{(t)})^2}{\delta\varepsilon^2|\mathcal{C}_4^{(t)}|}}, \frac{6ca^{(t)}(t-1)}{\delta\varepsilon^2|\mathcal{C}_4^{(t)}|^2}, \sqrt[3]{\frac{6b^{(t)}c\,(t-1)^2\left(\alpha\tau^{(t)} + 3(1-\alpha)(t-1)\right)}{4(1-\alpha)\delta\varepsilon^2|\mathcal{C}_4^{(t)}|^2}} \right\}$$

then the estimator $\varkappa^{(t)}$ returned by TS4C_2 *satisfies*

$$Pr\left(|\varkappa^{(t)} - |\mathcal{C}_4^{(t)}| < \varepsilon|\mathcal{C}_4^{(t)}|\right) > 1 - \delta.$$

As for Theorem 8.8, note that for $t \leq \min\{M_e, t_\Delta\}$ all the edges (resp., triangles) observed up to time t can be maintained in $\mathcal{S}_e$ (resp., $\mathcal{S}_\Delta$), and, hence, we have $\varkappa^{(t)} = |\mathcal{C}_k^{(t)}|$ and $\text{Var}\left[\varkappa^{(t)}\right] = 0$. The proof for Theorem 8.14 follows steps analogous to those discussed for Theorem 8.8.

Although the difference between TS4C_1 and TS4C_2 may appear of minor interest, our experimental analysis shows that it can lead to significantly different performances depending on the properties of the graph $G^{(t)}$. Intuitively, TS4C_2 emphasizes the importance of the triangle substructures compared to TS4C_1, thus resulting in better performance when the input graph is very sparse with low number of occurrences of 3 and 4-cliques.

8.5 Proofs of technical results regarding TS4C_2

In this section we present proofs for TS4C_2 algorithm discussed in Section 8.4.2.

Proof of Theorem 8.10. Let us define the event $E_\lambda = \lambda$ *is observed* on the stream by TS4C_2 using triangle $T_1 = \{e_1, e_2, e_4\}$ and triangle $T_2 = \{e_1, e_3, e_5\}$. Given the definition of TS4C_2 we have:

$$E_\lambda = E_{T_1} \wedge E_{T_2},$$

and hence:

$$p_\lambda = \Pr(E_\lambda) = \Pr(E_{T_1} \wedge E_{T_2}) = \Pr\left(E_{T_1}|E_{T_2}\right)\Pr(E_{T_2}).$$

In oder to study $\Pr(E_{T_2})$ we shall introduce event $E_{S(T_2)}$ ="*triangle T_2 is observed on the stream by* TS4C_2". From the definition of TS4C_2 we know that T_2 is observed on the stream iff when the last edge of T_2 is observed on the stream at $\max\{t_1, t_3, t_5\}$ the remaining to edges are in the edge sample. Applying Bayes's rule of total probability we have:

$$\Pr(E_{T_2}) = \Pr\left(E_{T_2}|E_{S(T_2)}\right)\Pr\left(E_{S(T_2)}\right).$$

and thus:

$$p_\lambda = \Pr\left(E_{T_1}|E_{T_2}\right)\Pr\left(E_{T_2}|E_{S(T_2)}\right)\Pr\left(E_{S(T_2)}\right). \tag{8.13}$$

In order for T_2 to be observed by $TS4c_2$ it is required that wen the last edge of T_2 is observed on the stream at $t_{1,3,5}^M$ its two remaining edges are kept in S_e. Lemma 7.1 we have:

$$\Pr\left(E_{S(T_2)}\right) = \frac{M_e}{t_{1,3,5}-1}\frac{M_e-1}{t_{1,3,5}-2}, \tag{8.14}$$

and:

$$\Pr\left(E_{T_2}|E_{S(T_2)}\right) = \frac{M_e}{\tau^{t_6}}. \tag{8.15}$$

Let us now consider $\Pr\left(E_{T_1}|E_{T_2}\right)$. In order for T_1 to be found in $\mathcal{S}_\Delta$ at t_6 it is necessary for T_1 to have been observed by TS4C$_2$. We thus have that $\Pr\left(E_{T_1}|E_{T_2}\right) = \Pr\left(E_{T_1}|E_{S(T_1)}, E_{T_2}\right)\Pr\left(E_{S(T_1)}|E_{T_2}\right)$

$$\Pr\left(E_{T_1}|E_{S(T_1)}\right) = \frac{M_\Delta - 1}{\tau^{t_6} - 1} \tag{8.16}$$

Let us now consider $\Pr\left(E_{S(T_1)}|E_{T_2}\right)$. while the content of $\mathcal{S}_\Delta$ itself does not influence the content of $\mathcal{S}_e$, the fact that T_2 is maintained in $\mathcal{S}_\Delta$ at t_6 implies that it has been observed on the stream at a previous time. We thus have $\Pr\left(E_{S(T_1)}|E_{T_2}\right) = \Pr\left(E_{S(T_1)}|E_{S(T_2)}\right)$. In order to study $p' = \Pr\left(E_{S(T_1)}|E_{S(T_2)}\right)$ it is necessary to distinguish the possible (5!) different arrival orders for edges e_1,e_2, e_3, e_4 and e_5. An efficient analysis we however reduce the number of cases to be considered to thirteen.

- $t^M_{1,2,4} \leq M_e$: in this case all edges of T_1 are observed on the stream before M_e so they are deterministically inserted in $\mathcal{S}_e$ and thus $p' = 1$.

- $t_1 > t^M_{2,3,4,5}$: in this case the edge e_1 that is shared by T_1 and T_2 is observed after e_2, e_3, e_4, e_5.

$$\begin{aligned} p' &= P(\{e_2, e_4\} \in {\mathcal{S}_e}^{(t_1)} | \{e_3, e_5\} \in {\mathcal{S}_e}^{(t_1)}) \\ &= P(e_2 \in \mathcal{S}_e(t_1) | \{e_3, e_4, e_5\} \in {\mathcal{S}_e}^{(t_1)}) \cdot P(e_4 \in {\mathcal{S}_e}^{(t_1)} | \{e_3, e_5\} \in {\mathcal{S}_e}^{(t_1)}) \\ &= min(1, \frac{M-3}{t_1 - 4}) \cdot min(1, \frac{M-2}{t_1 - 3}) \end{aligned}$$

- $t^M_{3,5} > t_1 > \max\{t^m_{3,5}, t_2, t_4\}$: in this case only one of the edges e_3,e_5 is observed after e_1, which is observed after all the remaining edges. Here we consider the case when e_3 is the edge to be observed last. The same procedure follows for e_5 as well.

$$\begin{aligned} p' &= P(\{e_2, e_4\} \in {\mathcal{S}_e}^{(t_1)} | e_5 \in {\mathcal{S}_e}^{(t_1)}) \\ &= P(e_2 \in \mathcal{S}_e(t_1) | \{e_4, e_5\} \in {\mathcal{S}_e}^{(t_1)}) \cdot P(e_4 \in {\mathcal{S}_e}^{(t_1)} | e_5 \in {\mathcal{S}_e}^{(t_1)}) \\ &= min(1, \frac{M-2}{t_1 - 3}) \cdot min(1, \frac{M-1}{t_1 - 2}) \end{aligned}$$

- $t^M_{3,5} > t^M_{2,4} > \max\{t^m_{3,5}, t^m_{2,4}, t_1\}$: in this case only one of the edges e_3, e_5 is observed after one of the edges e_2, e_4, which is observed after all the remaining edges. We consider the case when e_3 and e_2 are observed last. The same procedure follows for e_4 and e_5 as well.

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in {\mathcal{S}_e}^{(t_2)} | \{e_1, e_5\} \in {\mathcal{S}_e}^{(t_3)}) \\ &= P(e_1 \in \mathcal{S}_e(t_2) | \{e_1, e_4, e_5\} \in {\mathcal{S}_e}^{(t_3)}) \cdot P(e_4 \in {\mathcal{S}_e}^{(t_2)} | \{e_1, e_5\} \in {\mathcal{S}_e}^{(t_3)}) \\ &= 1 \cdot min(1, \frac{M-2}{t_2 - 3}) \end{aligned}$$

- $t^m_{3,5} > t_1 > t^M_{2,4}$: in this case both edges e_3,e_5 are observed after e_1, which is observed after all the remaining edges. Here we consider the case when $t_3 > t_5 > t_1$. The same procedure

follows for $t_5 > t_3 > t_1$.

$$\begin{aligned} p' &= P(\{e_2, e_4\} \in \mathcal{S}_e{}^{(t_1)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= P(e_2 \in \mathcal{S}_e(t_1) | e_4 \in \mathcal{S}_e(t_1), \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_4 \in \mathcal{S}_e{}^{(t_1)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= min(1, \frac{M-1}{t_1 - 2}) \cdot min(1, \frac{M}{t_1 - 1}) \end{aligned}$$

- $t^m_{3,5} > t^M_{2,4} > t_1$: in this case both edges e_3,e_5 are observed after one of the edges e_2,e_4, which is observed after e_1. We consider the case $t_2 > t_4$ and $t_3 > t_5$. The same procedure follows for $t_4 > t_2$ and $t_5 > t_3$

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= P(e_4 \in \mathcal{S}_e(t_2) | e_1 \in \mathcal{S}_e(t_2), \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_1 \in \mathcal{S}_e{}^{(t_3)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= min(1, \frac{M-1}{t_2 - 2}) \cdot 1 \end{aligned}$$

- $t^M_{2,4} > t_1 > \max\{M_e, t^m_{2,4}, t^M_{3,5}\}$: in this case both edges e_2, e_4 are observed after e_1, which is observed after the edge reservoir is filled and after all the remaining edges. We consider the case when $t_2 > t_4$. The same procedure follows for $t_4 > t_2$.

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= P(e_4 \in \mathcal{S}_e(t_2) | e_1 \in \mathcal{S}_e(t_2), \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_1 \in \mathcal{S}_e{}^{(t_2)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= \frac{M-1}{t_2 - 2} \cdot \frac{M-2}{t_1 - 3} \cdot \frac{t_1 - 1}{t_2 - 1} \end{aligned}$$

- $t^M_{2,4} > M_e > t_1 > \max\{t^m_{2,4}, t^M_{3,5}\}$:in this case both edges e_2, e_4 are observed after e_1, which is observed before the edge reservoir is filled and after all the remaining edges. We consider the case when $t_2 > t_4$. The same procedure follows for $t_4 > t_2$.

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= P(e_4 \in \mathcal{S}_e(t_2) | e_1 \in \mathcal{S}_e(t_2), \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_1 \in \mathcal{S}_e{}^{(t_2)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= \frac{M-1}{t_2 - 2} \cdot \frac{M_e}{t_2 - 1} \end{aligned}$$

- $t^M_{2,4} > t^M_{3,5} > \max\{M_e, t^m_{3,5}, t^m_{2,4}, t_1\}$: in this case only one of the edges e_2, e_4 is observed after one of the edges e_3, e_5 which is observed after the edge reservoir if filled and after all the remaining edges. We consider the case when $t_2 > t_4$ and $t_3 > t_5$. The same procedure follows for $t_4 > t_2$ and $t_5 > t_3$

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= P(e_1 \in \mathcal{S}_e(t_2) | e_4 \in \mathcal{S}_e(t_2), \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_4 \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= \frac{t_3 - 1}{t_2 - 1} \cdot \frac{t_3 - 2}{t_2 - 2} \cdot \frac{M_e - 2}{t_3 - 3} \end{aligned}$$

- $t^M_{2,4} > M_e > t^M_{3,5} > \max\{t^m_{3,5}, t^m_{2,4}, t_1\}$: in this case only one of the edges e_2, e_4 is observed after one of the edges e_3, e_5 which is observed before the edge reservoir if filled and after all the remaining edges. We consider the case when $t_2 > t_4$ and $t_3 > t_5$. The same procedure follows

for $t_4 > t_2$ and $t_5 > t_3$

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= P(e_1 \in \mathcal{S}_e(t_2) | e_4 \in \mathcal{S}_e(t_2), \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_4 \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= \frac{M_e}{t_2 - 1} \cdot \frac{M_e - 1}{t_2 - 2} \end{aligned}$$

- $t^m_{2,4} > t_1 > t^M_{3,5}$: in this case both edges e_2,e_4 are observed after e_1, which is observed after all the remaining edges. Here we consider the case when $t_2 > t_4 > t_1$. The same procedure follows for $t_4 > t_2 > t_1$.

$$\begin{aligned} p' &= P(\{e_2, e_4\} \in \mathcal{S}_e{}^{(t_1)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= P(e_2 \in \mathcal{S}_e(t_1) | e_4 \in \mathcal{S}_e(t_1), \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \cdot P(e_4 \in \mathcal{S}_e{}^{(t_1)} | \{e_3, e_5\} \in \mathcal{S}_e{}^{(t_1)}) \\ &= min(1, \frac{M_e - 1}{t_2 - 2}) \cdot min(1, \frac{M_e}{t_2 - 1}) \end{aligned}$$

- $t^m_{2,4} > t^M_{3,5} > \max\{M, t_1\}$: in this case both edges e_2,e_4 are observed after one of the edges e_3,e_5, which is observed after the edge reservoir is filled and after all the remaining edges.

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= P(e_4 \in \mathcal{S}_e(t_2) | e_1 \in \mathcal{S}_e(t_2), \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_1 \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= \frac{M_e - 1}{t_2 - 2} \cdot \frac{t_3 - 1}{t_2 - 1} \end{aligned}$$

- $t^m_{2,4} > M_e > t^M_{3,5} > t_1$: in this case both edges e_2,e_4 are observed after one of the edges e_3,e_5, which is observed before the edge reservoir is filled and after all the remaining edges.

$$\begin{aligned} p' &= P(\{e_1, e_4\} \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= P(e_4 \in \mathcal{S}_e(t_2) | e_1 \in \mathcal{S}_e(t_2), \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \cdot P(e_1 \in \mathcal{S}_e{}^{(t_2)} | \{e_1, e_5\} \in \mathcal{S}_e{}^{(t_3)}) \\ &= \frac{M_e - 1}{t_2 - 2} \cdot \frac{M_e}{t_2 - 1} \end{aligned}$$

The lemma follows combining the result for the values of p' with (8.14), (8.15) and (8.16) in (8.5). □

Applying this lemma to an analysis similar to the one used in the proof of Theorem 8.5 we prove that the estimations obtained using TS4C$_2$ are *unbiased*.

Proof of Theorem 8.12. Let t^* denote the first step at for which the number of triangle seen exceeds M_Δ. For $t \leq \min\{M_e, t^*\}$, the entire graph $G^{(t)}$ is maintained in $\mathcal{S}_e$ and all the triangles in $G^{(t)}$ are stored in $\mathcal{S}_\Delta$. Hence all the cliques in $G^{(t)}$ are deterministically observed by TS4C$_1$and we have $\varkappa^{(t)} = |\mathcal{C}_4^{(t)}|$.

Assume now $t > \min\{M_e, t^*\}$ and assume that $|\mathcal{C}_4^{(t)}| > 0$, otherwise, the algorithm deterministically returns 0 as an estimation and the thesis follows. For any 4-clique $\lambda \in \mathcal{C}_4^{(t)}$ which is observed by TS4C$_2$ with probability p_λ, consider a random variable X_λ which takes value p^{-1} iff λ is actually observed by TS4C$_2$ (i.e., with probability p_λ) or zero otherwise. We thus have $\mathbb{E}[X_\lambda] = p_\lambda^{-1} \Pr\left(X_\lambda = p_\lambda^{-1}\right) = p_\lambda^{-1} p_\lambda = 1$. Recall that every time TS4C$_2$ observes a 4-clique on

the stream it evaluates the probability p (according its correct value as shown in Lemma 8.10) of observing it and it correspondingly increases the running estimator by p^{-1}. We therefore can express the running estimator $\varkappa^{(t)}$ as:

$$\varkappa^{(t)} = \sum_{\lambda \in \mathcal{C}_4^{(t)}} X_\lambda \ .$$

From linearity of expectation, we thus have:

$$\mathbb{E}\left[\varkappa^{(t)}\right] = \sum_{\lambda \in \mathcal{C}_4^{(t)}} \mathbb{E}[X_\lambda] = \sum_{\lambda \in \mathcal{C}_4^{(t)}} p_\lambda^{-1} p_\lambda = |\mathcal{C}_4^{(t)}|.$$

□

The following proof for Theorem 8.13 follows a structure similar to the one presented for Theorem 8.6. The two proofs differ however reflecting the different way according to which 4-cliques are detected by TS4C$_2$ compared to TS4C$_1$.

Proof of Theorem 8.13. Assume $|\mathcal{C}_4^{(t)}| > 0$, otherwise TS4C$_1$ estimation is deterministically correct and has variance 0 and the thesis holds. For each $\lambda \in \mathcal{C}_4^{(t)}$ let $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, without loss of generality, let us assume the edges are disposed as in Figure 8.1. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. Let $t_{1,2,4} = \max\{t_1, t_2, t_4, M_e + 1\}$. Let us consider the random variable δ_λ which takes value p_λ^{-1} if the 4-clique λ is observed by TS4C$_2$ using triangles $T_1 = \{e_1, e_2, e_4\}$, $T_1 = \{e_1, e_3, e_5\}$ and the final edge e_6, or zero otherwise. Let p_λ denote the probability of such event.

Since, from Lemma 8.3 we know:

$$\mathrm{Var}\left[\delta_\lambda\right] = \mathrm{E}\left[\delta_\lambda^2\right] - \mathrm{E}\left[\delta_\lambda\right]^2 = p_\lambda - 1 - 1 \leq \prod_{i=0}^{1} \frac{\tau^{(t)} - 1}{M_\Delta - 1} \prod_{i=0}^{3} \frac{t - 1 - i}{M_e - i} - 1.$$

For the estimator $\varkappa^{(t)}$ maintained by TS4C$_2$ we thus have:

$$\begin{aligned}
\mathrm{Var}\left[\varkappa^{(t)}\right] &= \mathrm{Var}\left[\sum_{\lambda \in \mathcal{C}_4^{(t)}} \delta_\lambda\right] = \sum_{\lambda \in \mathcal{C}_4^{(t)}} \sum_{\gamma \in \mathcal{C}_4^{(t)}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] \\
&= \sum_{\lambda \in \mathcal{C}_4^{(t)}} \mathrm{Var}\left[\delta_\lambda\right] + \sum_{\substack{\lambda, \gamma \in \mathcal{C}_4^{(t)} \\ \lambda \neq \gamma}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] \\
&\leq |\mathcal{C}_4^{(t)}| \left(\prod_{i=0}^{1} \frac{\tau^{(t)} - i}{M_\Delta - i} \prod_{i=0}^{3} \frac{t - 1 - i}{M_e - i} - 1 \right) + \sum_{\substack{\lambda, \gamma \in \mathcal{C}_4^{(t)} \\ \lambda \neq \gamma}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] \qquad (8.17)
\end{aligned}$$

We now focus on the analysis of sum of covariances in the right hand side of the previous inequality. From the definition of covariance, we have:

$$\mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] = \mathrm{E}\left[\delta_\lambda \delta_\gamma\right] - \mathrm{E}\left[\delta_\lambda\right] \mathrm{E}\left[\delta_\gamma\right] = \mathrm{E}\left[\delta_\lambda \delta_\gamma\right] - 1,$$

where the last passage follows as, by construction, for all $\lambda \in \mathcal{C}_4^{(t)}$ we have $\mathrm{E}\left[\delta_\lambda\right] = 1$. To complete

the analysis of the covariance we will therefore consider the term $\mathrm{E}\left[\delta_\lambda \delta_\gamma\right]$. We have:

$$\begin{aligned}\mathrm{E}\left[\delta_\lambda \delta_\gamma\right] &= p_\lambda^{-1} p_\gamma^{-1} \mathrm{Pr}\left(\delta_\lambda = p_\lambda^{-1} \wedge \delta_\gamma = p_\gamma^{-1}\right) \\ &= p_\lambda^{-1} p_\gamma^{-1} \mathrm{Pr}\left(\delta_{\lambda_1} = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \mathrm{Pr}\left(\delta_\lambda = p_\lambda^{-1}\right) \\ &= p_\lambda^{-1} \mathrm{Pr}\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right). \end{aligned} \tag{8.18}$$

In order to conclude our analysis it will therefore be necessary to study the probability according to which TS4C$_2$ observes λ *conditioned* on the fact that γ was observed. We divide the possible pairs of 4-cliques depending on how many edges they share (if any). From Lemma 8.7 we have that any pair of 4-cliques λ and γ can share either one, three or no edges, hence:

1. λ and γ do not share any edge: As λ and γ do not share any edge, no edge of γ will be used by TS4C$_2$ to detect λ. Rather, if any of the edges (resp., triangles) of γ is included in $\S_e$ (resp., $\mathcal{S}_\Delta$), this would lessen the probability of TS4C$_2$ detecting λ as some of space in $\mathcal{S}_e$ or $\mathcal{S}_\Delta$ may be occupied by edges or triangle sub-structures for γ. Therefore we have $\mathrm{Pr}\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \leq \mathrm{Pr}\left(\delta_\lambda = p_\lambda^{-1}\right)$ and thus from (8.18):

$$\begin{aligned}\mathrm{E}\left[\delta_\lambda \delta_\gamma\right] &= p_\lambda^{-1} \mathrm{Pr}\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \\ &\leq p_\lambda^{-1} \mathrm{Pr}\left(\delta_\lambda = p_\lambda^{-1}\right) \\ &= 1; \end{aligned}$$

   We therefore have $\mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] \leq 0$. Hence we can conclude that the contribution of the covariances of the pairs of random variables corresponding to 4-cliques that do not share any edge to the summation in (8.17) is less or equal to zero.

2. λ and γ share exactly one edge $e^* = \lambda \cap \gamma$ as shown in Figure 8.2. Note first of all that if the shared edge is the last to be observed on the stream for either λ or *gamma*, then the same considerations presented for the case in which λ and γ do not share any edge apply an hence, $\mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] \leq 0$.

   In the following we assume that e^* is not the last edge observed on the stream for neither λ nor γ. Let us consider the event $E* =$"$e^* \cap T_1 \cap E^{(t_{1,2,4}-1)} \in \mathcal{S}_e^{t_{1,2,4}}$, or $e^* \cap T_2 \cap E^{(t_{1,3,5}-1)} \in \mathcal{S}_e^{t_{1,3,5}}$". Clearly $\mathrm{Pr}\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | \delta_{\gamma_2} = p_{\gamma_2}^{-1}\right) \leq \mathrm{Pr}\left(\delta_{\lambda_1} = p_{\lambda_1}^{-1} | E*\right)$. Recall from Lemma 8.3 that $\mathrm{Pr}\left(\delta_{\lambda_1} | E^*\right) = \mathrm{Pr}\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*\right) \mathrm{Pr}\left(T_1 \in \mathcal{S}_\Delta^{(t_6)} | E*\right) \mathrm{Pr}\left(S(T_1) | E^*\right)$, where $S(T_1)$ denotes the event "T_1 is observed on the stream by TS4C$_1$". By applying the law of total probability e have that $\mathrm{Pr}\left(T_1 \in \mathcal{S}_\Delta^{(t_6)} | E*\right) \leq \mathrm{Pr}\left(T_1 \in \mathcal{S}_\Delta^{(t_6)}\right)$. The remaining two terms are influenced differently depending on whether $e^* \in T_1$ or $e^* \in \{e_3, e_5\}$

   - if $e^* \in T_1$: we then have $\mathrm{Pr}\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*\right) \leq \mathrm{Pr}\left(\{e_3, e_5\} \subseteq \mathcal{S}_e^{(t_6)}\right)$. This follows from the properties of the reservoir sampling scheme as the fact that the edge e^* is in $\mathcal{S}_e$ means that one unit of the available memory space required to hold e_3 or e_5 is occupied, at least for some time, by e^*. If e^* is the last edge of T_1 observed in the stream we then

have:

$$\begin{aligned}\Pr\left(S(T_1)|E^*\right) &= \Pr\left(\{e_1,e_2,e_4\}\setminus\{e^*\}\in\mathcal{S}_e^{(}t_1,2,4)|E^*\right)\\ &= \Pr\left(\{e_1,e_2,e_4\}\setminus\{e^*\}\in\mathcal{S}_e^{(}t_1,2,4)\right)\\ &\leq \frac{M_e}{t_{1,2,4}-1}\frac{M_e-1}{t_{1,2,4}-2}\end{aligned}$$

Suppose instead that $e*$ is *not* the last edge of T_1. Assume further , without loss of generality, that $t_2 > \max\{t_1,t_4\}$. TS4C$_1$ observes T_1 iff $\{e_1,e_4\}\in\mathcal{S}_e^{(}t_2)$. As in E^* we assume that once observed e^* is *always* maintained in $\mathcal{S}_e$ until $t_{1,2,4}$ we have:

$$\begin{aligned}\Pr\left(S(T_1)|E^*\right) &= \Pr\left(\{e_1,e_4\}\setminus\{e^*\}\in\mathcal{S}_e^{(}t_1,2,4)|E^*\right)\\ &\leq \frac{M_e-1}{t_{1,2,4}-2}\end{aligned}$$

- if $e^*\in\{e_3,e_5\}$: we then have $\Pr\left(S(T_1)|E^*\right)\leq\Pr\left(S(T_1)\right)$. This follows from the properties of the reservoir sampling scheme as the fact that the edge e^* is in $\mathcal{S}_e$ means that one unit of the available memory space required to hold the first two edges of T_1 until $t_{1,2,4}$ is occupied, at least for some time, by e^*. Further, using Lemma 8.3 we have:
  - if $t_6\leq M_e$, then $\Pr\left(\{e_3,e_5\}\subseteq\mathcal{S}_e^{(t_6)}|E^*\right)=1$
  - if $\min\{t_3,t_5\}>t_{1,2,4}$, then $\Pr\left(\{e_3,e_5\}\subseteq\mathcal{S}_e^{(t_6)}|E^*\right)\geq\frac{M_e}{t_6-1}$,
  - if $\max\{t_3,t_5\}>t_{1,2,4}>\min\{t_3,t_5\}$, then
    $$\Pr\left(\{e_3,e_5\}\subseteq\mathcal{S}_e^{(t_6)}|E^*\right)\geq\max\{\frac{M_e-1}{t_6-2},\frac{M_e-2}{t_{1,2,4}-3}\frac{t_{1,2,4}-1}{t_6-1}\},$$
  - otherwise $\Pr\left(\{e_3,e_5\}\subseteq\mathcal{S}_e^{(t_6)}|E^*\right)\geq\frac{M_e-2}{t_{1,2,4}-3}\frac{t_{1,2,4}-1}{t_6-1}$.

Putting together these various results we have that

$$p_\lambda^{(-1)}\Pr\left(\delta_{\lambda_1}=p_{\lambda_1}^{-1}|\delta_{\gamma_2}=p_{\gamma_2}^{-1}\right)\leq p_\lambda^{(-1)}\Pr\left(\delta_{\lambda_1}=p_{\lambda_1}^{-1}|E*\right)\leq c\frac{t_6-1}{M_e},$$

with $c=s$. Hence we have $\mathrm{E}\left[\delta_{\lambda_1}\delta_{\gamma_2}\right]\leq\frac{c}{4}\frac{t_6-1}{M_e}\leq\frac{c}{4}\frac{t-1}{M_e}$. We can thus bound the contribution to the third component of (8.5) given by the pairs of random variables corresponding to 4-cliques that share one edge as:

$$2a^{(t)}\left(c\frac{t-1}{M_e}-1\right), \tag{8.19}$$

where $a^{(t)}$ denotes the number of unordered pairs of 4-cliques which share one edge in $G^{(t)}$.

3. λ and γ share three edges $\{e_1^*,e_2^*,e_3^*\}$ which form a triangle sub-structure for both λ and γ. Let us refer to Figure 8.3, without loss of generality let T_1 denote the triangle shared between the two cliques. We distinguish the kind of pairs for the random variables δ_{λ_i} and δ_{γ_j} cases:

   - $\delta_{\lambda_i}=p_{\lambda_i}^{-1}$ if $T_1\in\mathcal{S}_\Delta^{t_6-1}\wedge\{e_3,e_5\}\subseteq\mathcal{S}_e^{t_6-1}$ and $\delta_{\gamma_j}=p_{\gamma_j}^{-1}$ if $T_1\in\mathcal{S}_\Delta^{t_\gamma-1}\wedge\{g_3,g_5\}\subseteq\mathcal{S}_e^{t_\gamma-1}$, where t_γ denotes the time step at which the last edge of γ is observed. This is the case for which the random variables δ_{λ_i} and δ_{γ_j} corresponds to FourEst observing λ and γ using the *shared triangle* T_1. Let us consider the event $E*=$"$T_1\in\mathcal{S}_\Delta^{(t_6)}$. Clearly $\Pr\left(\delta_{\lambda_i}=p_{\lambda_i}^{-1}|\delta_{\gamma_j}=p_{\gamma_j}^{-1}\right)\leq\Pr\left(\delta_{\lambda_i}=p_{\lambda_i}^{-1}|E*\right)$. In this case we have

$\Pr\left(T_1 \in \mathcal{S}^{(t_6)} | E^*\right) \leq 1$, while $\Pr\left(\{e_3, e_5\} \in \mathcal{S}_e^{(t_6)} | E^*\right) \leq Prob\{e_3, e_5\} \in \mathcal{S}_e^{(t_6)}$. This second fact follows from the properties of the reservoir sampling scheme as the fact that the edges e_1^*, e_2^* and e_3^* are in $\mathcal{S}_e$ *at least* for the time required for T_1 to be observed, means that at least two unit of the available memory space required to hold the edges e_3, e_5 are occupied, at least for some time. Putting together these various results we have that $p_{\lambda_i}^{(-1)} Prob\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1} \leq p_{\lambda_i}^{(-1)} \Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*\right) \leq c \left(\frac{t_6 - 1}{M_e}\right)^2 \frac{\tau^{(t)}}{\mathcal{S}_\Delta}$. Hence we have $\mathrm{E}\left[\delta_{\lambda_i} \delta_{\gamma_j}\right] \leq \frac{c}{4} \left(\frac{t_6 - 1}{M_e}\right)^2 \frac{\tau^{(t)}}{\mathcal{S}_\Delta}$ and $\mathrm{Cov}\left[\delta_{\lambda_i}, \delta_{\gamma_j}\right] \leq \frac{c}{4} \left(\frac{t_6 - 1}{M_e}\right)^2 \frac{\tau^{(t)}}{\mathcal{S}_\Delta} - \frac{1}{4}$.

- in all the remaining cases, then the random variables δ_{λ_i} and δ_{γ_j} corresponds to FourEst not observing λ and γ using the shared triangle T_1 for both of them. Let T^* denote the triangle sub-structure used by TS4C$_1$ to count λ with respect to δ_{λ_i}. Let us consider the event E^* ="$\{e_1^*, e_2^*, e_3^*\} \cap T_1 \cap E^{(t_{1,2,4}-1)} \in \mathcal{S}_e^{t_{1,2,4}}$, $T_1 \in \mathcal{S}_\Delta^{(t_6)}$ unless one of its edges is the last edge of T^* observed on the stream, and $\{e_1^*, e_2^*, e_3^*\} \cap \{e_3, e_5\} \cap E^{(t_6-1)} \in S_e^{(t_6)}$ if $e^* \in \{e_3, e_5\}$", where $E^{(t)}$ denotes the set of edges observed up until time t included. Clearly $\Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}\right) \leq \Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*\right)$. Note that in this case $|\{e_1^*, e_2^*, e_3^*\} \cap T_1| + |\{e_1^*, e_2^*, e_3^*\} \cap \{e_3, e_5\}|$. By analyzing $\Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*\right)$ in this case using similar steps as the ones described for the other sub-cases we have $p_{\lambda_i}^{(-1)} \Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | \delta_{\gamma_j} = p_{\gamma_j}^{-1}\right) \leq p_{\lambda_i}^{(-1)} \Pr\left(\delta_{\lambda_i} = p_{\lambda_i}^{-1} | E*\right) \leq c \left(\frac{t_6 - 1}{M_e}\right)^3$. Hence we have $\mathrm{E}\left[\delta_{\lambda_i} \delta_{\gamma_j}\right] \leq \frac{c}{4} \left(\frac{t_6 - 1}{M_e}\right)^3$ and $\mathrm{Cov}\left[\delta_{\lambda_i}, \delta_{\gamma_j}\right] \leq \frac{c}{4} \left(\frac{t_6 - 1}{M_e}\right)^3 - \frac{1}{4}$.

We can thus bound the contribution to the third component of (8.5) given by the pairs of random variables corresponding to 4-cliques that share three edges as:

$$2b^{(t)} \left(c \left(\frac{t-1}{M_e}\right)^2 \left(\frac{1}{4}\frac{\tau^{(t)}}{M_\Delta} + \frac{3}{4}\frac{t-1}{M_e}\right) - 1 \right), \tag{8.20}$$

where $b^{(t)}$ denotes the number of unordered pairs of 4-cliques which share three edges in $G^{(t)}$.

The Theorem follows by combining (8.7), (8.19) and (8.20) in (8.5). □

8.6 Comparison with single sample approach

Given a certain fixed amount M of available memory space, our TieredSampling approach suggest that the user *allocates* such memory into multiple tiers of reservoir samples, in order to exploit the sparsity of the sub-structures of the motif of interest. However, it is only natural to wonder how this approach compares to an alternative, somewhat simpler, strategy that maintains a *single* sample of edges and that relies *only* on the edges in the sample to detect occurrences of the motif in $G^{(t)}$.

To quantify the advantage of our TieredSampling approach, we construct and fully analyze algorithm FourEst that uses a single edges sample strategy. We thoroughly compare the performance achieved by TS4C$_1$ with the performances of an algorithm, named FourEst, that uses a single sample strategy. We analyze the estimator provided by FourEst compare the analytical

ALGORITHM 17 FourEst

Input: Edge stream Σ, integer $M \geq 6$
Output: Estimation of the number of 4-cliques $\varkappa$
$\mathcal{S}_e \leftarrow \emptyset$, $t \leftarrow 0$, $\varkappa \leftarrow 0$
for each element (u, v) from Σ **do** ▷ Process each edge (u,v) coming from the stream in discretized timesteps
  $t \leftarrow t + 1$
  Update4Cliques(u, v) ▷ Update the 4-clique estimator by considering the new 4-cliques closed by edge (u,v)
  SampleEdge$((u, v), t)$ ▷ Update edge sample with (u,v) according to RS scheme

function Update4Cliques$((u, v), t)$
  $\mathcal{N}^{\mathcal{S}}_{u,v} \leftarrow \mathcal{N}^{\mathcal{S}}_{u} \cap \mathcal{N}^{\mathcal{S}}_{v}$
  for each element (x, w) from $\mathcal{N}^{\mathcal{S}}_{u,v} \times \mathcal{N}^{\mathcal{S}}_{u,v}$ **do** ▷ For each 4-clique formed by (u, v) and 5 edges in the S_e
    if (x, w) in $\mathcal{S}_e$ **then** ▷ Calculate the probability of observing the new formed 4-clique
      if $t \leq M$ **then**
        $p \leftarrow 1$
      else
        $p \leftarrow min\{1, \frac{M(M-1)(M-2)(M-3)(M-4)}{(t-1)(t-2)(t-3)(t-4)(t-5)}\}$
      $\varkappa \leftarrow \varkappa + p^{-1}$ ▷ Increase 4-clique estimator by the inverse of the probability of observing the new 4-clique

bound on the variance of the estimator and we then compare their performance on both synthetic and real-world data (in Section 8.8).

8.6.1 Edge sampling approach - FourEst

FourEst (FOUR clique ESTimation) maintains an uniform random sample $\mathcal{S}$ of size M of the edges observed over the stream using the reservoir sampling scheme, in order to estimate the number of four cliques in $G^{(t)}$. This algorithm is a natural extension of the technique discussed in [51] for triangle counting.

At each time step t, FourEst maintains a *running estimation* $\varkappa^{(t)}$ of $|\mathcal{C}_4^{(t)}|$. Clearly $\varkappa^{(0)} = 0$. Every time a new edge $e_t = (u, v)$ is observed on the stream, FourEst verifies whether $e_{(t)}$ completes any 4-cliques with the edges currently in $\mathcal{S}_e^{(t)}$. If that is the case, the estimator $\varkappa$ is increased by the reciprocal of the probability $p = \min\{1, \prod_{i=0}^{4} \frac{M-i}{t-1-i}\}$ of observing that same 4-clique (from Lemma 8.1) using FourEst. Finally, the algorithm updates $\mathcal{S}_e$ according to the reservoir sampling scheme discussed in Section 8.2. The pseudocode for FourEst is presented in Algorithm 17.

Analysis of the FourEst estimator: Before presenting the proof of the *unbiasedness* of the estimations obtained using FourEst in Theorem 8.16, we introduce Lemma 8.15 which characterizes the probability of a 4-clique being observed by algorithm FourEst. The proofs of the following results share the structure of the proofs of corresponding results for TS4C$_1$ and TS4C$_2$.

Lemma 8.15. *Let $\lambda \in \mathcal{C}_4^{(t)}$ with $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. Assume, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. λ is observed by* FourEst *at time t_6 with probability:*

$$p_\lambda = \begin{cases} 0 & if\ |M| < 5, \\ 1 & if\ t_6 \leq M+1, \\ \prod_{i=0}^{5} \frac{M-i}{t-i-1} & if\ t_6 > M+1. \end{cases} \tag{8.21}$$

Proof. Clearly FourEst can observe λ *only* at the time step at which the last edge e_6 is observed on the stream at t_6. Further, from its construction, FourEst observed a 4-clique λ if and only if when its large edge is observed on the stream its remaining five edges are kept in the edge reservoir $\mathcal{S}$. $\mathcal{S}$ is a uniform edge sample maintained by means of the reservoir sampling scheme. From Lemma 7.1 we have that the probability of any five elements observed on the stream *prior* to t_6 being in $\mathcal{S}$ at the beginning of step t_6 is given by:

$$\Pr\left(\{e_1, e_2, e_3, e_4, e_5, e_6\} \subseteq\right) = \begin{cases} 0 & if\ |M| < 5, \\ 1 & if\ t_6 \leq M+1, \\ \prod_{i=0}^{4} \frac{M-i}{t-i-1} & if\ t_6 > M+1. \end{cases}$$

The lemma follows. □

As for the TieredSampling algorithms, the estimator obtained using FourEst is unbiased.

Theorem 8.16. *Let $\varkappa^{(t)}$ the estimated number of 4-cliques in $G^{(t)}$ computed by* FourEst *using memory of size M. $\varkappa^{(t)} = |\mathcal{C}_4^{(t)}|$ if $t \leq M+1$ and $\mathbb{E}\left[\varkappa^{(t)}\right] = |\mathcal{C}_4^{(t)}|$ if $t > M+1$.*

Proof. Recall that when a new edge e_t is observed on the stream FourEst updates the estimator $\varkappa^{(t)}$ *before* deciding whether the new edge is inserted in $\mathcal{S}$. For $t \leq M+1$, the entire graph $G^{(t)} \setminus \{e_t\}$ is maintained in $\mathcal{S}$, thus whenever an edge e_t is inserted at time $t \leq M+1$, FourEst observes *all* the triangles which include e_t in $G^{(t)}$ with probability 1 thus increasing $\varkappa$ by one. By a simple inductive analysis we can therefore conclude that for $t \leq M+1$ we have $\varkappa^{(t)} = |\mathcal{C}_4^{(t)}|$.

Assume now $t > M+1$ and assume that $|\mathcal{C}_4^{(t)}| > 0$, otherwise, the algorithm deterministically returns 0 as an estimation and the thesis follows. Recall that every time FourEst observes a 4-clique on the stream a time t it computes the probability $p = \prod_{i=0}^{4} \frac{M-i}{t-i-1}$ of observing it and it correspondingly increases the running estimator by p^{-1}. From Lemma 7.1, the probability p computed by FourEst does indeed correspond to the correct probability of observing a 4-clique at time t. For any 4-clique $\lambda \in \mathcal{C}_4^{(t)}$ which is observed by TS4C$_1$with probability p_λ, consider a random variable X_λ which takes value p^{-1} iff λ is actually observed by FourEst (i.e., with probability p_λ) or zero otherwise. We thus have $\mathbb{E}[X_\lambda] = p_\lambda^{-1} \Pr\left(X_\lambda = p_\lambda^{-1}\right) = p_\lambda^{-1} p_\lambda = 1$.
We therefore can express the running estimator $\varkappa^{(t)}$ as: $\varkappa^{(t)} = \sum_{\lambda \in \mathcal{C}_4^{(t)}} X_\lambda$.
From linearity of expectation, we thus have:

$$\mathbb{E}\left[\varkappa^{(t)}\right] = \sum_{\lambda \in \mathcal{C}_4^{(t)}} \mathbb{E}[X_\lambda] = \sum_{\lambda \in \mathcal{C}_4^{(t)}} p_\lambda^{-1} p_\lambda = |\mathcal{C}_4^{(t)}|.$$

□

We now show an *upper bound* to the variance of the FOUREST estimations for $t > M$ (for $t \leq M$ we have $\varkappa^{(t)} = |\mathcal{C}_4^{(t)}|$ and thus the variance of $\varkappa^{(t)}$ is zero), and a corresponding concentration bound.

Theorem 8.17. *For any time $t > M + 1$, we have*

$$Var\left[\varkappa^{(t)}\right] \leq |\mathcal{C}_4^{(t)}| \left(\left(\frac{t-1}{M} \right)^5 - 1 \right) + a^{(t)} \left(\frac{t-1}{M} - 1 \right) + b^{(t)} \left(\left(\frac{t-1}{M} \right)^3 - 1 \right),$$

where $a^{(t)}$ (resp., $b^{(t)}$) denotes the number of unordered pairs of 4-cliques which share one edge (resp., three edges) in $G^{(t)}$. Thus, for any $\varepsilon, \delta \in (0,1)$, if

$$M > (t-1) \max\left\{ \sqrt[5]{\frac{3}{\delta\varepsilon^2 |\mathcal{C}_4^{(t)}|}}, \frac{3a^{(t)}}{\delta\varepsilon^2 |\mathcal{C}_4^{(t)}|^2}, \sqrt[3]{\frac{3b^{(t)}}{\delta\varepsilon^2 |\mathcal{C}_4^{(t)}|^2}} \right\}$$

then $Pr\left(|\varkappa^{(t)} - |\mathcal{C}_4^{(t)}| < \varepsilon |\mathcal{C}_4^{(t)}| \right) > 1 - \delta$.

Proof. Assume $|\mathcal{C}_4^{(t)}| > 0$ and $t > M + 1$, otherwise (from Theorem 8.16) TS4C$_1$ estimation is deterministically correct and has variance 0 and the thesis holds. For each $\lambda \in \mathcal{C}_4^{(t)}$ let $\lambda = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, without loss of generality let us assume the edges are disposed as in Figure 8.1. Assume further, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_6 > \max\{t_i, 1 \leq i \leq 5\}$. Let us consider the random variable δ_λ (which takes value p_λ^{-1} if the 4-clique λ is observed by FOUREST, or zero otherwise. From Lemma 8.15, we have:

$$p_\lambda = \Pr\left(\delta_\lambda = p_\lambda^{-1} \right) = \prod_{i=0}^{4} \frac{M-i}{t-i-1},$$

and thus:

$$\mathrm{Var}\left[\delta_\lambda\right] = p_\lambda^{-1} - 1.$$

We can express the estimator $\varkappa^{(t)}$ as $\varkappa^{(t)} = \sum_{\lambda \in \mathcal{C}_4^{(t)}}$. We therefore have:

$$\mathrm{Var}\left[\varkappa^{(t)}\right] = \mathrm{Var}\left[\sum_{\lambda \in \mathcal{C}_4^{(t)}} \delta_\lambda \right] = \sum_{\lambda \in \mathcal{C}_4^{(t)}} \sum_{\gamma \in \mathcal{C}_4^{(t)}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] = \sum_{\lambda \in \mathcal{C}_4^{(t)}} \mathrm{Var}\left[\delta_\lambda\right] + \sum_{\substack{\lambda,\gamma \in \mathcal{C}_4^{(t)} \\ \lambda \neq \gamma}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right]$$

$$\leq |\mathcal{C}_4^{(t)}| \left(\prod_{i=0}^{4} \frac{t-1-i}{M-i} - 1 \right) + \sum_{\substack{\lambda,\gamma \in \mathcal{C}_4^{(t)} \\ \lambda \neq \gamma}} \mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] \quad . \tag{8.22}$$

From Lemma 8.7, we have that two distinct cliques λ and γ can share one, three or no edges. In analyzing the summation of covariance terms appearing in the right-hand-side of (8.5) we shall therefore consider separately the pairs that share respectively one, three or no edges.

- λ and γ do not share any edge:

$$\begin{aligned} \mathrm{E}\left[\delta_\lambda \delta_\gamma\right] &= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} \wedge \delta_\gamma = p_\gamma^{-1} \right) \\ &= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1} \right) \Pr\left(\delta_\lambda = p_\lambda^{-1} \right) \end{aligned}$$

  The term $\Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1} \right)$ denotes the probability of FOUREST observing λ conditioned of the fact that γ was observed. Note that as λ and γ do not share any edge, no edge of

γ will be used by FourEst to detect λ. Rather, if any edge of γ is included in $\mathcal{S}$, this lowers the probability of FourEst detecting λ as some of space available in $\mathcal{S}$ may be occupied by edges of γ. Therefore we have $\Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \leq \Pr\left(\delta_\lambda = p_\lambda^{-1}\right)$ and thus:

$$\begin{aligned}
\mathrm{E}\left[\delta_\lambda \delta_\gamma\right] &= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \Pr\left(\delta_\gamma = p_\gamma^{-1}\right) \\
&\leq p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1}\right) \Pr\left(\delta_\gamma = p_\gamma^{-1}\right) \\
&\leq p_\lambda^{-1} p_\gamma^{-1} p_\lambda p_\gamma \\
&\leq 1.
\end{aligned}$$

As $\mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] = \mathrm{E}\left[\delta_\lambda \delta_\gamma\right] - 1$, we therefore have $\mathrm{Cov}\left[\delta_\lambda, \delta_\gamma\right] \leq 0$. Hence we can conclude that the contribution of the covariances of the pairs of random variables corresponding to 4-cliques that do not share any edge to the summation in (8.5) is less or equal to zero.

- λ and γ share exactly one edge $e^* = \lambda \cap \gamma$ (as shown in Figure 8.2). Let us consider the event E^* ="$e^* \in \mathcal{S}_{t_6}$ unless e^* is observed at t_6". Clearly $\Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \leq \Pr\left(\delta_\lambda = p_\lambda^{-1} | E*\right)$. Recall from Lemma 8.15 that $\Pr\left(\delta_\lambda | E^*\right) = \Pr\left(\{e_1, \ldots, e_5\} \subseteq \mathcal{S}_e^{(t_6)} | E^*\right)$. We can distinguish two cases: (a) $e^* = e_6$: in this case we have $\Pr\left(\delta_\lambda | E^*\right) = \Pr\left(\{e_1, \ldots, e_5\} \subseteq \mathcal{S}_e^{(t_6)}\right) = p_\lambda$; (b) $e^* \neq e_6$: in this case we have $\Pr\left(\delta_\lambda | E^*\right) = \Pr\left(\{e_1, \ldots, e_5\} \setminus \{e^*\} \subseteq \mathcal{S}^{(t_6)} | e^* \in \mathcal{S}_e^{(t_6)}\right) = \prod_{i=0}^{3} \frac{M-1-i}{t_6-2-i}$. We can therefore conclude:

$$\begin{aligned}
\mathrm{E}\left[\delta_\lambda \delta_\gamma\right] &= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \Pr\left(\delta_\gamma = p_\gamma^{-1}\right) \\
&\leq p_\lambda^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \\
&\leq \frac{t_6 - 1}{M}.
\end{aligned}$$

We can thus bound the contribution to covariance summation in (8.22) given by the pairs of random variables corresponding to 4-cliques that share one edge as:

$$a^{(t)} \left(\frac{t-1}{M} - 1\right), \tag{8.23}$$

where $a^{(t)}$ denotes the number of unordered pairs of 4-cliques which share one edge in $G^{(t)}$.

- λ and γ share three edges $\{e_1^*, e_2^*, e_3^*\}$ which form a triangle sub-structure for both λ and γ. Let us refer to Figure 8.3. Let us consider the event E^* ="$\{e_1^*, e_2^*, e_3^*\} \cap E^{(t_6-1)} \subseteq \mathcal{S}^{(t_6)}$. Clearly $\Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \leq \Pr\left(\delta_\lambda = p_\lambda^{-1} | E*\right)$. Recall that $\Pr\left(\delta_\lambda | E^*\right) = \Pr\left(\{e_1, \ldots, e_5\} \subseteq \mathcal{S}^{(t_6)} | E^*\right)$. We can distinguish two cases:

  (a) $e_6 \cap \{e_1^*, e_2^*, e_3^*\} \neq \emptyset$: in this case we have $|\{e_1, \ldots, e_5\} \setminus \left(\{e_1^*, e_2^*, e_3^*\} \setminus \{e_6\}\right)| = 3$ hence $\Pr\left(\delta_\lambda | E^*\right) = \Pr\left(\{e_1, \ldots, e_5\} \setminus \left(\{e_1^*, e_2^*, e_3^*\} \setminus \{e_6\}\right) | \left(\{e_1^*, e_2^*, e_3^*\} \setminus \{e_6\}\right) \subseteq \mathcal{S}^{(t_6)}\right) = \prod_{i=0}^{2} \frac{M-2-i}{t_6-3-i}$;

  (b) $e_6 \cap \{e_1^*, e_2^*, e_3^*\} = \emptyset$: in this case we have $|\{e_1, \ldots, e_5\} \setminus \left(\{e_1^*, e_2^*, e_3^*\} \setminus \{e_6\}\right)| = 2$ hence $\Pr\left(\delta_\lambda | E^*\right) = \Pr\left(\{e_1, \ldots, e_5\} \setminus \{e_1^*, e_2^*, e_3^*\} | \{e_1^*, e_2^*, e_3^*\}) \subseteq \mathcal{S}^{(t_6)}\right) = \prod_{i=0}^{1} \frac{M-3-i}{t_6-4-i}$;. For the

pairs of 4-cliques which share three edge we therefore have:

$$\begin{aligned}\mathrm{E}\left[\delta_\lambda \delta_\gamma\right] &= p_\lambda^{-1} p_\gamma^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \Pr\left(\delta_\gamma = p_\gamma^{-1}\right) \\ &\leq p_\lambda^{-1} \Pr\left(\delta_\lambda = p_\lambda^{-1} | \delta_\gamma = p_\gamma^{-1}\right) \\ &\leq \prod_{i=0}^{2} \frac{t-1-i}{M-i}.\end{aligned}$$

We can thus bound the contribution to covariance summation in (8.22) given by the pairs of random variables corresponding to 4-cliques that share one edge as:

$$b^{(t)} \left(\prod_{i=0}^{2} \frac{t-1-i}{M-i} - 1 \right) = b^{(t)} c' \left(\frac{t}{M} \right)^5 \tag{8.24}$$

where $b^{(t)}$ denotes the number of unordered pairs of 4-cliques which share three edges in $G^{(t)}$ and $c' = (M-1)(M-2)/M^2$.

The bound on the variance follows form the previous considerations and by combining by combining (8.23) and (8.24) in (8.22). Finally, the concentration bound is obtained by applying Chebyshev's inequality [164, Thm. 3.6]. The proof follows a reasoning analogous to that in the proof of Theorem 8.8. □

8.6.2 Variance comparison

Although the upper bounds obtained in Theorems 8.6, 8.13, and 8.17 cannot be compared directly, they still provide some useful insight on which algorithm may be performing better according to the properties of $G^{(t)}$.

Let us consider the first, dominant, terms of each of the variance bounds, that is $|\mathcal{C}_4^{(t)}| \left(\left(\frac{t-1}{M_e} \right)^4 \frac{\tau^{(t)}}{M_\Delta} - 1 \right)$ for TS4C$_1$, $|\mathcal{C}_4^{(t)}| \left(c \left(\frac{t-1}{M_e} \right)^4 \left(\frac{\tau^{(t)}}{M_\Delta} \right)^2 - 1 \right)$ for TS4C$_2$, and $|\mathcal{C}_4^{(t)}| \left(\left(\frac{t-1}{M} \right)^5 - 1 \right)$ for FourEst. All the bounds share a similar dependence from the number of pairs of cliques that share one or three edges in the second and third term. Due to the fact that TS4C$_1$ splits the memory into two levels (in particular, with $M_e = 4M/5$) we have a higher overall contribution for these terms.

While TS4C$_1$ exhibits a slightly higher constant multiplicative term cost due to the splitting of the memory in the TieredSampling approach, the most relevant difference is however given by the term $\frac{\tau^{(t)}}{M}$ appearing in the bound for TS4C$_1$ compared with an additional $\frac{t-1}{M}$ appearing in the bound for FourEst. Recall that $\tau^{(t)}$ denotes here the number of triangles *observed* by the algorithm up to time t. Due to the fact that the probability of observing a triangle decreases quadratically with respect to the size of the graph t, we expect that $\tau < t$ and, for sparser graphs for which 3 and 4-cliques are indeed "*rare patterns*", we actually expect $\tau^{(t)} << t$. Therefore, under these circumstances, we would expect $M/5\tau >> M/t$.

This is the critical condition for the success of the TieredSampling approach. If the substructure selected as a tool for counting the motif of interest is not "*rare enough*" then there is no benefit in devoting a certain amount of the memory budget to maintaining a sample of occurrences

of the sub-structure. Such problem would, for instance, arise when using the TieredSampling approach for counting triangles using *wedges* (i.e., two-hop paths) as a sub-structure, as attempted in [117], as in most real-world graph the number of wedges is much greater of the number of edges themselves making them not suited to be used as a sub-structure.

8.6.3 Experimental evaluation over random graphs

In this section, we compare the performances of our TieredSampling algorithms of TS4C$_1$ and TS4C$_2$ with the performance of the single sample approach FourEst, on randomly generated graphs. In particular, we analyze random graph based on a variation [2] of the Barábasi-Albert random graph [8] model, which exhibits the same *scale-free* property observed in many real-world graphs of interest such as social networks. The graph are generated as follows: the initial graph is a *star graph* with $m+1$ nodes and m edges where a node (i.e., the *center* of the star) is connected to the remaining m ones. Then, n vertices are added one at a time. When the i-th node is added, for $m+1 \leq i \leq n$, it is connected to m vertices among the $i-1$ ones already added which are chose with a probability that is *proportional* to the number of *links* (i.e., edges) that the existing nodes already have. In particular, the probability that the i-th new node is connected to node j, for $1 \leq j \leq i-1$ is $p_i, j = d_j / \sum_{\ell=1}^{i-1} d_\ell$, where d_j denotes the degree of the j-th node before the insertion of the i-th node. The graph constructed at the end of this process has $n+m$ total vertices and nm total edges. The corresponding graph stream can be constructed by simulating the generating process of the random graph and by selecting randomly the order of the edges among the m that are generated for each node insertion. Such variation allows the study of a random preferential attachment graph without starting from a densely connected initial component.

In our experiments, we set $n = 20000$ and we consider various values for m from 50 to 2000 in order to compare the performances of the two approaches as the number of edges (and thus triangles) increases and the generated graph grows denser. We start with a graph with The algorithms use a memory space whose size corresponds to 5% of the number of edges in the graph nm. While FourEst devotes the entire available memory to maintaining an edges reservoir sample, the TieredSampling algorithms will split the available space between the edge sample $\mathcal{S}_e$ and the triangle samples ($\mathcal{S}_\Delta$) according to the *splitting criteria* discusses in the respective sections (i.e., for TS4C$_1$: $M_e = 4M/5$ and $M_\Delta = M/5$; for TS4C$_2$ $M_e = 2M/3$ and $M_\Delta = M/3$).

We compare the accuracy of TieredSampling and FourEst approaches on random graphs using the standard Mean Average Percentage Error *MAPE* [110] as defined in Section 8.2.

In Figure 8.4, we compare the average MAPE of FourEst, TS4C$_1$ and TS4C$_2$ for Barábasi-Albert random graphs with 20000 nodes and various values of m ranging from 50 to 2000. In columns 2,3 and 5 of Table 8.2 we present the average of the MAPEs of the ten runs for FourEst, TS4C$_1$ and TS4C$_2$. In column 4 (resp., 6) of Table 8.2 we report the percent reduction/increase in terms of the average MAPE obtained by TS4C$_1$ (resp., TS4C$_2$) with respect to FourEst.

[2] We use the version provided by the *NetworkX* package `https://networkx.github.io/documentation/networkx-1.9.1/reference/generated/networkx.generators.random_graphs.barabasi_albert_graph.html`

m	FourEst	$TS4C_1$	Change $TS4C_1$	$TS4C_2$	Change $TS4C_2$
50	0.8775	0.5862	-33.19%	0.5222	-40.49%
100	0.3054	0.1641	-46.27%	0.1408	-53.90%
150	0.1521	0.0937	-38.34%	0.0917	-39.70%
200	0.0899	0.0599	-33.39%	0.0549	-38.95%
300	0.0486	0.0346	-28.80%	0.0417	-14.19%
400	0.0289	0.0249	-13.87%	0.0261	-9.32%
500	0.0221	0.0197	-11.28%	0.0239	8.08%
750	0.0134	0.0132	-1.57%	0.0181	34.90%
1000	0.0088	0.0099	13.14%	0.0146	66.36%

Table 8.2: Comparison of MAPE of FourEst. $TS4C_1$ and $TS4C_2$ for Barábasi-Albert graphs.

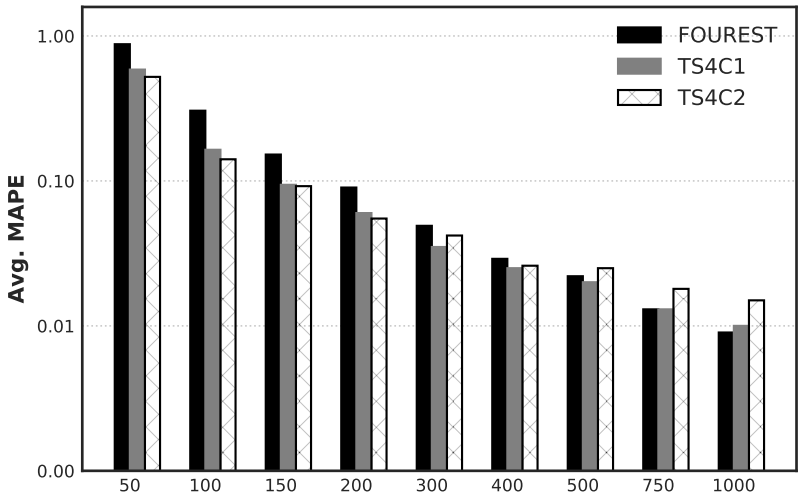


Figure 8.4: Average MAPE on Barábasi-Albert random graphs, with $n = 20000$ and various values of m.

Both TieredSampling algorithms consistently outperform FourEst for values of m up to 400, that is for fairly sparse graphs for which we expect 3 and 4 cliques to be rare patterns. The advantage over FourEst is particularly strong for values of m up to 200 with reductions of the average MAPE up to 30%. For denser graphs, i.e. $m \geq 750$, FourEst outperforms *both* TieredSampling algorithms. This in consistent with the intuition discussed in Section 8.6.2, as for denser graphs triangles are not "*rare enough*" to be worth saving over edges. Note however that in these cases the quality of *all* the estimators is very high (i.e., MAPE$\leq$ 1%). We can also observe that TS4C$_2$ outperforms TS4C$_1$ for $m \leq 200$. Vice versa, TS4C$_1$ outperforms TS4C$_2$ (and FourEst) for $300 \leq m \leq 750$. Since, as discussed in Section 8.4.2, TS4C$_2$ gives "*more importance*" to triangles, it works particularly well when the graph is very sparse, and triangles are particularly rare. As the graph grows denser (and the number of triangles increases), TS4C$_1$ performs better until, for highly dense graphs FourEst produces the best estimates.

8.7 Adaptive Tiered Sampling Algorithm

An appropriate partition of the available memory between the layers used in the TieredSampling approach is crucial for the success of the algorithm: while assigning more memory to the triangle sample allows to maintain more sub-patterns, removing too much space from the edge sample reduces the probability of observing new triangles.

While in Section 8.4.1 (resp., Section 8.4.2) we provide a general rule according to which to decide how to split the available memory space for TS4C$_1$ (resp., TS4C$_2$), such partition may not always lead to the best possible results. For instance, if the graph being observed is particularly sparse, assigning a large portion of the memory to the triangles would result in a considerable waste of memory space due to the low probability of observing triangles. Further, as discussed in Section 8.6, depending on the properties of $G^{(t)}$ an approach based on simply maintaining a sample of the edges could perform better than the TieredSampling algorithms. As in the graph streaming setting these properties are generally not known a priori, nor stable through the graph evolution, a fixed memory allocation policy appears not to be the ideal solution.

In this section, we present ATS4C, an *adaptive* variation of our TS4C$_2$ algorithm, which *dynamically* analyzes the properties of $G^{(t)}$ through time and consequently decides how to allocate the available memory.

Algorithm description: We present a step by step pseudocode description of ATS4C in Algorithm 18.ATS4C has two main "*execution regimens*": the "*initial regimen*" **(R1)** for which it behaves exactly as FourEst, and the "*stable regimen*" **(R2)** for which it behaves similarly to TS4C$_2$. **(R1)** is the initial regimen for ATS4C. Once the algorithm switches to **(R2)** it does not ever switch back to **(R1)**. ATS4C maintains an estimate $\varkappa^{(t)}$ of the number of 4-cliques observed on the stream up to time t. ATS4C increases $\varkappa$ each time a 4-clique is observed according to the same method

discusses for FOUREST while in **(R1)** (by invoking UPDATE4CLIQUEFIRSTREGIMEN, line 13 of Algorithm 18), and according to the same method discussed for TS4C$_2$while in **(R3)** (by invoking UPDATE4CLIQUESECONDREGIMEN, line 26 of Algorithm 18). Analogously to what done for the other algorithms discusses so far, towards guaranteeing that $\varkappa$ is an *unbiased estimator*, each time a 4-clique is detected ATS4C computes the probability p of such detection and increases the estimate by p^{-1}.

While in **(R1)**, every M time steps ATS4C decides, based on the number of triangles observed so far, whether to switch from **(R1)** to **(R2)**. Recall that, from Lemma 7.1 (resp., Lemma 8.11) the probability of a 4-clique whose last edge is observed at time t being observed by TS4C$_2$ (resp., FOUREST) is approximately $p_\alpha = (\min\{1, \alpha M/t\})^4(\min\{1, (1-\alpha)M/\tau^{(t)}\})^2$ (resp., $p_s = (\min\{1, M/t-1\})^5$), where $2/3 < \alpha < 1$ (resp., $0 < 1-\alpha < 1/3$) denotes the fraction of the available memory which is assigned to the edge sample (resp., triangle sample).SWITCH determines which partition of the available space between edge only sample and triangle sample would maximize the approximate probability of detecting a 4-clique using TS4C$_2$. That is, SWITCH computes $\alpha^* = \text{argmax}_{\alpha\in[2/3,1)} p_\alpha$ (line 31 Algorithm 18). SWITCH then evaluates if the approximate probability of detecting a clique using the FOUREST approach is higher than that achievable using TS4C$_2$ while partitioning the available memory according to α^* (line 32). If that is the case (i.e., $p_s > p_{\alpha^*}$), SWITCH returns zero and ATS4C elects to remain in **(R1)** (line 11). Vice versa, if $p_s \leq p_{\alpha^*}$, SWITCH returns the value α^* and ATS4C transitions to **(R2)**): the triangle reservoir $\mathcal{S}_\Delta$ is assigned $(1-\alpha^*)M$ memory space, and it is filled with the triangles composed by the edges *currently* in the edge reservoir, using the reservoir sampling scheme.
Finally the edge reservoir $\mathcal{S}_e$ is constructed by selecting $\alpha^* M$ of the edges in the current sample uniformly at random, thus ensuring that $\mathcal{S}_e$ is an *uniform sample* (lines 7-11). Once ATS4C switches to **(R2)** it never goes back to **(R1)**.

While in **(R2)**, as long as $|\mathcal{S}_\Delta| < M/3$, every M time steps ATS4C evaluates whether it is opportune to assign a higher fraction of the available memory to $\mathcal{S}_\Delta$. Let $t = iM$, rather than just using the information of the number of triangles seen so far $\tau^{(iM)}$, ATS4C computes a "*prediction*" of the total number of triangles seen until $(i+1)M$ assuming that the number of triangles seen during the next M steps will equal the number of triangles seen during the last M steps (line 53), that is $\tau^{(\tilde{(i+1)M})} = 2\tau^{(iM)} - \tau^{((i-)M)}$. ATS4C then evaluates the partitioning of the available memory space among the two tiers which would maximize the approximate probability of detecting a 4-clique at time $(i+1)M$, i.e., $\alpha^* = \text{argmax}_{a\in[2/3,1)}(\min\{1, \alpha M/(i+1)M)\})^4(\min\{1, (1-\alpha)M/\tau^{(\tilde{(i+1)M})}\})^2$. Let α denote the split being used by ATS4C at $t = iM$:

- if $\alpha^* > \alpha$, ATS4C determines that increasing the memory space devoted to the triangles sample would not improve the algorithm's likelihood of detecting 4-cliques. ATS4C continues its execution with no further operations (ATS4C never reduces the memory space assigned to $\mathcal{S}_\Delta$);

- otherwise, if $\alpha^* < \alpha$, ATS4C removes $(\alpha - \alpha^*)M$ edges from $\mathcal{S}_e$ selected uniformly at random, thus ensuring that $\mathcal{S}_e$ is still an uniform sample of the edges observed on the stream, and the

ALGORITHM 18 ATS4C - Adaptive Version of TieredSampling

Input: Insertion-only edge stream Σ, M

$\mathcal{S}_e \leftarrow \emptyset$, $\mathcal{S}_\Delta \leftarrow \emptyset$, $\mathcal{S}'_\Delta \leftarrow \emptyset$, $M'_\Delta \leftarrow 0$, $t \leftarrow 0$, $t_\Delta \leftarrow 0$, $\sigma \leftarrow 0$, $r \leftarrow 1$
for each element (u, v) from Σ **do**
  $t \leftarrow t + 1$
  if $r = 1$ **then** ▷ Operations while in **(R1)**
    if $t\%M = 0$ **then** ▷ Every M time steps evaluate whether to switch from **(R1)** to **(R2)**
      $\alpha \leftarrow$ SWITCH ▷ Compute recommended splitting coefficient
      if $\alpha > 0$ **then**
        $r \leftarrow 2$ ▷ Switch to **(R2)**
        $\mathcal{S}_\Delta \leftarrow$ CreateTriangleReservoir(α) ▷ Create and populate triangles sample $\mathcal{S}_\Delta$
        $\mathcal{S}_e \leftarrow$ Subsample($\alpha, \mathcal{S}_e$) ▷ Select uniformly at random a subset of size αM of $\mathcal{S}_e$
        Update4CliquesSecondRegimen($(u, v), t$) ▷ Update 4-clique estimate
    else
      Update4CliquesFirstRegimen($(u, v), t$) ▷ Remain in **(R1)**, update 4-clique estimate
  else if $r = 2$ **then** ▷ Operations while in **(R2)**
    if $t\%M = 0$ **then** ▷ Every M time steps update split proportions between S_e and S_Δ
      UpdateMemory ▷ Decide whet er to assign more space to S_Δ
    if $t'_\Delta < M_\Delta$ **then**
      UpdateTriangles($(u, v), t'_\Delta, S'_\Delta$)
    $p_\Delta \leftarrow min(1, \frac{M_\Delta}{t_\Delta})$, $p'_\Delta \leftarrow min(1, \frac{M'_\Delta}{t'_\Delta})$
    if $p_\Delta < p'_\Delta$ **then** ▷ Merge main triangle sample $\mathcal{S}_\Delta$ and temporary $\mathcal{S}'_\Delta$
      $S_{merged} \leftarrow \emptyset$
      for each $(x, y, z) \in S_\Delta$ **do**
        if FlipBiasedCoin($\frac{p'_\Delta}{p_\Delta}$) = heads **then** ▷ Select the triangles to be moved from $\mathcal{S}'_\Delta$ to $\mathcal{S}_\Delta$
          $S_{merged} \leftarrow S_{merged} \cup (x, y, z)$
      $S_\Delta \leftarrow S_{merged} \cup S'_\Delta$
    Update4CliquesSecondRegimen($(u, v), t$) ▷ Update 4-clique estimate
    UpdateTriangles($(u, v), S_\Delta$) ▷ Update triangles sample
  SampleEdge($(u, v), t$) ▷ Update edges sample
function Switch ▷ Decide whether to switch from **(R1)** to **(R2)**
  $p_s \leftarrow min(1, \frac{M}{t})^5$
  $p_\alpha \leftarrow min(1, \alpha\frac{M}{t})^4 \cdot min(1, (1-\alpha)\frac{M}{t_\Delta})^2$ **where** $a = argmax_{\alpha \in [\frac{2}{3}, 1]} p_\alpha$
  if $p_s < p_\alpha$ **then**
    return α ▷ ATS4C switches from **(R1)** to **(R2)**
  else
    return 0 ▷ ATS4C remains in **(R1)**

freed space is assigned to $\mathcal{S}_\Delta$. Let us denote this space as $\mathcal{S}'_\Delta$.

As ATS4C progresses and observes new triangles it fills $\mathcal{S}'_\Delta$ using the reservoir sampling scheme. That is $\mathcal{S}'_\Delta$ is used a *temporary buffer* to store observed triangle patterns. $\mathcal{S}_\Delta$ and $\mathcal{S}'_\Delta$ are then *merged* at the first time step for which the probability p_Δ of a triangle seen before the creation of $\mathcal{S}'_\Delta$ being in $\mathcal{S}_\Delta$ becomes lower than the probability $p_{\Delta'}$ of a triangle observed after the creation

36: **function** CreateTriangleReservoir(α)
37: $i \leftarrow 1$
38: $M_\Delta = (1-\alpha)M$
39: **while** $|S_\Delta| < M_\Delta$ **do**
40: $(x,y)^{(i)} \leftarrow$ *Edge observed at time* i
41: **for each** element z from $\mathcal{N}^{\mathcal{S}^{(i)}_{x,y}}$ **do**
42: $t_\Delta \leftarrow t_\Delta + 1$
43: $S_\Delta \leftarrow S_\Delta \cup (x,y,z)$
44: **while** $|S| > M - M_\Delta$ **do**
45: $(v,w) \leftarrow$ *random edge from* S
46: $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(v,w)\}$
47: **function** UpdateMemory
48: $t_\Delta^{(i+1)M} = 2t_\Delta^{(iM)} - t_\Delta^{((i-1)M)}$ ▷ Prediction of number of triangles at (i+1) step
49: $p_\alpha \leftarrow min(1, \alpha\frac{M}{t})^4 \cdot min(1, (1-\alpha)\frac{M}{t_\Delta^{(i+1)M}})^2$ **where** $a = argmax_{\alpha \in [\frac{2}{3},1]} p_\alpha$
50: **if** $\alpha < \frac{M_\Delta}{M}$ **then**
51: **for** $i \in [1, \frac{M_\Delta}{M} - \alpha]$ **do**
52: $(v,w) \leftarrow$ *random edge from* S
53: $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(v,w)\}$
54: $M_\Delta \leftarrow M_\Delta + 1$, $M'_\Delta \leftarrow M'_\Delta + 1$
55: **function** UpdateTriangles($(u,v),t$) ▷ Update $\mathcal{S}_\Delta$
56: $\mathcal{N}^{\mathcal{S}}_{u,v} \leftarrow \mathcal{N}^{\mathcal{S}}_u \cap \mathcal{N}^{\mathcal{S}}_v$
57: **for each** element w from $\mathcal{N}^{\mathcal{S}}_{u,v}$ **do**
58: $t_\Delta \leftarrow t_\Delta + 1$
59: SampleTriangle(u,v,w)
60: **function** Update4CliquesFirstRegimen($(u,v),t$))
61: Update $\varkappa$ using procedure Update4Cliques($(u,v),t$) of FourEst algorithm
62: **function** Update4CliquesSecondRegimen($(u,v),t$)
63: Update $\varkappa$ using procedure Update4Cliques($(u,v),t$) of TS4C$_2$ algorithm

of $\mathcal{S}'_\Delta$ being in it. The merged triangle sample contains all the triangles in $\mathcal{S}'_\Delta$, while the triangles in $\mathcal{S}_\Delta$ are moved to it with probability $p_{\Delta'}/p_\Delta$. This ensures that after the merge all the triangles seen on the stream are kept in the triangle reservoir with probability $p_{\Delta'}$. After the merge, ATS4C operates the samples as described in TS4C$_2$. Finally, ATS4C increases the memory space for $\mathcal{S}_\Delta$ only if all the currently assigned space is used. Once $\mathcal{S}_\Delta$ is filled, ATS4C maintains it a reservoir sample for the triangle observed on the stream (invoking the UpdateTraingles function, line 55).

The analysis of ATS4C presents several additional challenges compared to those of our previous algorithms TS4C$_1$ and TS4C$_2$, due to the interplay between execution regimens. Still, as during **(R2)** (resp., **(R1)**) ATS4C behaves effectively as TS4C$_2$ (resp., FourEst), albeit with some further complication due to the adjustment of the assignment of memory between layer, whenever a 4-clique is observed, ATS4C may compute *exactly* the probability of observing such clique, by relying on the properties of TS4C$_2$ and FourEst.

Theorem 8.18. *The estimator $\varkappa$ returned by* ATS4C *is unbiased, that is $\varkappa^{(t)} = |\mathcal{C}_k^{(t)}|$ if $t \leq M$. Further, $\mathbb{E}\left[\varkappa^{(t)}\right] = |\mathcal{C}_k^{(t)}|$ if $t \geq M$.*

The proof of Theorem 8.18, follows a reasoning analogous to that discussed in similar results for the unbiasedness of TS4C$_2$ and FourEst, and relies on the fact that, as discussed in the algorithm description, whenever a 4-clique is observed, ATS4C can compute *exactly* the probability of observing such clique.

We show how the adaptive assignment of the available memory realized by ATS4C succeeds in combining the advantages of the single edge sample approach (e.g., FourEst), and of the multi-tier prototype-based approach of TieredSampling via experimental evaluation on real-world graphs presented in Section 8.8.2.

8.8 Experimental Evaluation

In this section, we evaluate through extensive experiments the performance of our proposed TieredSampling method when applied for counting 4-cliques and 5-cliques in large graphs observed as streams. We use several real-world graphs with size ranging from 10^6 to 10^8 edges (see Table 8.3 for a complete list). All graphs are treated as undirected. The edges are observed on the stream according to the values of the associated timestamps if available, or in random order otherwise. In order to evaluate the accuracy of our algorithms, we compute the "*ground truth*" exact number of 4-cliques (resp., 5-cliques) for each time step using an exact algorithm that maintains the entire $G^{(t)}$ in memory. Besides verifying the accuracy of the proposed TieredSampling methods, an important goal of the experimental analysis is to ascertain the benefits of our multi-tier method compared to approaches for which the estimate is achieved by maintaining simply a reservoir sample of the observed edges. Therefore, our main term of comparison will be the FourEst algorithm which is a generalization of the TRIEST algorithm [51]. Our experiments are implemented in Python and their source code can be found in (*https://github.com/erisaterolli/TieredSampling*). The experiments were run on the Brown University CS department cluster[3], where each run employed a single core and used at most 4GB of RAM.

The section is organized as follows: we first evaluate the performance of TS4C$_1$ and TS4C$_2$ and we compare them with the estimations obtained using FourEst. We then present practical examples that motivate the necessity for the adaptive version of our TieredSampling approach, and we show how our ATS4C manages to capture the best of the single and multi-level approach. Finally, we show how the TieredSampling approach can be generalized in order to count structures other than 4-cliques.

8.8.1 Counting 4-Cliques

We estimate the global number of 4-cliques on insertion-only streams, starting as empty graphs and for which an edge is added at each time step, using algorithms FourEst, TS4C$_1$ and TS4C$_2$. As discussed in Section 8.4.1 (resp., Section 8.4.2), in TS4C$_1$ (resp., TS4C$_2$) we split the total available

[3]https://cs.brown.edu/about/system/services/hpc/grid/

Graph	Nodes	Edges	Exact 4-clique count	TieredSampling $M = 0.05\|E\|$	Source
DBLP	986,324	3,353,618	40,675,407	41,750,428	[26]
Patent (Cit)	2,745,762	13,965,132	3,296,890	3,294,045	[51]
LastFM	681,387	30,311,117	46,201,534,449	49,189,084,815	[51]
Live Journal	5,363,186	49,514,271	16,121,317,106	16,035,963,528	[26]
Hollywood	1,917,070	114,281,101	728,184,767,782	727,002,104,249	[26]
Orkut	3,072,441	117,185,083	3,221,163,953	3,210,957,578	[162]
Twitter	25,080,769	100,000,000	19,920,704	20,562,810	[26] [51]

Table 8.3: Graphs used in the experiments

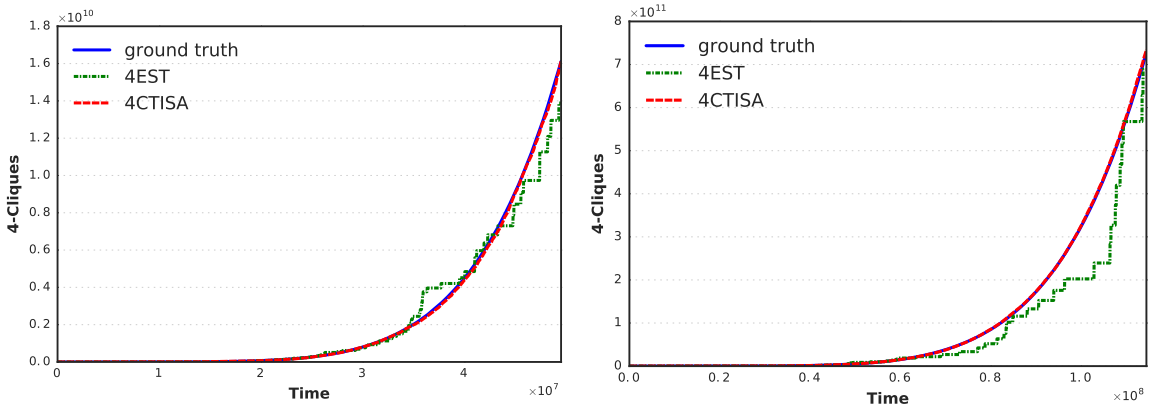
	M = 0.01				M = 0.02				M = 0.05			
	FOUR EST	TS4C1	TS4C2	TS Change	FOUR EST	TS4C1	TS4C2	TS Change	FOUR EST	TS4C1	TS4C2	TS Change
DBLP	0.942	0.658	0.599	-36.00%	0.915	0.336	0.526	-63.28%	0.114	0.069	0.105	-39.47%
Holly wood	0.056	0.029	0.023	-59.00%	0.013	0.016	0.01	-23.08%	0.004	0.006	0.002	-50.00%
Last FM	0.07	0.20	0.196	65.00%	0.022	0.239	0.10	354.54%	0.003	0.0304	0.032	866.67%
Live Journal	0.295	0.079	0.148	-73.00%	0.098	0.02	0.025	-79.59%	0.019	0.006	0.006	-68.42%
Orkut	0.789	0.224	0.086	-89.00%	0.116	0.059	0.058	-50.00%	0.013	0.021	0.008	-38.46%
Patent Cit	0.954	0.767	0.188	-80.00%	0.473	0.304	0.141	-70.19%	0.112	0.113	0.039	-65.18%
Twitter	0.928	0.896	0.511	-45.00%	0.928	0.481	0.428	-53.88%	0.205	0.076	0.088	-62.93%

Table 8.4: Average MAPE of various approaches for all graphs with available memories of 1%, 2% and 5% of the size of the graph. In the TS Change column we report the decrease/increase in MAPE of the best performing TieredSampling algorithm among TS4C_1 and TS4C_2 compared to the single reservoir algorithm FourEst.

memory space M as $|\mathcal{S}_e| = 4M/5$ and $|\mathcal{S}_\Delta| = M/5$ (resp., $|\mathcal{S}_e| = 2M/3$ and $|\mathcal{S}_\Delta| = M/3$). The experimental results show that these fixed memory splits perform well for most cases. We then experiment with an adaptive splitting mechanism that handles the remaining cases.

In Fig. 8.5 we present the estimation obtained by averaging 10 runs of respectively TS4C_1, TS4C_2 and FourEst using total memory space $M = 5 \times 10^5$ for the LiveJournal and Hollywood graphs (i.e., respectively using less than 1% and 0.5% of the graph size). While the average of the runs for TS4C_1 and TS4C_2 are almost *indistinguishable* from the ground truth, the quality of the estimator obtained using FourEst considerably worsens as the graph size increases.

In Table 8.4, we report the average MAPE performance over 5 runs for TS4C_1, TS4C_2 and FourEst for graphs listed in Table 8.3. For each graph, we assign a different total memory space equal to a 1%, 2% and 5% fraction of the total number of edges (i.e., of the size of the graph). Except for LastFM, our TieredSampling algorithms clearly and consistently outperform FourEst for all the considered available memory sizes with the average MAPE reduced by up to 89%. While the performances of all algorithms are improved (in terms of MAPE reduction) as the size of the available memory increases, the improvement achieved by or TieredSampling algorithms with respect to FourEst appear to consistent for the various memory sizes being considered. While on most graphs



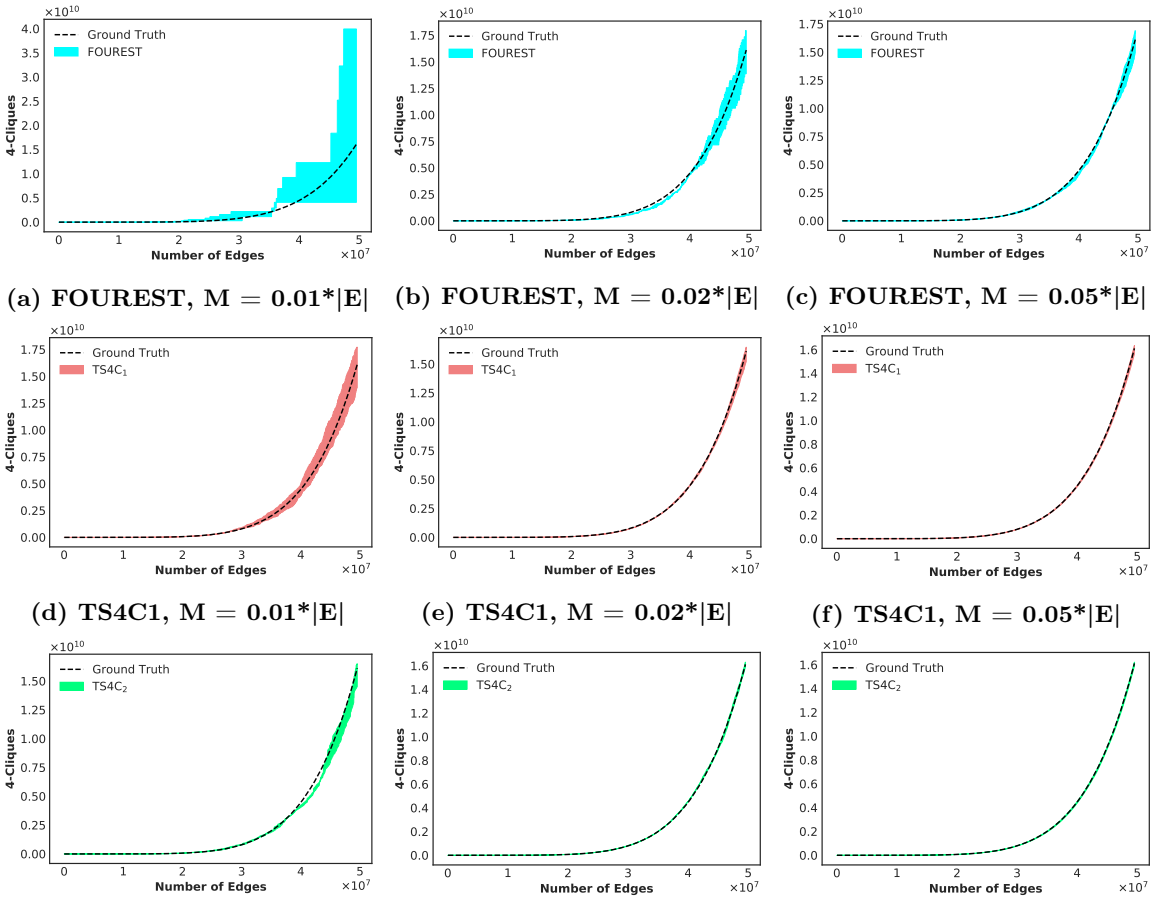
(a) Live Journal **(b) Hollywood**

Figure 8.5: Comparison of $|\mathcal{C}_4^{(t)}|$ estimates obtained using TS4C$_1$, TS4C$_2$ and FourEst with $M = 5 \times 10^5$,

TS4C$_1$ and TS4C$_2$ produce similar estimates (and, hence, MAPE), it can be noted that TS4C$_2$ clearly outperforms TS4C$_1$ in graphs for which the number of 4-cliques is low compared to the number of edges, such as Patent (Cit) and Twitter. In these graphs, triangles are "*rare enough*" so that assigning a greater fraction of the available memory to maintain their occurrences and relying on observed triangles to count 4-cliques leads to tighter estimates. LastFM is the only graph for which FourEst (considerably) outperforms the TieredSampling algorithms. Such phenomenon is due to the high density of the graph $|E|/|V| > 500$ and to the fact that for the LastFM graph triangles are not a rare enough sub-structure to justify the choice of maintaining them over simple edges.

We analyze the variance reduction achieved using our TieredSampling algorithms by comparing the empirical variance observed over ten runs for all graphs using available memory M equal to 1%, 2% and 5% of the size of the graphs. The results for Live Journal graph are reported in Fig. 8.6. While for both TS4C$_1$ and TS4C$_2$ the minimum and maximum estimators are close to the ground truth throughout the evolution of the graph even in cases having a small amount of available memory, FourEst estimators exhibit very high variance especially towards the end of the stream and when the available memory is small. This trend is consistently observed for all the other graphs. As an illustrative example, we show the variance of all algorithms for Twitter with an available memory size of 5% of the total graph.

Despite the differences between TS4C$_1$ and TS4C$_2$, their variances appear to be strikingly similar for the evaluated graph, with TS4C$_2$ exhibiting slightly better performance, especially during the first half of the stream. This is due to the rarity of the triangle prototype pattern used in the detection of 4-cliques for the Hollywood graph: compared to TS4C$_1$, TS4C$_2$ devotes a higher fraction of the available memory to $\mathcal{S}_\Delta$ which leads to a higher likelihood of detecting 4-cliques and, hence, a lower empirical variance.



(a) FOUREST, M = 0.01*|E| **(b) FOUREST, M = 0.02*|E|** **(c) FOUREST, M = 0.05*|E|**

(d) TS4C1, M = 0.01*|E| **(e) TS4C1, M = 0.02*|E|** **(f) TS4C1, M = 0.05*|E|**

(g) TS4C2, M = 0.01*|E| **(h) TS4C2, M = 0.02*|E|** **(i) TS4C2, M = 0.05*|E|**

Figure 8.6: Variance of various approaches for Live Journal with available memories of 1%, 2% and 5% of the size of the graph.

Our experiments not only verify that TS4C$_1$ and TS4C$_2$ allow to obtain good quality estimations which are in most cases (except for the LastFM graph) superior to the ones achievable using a single sample strategy, but also validate the general intuition underlying the TieredSampling approach.

Both TieredSampling algorithms are extremely scalable, showing average update times in the order of hundreds microseconds for all graphs as shown in Table 8.5.

8.8.2 Adaptive Tiered Sampling

In Section 8.8.1, we showed that TS4C$_2$, allows to obtain high quality estimations for the number of 4-cliques outperforming in most cases both TS4C$_1$ and FourEst. These results were obtained splitting the available memory such that $|\mathcal{S}_e| = 2M/3$ and $|\mathcal{S}_\Delta| = M/3$. As discussed in Section 8.7, while this is a useful general rule, depending on the properties of the graph, different splitting rules may yield better results. We verify this fact by evaluating the performance of TS4C$_2$ when used to analyze the Patent(Cit) graph using different assignments of the total space $M = 5 \times 10^5$ to the

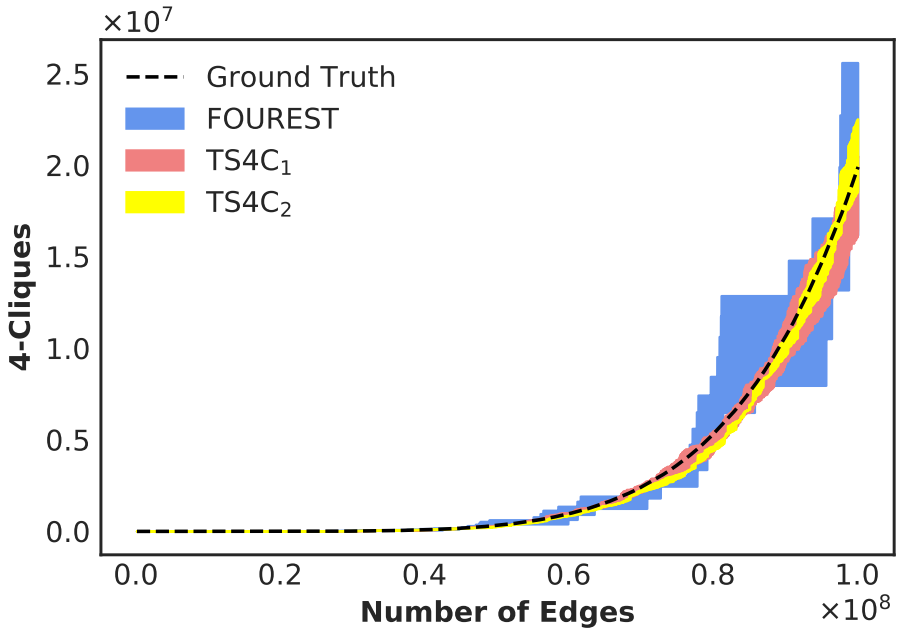


Figure 8.7: Variance of various approaches for Twitter with available memory of 5% of the size of the graph.

Dataset	FourEst	TS4C$_1$	TS4C$_2$
DBLP	6.56	19.6	15.8
Hollywood	10.95	15.75	19.95
Lastfm	17.65	45.62	64.92
Live Journal	9.22	10.72	16.49
Orkut	5.55	8.93	7.45
PatentCit	9.25	11.95	12.12
Twitter	9.08	20.49	17.77

Table 8.5: Average update time for all graphs measured in microseconds.

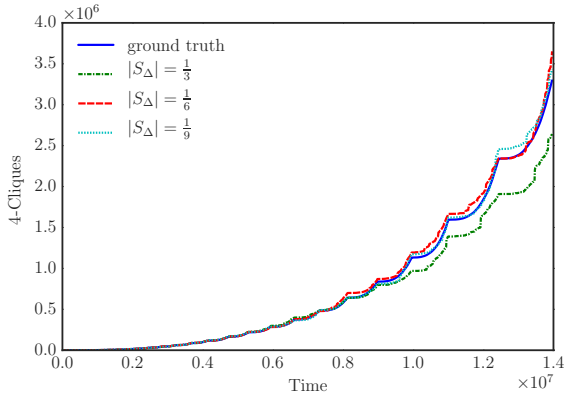
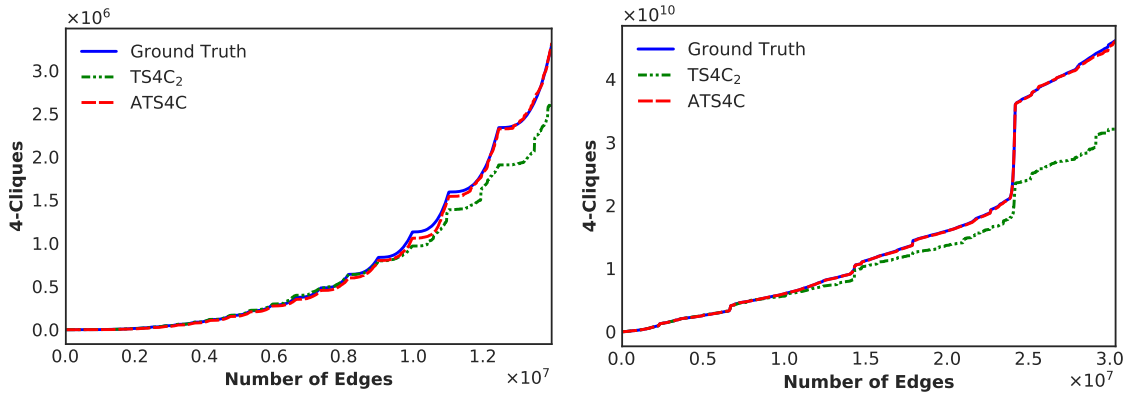


Figure 8.8: $|\mathcal{C}_4^{(t)}|$ estimations for Patent (Cit) using TS4C$_2$ with $M = 5{\times}10^5$ and different memory space assignments among tiers.



(a) Patent(Cit.) **(b) [LastFM**

Figure 8.9: Comparison of $|\mathcal{C}_4^{(t)}|$ estimates obtained using ATS4C, TS4C$_2$, and FourEst with $M = 5 \times 10^5$.

two levels. The results in Fig. 8.8 show that decreasing the space assigned to the triangle sample from $M/3$ to $M/9$ allows to achieve estimates that are closer to the ground truth leading to a 31% reduction of the average MAPE. Due to the sparsity of the Patent(Cit) graph, TS4C$_2$ observes a very small number of triangles for a large part of the stream. Assigning a large fraction of the memory space to $\mathcal{S}_\Delta$ is thus inefficient as the probability of observing new triangles is reduced, and the space assigned to $\mathcal{S}_\Delta$ is not fully used.

To overcome such difficulties, in Section 8.7 we introduced ATS4C, an *adaptive* version of TS4C$_2$, which allows to dynamically adjust the use of the available memory space based on the properties of the graph being observed. We experimentally evaluate the performance of ATS4C over ten runs on the Patent(Cit) and the LastFM graphs and we compare it with TS4C$_2$ and FourEst using $M = 5 \times 10^5$.

As shown in Fig. 8.9, for both graphs, ATS4C produces estimates that are nearly indistinguishable from the ground truth. ATS4C clearly outperforms TS4C$_2$ (with $|\mathcal{S}_\Delta| = \frac{M}{3}$) on Patent(Cit) where triangles are sparse motifs achieving an $\sim 85\%$ reduction of the average MAPE compared to TS4C$_2$. ATS4C returns high quality estimations even for the LastFM graph, for which the single level approach FourEst outperforms TS4C$_1$ and TS4C$_2$.

8.9 Generalizing the TieredSampling approach

While so far we focused on counting the number of occurrences of 4-cliques in a graph stream, the TieredSampling approach can be generalized towards estimating the counts of a wide variety of graphlets. In this section, we discuss an heuristic which allows generalizing the approach discussed so far to a general graphlet. As an example, we discuss and application of the TieredSampling approach that yields high-quality estimates of the number of 5-cliques in graph streams, named TS5C.

8.9.1 The TieredSampling framework

Let $G' = (V', E')$ be the graph or pattern of interest for whom we aim to estimate the number of occurrences in the graph stream. Towards obtaining an algorithm $\mathcal{A}$, which provides such estimates, we apply the TieredSampling approach following the following steps:

1. **Identify a "prototype" sub-pattern:** We identify a sub-pattern of G', henceforth denoted as $G'' = (V'', E'')$ such that $V'' \subseteq V'$ and $E'' \subseteq E'$. G'' will play a role analogous to that played by triangles towards counting 4-cliques in the stream. Such sub-pattern serves as a *prototype* to aid in the detection of instances of the graphlet of interest. Part of the memory available to algorithm $\mathcal{A}$ will be allocated to maintaining a reservoir sample of the instances of G'' observed on the stream by $\mathcal{A}$ so far, henceforth named $\mathcal{S}_P$, while the remaining available memory will be used as a uniform sample of the edges observed on the stream, henceforth named $\mathcal{S}_e$. Both samples are maintained by $\mathcal{A}$ according to the reservoir sampling mechanism, following steps analogous to those discussed for TS4C$_1$ and TS4C$_2$. Further, we assume that whenever an occurrence of G'' is selected to be included in $\mathcal{S}_P$, $\mathcal{A}$ maintains information on the time step at which it was detected.

   Algorithm $\mathcal{A}$ operates similarly to TS4C$_1$ and TS4C$_2$: whenever an edge e is observed in the steam at time t $\mathcal{A}$ verifies whether an instance of G' can be obtained by combining e with one of the prototypes in $\mathcal{S}_P^{(t)}$ and with *exactly* $|E'|-|E''|-1$ edges in $\mathcal{S}_e^{(t)}$. $\mathcal{A}$ maintains an estimate $\varkappa^{(t)}$ of the number of occurrences of G' in the stream up to time t which is updated each time an instance of G' is detected. Then $\mathcal{A}$ determines whether an instance of the prototype G'' can be detected combining e with $|E''|-1$ edges in $\mathcal{S}_e^{(t)}$. Each detected occurrence G'' is submitted to the reservoir $\mathcal{S}_P$. Finally, $\mathcal{A}$ inserts e into the reservoir sample $\mathcal{S}_e$.

While G'' can be chosen arbitrarily among the sub-graphs of G' some properties appear to be desirable: (i) G'' should be *rare enough* so that it is worth maintaining instances of it in memory but not *overly rare* so that part of the available memory would not be utilized (unless using the adaptive version of TieredSampling); (ii) G'' should cover a significant fraction of the edges of G'', that is, it should be possible to complete an instance of G' by adding a small number of edges to G'; (iii) G'' should be *connected*. Property (i) is desirable since, as shown for the case of 4-cliques, under such circumstances $\mathcal{A}$ achieves efficient use of the available memory and the benefit of maintaining *rare* instances of the prototype towards detecting G'. Property (ii) is desirable as it allows for a simpler and less computational intensive detection of instances of G' by composing instances of G'' and a few remaining edges. Finally, property (iii) is desirable as connected sub-graphs are, in general, rarer than non-connected graphs with the same number of edges, and they allow for a simpler - less computationally intensive - algorithmic criteria for detecting occurrences of G' by completing occurrences of the prototype G''. While any subgraph satisfying these properties appears desirable to be used as a prototype, as a heuristic, choosing as G'' a clique sub-graph of G' satisfies the desired properties.

2. **Bounding the probability of observing G':** Algorithm $\mathcal{A}$ maintains an estimator $\varkappa(t)$ of the number of occurrences of G' observed on the stream up to step t. Ideally, $\mathcal{A}$ would proceed by increasing $\varkappa(t)$ each time an occurrence of G' is detected on the stream by an amount which corresponds to the reciprocal of the probability of such occurrence being observed by $\mathcal{A}$. As discussed in Theorem 8.5 (resp., Theorem 8.12) for TS4C$_1$ (resp., TS4C$_2$), this does indeed guarantee $\varkappa(t)$ to be an *unbiased estimate* of the number of occurrences of G'. However, computing *exactly* the probability of observing G' and G'' is, in general, rather complicated as it requires breaking down a high number of sub-cases each tied to the order according to which the edges are observed on the stream and whose number does, therefore, grow *exponentially* with respect to the number of edges of G'. While it is possible to reduce the number of these cases by matching some common patterns, the analysis is still rather complex and not easily generalize from one sub-graph G' to another G'. Rather than computing the exact value, using an approximate of the probability according to whom $\mathcal{A}$ observes an instance of G' (resp., G'') is generally sufficient to obtain good quality, albeit generally non-unbiased, estimates for the count of occurrences of G' while considerably reducing the effort of the analysis.

   Let t denote the time step at whom the last edge e of an occurrence of G', denoted as $\hat{G}' = (\hat{V}', \hat{E}')$, is observed on the stream. Based on the choice of the prototype, it will be possible for $\mathcal{A}$ to detect $\hat{G}'$ by composing an occurrence of the prototype G'' (whose edges has been previously observed on the stream),which may have been previously detected and maintained in the reservoir sample used to maintain instances of G'', with some edges currently stored in the edge reservoir sample, and e itself. Assume that $\mathcal{A}$ detects by composing an occurrence of G'', henceforth denoted as $\hat{G}'' = (\hat{V}'', \hat{E}'')$. stored in $\mathcal{S}_{\Delta}^{(t)}$, $|E'| - |E''| - 1$ edges of $\hat{G}''$ in $\mathcal{S}_{e}^{(t)}$, and the edge e itself. Let p denote the probability of $\mathcal{A}$ observing $\hat{G}'$ in such a way, $\mathcal{A}$

approximates p as

$$\tilde{p} = \Pr\left(\hat{G}'' \in \mathcal{S}_\Delta\right) \Pr\left(\hat{V}' \setminus (\hat{V}'' \cup \{e\}) \subseteq S_e^{(t)}\right)$$
$$= \Pr\left(\hat{G}'' \in \mathcal{S}_\Delta | \hat{G}'' \text{ was observed by } \mathcal{A}\right) \Pr\left(\hat{G}'' \text{ was observed by } \mathcal{A}\right) \Pr\left(\hat{V}' \setminus (\hat{V}'' \cup \{e\}) \subseteq \mathcal{S}_e^{(t)}\right).$$

- $\mathcal{A}$ approximates $\Pr\left(\hat{G}'' \in \mathcal{S}_\Delta | \hat{G}'' \text{ was observed by } \mathcal{A}\right)$, as the probability that, conditioned on $\hat{G}''$ having been observed by $\mathcal{A}$, $\hat{G}''$ is stored in the the reservoir $\mathcal{S}_P$ at time t, that is:
$$\Pr\left(\hat{G}'' \in \mathcal{S}_\Delta | \hat{G}'' \text{ was observed by } \mathcal{A}\right) \approx \min\left\{1, \frac{|\mathcal{S}_P|}{\tau^{(t)}}\right\},$$
where $|\mathcal{S}_P|$ denotes the size of the fraction of the memory space allocated to $\mathcal{S}_P$ and $\tau^{(t)}$ denotes the number of instances of G'' detected by $\mathcal{A}$ up to time $t-1$.
- $\mathcal{A}$ approximates $\Pr\left(\hat{G}'' \text{ was observed by } \mathcal{A}\right)$ as the probability that when the last edge of $\hat{G}''$ arrived on the stream its remaining $|E''|-1$ were stored in $\mathcal{S}_e$. That is:
$$\Pr\left(\hat{G}'' \text{ was observed by } \mathcal{A}\right) \approx \left(\min\left\{1, \frac{|\mathcal{S}_e|}{t'-1}\right\}\right)^{|E''|-1},$$
where $|\mathcal{S}_e|$ denotes the size of the fraction of the memory space allocated to $\mathcal{S}_e$ and t' is the time step at which the last edge of $\hat{G}''$ was observed on the stream. The approximate used here is a simplified version of the exact value computed in Lemma 8.1.
- $\mathcal{A}$ approximates $\Pr\left(\hat{V}' \setminus (\hat{V}'' \cup \{e\})\right)$ as the probability of the $|E'| - |E''| - 1$ edges in $\hat{V}' \setminus (\hat{V}'' \cup \{e\}$ being included in $\mathcal{S}_\urcorner$. That is:
$$\Pr\left(\hat{V}' \setminus (\hat{V}'' \cup \{e\})\right) \approx \left(\min\left\{1, \frac{|\mathcal{S}_e|}{t-1}\right\}\right)^{|E'|-|E''|-1}.$$
The approximate used here is a simplified version of the exact value computed in Lemma 8.1.

Hence, we have:

$$\tilde{p} = \min\left\{1, \frac{|\mathcal{S}_P|}{\tau^{(t)}}\right\} \left(\min\left\{1, \frac{|\mathcal{S}_e|}{t'-1}\right\}\right)^{|E''|-1} \left(\min\left\{1, \frac{|\mathcal{S}_e|}{t-1}\right\}\right)^{|E'|-|E''|-1}.$$

In general we have that $p \neq \tilde{p}$. In most cases $\tilde{p}$ is lower than than the actual value p.

3. **Determining increments to the estimate $\varkappa$:** The computed value $\tilde{p}$ approximates the probability of $\mathcal{A}$ detecting $\hat{G}'$ specifically using the instance $\hat{G}''$ of the prototype completed with edges form $\mathcal{S}_e$. However, in general it will be possible for $\mathcal{A}$ to detect the same occurrence $\hat{G}'$ of the pattern of interest. A similar circumstance was discussed for TS4C_1, as the algorithm can detect a 4-clique in two possible ways (i.e., using two possible triangle sub-graphs). Correcting for such phenomena is of crucial importance towards avoiding overestimating the number of occurrences of G'. In order to do so, it is necessary to determine the number of possible ways for $\mathcal{A}$ to detect $\hat{G}'$ given the order of the edges on the stream. Let e denote the last edge of $\hat{G}'$ observed on the stream. If/when $\mathcal{A}$ detects $\hat{G}'$, it counts the number of possible distinct occurrences of the prototype G'' in the sub-graph $(\hat{V}', \hat{E}' \setminus \{e\})$ (i.e., the sub-graph of $\hat{G}'$ obtained by removing e). By construction of $\mathcal{A}$, such number, henceforth denoted as $c(\hat{G}')$

corresponds to the number of possible ways according to whom $\mathcal{A}$ may detect $\hat{G}'$ by combining e, an occurrence of the prototype G'', and exactly $|V'| - |V''| - 1$ edges from $\mathcal{S}_e$. $c(\hat{G}')$ can be derived only from e and $\hat{G}'$ which are both known to $\mathcal{A}$ if/when $\hat{G}'$ is detected.

Whenever $\mathcal{A}$ observes an instance of G' it increases $\varkappa$ by $\tilde{p}^{-1}/c(\hat{G}')$. For each occurrence $\hat{G}'$ of G' let $\varkappa_{hatG'}$ be the random variable which corresponds increment to the estimate $\varkappa$ due to $\hat{G}'$. That is, $\varkappa = \sum_{\text{occurrences of G'}} \varkappa_{\hat{G}'}$. Towards $\varkappa$ being an unbiased estimate we therefore desire $\mathrm{E}\left[\varkappa_{\hat{G}'}\right] = 1$. By construction of $\mathcal{A}$ we have:

$$\begin{aligned}
\mathrm{E}\left[\varkappa_{\hat{G}'}\right] &= \sum_{\text{ways for } \mathcal{A} \text{ to detect } \hat{G}'} \frac{\tilde{p}^{-1}}{c(\hat{G}')} \Pr\left(\mathcal{A} \text{ detects } \hat{G}' \text{ in the } i\text{-th way}\right) \\
&\approx \frac{1}{c(\hat{G}')} \sum_{\text{ways for } \mathcal{A} \text{ to detect } \hat{G}'} \tilde{p}^{-1}\tilde{p} \\
&= \frac{c(\hat{G}')}{c(\hat{G}')} \\
&= 1.
\end{aligned}$$

While, for different combinations of prototypes and edges (i.e., the *ways*) the probabilities of detecting $\hat{G}'$, using the approximate $\tilde{p}$ allows for a considerable simplification of the analysis (and, in turn, of the algorithm) while still providing good estimates. While the criteria used deciding how to increment $\varkappa$ aims to have it function as an unbiased estimate of the number of occurrences of G', due to the use of the approximate $\tilde{p}$, $\varkappa$ is not, in general, an actual unbiased estimator. Since, as previously discussed, we have that, in general, $\tilde{p} < p$, our estimator $\varkappa$ will generally *slightly underestimate* the number of occurrences of the pattern of interest.

4. **Partition of the available memory space among reservoir tiers:** Following the heuristic already used for determining how to partition the available space discussed for TS4C$_1$ and TS4C$_2$, toward maximizing the probability of $\mathcal{A}$ detecting instances of G', we assign $|\mathcal{S}_P|$ and $|\mathcal{S}_\urcorner|$ in such a way that $\tilde{p}$ is maximized, while setting t, t' and $\tau^{(t)}$ as constants.

By following the outlined steps it is possible to deploy the TieredSampling approach to obtain an algorithm $\mathcal{A}$ that produces high-quality estimated of the number of occurrences of the desired pattern G'. While the construction of $\mathcal{A}$ previously discussed lead to obtaining an algorithm for which the memory space is statically distributed between the edge-only reservoir and the prototype reservoir, a generalization of the adaptive algorithm ATS4C can be obtained following similar reasoning.

8.9.2 Using TieredSampling to count 5-Cliques

To demonstrate the generality of our TieredSampling approach, we present TS5C, a one-pass counting algorithms for 5-clique in a stream. Following the steps outlined in the previous section, TS5C selects a 4-clique as a prototype sub-pattern to detect 5-cliques. The algorithm maintains two reservoir samples, one for edges and one for 4-cliques. When the currently observed edge completes

ALGORITHM 19 TS5C - Tiered Sampling for 5-Clique counting

Input: Insertion-only edge stream Σ, integers M, M_C
$\mathcal{S}_e \leftarrow \emptyset$, $\mathcal{S}_P \leftarrow \emptyset$, $t \leftarrow 0$, $\tau \leftarrow 0$, $\varkappa \leftarrow 0$
for each element (u, v) from Σ **do** ▷ Process each edge (u, v) coming from the stream in discretized timesteps
  $t \leftarrow t + 1$
  UPDATE5CLIQUES(u, v) ▷ Update the 5-clique estimator by considering the new 5-cliques closed by edge (u,v)
  UPDATE4CLIQUES(u, v) ▷ Update the 4-clique sample with the new 4-clique observed according to RS scheme
  SAMPLEEDGE$((u, v), t)$ ▷ Update the edges sample with the edge (u, v) according to RS scheme

function UPDATE5CLIQUES$((u, v), t)$
  for each 4-clique $(u, x, w, z, t') \in \mathcal{S}_P$ **do**
    if $(v, x) \in \mathcal{S}_e \wedge (v, w) \in \mathcal{S}_e \wedge (v, z) \in \mathcal{S}_e$ **then**
      $\tilde{p} \leftarrow \min\left\{1, \frac{M_P}{\tau}\right\}\left(\min\left\{1, \frac{M_e}{t-1}\right\}\right)^3\left(\min\left\{1, \frac{M_e}{t'-1}\right\}\right)^5$ ▷ Approximate probability of detecting a 5-clique
      $\varkappa \leftarrow \varkappa + \tilde{p}^{-1}/2$ ▷ Estimate update accounting for two possible ways of detecting 5-cliques
  for each 4-clique $(v, x, w, z, t') \in \mathcal{S}_\Delta$ **do**
    if $(u, x) \in \mathcal{S}_e \wedge (u, w) \in \mathcal{S}_e \wedge (u, z) \in \mathcal{S}_e$ **then**
      $\tilde{p} \leftarrow \min\left\{1, \frac{M_P}{\tau}\right\}\left(\min\left\{1, \frac{M_e}{t-1}\right\}\right)^3\left(\min\left\{1, \frac{M_e}{t'-1}\right\}\right)^5$
      $\varkappa \leftarrow \varkappa + \tilde{p}^{-1}/2$

function UPDATE4CLIQUES$((u, v), t)$
  $\mathcal{N}^{\mathcal{S}}_{u,v} \leftarrow \mathcal{N}^{\mathcal{S}}_u \cap \mathcal{N}^{\mathcal{S}}_v$
  for each pair (w, z) from $\mathcal{N}^{\mathcal{S}}_{u,v} \times \mathcal{N}^{\mathcal{S}}_{u,v}$ **do**
    if $(w, z) \in \mathcal{S}_e$ **then**
      $t_C \leftarrow t_C + 1$
      SAMPLE4CLIQUE(u, v, w, z, t)

a 4-clique with edges in the edge sample the algorithm attempts to insert it to the 4-cliques reservoir sample. A 5-clique is counted when the current observed edge completes a 5-clique using one 4-clique in the reservoir sample and 3 edges in the edge sample. Given the choice of the prototype, each occurrence of a 5-clique may be detected by TS5C in at most two different ways. The probability of each detection is approximated as

$$\tilde{p} = \min\left\{1, \frac{M_P}{\tau}\right\}\left(\min\left\{1, \frac{M_e}{t'-1}\right\}\right)^5\left(\min\left\{1, \frac{M_e}{t-1}\right\}\right)^3$$

where M_e (resp., M_P) denotes the memory space assigned to the edge-only (resp., the prototype/ 4-cliques) reservoir, t (resp., t') denotes the time step at which the last edge of the 5-clique (reps., prototype 4-clique being used) arrived on the stream, and $\tau^{(t)}$ denotes the number of 4-clique prototypes detected by TS5C up to time t. The approximate $\tilde{p}$ is obtained following the breakdown discussed in step (3) of Section 8.1:

- $\min\left\{1, \frac{M_P}{\tau}\right\}$ approximates the probability of the 4-clique prototype used to detect the occurrence of a 5-clique to be in $\mathcal{S}_P$ at time t conditioned on the fact that it was observed by TS5C;
- $\left(\min\left\{1, \frac{M_e}{t'-1}\right\}\right)^5$ approximates the probability that the 4-clique prototype used to detect the occurrence of a 5-clique and whose last edge arrived on the stream at time t' was observed by TS5C (i., the probability that when the last edge of the 4-clique was observed on the stream the remaining 5 edges were included in $\mathcal{S}_e^{(t')}$);
- $\left(\min\left\{1, \frac{M_e}{t-1}\right\}\right)^3$ approximates the probability that the remaining 3 edges used in the detection of the 5-clique are included in $\mathcal{S}_e^{(t)}$.

Other details of the construction of TS5C follow the blueprint outlined in the previous section. The pseudocode of TS5C is presented in Algorithm 19. The use of the approximate $\tilde{p}$ allows to adapt the TieredSampling paradigm to the task of counting 5-cliques while considerably reducing the analysis effort. In contrast, the exact analysis would require a breakdown of a number of cases corresponding to all the possible ordering of arrival the 11 edges in a 5-clique.

While the use of approximate values of the probability of detecting instances of the pattern of interest leads to the estimate $\varkappa$ not being unbiased, in the following experimental evaluation we show that the computed estimate is still very close to the ground truth value. Our experiments compare the performance of TS5C to that of a standard one tier edge reservoir sample algorithm FiveEst (refer to Algorithm 20 in the latter part of this Section), similar to FourEst. We evaluate the average performance over 10 runs of the two algorithms on the DBLP graph using $M = 3 \times 10^5$. The results are presented in Fig. 8.10. TS5C clearly outperforms FiveEst, despite the latter providing an *unbiased estimate* (Theorem 8.20), in obtaining much better estimations of the ground truth value $|\mathcal{C}_5^{(t)}|$ achieving an $\sim 56\%$ reduction of the average MAPE.

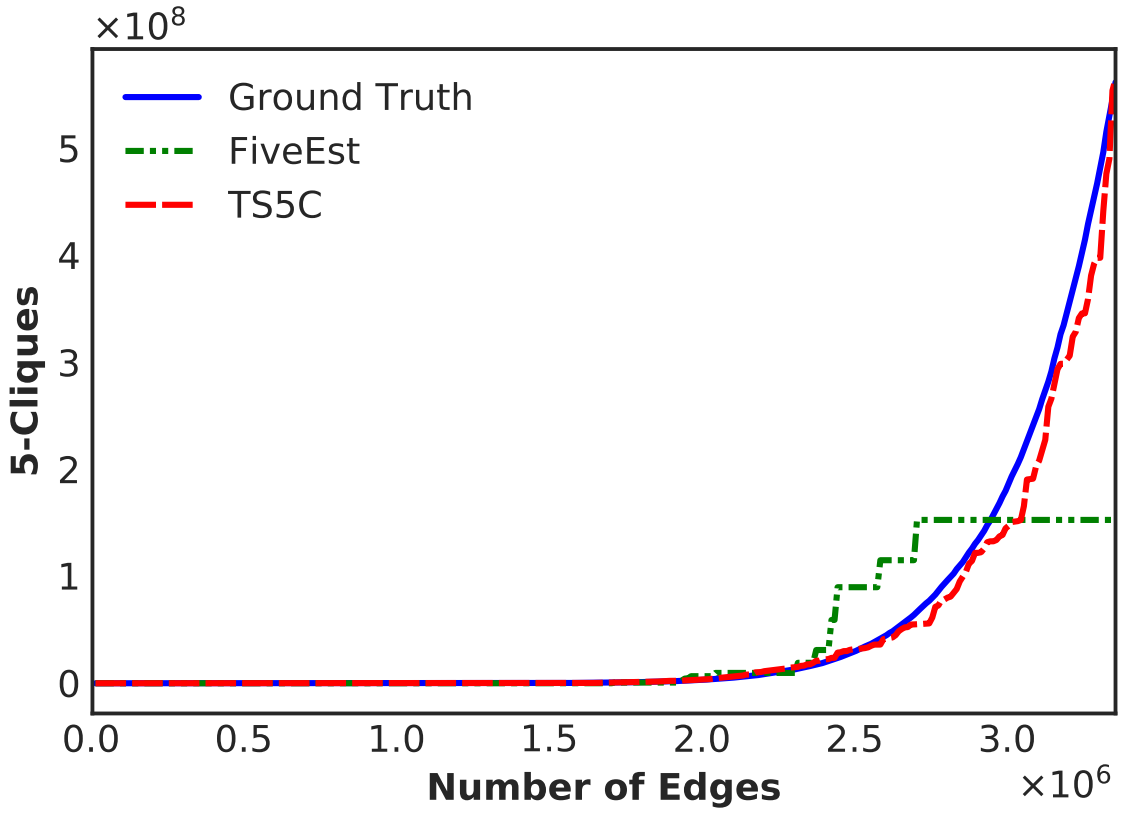


Figure 8.10: Comparison of $|\mathcal{C}_5^{(t)}|$ estimates for the DBLP graph obtained using TS5C and FiveEst with $M = 3 \times 10^5$. Out of 567,495,440 5-cliques that are present in the DBLP graph, the single reservoir algorithm FiveEst estimates a number of 153,979,072 and TS5C estimates of number of 565,173,908 5-cliques.

FiveEst: 5-clique counting using an edge-only sample

Lemma 8.19. *Let $\lambda \in \mathcal{C}_5^{(t)}$ with $\lambda = \{e_1, \ldots, e_{10}\}$. Assume, without loss of generality, that the edge e_i is observed at t_i (not necessarily consecutively) and that $t_{10} > \max\{t_i, 1 \leq i \leq 9\}$. λ is observed by* FiveEst *at time t_{10} with probability:*

$$p_\lambda = \begin{cases} 0 & if |M| < 9, \\ 1 & if\ t_{10} \leq M+1, \\ \prod_{i=0}^{9} \frac{M-i}{t-i-1} & if\ t_{10} > M+1. \end{cases} \tag{8.25}$$

Theorem 8.20. *Let $\varkappa^{(t)}$ the estimated number of 5-cliques in $G^{(t)}$ computed by* FiveEst *using memory of size $M > 9$. $\varkappa^{(t)} = |\mathcal{C}_5^{(t)}|$ if $t \leq M+1$ and $\mathbb{E}\left[\varkappa^{(t)}\right] = |\mathcal{C}_5^{(t)}|$ if $t > M+1$.*

The proof for Lemma 8.19 (resp., Theorem 8.20), closely follows the steps of the proof of Lemma 8.15 (resp., Theorem 8.16).

ALGORITHM 20 FIVEEST - Single Reservoir Sampling for 5-cliques counting

Input: Edge stream Σ, integer $M \geq 6$
Output: Estimation of the number of 5-cliques $\varkappa$
$\mathcal{S}_e \leftarrow \emptyset$, $t \leftarrow 0$, $\varkappa \leftarrow 0$
for each element (u, v) from Σ **do** ▷ Process each edge (u, v) coming from the stream in discretized timesteps
  $t \leftarrow t + 1$
  UPDATE5CLIQUES(u, v) ▷ Update the 5-clique estimator by considering the new 5-cliques closed by edge (u, v)
  SAMPLEEDGE$((u, v), t)$ ▷ Update the edges sample with the edge (u, v) according to RS scheme

function UPDATE5CLIQUES(u, v)
  $\mathcal{N}_{u,v}^{\mathcal{S}} \leftarrow \mathcal{N}_u^{\mathcal{S}} \cap \mathcal{N}_v^{\mathcal{S}}$
  for each element (x, w, z) from $\mathcal{N}_{u,v}^{\mathcal{S}} \times \mathcal{N}_{u,v}^{\mathcal{S}} \times \mathcal{N}_{u,v}^{\mathcal{S}}$ **do**
    if $\{(x, w), (x, z), (w, z)\} \subseteq \mathcal{S}_e$ **then**
      if $t \leq M + 1$ **then**
        $p \leftarrow 1$
      else
        $p \leftarrow \prod_{i=0}^{9} \frac{M-i}{t-i-1}$
      $\varkappa \leftarrow \varkappa + p^{-1}$

8.10 Conclusions

In this work, we study the problem of counting sparse motifs in large-scale graphs streams using multi-layer (tiered) reservoir sampling. We developed TIEREDSAMPLING, a novel technique for approximate counting sparse motifs in massive graphs whose edges are observed in a one-pass stream. We fully analyze and demonstrate the advantage of our method in a specific application for counting number the of 4-cliques in a graph using a two sample approach. Through extensive experimental analysis, we show that the proposed algorithms produce high quality and low variance approximations for the number of 4-cliques for large graphs, both synthetic and real-world, with up to hundred of millions edges. We present both analytical proofs and experimental results, demonstrating the advantage of our method in counting sparse motifs compared to the standard methods of using just a single edge reservoir sample.

We present a simple process that allows generalizing the TIEREDSAMPLING approach to provide estimates of the count for any subgraph of interest while considerably reducing the effort for the analysis by using opportune approximations. We showcase the effectiveness of this approach by presenting TS5C, an application of TIEREDSAMPLING for counting the number of 5-cliques in a graph stream.

With the growing interest in discovering and analyzing large motifs in massive-scale graphs in social networks, genomics, and neuroscience, we expect to see further applications of our technique.

Chapter 9

Reconstructing Hidden Permutations Using the Average-Precision (AP) Correlation Statistic[1]

In this Chapter, we study the problem of learning probabilistic models for permutations, where the order between highly ranked items in the observed permutations is more reliable (i.e., consistent in different rankings) than the order between lower ranked items, a typical phenomena observed in many applications such as web search results and product ranking. We introduce and study a variant of the Mallows model where the distribution is a function of the widely used Average-Precision (AP) Correlation statistic, instead of the standard Kendall's tau distance. We present a generative model for constructing samples from this distribution and prove useful properties of that distribution. Using these properties we develop an efficient algorithm that provably computes an asymptotically unbiased estimate of the center permutation, and a faster algorithm that learns with high probability the hidden central permutation for a wide range of the parameters of the model. We complement our theoretical analysis with extensive experiments showing that unsupervised methods based on our model can precisely identify ground-truth clusters of rankings in real-world data. In particular, when compared to the Kendall's tau based methods, our methods are less affected by noise in low-rank items.

[1] A preliminary version of the results presented in this chapter appeared in the proceedings of the 30-th AAAI Conference on Artificial Intelligence. This is joint work with Alessandro Epasto, Professor Fabio Vandin and Professor Eli Upfal.

9.1 Introduction

Probabilistic models of ranking data have been studied extensively in statistics [153], machine learning [9] and theoretical computer science [30, 42]. Applications of ranking models include understanding user preferences in electoral systems, ordering web search results, aggregating crowd-sourcing data, and optimizing recommendation systems results [233, 197, 61, 215, 34, 241, 159].

Most of the analytic work in this area has focused on the *Mallows model* [153] which defines a probability distribution over a set of permutations of n elements given a fixed *center* permutation π (the *ground truth*), and a *dispersion* parameter $\beta > 0$. The probability of a permutation σ in Mallows model is $\Pr_{\mathcal{M}(\beta,\pi)}(\sigma) = N_\beta^{-1} \exp(-\beta d_K(\pi, \sigma))$, where $d_K(\pi, \sigma)$ is the Kendall's tau distance between π and σ, and N_β is a normalization factor independent of σ (see Section 9.3 for more details). Since the Kendall's tau distance simply counts the number of pairs of items whose order is inverted in σ with respect to their order in π, a permutation with inversions of elements occupying positions towards the end of the ranking has the same probability as a permutation with the same number of inverted pairs near the top of the ranking.

In many practical applications, such as web search ranking, voter preferences surveys, consumer evaluation systems and recommendation systems we expect the input samples to provide a more reliable order for items at the top of the ranking than for "less important" items at the bottom of the ranking [159, 233, 241]. The limitation of the Kendall's tau distance and the associated Mallows model, when applied to such data has been pointed out in a number of recent works [233, 241, 136]. Note that the Spearman's footrule distance [216], another widely used statistic, suffers from the same problem.

In this work we address the limitation of the standard Mallows distribution by proposing an alternative exponential distribution of permutations, $\mathcal{M}_{\mathcal{AP}}(\beta, \pi)$, in which the Kendall's tau distance of the Mallows distribution is replaced by a new distance based on the widely used Average Precision (AP) correlation statistic [241]. The AP statistic (and hence the AP model) takes into account not only the number of miss-ranked pairs but also the locations (and hence the importance) of the miss-placed pairs in the ranking. Let $[n]$ be the set of natural numbers $\{1, \ldots, n\}$, and let S_n be the set of all permutations over $[n]$. Given a fixed *center* permutation $\pi \in S_n$ and a *dispersion* parameter $\beta > 0$, the probability associated with a permutation $\sigma \in S_n$ according to the new model is $\Pr_{\mathcal{M}_{\mathcal{AP}}(\beta,\pi)}(\sigma) = Z_\beta^{-1} \exp(-\beta d_{\mathcal{AP}}(\pi, \sigma))$, where $d_{\mathcal{AP}}(\pi, \sigma)$ is the AP-distance between π and σ and Z_β is a normalization factor independent of σ (see Section 9.3 for more details). To the best of our knowledge, no work has addressed the problem of modeling permutations with provable guarantees subject to such properties.

We now summarize the main contributions of this work:

- We introduce a novel variant of the standard Mallows model based on the more nuanced AP statistic.
- We introduce a generative process for this probability distribution. This process, besides defining an efficient algorithm for sampling permutations from the distribution, is a useful tool

in establishing formal properties of our model.

- We provide bounds on the probability of swapping elements and use these bounds to quantify the number of samples required for learning the parameters of the model.
- We design efficient algorithms that given $\mathcal{O}(\log n/(n\beta))$ samples learn the central permutation with high probability for $\beta = \Omega(1/n)$. We also show an alternative algorithm that efficiently computes an asymptotically unbiased and consistent estimator of the central permutation for any $\beta > 0$.
- We experimentally evaluate our model and algorithms using both synthetic and real-world data. We show experimentally the accuracy of our estimators in the reconstruction of the hidden permutation with a limited number of samples. We also show with real data that unsupervised methods based on our model can precisely (and efficiently) identify ground-truth clusters of rankings with tens of thousand of elements. Moreover we show that our method is less affected by noise in the lowest positions of the rankings, which is more likely to occur [233], than the Kendall-based ones. Finally we show that simple supervised classification algorithms based on the AP statistic outperform classifications based on the Kendall-based distance measure.

The new $\mathcal{M}_{\mathcal{AP}}(\beta, \pi)$ model is significantly more challenging for mathematical analysis than the Mallows model. The $\mathcal{M}_{\mathcal{AP}}(\beta, \pi)$ model does not have many of the simplifying features used in the analysis of Mallows model, such as translational invariant of the distribution [42]. Nevertheless, as we show in this work, the model remains sufficiently mathematically tractable to derive non-trivial theoretical results. Furthermore, while many of the results for the Mallows model require a constant β, our analysis applies to $\beta = \beta(n)$ that is decreasing in n.

The presentation is structured as follows. Section 9.2 reviews part of the literature which is relevant for our problem. Section 9.3 formally defines the $\mathcal{M}_{\mathcal{AP}}(\beta, \pi)$ model and provides some intuition on the behavior of the AP model. Section 9.4 defines a generative process for the distribution that will be pivotal in the analysis of our model. Section 9.5 presents the in depth study the mathematical properties of our model and the learning algorithms designed on this study. Section 9.6 presents an experimental evaluation of our method on real and synthetic data. Finally, Section 9.7 draws conclusions and hints at possible future directions stemming from our work.

9.2 Related Work

Ranking problems have been studied extensively since at least the 18th century works of Borda and Condorcet on social choice theory [34, 197]. We discuss here only the most relevant works derived from Mallows' original work.

Properties of the Mallows model [153] were studied in [55, 79]. Tail bounds on the displacement of an element were studied in [31, 21]. Reconstructing the hidden central permutation was studied in [31, 158, 42, 151]. Finding the maximum likelihood permutation is equivalent to the well

known rank aggregation problem which is NP-hard [11] and has a polynomial-time approximation scheme [126]. Several heuristics were introduced in order to solve this problem [158, 203] with no provable guarantees. The problem of learning a mixture of Mallows models was studied in [9, 42]. The Mallows model has also been extended in various ways by generalizing the Kendall's tau distance in order to weigh the number of inversions with respect to previously defined parameters for each element [78, 79]. In the Recursive Inversion Model [157] different weights are assigned to the inversions of elements, requiring a priori specification of $n-1$ weights. A priori assigning such weights is not easy. In contrast, the AP model we consider gradually and coherently weighs the elements based on their positions.

Recent work [186] generalizes the Mallows model to other distances in which the maximum likelihood can be efficiently computed. [209] proposed a weighting scheme for Kendall's tau distance where the weight of errors depends on the ranks of the inverted items. [136] later extended Shieh's model defining an even more sophisticated scheme that takes into account positions of the items, weights, and their similarity. Various works [71, 102] addressed the issue of assigning different weights to various elements by restricting the computation of some statistics only on specific parts of the ranking.

The AP correlation statistic, which is the focus of this work, was introduced in [241, 240]. Since its introduction, AP correlation has been widely used in the literature [233, 198, 205, for instance, and see the references of these papers]. Recently [233] generalized the AP correlation and other metrics to handle ties and other ways to weighting inversions. We believe that our approach could be adapted to the corresponding extension of the AP-distance. To the best of our knowledge no prior work has addressed the problem of provably reconstructing the center permutation of a generalized Mallows model using AP distance or similar more nuanced measures.

9.3 The AP model

For $\pi \in S_n$ let π_i be the i-th element in the order defined by π, and $\pi(i)$ is the position of $i \in [n]$ in π. We use the notation $i <_\pi j$ to indicate that π ranks i before j, or equivalently, that $\pi(i) < \pi(j)$. Finally we use $\pi[t]$ to denote the prefix of the t highest ranked elements in π, i.e. $\pi[t] = (\pi_1, \ldots, \pi_t)$.

Our new model uses the following *AP distance* of a permutation σ from permutation π:

$$d_{\mathcal{AP}}(\pi, \sigma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E_{ij} \frac{n}{2(j-1)},$$

where $E_{ij} = 1$ iff item π_i is ranked after item π_j in permutation σ, and $E_{ij} = 0$ otherwise. Note that the AP-distance $d_{\mathcal{AP}}(\pi, \sigma)$ is defined with respect to a ground truth π and it is therefore not symmetric. Notice that the Kendall's tau distance $d_K(\pi, \sigma)$ can be expressed in the same form by assigning cost 1 (instead of $\frac{n}{2(j-1)}$) for each inversion, i.e. $d_K(\pi, \sigma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E_{ij}$.

The AP-distance has the same range and extreme points as the Kendall's tau distance: $0 \leq d_{\mathcal{AP}}(\pi, \sigma) \leq \binom{n}{2}$, $d_{\mathcal{AP}}(\pi, \sigma) = 0$ iff $\pi = \sigma$, and $d_{\mathcal{AP}}(\pi, \sigma) = \binom{n}{2}$ iff σ is obtained by reversing π. However, the AP-distance assigns weights to inverted pairs in σ which depend on the locations of

the items in π. More specifically, the cost assigned to inversion of elements (π_i, π_j) for $i < j$ is $\frac{n}{2(j-1)}$. The cost assigned by AP is strictly higher than the one given by tau distance (always 1) for elements in the first half of the ranking, i.e. for $j < \frac{n}{2} + 1$ and strictly lower than 1 for $j > \frac{n}{2} + 1$.

Given this distance function we define the AP model $\mathcal{M}_{\mathcal{AP}}(\beta, \pi)$:

Definition 9.0.1 (AP model)**.** *The* AP model *is a probability distribution over S_n. Given a fixed* center *permutation π and a* dispersion *parameter $\beta > 0$, the probability of a permutation $\sigma \in S_n$ is*

$$\Pr_{\mathcal{M}_{\mathcal{AP}}(\beta,\pi)}(\sigma) = Z_\beta^{-1} \exp(-\beta d_{\mathcal{AP}}(\pi, \sigma)), \tag{9.1}$$

where $Z_\beta = \sum_{\sigma \in S_n} \exp\left(-\beta d_{\mathcal{AP}}(\pi, \sigma)\right)$ is a normalization coefficient independent of σ.

Notice that our model differs from the traditional Mallows model $\mathcal{M}(\beta, \pi)$ only in the use of the AP distance $d_{\mathcal{AP}}(\pi, \sigma)$ instead of the Kendall's tau distance $d_K(\pi, \sigma)$.

Before studying the details of the model we show that, as in the traditional Mallows model, finding the maximum likelihood center permutation for an *arbitrary* multi-set of permutations is NP-hard.

More precisely, we define the `AP-ML` problem as follows. Given an arbitrary multiset P of elements of S_n (and $\beta > 0$) find the permutation $\pi_{AP} \in S_n$ such that:

$$\pi_{AP} = \arg\max_{\pi \in S_n} \prod_{\sigma \in P} \Pr_{\mathcal{M}_{\mathcal{AP}}(\beta,\pi)}(\sigma)$$

Theorem 9.1. *The `AP-ML` problem is NP-hard*

Proof. We show that this problem is NP-Hard for an arbitrary multiset P by reducing an instance of the K-LM problem, which has been shown to be NP-Hard in [30], to a corresponding instance of the AP-ML problem.

Reduction from Kendall-ML to the AP-ML Given the multiset P of permutations over $[n]$, we create the set $F = \{f_1, \ldots, f_N\}$ composed by $N = |P|n^3$ new elements such that $F \cap [n] = \emptyset$. We denote the as $S_{F \cap [n]}$ the set of permutations generated by $F \cap [n]$.

First, we create a multiset $\bar{P}'$ which is composed by $M = n^{10}$ copies of the pair of permutations $(f_1, \ldots, f_N, 1, \ldots n)$ and $(f_1, \ldots, f_N, n, \ldots 1)$.

We then create a multiset $\bar{P}''$ such that for each permutation $\pi \in P$ we have a permutation $\bar{\pi} \in \bar{P}''$ which is obtained by prefixing the permutation π with $(f_1, \ldots, f_N)$.

We finally the set $\bar{P} \subseteq S_{F \cap [n]}$ as $\bar{P} = \bar{P}' \cup \bar{P}''$.

Lemma 9.2. *Let π_{AP} be the solution of the AP-ML problem on the set $\bar{P}$. We then have $\pi_{AP}(i) = f_i$ for $i = 1, \ldots, N$.*

Proof. We shall first verify that the the elements form F occupy the first N positions in π_{AP}. If this is not the case then there must exist a pair of elements $f_i \in F$ and $i \in [n]$ s.t. $\pi_{AP}(i) = \pi_{AP}(f_i) + 1$. Let π' be the ranking obtained by just inverting these two elements, given the construction of $\bar{P}$, for any $\bar{\sigma} \in \bar{P}$ we have $d_{\mathcal{AP}}(\pi', \sigma) < d_{\mathcal{AP}}(\pi_{AP}, \sigma)$ which leads to a contradiction. A similar proof allows us to conclude that we indeed have $\pi_{AP}(i) = f_i$ for $i = 1, \ldots, N$. The lemma follows. □

Lemma 3 implies that the elements in $[n]$ occupy the last n positions of the solution of the AP-ML problem oh the set $\bar{P}$. Let π_K be the sub-permutation obtained by considering just the elements of $[n]$ in π_{AP}.

Lemma 9.3. *The permutation π_K is a correct solution K-ML problem for the multiset P.*

Proof. The proof is by contradiction. Suppose there exists a permutation $\pi'_K \in S_n$ such that $\sum_{\sigma\in P} d_K\left(\pi'_K,\sigma\right) < \sum_{\sigma\in P} d_K\left(\pi_K,\sigma\right)$. Let us then consider the permutation $\pi'_{AP} = (f_1,\ldots,f_N,\pi'_K) \in S_{F\cap[n]}$: from Lemma 3 we have that for any $\bar{\sigma} \in \bar{P}$ there are no inversions between elements form F and element from $[n]$ in π'_{AP} with respect to $\bar{\sigma}$. The possible variation in the value of $\sum_{\sigma\in\bar{P}} d_{\mathcal{AP}}\left(\pi'_{AP},\sigma\right)$ with respect to $\sum_{\sigma\in\bar{P}} d_{\mathcal{AP}}\left(\pi_{AP},\sigma\right)$ can therefore depend just on inversions involving elements from $[n]$. Since, by the construction of the permutations in $\bar{P}$, these elements appear in the positions $N+1, N+2,\ldots,N+n$, the inversions involving them can have weight at most $1/N = 1/|P|n^3$ and at least $1/(N+n) = 1/(|P|n^3+n)$ according to the AP measure. The contribution to $\sum_{\sigma\in\bar{P}} d_{\mathcal{AP}}\left(\pi_{AP},\sigma\right)$ due just to the elements from $[n]$ is therefore at least $\sum_{\sigma\in P} d_K\left(\pi_K,\sigma\right)/(N+n)$ while the contribution to $\sum_{\sigma\in\bar{P}} d_{\mathcal{AP}}\left(\pi'_{AP},\sigma\right)$ due just to the elements from $[n]$ is at most $\sum_{\sigma\in P} d_K\left(\pi'_K,\sigma\right)/(N)$. Since we have $\sum_{\sigma\in P} d_K\left(\pi'_K,\sigma\right) < \sum_{\sigma\in P} d_K\left(\pi_K,\sigma\right) \leq |P|n^2/2$ we can conclude:

$$\frac{\sum_{\sigma\in P} d_K\left(\pi_K,\sigma\right)}{N+n} > \frac{\sum_{\sigma\in P} d'_K\left(\pi_K,\sigma\right)}{N}$$

If that is the case, then the permutation $\pi'_{AP} = (f_1,\ldots,f_N,\pi'_K) \in S_{F\cap[n]}$ is such that:

$$\sum_{\sigma\in\bar{P}} d_{\mathcal{AP}}\left(\pi',\sigma\right) < \sum_{\sigma\in\bar{P}} d_{\mathcal{AP}}\left(\pi_{AP},\sigma\right).$$

Since π_{AP} is a solution of the AP-ML problem for $\bar{P}$ we have a contradiction. The lemma follows. □

□

Note that for any $\beta > 0$ we have that a permutation π which is a solution of the AP-ML problem for P is the one that minimizes the average AP distance to the multi-set P, that is also NP-hard to find.

We stress that the previous NP-hardness result holds for arbitrary multi-sets (i.e. not generated by the AP model distribution). As we will see in the rest of this Chapter, when the permutations are generated according to the AP model distribution, it is possible to learn important properties of the model in polynomial time.

9.4 Generative Process for $\mathcal{M}_{\mathcal{AP}}(\beta,\pi)$

We give a randomized algorithm for constructing a permutation $\sigma \in S_n$ according to the $\mathcal{M}_{\mathcal{AP}}(\beta,\pi)$ model. The algorithm is based on the one presented in [55] and provides useful insight on the probability distribution for the AP model. Recall that $\pi[i]$ refers to the prefix of length i of the permutation π.

The $\mathcal{M}_{\mathcal{AP}}(\beta,\pi)$ Generative Process: Let $\pi=\pi_1,\pi_2,\ldots,\pi_n$ be the central permutation. We generate a random permutation σ in n steps. At step i we produce a permutation σ^i in the following way:

1. $\sigma^1=(\pi_1)$;
2. for $i=2$ to n do:
   (a) choose a random value $0\leq r\leq i-1$ according to the probability
   $$\Pr(r=j)=\frac{1}{B_i}\exp\left(-\frac{\beta n}{2}\frac{j}{i-1}\right),$$
   where
   $$B_i=\sum_{j=0}^{i-1}\exp\left(-\frac{\beta n}{2}\frac{j}{i-1}\right).$$
   (b) Generate a permutation σ^i from σ^{i-1} by placing item π_i after item σ^{i-1}_{i-1-r}. i.e., for $1\leq t\leq i-1-r$ we have $\sigma^i_t=\sigma^{i-1}_t$, then $\sigma^i_{i-r}=\pi_i$, and for $i-r+1\leq t\leq i$ we have $\sigma^i_t=\sigma^{i-1}_{t-1}$. If $r=i-1$, we have $\sigma^i_1=\pi_i$.
   (c) Finally output $\sigma=\sigma^n$.

Theorem 9.4 (Correctness AP generative process)**.** *For $i=1,\ldots,n$, the permutation σ^i generated by the $\mathcal{M}_{\mathcal{AP}}(\beta,\pi)$ Generative Process has distribution $\mathcal{M}_{\mathcal{AP}}\left(\frac{n}{i}\beta,\pi[i]\right)$. In particular, the output permutation $\sigma=\sigma^n$ has distribution $\mathcal{M}_{\mathcal{AP}}(\beta,\pi)$.*

Proof. In the following let $p_{i,i-j}$, denotes the probability that element π_i has been inserted in position $1\leq i-j\leq i$ of the permutation σ^i during the i-th step of the generative process for $0\leq j<i\leq n$. We have:
$$p_{i,i-j}=\frac{1}{B_i}\exp\left(-\frac{\beta n}{2}\frac{j}{i-1}\right) \tag{9.2}$$
The proof is by induction on i. In the base case for $i=1$ we have $\sigma^1=\pi^1$ with probability one. Since the the only possible permutation has distance zero from the center the statement is therefore verified. For $i>1$, let σ^i be the sequence given as output of the process at step i for which the element π_i is placed in position $i-j$ of the corresponding permutation σ^{i-1}. By inductive hypothesis, the probability of σ^{i-1} being returned by the process at step $i-1$ is given by:
$$\Pr_{\mathcal{M}_{\mathcal{AP}}\left(\frac{n}{i-1}\beta,\pi[i-1]\right)}(\sigma^{i-1})=\frac{1}{Z_\beta[i-1]}\exp(-\beta d_{\mathcal{AP}}\left(\pi[i-1],\sigma^{i-1}\right)), \tag{9.3}$$
where $Z_\beta[i]=\sum_{\sigma^i}\exp\left(-\beta d_{\mathcal{AP}}\left(\pi[i],\sigma^i\right)\right)$

We have:
$$d_{\mathcal{AP}}\left(\pi[i],\sigma^i\right)=d_{\mathcal{AP}}\left(\pi[i-1],\sigma^{i-1}\right)+\frac{j}{i-1} \tag{9.4}$$
with $d_{\mathcal{AP}}\left(\pi[1],\sigma^1\right)=0$. We can express $Z_\beta[i]$ as a function of $Z_\beta[i-1]$ as follows:
$$\begin{aligned}Z_\beta[i] &= \sum_{\sigma^i}\exp\left(-\beta d_{\mathcal{AP}}\left(\pi\left[i\right],\sigma^i\right)\right)\\ &= \sum_{\sigma^{i-1}}\exp\left(-\beta d_{\mathcal{AP}}\left(\pi\left[i-1\right],\sigma^{i-1}\right)\right)\sum_{j=0}^{i-1}\exp\left(-\beta\tfrac{n}{2}\tfrac{j}{i-1}\right)\end{aligned}$$
And therefore:
$$Z_\beta[i]=Z_\beta[i-1]B_i \tag{9.5}$$

The probability of obtaining σ^i as output of the process at step i is given by the product of the probability of of obtaining σ^{i-1} as output of the process at step $i-1$ times the probability of placing element π_i in position $i-j$:

$$\begin{aligned}\Pr_{\mathcal{M}_{\mathcal{AP}}\left(\frac{n}{i-1}\beta,\pi[i-1]\right)}(\sigma^{i-1})p_{i,i-j} &= \frac{1}{Z_\beta[i-1]}\exp(-\beta d_{\mathcal{AP}}\left(\pi[i-1],\sigma^{i-1}\right))\frac{1}{B_i}\exp\left(-\frac{\beta n}{2}\frac{j}{i-1}\right)\\ &= \frac{1}{Z_\beta[i-1]B_i}\exp(-\beta\left(d_{\mathcal{AP}}\left(\pi[i-1],\sigma^{i-1}\right)+\frac{j}{i-1}\right)\end{aligned}$$

From (9.4) and (9.5) we therefore have:

$$\Pr_{\mathcal{M}_{\mathcal{AP}}\left(\frac{n}{i-1}\beta,\pi[i-1]\right)}(\sigma^{i-1})p_{i,i-j} = \frac{1}{Z_\beta[i]}\exp(-\beta d_{\mathcal{AP}}\left(\pi[i],\sigma^i\right)) = \Pr_{\mathcal{M}_{\mathcal{AP}}\left(\frac{n}{i}\beta,\pi[i]\right)}(\sigma^i)$$

The theorem follows. □

Moreover, we can also compute the normalization coefficient Z_β of the model is given by:

Corollary 9.5. *For $n=1$, $Z_\beta=1$. Otherwise, for $n>1$,*

$$Z_\beta = \prod_{i=2}^{n}\frac{\exp\left(-\frac{\beta n}{2}\frac{i}{i-1}\right)-1}{\exp\left(-\frac{\beta n}{2}\frac{1}{i-1}\right)-1}.$$

9.5 Reconstructing the Center Permutation in the AP model

In this section we study the problem of reconstructing the hidden center permutation given a set of samples drawn from the AP model.

Problem 9.6 (Permutation Reconstruction in the AP model)**.** *Given T i.i.d. samples obtained from $\mathcal{M}_{\mathcal{AP}}(\beta,\pi)$ with unknown parameters $\beta>0$ and $\pi\in\mathcal{S}_n$, compute a best estimate for the center permutation π.*

A minimum requirement for solving the problem is $\beta>0$, since for $\beta=0$, the distribution is uniform over S_n and all centers define the same distribution.

Our first goal is to show that with a sufficient number of samples we can reconstruct the center permutation for any $\beta>0$, even for $\beta=\beta(n)\overset{n\to\infty}{\longrightarrow}0$. To achieve this goal we design a rank-based algorithm that for any set of samples outputs a permutation that is an unbiased estimate of the central permutation. As we increase the number of samples, the sequence of estimates converges; thus the estimate must be consistent and therefore converges to the central permutation.

For a more practical result we present a comparison-based algorithm that for any $\beta>\frac{2\ln(2)}{n}$, and given $O(\log(n))$ independent samples from the distribution, computes w.h.p.[2] the correct center permutation.

[2]We say that an event happens with high probability (w.h.p.) if it happens with probability $\geq 1-\frac{1}{n}$.

9.5.1 Rank-based algorithm for $\beta > 0$

Theorem 9.7 (Probability of a swap between elements adjacent in π)**.** *Let σ be obtained from $\mathcal{M}_{\mathcal{AP}}(\beta, \pi)$, with $\pi \in \mathcal{S}_n$ and $\beta > 0$. For any $1 \leq i \leq n-1$, let $Pr\left(S_{i,i+1}\right)$ be the probability of two elements occupying the positions i and $i+1$ in π are swapped in σ. We have:*

$$Pr\left(S_{i,i+1}\right) \leq 1/2$$

with equality iff $\beta = 0$.

Proof. We compute the probability of swap using our generative process. The relative order between π_i and π_{i+1} in σ is determined when π_{i+1} is inserted in σ^{i+1} in the generating process. Later insertions of items cannot change that order. Thus, the probability of a swap between elements π_{i-1} and π_i in σ is given by the probability of inserting π_i in a position smaller or equal to the one in which the element π_{i-1} was previously inserted.

Let $q = e^{-\beta n/2}$ than $q \leq 1$ with equality iff $\beta = 0$. The generative process inserts the element π_i in position $1 \leq i - j \leq i$ during the i-th step with probability

$$p_{i,i-j} = B_i^{-1} \exp\left(-\frac{\beta n}{2}\frac{j}{i-1}\right) = B_i^{-1} q^{\frac{j}{i-1}}.$$

$$\begin{aligned}
&\text{Thus, } \Pr\left(\sigma(\pi_{i+1}) < \sigma(\pi_i)\right) \\
&= \frac{1}{B_i B_{i+1}} \sum_{j=0}^{i} q^{\frac{j}{i}} \left(\sum_{t=0}^{j-1} q^{\frac{t}{i-1}}\right) = \frac{\sum_{j=0}^{i}\sum_{t=0}^{j-1} q^{\frac{j}{i}+\frac{t}{i-1}}}{\sum_{j=0}^{i}\sum_{t=0}^{i-1} q^{\frac{j}{i}+\frac{t}{i-1}}} \\
&= \frac{\sum_{j=0}^{i}\sum_{t=0}^{j-1} q^{\frac{j}{i}+\frac{t}{i-1}}}{\sum_{j=0}^{i}\sum_{t=0}^{j-1} q^{\frac{j}{i}+\frac{t}{i-1}} + \sum_{j=0}^{i}\sum_{t=j}^{i-1} q^{\frac{j}{i}+\frac{t}{i-1}}} = \frac{A}{A+B}
\end{aligned}$$

We can now match each element $q^{\frac{j}{i}+\frac{t}{i-1}}$ in B to element $q^{\frac{t+1}{i}+\frac{j}{i-1}}$ in A. It is easy to verify that $q^{\frac{j}{i}+\frac{t}{i-1}} \geq q^{\frac{t+1}{i}+\frac{j}{i-1}}$ with equality either if $q = 1$ or $t = i-1$ and $j = 0$. . This implies $A \leq B$ with equality iff $\beta = 0$, proving the claim. □

Note that the above result applies only for adjacent items in the permutation π, and only with probability $\leq 1/2$. Thus, the result does not imply a total order. Recall that in general, $Pr(X \geq Y) > 1/2$ does not imply $E[X] \geq E[Y]$ or vice versa. Nevertheless, in this model we can prove a total order on the expected position of items in the observed permutations. In the following, let $\sigma(\pi_i)$ denote the position of the element π_i in a permutation $\sigma \sim M_{AP}(\pi, \beta)$. Note the distinction with σ_i which denotes the element at the i-th position in σ instead.

Theorem 9.8. *For any $\beta > 0$ and $1 \leq i \leq n-1$,*

$$E[\sigma(\pi_i)] < E[\sigma(\pi_{i+1})].$$

Proof. Consider the process of generating the permutation $\sigma \sim M_{AP}(\pi, \beta)$. While the order between π_{i-1} and π_i is fixed when π_i is in inserted to the (partial) permutation, their final locations are influenced by the insertions of elements $\pi_{i+1}, \ldots, \pi_n$. In particular, the distance between π_{i-1} and π_i is increased when an element is inserted between them.

We now use the mapping defined in the proof of the previous theorem, mapping the event where π_{i-1} was inserted in position $i-1-j$ and π_i was inserted in position $i-t$ with $t \leq j$ to the event

with of lower or equal probability in which π_{i-1} was placed in position $i-1-t$ while π_i was inserted in position $i-j+1$. The interval between the two elements in both cases starts at location $i-1+j$. When $\sigma(\pi_{i-1}) < \sigma(\pi_i)$ the length of the interval is $j-t+1$ and when $\sigma(\pi_i) < \sigma(\pi_{i-1})$ the length of the interval is $j-t$. Since the insertion of the elements $\pi_{i+1}, \ldots, \pi_n$ is independent of the locations of π_{i-1} and π_i the expected increase in the location of $\sigma(\pi_i)$ is no less than the expected increase in the location of $\sigma(\pi_{i-1})$. □

We will now describe a rank-based algorithm for the permutation reconstruction problem which takes as input a sample $\sigma^1, \ldots, \sigma^s$ of s independent random permutations from $\mathcal{M}_{\mathcal{AP}}(\beta, \pi)$ and computes an estimate π_{Rank} for the central permutation π.

For each element $\pi_1, \ldots, \pi_n$ we compute the respective average rank $\overline{\sigma(\pi_i)} = \frac{1}{s}\sum_{j=1}^{s} \sigma^j(\pi_i)$. We then build the permutation π_{Rank} by ordering the elements π_i according to their average ranks $\overline{\sigma(\pi_i)}$. The running time of the proposed algorithm is $O(sn + n\log n)$ where $O(sn)$ time is needed to computed the average rank for each element, and $O(n\log n)$ time is required by the sorting algorithm. Since the average ranks converge to their expectations we have:

Theorem 9.9. *The ranking π_{Rank} is an asymptotically unbiased and consistent estimator.*

Since the ranks are integers we can conclude that:

Corollary 9.10. *For any value of $\beta > 0$, and with a sufficient number of samples, the reconstructed permutation π_{Rank} equals the central ranking π .*

9.5.2 Comparison-Based algorithm for $\beta \in \Omega(\frac{1}{n})$

We now present an efficient algorithm for $\beta > \frac{c\ln(2)}{n}$, $c > 2$ independent of n, which relies on the following result:

Theorem 9.11 (Probability of a swap between any pair of elements from π)**.** *Let σ be obtained from $\mathcal{M}_{\mathcal{AP}}(\beta, \pi)$, with $\pi \in \mathcal{S}_n$ and $\beta > 0$. Given a pair (i,k) with $1 \le i < n$ and $0 \le k < n-i$, let $Pr(S_{i,i+k+1})$ be the probability of two elements occupying the positions i and $i+k+1$ in π are swapped in σ.*

$$Pr(S_{i,i+k+1}) \le \exp\left(-\frac{\beta n}{2}\right)\exp\left(-\frac{\beta n}{2}\frac{i}{i+k}\right).$$

In the following, we use $p_{i,k}$ to denote the probability that the element π_i has been inserted in position $1 \le k \le i$ of the permutation σ^i during the i-th step of the generative process. Let $k = i-j$ from equation (9.2) we have:

$$
\begin{aligned}
p_{i,k} = p_{i,i-j} &= \frac{1}{B_i} \exp\left(-\frac{\beta n}{2}\frac{j}{i-1}\right) \\
&= \frac{\exp\left(\frac{\beta n}{2}\right)}{\exp\left(\frac{\beta n}{2}\right)} \frac{\exp\left(-\frac{\beta n}{2}\frac{j}{i-1}\right)}{\sum_{k=0}^{i-1} \exp\left(-\frac{\beta n}{2}\frac{k}{i-1}\right)} \\
&= \frac{\exp\left(\frac{\beta n}{2}\frac{i-1-j}{i-1}\right)}{\sum_{k=0}^{i-1} \exp\left(\frac{\beta n}{2}\frac{i-1-k}{i-1}\right)} \\
&= \exp\left(\frac{\beta n}{2}\frac{k-1}{i-1}\right) \frac{\exp\left(\frac{\beta n}{2(i-1)}\right)-1}{\exp\left(\frac{\beta n}{2(i-1)}i\right)-1}
\end{aligned}
$$

We first introduce the following lemma which will prove useful in the following derivations.

Lemma 9.12. *Let $\sigma^i \sim \mathcal{M}_{\mathcal{AP}}\left(\frac{n}{i}\beta_n, \pi[i]\right)$ be a permutation produced at the i-th step of the algorithm and let $\sigma^i(\pi_i) \in \{1, 2, \ldots, i\}$ be the position in which the element π_i added during the i-th step. We then have:*

$$
Pr\left(\sigma^i(\pi_i) \leq j\right) = \frac{\exp\left(\frac{\beta n}{2(i-1)}j\right)-1}{\exp\left(\frac{\beta n}{2(i-1)}i\right)-1}. \tag{9.6}
$$

Proof. We have:

$$
\begin{aligned}
\Pr\left(\sigma^i(\pi_i) \leq j\right) &= \sum_{k=1}^{j} \Pr\left(\sigma^i(\pi_i) = k\right) \\
&= \sum_{k=1}^{j} p_{i,k} \\
&= \sum_{k=1}^{j} \frac{\exp\left(\frac{\beta n}{2(i-1)}(k-1)\right)\left(\exp\left(\frac{\beta n}{2(i-1)}\right)-1\right)}{\exp\left(\frac{\beta n}{2(i-1)}i\right)-1} \\
&= \frac{\exp\left(\frac{\beta n}{2(i-1)}\right)-1}{\exp\left(\frac{\beta n}{2(i-1)}i\right)-1} \sum_{k=1}^{j} \exp\left(\frac{\beta n}{2(i-1)}(k-1)\right) \\
&= \frac{\exp\left(\frac{\beta n}{2(i-1)}\right)-1}{\exp\left(\frac{\beta n}{2(i-1)}i\right)-1} \sum_{y=0}^{j-1} \exp\left(\frac{\beta n}{2(i-1)}(y)\right) \\
&= \frac{\exp\left(\frac{\beta n}{2(i-1)}\right)-1}{\exp\left(\frac{\beta n}{2(i-1)}i\right)-1} \frac{\exp\left(\frac{\beta n}{2(i-1)}j\right)-1}{\exp\left(\frac{\beta n}{2(i-1)}\right)-1} \\
&= \frac{\exp\left(\frac{\beta n}{2(i-1)}j\right)-1}{\exp\left(\frac{\beta n}{2(i-1)}i\right)-1}
\end{aligned}
$$

□

Proof of Theorem 9.11 The probability $\Pr\left(S_{i,i+k+1}\right)$ can be upper bounded by using law of total probability conditioning on the position in which the i-th element was inserted at the i-th step. The probability of π_i and π_{i+k+1} to be swapped in σ depends on the position occupied by π_i when π_{i+k+1} is inserted according to the Generative Process described in Section 9.4. In particular said probability increases monotonically with the index of the position occupied by π_i prior to the insertion of π_{i+k+1}, a sketch of the proof for this statement follows. If prior to the insertion of π_{i+k+1} element π_i is occupying position $r \leq i+k$, then the swap occurs iff π_{i+k+1} is inserted in a position with index smaller or equal than r. If instead prior to the insertion π_i is occupying position $r+1$, then the swap occurs iff π_{i+k+1} is inserted in any position with index smaller or equal than $r+1$. Since the probability of inserting π_{i+k+1} in in a position with index smaller or equal than $r+1$ is strictly greater than the probability of inserting π_{i+k+1} in in a position with index smaller or equal than r, the statement is verified. Next, we observe that the position occupied by element π_i when π_{i+k+1} is inserted depends on the position in which π_i was inserted during the i-th step. In particular, if π_i was inserted in position j during the the i-th step, the leftmost position it can occupy at step $i+k+1$ is $j+k$. This occurs iff all the k elements inserted between the i-th and the $i+k+1$-th step are swapped with respect to the i-th element. Following our previous observations, we can then conclude that the probability of the i-th and the $i+k+1$-th elements being swapped in σ given that the i-th was inserted in position j is upper bounded by the probability of swapping the elements given that the i-th is occupying the leftmost rightmost possible position, hence $j+k$, whenπ_{i+k+1} is inserted. We thus have:

$$\begin{aligned}
\Pr\left(S_{i,i+k+1}\right) &\leq \sum_{j=1}^{i} p_{i,j} \Pr\left(\sigma^{i+k+1}(\pi_{i+k+1}) \leq j+k\right) \\
&= \sum_{j=1}^{i} \frac{\exp\left(\frac{\beta n(j-1)}{2(i-1)}\right)\left(\exp\left(\frac{\beta n}{2(i-1)}\right)-1\right)}{\exp\left(\frac{\beta n}{2(i-1)} i\right)-1} \frac{\exp\left(\frac{\beta n}{2(i+k)}(j+k)\right)-1}{\exp\left(\frac{\beta n}{2(i+k)}(i+k+1)\right)-1}
\end{aligned}$$

Since $\exp\left(\frac{\beta n}{2(i-1)}\right) < \exp\left(\frac{\beta n}{2(i-1)} i\right)$ and $\exp\left(\frac{\beta n}{2(i+k)}(j+k)\right) < \exp\left(\frac{\beta n}{2(i+k)}(i+k+1)\right)$ we have:

$$\begin{aligned}
\Pr\left(S_{i,i+k+1}\right) &\leq \sum_{j=1}^{i} \frac{\exp\left(\frac{\beta n(j-1)}{2(i-1)}\right)\exp\left(\frac{\beta n}{2(i-1)}\right)}{\exp\left(\frac{\beta n}{2(i-1)} i\right)} \frac{\exp\left(\frac{\beta n}{2(i+k)}(j+k)\right)}{\exp\left(\frac{\beta n}{2(i+k)}(i+k+1)\right)} \\
&\leq \sum_{j=1}^{i} \frac{\exp\left(\frac{\beta n}{2}\left(\frac{j}{i-1}+\frac{j+k}{i+k}\right)\right)}{\exp\left(\frac{\beta n}{2}\left(\frac{i}{i-1}+\frac{i+k+1}{i+k}\right)\right)} = \sum_{j=1}^{i} \frac{\exp\left(\frac{\beta n}{2}\left(\frac{2ij+kj+ik-j-k}{(i+k)(i-1)}\right)\right)}{\exp\left(\frac{\beta n}{2}\left(\frac{2i^2+2ik-k-1}{(i+k)(i-1)}\right)\right)} \\
&\leq \sum_{j=1}^{i} \exp\left(\frac{\beta n}{2}\left(\frac{2ij+kj+ik-j-k-2i^2-2ik+k+1}{(i+k)(i-1)}\right)\right) \\
&\leq \sum_{j=1}^{i} \exp\left(\frac{\beta n}{2}\left(\frac{2ij+kj-j-2i^2-ik+1}{(i+k)(i-1)}\right)\right) \\
&\leq \sum_{j=1}^{i} \exp\left(\frac{\beta n}{2}\left(\frac{2ij+kj-j-2i^2-ik+1+i-i}{(i+k)(i-1)}\right)\right) \\
&\leq \sum_{j=1}^{i} \exp\left(-\frac{\beta n}{2}\left(\frac{(i-j)(2i+k-1)+(i-1)}{(i+k)(i-1)}\right)\right) \\
&\leq \exp\left(-\frac{\beta n}{2(i+k)}\right) \sum_{y=0}^{i-1} \exp\left(-\frac{\beta n}{2}\left(\frac{2i+k-1}{(i+k)(i-1)}\right) y\right) \\
&\leq \exp\left(-\frac{\beta n}{2(i+k)}\right) \frac{\exp\left(-\frac{\beta n}{2}\left(\frac{2i+k-1}{(i+k)(i-1)} i\right)\right)-1}{\exp\left(-\frac{\beta n}{2}\left(\frac{2i+k-1}{(i+k)(i-1)}\right)\right)-1}
\end{aligned}$$

Again since $\exp\left(-\frac{\beta n}{2}\left(\frac{2i+k-1}{(i+k)(i-1)}\right)\right) > \exp\left(-\frac{\beta n}{2}\left(\frac{2i+k-1}{(i+k)(i-1)} i\right)\right)$ we have:

$$\begin{aligned}
\Pr\left(S_{i,i+k+1}\right) &\leq \exp\left(-\frac{\beta n}{2(i+k)}\right) \frac{\exp\left(-\frac{\beta n}{2}\left(\frac{2i+k-1}{(i+k)(i-1)} i\right)\right)}{\exp\left(-\frac{\beta n}{2}\left(\frac{2i+k-1}{(i+k)(i-1)}\right)\right)} \\
&\leq \exp\left(-\frac{\beta n}{2(i+k)}\right) \exp\left(-\frac{\beta n}{2}\left(\frac{2i+k-1}{(i+k)(i-1)}(i-1)\right)\right) \\
&\leq \exp\left(-\frac{\beta n}{2(i+k)}\right) \exp\left(-\frac{\beta n}{2}\left(\frac{(i+k)+(i-1)}{i+k}\right)\right) \\
&\leq \exp\left(-\frac{\beta n}{2}\right) \exp\left(-\frac{\beta n}{2}\frac{i}{i+k}\right)
\end{aligned}$$

□

Corollary 9.13. *For any $0 < q < 1/2$, and a pair (i,k) with $1 \leq i < n$ and $0 \leq k < n-i$, the probability that π_i and π_{i+k+1} are swapped in $\sigma \sim \mathcal{M}_{\mathcal{AP}}(\beta, \pi)$, with $\beta > \frac{2\ln\left(\frac{1}{q}\right)(i+k)}{(2i+k)n}$ is at most:*

$$Pr\left(S_{i,i+k+1}\right) \leq q.$$

In particular, for $\beta > \frac{c\ln(2)}{n}$ with $c > 2$ independent of n, we have that for any pair (i,k) with $1 \leq i < n$ and $0 \leq k < n-i$:

$$Pr\left(S_{i,i+k+1}\right) \leq \frac{1}{2}.$$

With $O(\lceil \log n/(\beta n) \rceil)$ samples we can therefore run any $O(n \log n)$ sorting algorithm, repeating every comparisons $O(\lceil \log n/(\beta n) \rceil)$ times to guarantee that with probability $1 - O(1/n)$ all pairs are placed in the right order. The total running time of this algorithm is $O(n \log n \log n/(\beta n))$. Note

that with a careful analysis of the dependencies between the comparisons one can apply the noisy sorting algorithm in [74] to obtain an $O(n \log n)$ algorithm.

9.6 Experimental evaluation

We used both synthetic and real-world data to evaluate our algorithms and to explore the use of the AP distance as an alternative to the standard Kendall's tau distance in learning a true order from noisy sample data. We used two real-world datasets. First, we obtained a new dataset with long permutations and ground truth information, by automatically ranking Wikipedia pages with multiple ground-truth classes (a similar approach was used in [181] for linguistic graphs). We believe that our practical and efficient approach for generating classified permutations datasets will be useful in other studies. Second, we used publicly available data from human cancer studies with permutations with binary classifications.

We implemented the algorithms in C++. Each run used a single core and less than 4GB of RAM.[3]

9.6.1 Reconstructing the Hidden Permutation

In this experiment, we generated, using the algorithm defined in Section 9.4, a set P of i.i.d. permutations of size $n = 100$ from the $\mathcal{M}_{\mathcal{AP}}(\beta, \pi)$ model with different settings of β and size of $|P|$.[4] We obtained an estimate π^* of the central permutation π by applying our estimation algorithm to the set P. We then tested the quality of the estimate π^* using two measures: Correctness (Corr.), the fraction of correctly ranked pairs $i, j \in [n]$ s.t. $\pi^*(\pi_i) < \pi^*(\pi_j)$ for $i < j$; Precision at 10 (P. at 10), the fraction of elements in the first 10 positions of π^* that are in $\{\pi_1, ..\pi_{10}\}$. Both of the measures range from 0 (π^* is the inverse of π) to 1 ($\pi^* = \pi$).

The results are shown in Figure 9.1 for the algorithm in Section 9.5.2 and are consistent with our theoretical analysis (we report averages over 300 runs of the experiment). Results for the Rank-based algorithm in Section 9.5.1 are very close and thus omitted. We observe that for both algorithms fewer samples are required to correctly reconstruct π as β grows, as expected. Notice that the algorithms achieve high precision even with few samples for $\beta > \frac{1}{n}$. For the more challenging settings (e.g., $\beta = 0.01$) we observe fairly high precision with more samples. Notice also the high precision in the reconstruction of the first (and most important) positions of the central ranking even in the higher variance experiment. This confirms our theoretical insight discussed in Section 9.5 according to which the AP model has lower variance then the Mallows model using Kendall's tau distance in the first positions of the ranking.

[3] All our algorithms are easily parallelizable, if needed, but we did not pursue this direction.

[4] W.l.o.g., we set π to the identity permutation $(1, \ldots, 100)$.

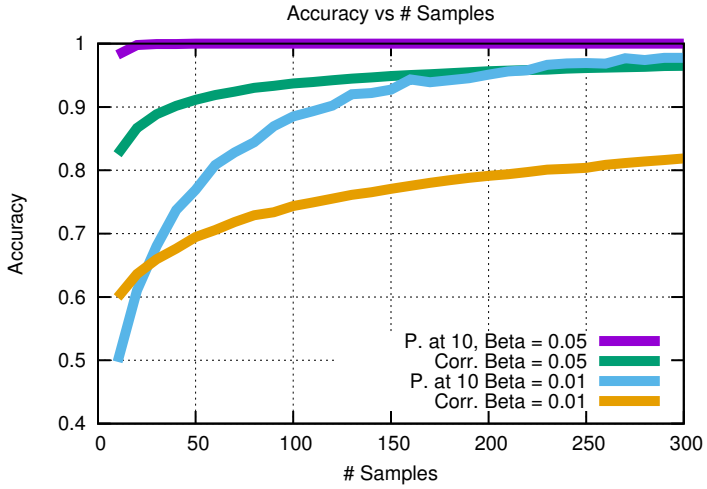


Figure 9.1: Accuracy on synthetic data

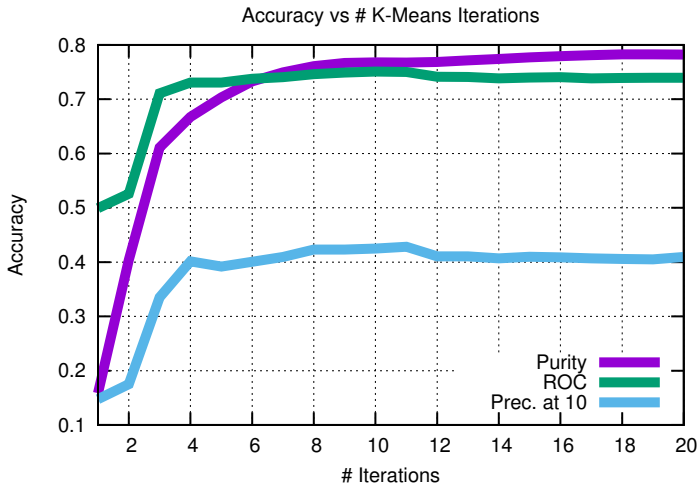


Figure 9.2: Accuracy vs. # Iterations in K-means with Real Data

9.6.2 Experiments with real-data

Clustering. We now assess the ability of a simple unsupervised clustering algorithm based on k-means, and equipped with our AP model estimators, to correctly cluster permutations from real-data for which we possess ground truth labeling. We address this problem in the context of web-pages rankings. In this context ranking correctly web-pages as well classifying pages in ground-truth categories is key to provide high quality recommendations.

Numerous algorithms, like the PageRank, are known to induce rankings over web-pages. In this experiment we use the well known Personalized PageRank [101] random walk with different seed pages to produce random permutations with ground-truth classes. Note that the AP model finds a natural application in this context as the first results in the permutations are more meaningful than

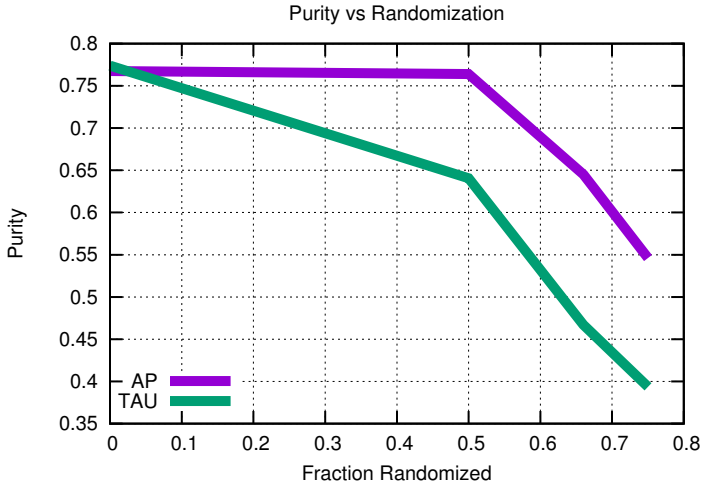


Figure 9.3: Purity vs. Level of Randomization in K-means with Real Data

the last ones.

More precisely, we used a snapshot of the entire English Wikipedia graph [27] with approximately 4.2 million nodes and 101 million edges; where nodes represent pages and edges represent hyperlinks. Each page in our dataset is classified with ground-truth categories ($\approx$ 30 thousand classes) in the `YAGO2` [105] knowledge-base taxonomy. We select a set C of `YAGO2` categories. Then for, each category $c \in C$, we create t rankings of nodes with ground-truth category c as follows: for t times, we select a page u in category c uniformly at random (u.a.r.), then we compute the the Personalized PageRank (PPR) ranking using u as seed of the walk. We set $\alpha = 0.15$ as the jump-back probability in all the walks. Notice that each of these rankings is biased towards the origin node and hence, toward the nodes belonging to category c of the seed node. This produces a set P of $|P| = t|C|$ permutations over nodes in the graph with ground-truth classes.[5]

Given rankings P as input (without their ground-truth category) our aim is to cluster them as homogeneously as possibly by category, and to derive a representative ranking from each cluster. To do so we apply a heuristic based on the k-means algorithm as follows. Given the permutations P and a value k, we first assign the permutations to k clusters C_i for $i \in [k]$ u.a.r.. Then, we apply the following two procedures in order for a certain number of iterations (or until convergence). **Step 1:** For each C_i we compute the center permutation $\bar{\pi}^i$ using the comparison-based algorithm in Section 9.5.2 over the permutations in C_i. **Step 2:** For each permutation $\sigma^j \in P$ we assign σ^j to the cluster C_i whose center $\bar{\pi}^i$ minimizes the AP cost function $d_{AP}(\bar{\pi}^i, \sigma^j)$. Finally, the output of the algorithm is a partition of the rankings in k clusters as well as a center permutation $\bar{\pi}^i$ for each cluster.

[5]We restrict these rankings to contain only the nodes associated with some category $c \in C$.

For our quantitative analysis, we selected $t = 200$ u.a.r. nodes for each of the eight well-represented categories in the Wikipedia taxonomy.[6] In this experiment $|P| = 1600$ and each permutation has $n \approx 26{,}000$ elements. To evaluate the obtained clusters, we used the purity score– i.e., the fraction of correctly classified rankings if each ranking in cluster C_i is assigned to a majority ground-truth class in C_i (purity 1 is obtained only for the perfect partitioning). Figure 9.2 shows the evolution of purity over the iterations setting $k = 8$ and averaging over 20 runs. Notice that purity converges to a high level (close to 80% percent) in a few iterations, showing that the obtained clusters are indeed close to the ground truth classes. We then evaluate the correctness of the obtained $k = 8$ centers' rankings. To do so we use the standard ROC and Precision at 10 measure obtained by considering each position of the central ranking of a cluster as correct if it belongs to a node of the majority class of that cluster. The results are shown in Figure 9.2. Notice that the algorithm converges quickly to good quality central rankings (ROC significantly higher than 50%, precision close to 40%).

Finally we tested the ability of our algorithms (and of the AP distance) to overcome noise in the lowest position of the permutations — which are more noisy in practice [233]. Consider a ranking σ in P and let c be the category to which it is assigned. We randomized the lowest γn positions, for $\gamma \in (0, 1)$, as follows: we kept the first $(1 - \gamma)n$ positions intact and then we added the remaining elements in the order of a u.a.r. ranking chosen from a *different* category. This randomization was applied to every ranking. As γ grows, the described randomization increases the amount of noise in the lower part of the rankings making them less related to the original category (and more biased towards a random different category). We then applied the k-means algorithm as before. In this experiment we compared also the result obtained by the same algorithm using the Kendall's tau distance instead of the AP distance (in Step 2) to assign permutations to centers (all the rest left equal). Figure 9.3 shows the purity reached after 10 iterations of k-means using both measures depending on the amount of randomization. We noticed that the results without randomization are qualitatively similar using the two measures. However, while the AP-based algorithm is almost unaffected even when the lowest 50% of rankings are randomized, the Kendall's tau one degrades its performance quickly. The AP maintains good purity in the clusters found even for a 75% randomization of the rankings. This confirms experimentally the intuition that the AP distance is less affected by the noise occurring in the lowest positions with respect to the Kendall's tau distance.

Classification. We now turn our attention to a supervised learning algorithm for classification based on our model. We used 10 publicly available datasets from human cancer research. From each dataset, we have a series of real-valued feature vectors measuring gene expressions or proteomic spectra — see [119] for details on the dataset. Each feature vector is assigned to one of two binary classes (e.g., "Normal vs. Tumor", "Non-relapse vs. Relapse").

[6] We used scientific fields (computer scientists, neuroscientists, mathematicians), art-related (American actors, singers, English writers) and other categories (American politicians, Olympic athletes). For this test we discarded from the sample nodes belonging to more than one category in the list.

Dataset	Prec. Tau	Prec. AP
BC1	0.662	**0.674**
BC2	**0.621**	0.601
CT	0.848	**0.868**
LA1	0.666	**0.685**
LC2	0.986	**0.993**
MB	0.613	**0.648**
OV	**0.836**	0.817
PC1	0.657	**0.667**
PC2	**0.499**	0.493
Average	0.709	**0.716**

Table 9.1: Classification Results. Average precision of the classification algorithm in the human cancer dataset. The best result for each dataset is highlighted in bold.

It has been observed that in the context of high-dimensional gene expressions, the relative order of the features is more important than their absolute magnitude [119, 85], since the relative order is less affected by pre-processing designed to overcome technological biases. Hence, as in [119], for each vector in our dataset we obtain a permutation over the set of the features by sorting them in decreasing order of value (ties broken randomly). We split each dataset in training and testing sets using 5-fold cross-validation.

We applied the following simple classification algorithm. For each class we reconstructed the center permutation using our comparison-based algorithm (in Section 9.5.2) on the set of permutations belonging to that class in the training set. We then classified each permutation in the testing set with the class of the center permutation having minimum distance to that permutation. We used as distance either AP or Kendall's tau. The results are reported in Table 9.1 where we show the average classification precision over 10 independent runs of the 5-fold cross-validation (a total of 50 independent test/train experiments). The results show that the AP distance improves over the Kendall's tau distance in most datasets, and the average precision improves as well.

These experiments confirm our original motivation for the study of the AP model suggesting that items in the beginning of the permutations are likely to have higher importance and less noise in both supervised and unsupervised learning tasks.

9.7 Conclusion

In this work we have introduced a novel stochastic permutation model based on a more nuanced permutation statistic measure that is widely used in practice. Despite its sophistication is still amenable to formal analysis. This allows us to define provably correct algorithms to learn the parameters of this model efficiently.

We believe that this is a first step towards defining even more sophisticated (and arguably more realistic) probabilistic models of rankings for which many of the results achieved in the traditional Mallows model literature could be extended.

As a future work we would like to define models that allow both ties in the rankings—which are very frequent in many practical applications—and more general cost functions as those recently defined in [233].

Chapter 10

Conclusion and Future Directions

In this thesis, we have shown how probabilistic modeling and sampling-based methods can be successfully used to tackle a wide array of challenges arising while analyzing data due to its availability. In the case for which there is low available data, difficulties arise when trying to extract *statistically significant* pattern from the data while reducing the risk of overfit, that is, the risk of erroneously inferring that properties observed on the observed data *generalize* to the entire population. In the case of a massive dataset, computational challenges arise from the cost of manipulating and moving large amounts of data towards computing the quantities of interest.

The approaches discussed in this thesis share an emphasis on the use of estimates computed using the given samples, which are then used to infer properties of the overall population. This helps in both the settings previously discussed: it allows extract insights on the large population from a smaller sample of available data, and it allows to operate on a manageable subset of a large dataset, thus considerably reducing computational requirements and effort while still extracting significant information. Towards validating the insights obtained from such estimates, it is crucial to characterize their quality by bounding the probability of observing *large deviations* for the observed values with respect to the corresponding ground truth values. In our work, we use different methods depending on the specific task at hand. In particular, we build from tools form statistical learning theory that yield uniform convergence bounds on the error distribution for a class of query functions. These bounds relate the strength of the guarantees (accuracy and confidence) to notions of complexity of the class of query functions such as the Vapnik–Chervonenkis dimension and Rademacher complexity. We also use important sampling schemes such as reservoir sampling and random pairing in our analysis of massive graph streams. All our efforts aim to produce efficient algorithms which yield results whose quality satisfies rigorous probabilistic guarantees.

Due to the intrinsic generality of the used approached we believe our result can be further developed in multiple interesting ways: we aim to improve our multiple hypotheses testing procedure for both the standard and adaptive case (`RadeFWER`, `RadeFDR`, `RadaBound`) by using tighter analysis of the generalization error based on *variance-aware* uniform convergence bound. In particular, we aim to integrate the empirical variance of the class of functions corresponding to the hypotheses

being tested in the bound on the generalization error of the estimates. By doing so, we aim to have tighter generalization bounds and, in turn, increase the statistical power of the proposed procedures. On a similar note, we aim to integrate the use of additional information on the hypotheses being tested in our procedures. In some settings, it would appear reasonable for the analyst to have some prior information or belief regarding the population and/or the set of hypotheses being tested. Such knowledge may, for example, manifest itself into the willingness to accept some observation as statistically significant even in circumstances for which the same observation would not be considered as such without prior knowledge. Similar approaches are used in the classical multiple hypothesis scenario by assigning weights to the p-values obtained for certain hypotheses. We believe that by opportunely adapting our current procedures, we will be able to allow for the integration of such user knowledge.

We also aim the extend on the graph motifs analysis framework introduced in TRIÉST and Tiered Sampling. A crucial aspect of modern social networks is their intrinsic *dynamic* nature. Social networks are a prime example of this as an incredibly high number of communications between users are established and closed every minute. As studied in our work on counting triangles in fully dynamic graph streams [51], characterizing such dynamic behavior introduces several challenges. It is then perhaps surprising that when studying the number of occurrences of a given sub-graph, the only property of interest of such a pattern is its topology. Several recent works have proposed a different characterization of the dynamic nature of sub-graph occurrences referred as "*temporal motifs*" [174, 169, 141, 147]. While the definitions proposed so far in the literature have some differences, they share the core idea of characterizing the sup-patterns of interest not only according to their topology (e.g., a triangle, 4-clique) but also according to their "*temporal profile*". The temporal profile is a property that characterizes the occurrence of the sub-pattern with respect to the temporal properties of the edges. An example of a possible temporal profile would be a minimum time span between the arrival of the first and of the last edge composing the pattern of interest. Such an example would be particularly appropriate for applications for which we care particularly about interactions (or communities) that are established in a short time, possibly as a result of some event. Another possible category of temporal profiles would characterize the *order* according to which the pattern is realized. For instance, when considering 4-cliques, we may be interested only in the cases for which one of the nodes is connected with the remaining three nodes, while the remaining communications between nodes are established afterward. Such a temporal profile would allow characterizing situations in which closely connected communities are generated as a result of the action of a "*trend setter*" party. Different notions of temporal profile share a common challenge due to the inclusion of "*time*" as a requirement for a pattern occurrence. This is particularly important in the analysis of large social networks as we can observe a large number of interactions over fairly large time-span. As an interesting direction for future work, we aim to deploy the paradigm of Trièst and Tiered Sampling to the study of *temporal graph motifs*, that is, to count the occurrences of sub-graph patterns characterized not only by their topology but also by their temporal behavior. Such results would provide means of a characterization of the evolution

of the properties of large graphs through time. We plan to build on more refined sampling schemes, such as *Weighted Reservoir Sampling* by Chao [39], in order to improve the quality of our estimates. We believe Tiered Sampling to be particularly useful for detecting temporal motifs for which the temporal profile consists of a characterization of the order according to which a given target pattern evolves over time. This is due to the possibility of selecting a rare *prototype temporal motif* to be used to detect the entire motif.

Bibliography

[1] Ehud Aharoni and Saharon Rosset. Generalized α-investing: definitions, optimality results and application to public databases. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):771–794, 2014.

[2] Nesreen K Ahmed, Nick Duffield, Jennifer Neville, and Ramana Kompella. Graph Sample and Hold: A framework for big-graph analytics. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1446–1455. ACM, 2014.

[3] Nesreen K. Ahmed, Nick Duffield, Jennifer Neville, and Ramana Kompella. Graph sample and hold: A framework for big-graph analytics. In *KDD '14*, pages 1446–1455, 2014.

[4] Nesreen K Ahmed, Nick Duffield, Theodore Willke, and Ryan A Rossi. On sampling from massive graph streams. *arXiv preprint arXiv:1703.02625*, 2017.

[5] Nesreen K Ahmed, Nick Duffield, and Liangzhen Xia. Sampling for approximate bipartite network projection. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3286–3292, 2018.

[6] Nesreen K Ahmed, Jennifer Neville, Ryan A Rossi, and Nick Duffield. Efficient graphlet counting for large networks. In *ICDM'15*, pages 1–10. IEEE, 2015.

[7] Mikel Aickin and Helen Gensler. Adjusting for multiple testing when reporting research results: the bonferroni vs holm methods. *American journal of public health*, 86(5):726–728, 1996.

[8] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[9] Pranjal Awasthi, Avrim Blum, Or Sheffet, and Aravindan Vijayaraghavan. Learning mixtures of ranking models. In *Advances in Neural Information Processing Systems*, pages 2609–2617, 2014.

[10] Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '02, pages 623–632. SIAM, 2002.

[11] John Bartholdi III, Craig A. Tovey, and Michael A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and welfare*, 6(2):157–165, 1989.

[12] Peter L. Bartlett, Stéphane Boucheron, and Gábor Lugosi. Model selection and error estimation. *Mach. Learn.*, 48:85–113, 2002.

[13] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, March 2003.

[14] Luca Becchetti, Paolo Boldi, Carlos Castillo, and Aristides Gionis. Efficient algorithms for large-scale local triangle counting. *ACM Transactions on Knowledge Discovery from Data*, 4(3):13:1–13:28, 2010.

[15] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995.

[16] Yoav Benjamini, Abba M Krieger, and Daniel Yekutieli. Adaptive linear step-up procedures that control the false discovery rate. *Biometrika*, 93(3):491–507, 2006.

[17] Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001.

[18] Yoav Benjamini, Daniel Yekutieli, et al. The control of the false discovery rate in multiple testing under dependency. *The annals of statistics*, 29(4):1165–1188, 2001.

[19] D. A Berry et al. Bayesian perspectives on multiple comparisons. *Journal of Statistical Planning and Inference*, 82(1–2), 1999.

[20] Jonathan W. Berry, Bruce Hendrickson, Randall A. LaViolette, and Cynthia A. Phillips. Tolerating the community detection resolution limit with edge weighting. *Physical Review E*, 83(5):056119, 2011.

[21] Nayantara Bhatnagar and Ron Peled. Lengths of monotone subsequences in a Mallows permutation. *Probability Theory and Related Fields*, pages 1–62, 2014.

[22] Carsten Binnig, Lorenzo De Stefani, Tim Kraska, Eli Upfal, Emanuel Zgraggen, and Zheguang Zhao. Toward sustainable insights, or why polygamy is bad for you. In *CIDR 2017, 8th Biennial Conference on Innovative Data Systems Research, Chaminade, CA, USA, January 8-11, 2017, Online Proceedings*, 2017.

[23] Gilles Blanchard and Etienne Roquain. Adaptive fdr control under independence and dependence. *arXiv preprint arXiv:0707.0536*, 2007.

[24] Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *International Conference on Machine Learning*, pages 1006–1014, 2015.

[25] Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 587–596. ACM, 2011.

[26] Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks. In *WWW'11*. ACM, 2011.

[27] Paolo Boldi and Sebastiano Vigna. The webgraph framework i: compression techniques. In *Proceedings of the 13th conference on World Wide Web*, pages 595–602. ACM, 2004.

[28] Carlo E Bonferroni. *Teoria statistica delle classi e calcolo delle probabilita*. Libreria internazionale Seeber, 1936.

[29] Ilaria Bordino, Debora Donato, Aristides Gionis, and Stefano Leonardi. Mining large networks with subgraph counting. In *ICDM'08*, pages 737–742. IEEE, 2008.

[30] Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *Proceedings of the nineteenth ACM-SIAM symposium on Discrete algorithms*, pages 268–276. Society for Industrial and Applied Mathematics, 2008.

[31] Mark Braverman and Elchanan Mossel. Sorting from noisy information. *arXiv preprint*, 2009.

[32] Marco Bressan, Flavio Chierichetti, Ravi Kumar, Stefano Leucci, and Alessandro Panconesi. Motif counting beyond five nodes. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(4):1–25, 2018.

[33] Marco Bressan, Stefano Leucci, and Alessandro Panconesi. Motivo: fast motif counting via succinct color coding and adaptive sampling. *Proceedings of the VLDB Endowment*, 12(11):1651–1663, 2019.

[34] Eric Brian. Condorcet and Borda in 1784. Misfits and documents. *Journal Electronique d'Histoire des Probabiltés et de la Statistique*, 4(1), 2008.

[35] Laurent Bulteau, Vincent Froese, Konstantin Kutzkov, and Rasmus Pagh. Triangle counting in dynamic graph streams. *Algorithmica*, 76(1):259–278, sep 2016.

[36] AE Burgess, RF Wagner, RJ Jennings, and Horace B Barlow. Efficiency of human visual signal discrimination. *Science*, 214(4516):93–94, 1981.

[37] Luciana S. Buriol, Gereon Frahling, Stefano Leonardi, Alberto Marchetti-Spaccamela, and Christian Sohler. Counting triangles in data streams. In *Proceedings of the 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '06, pages 253–262. ACM, 2006.

[38] Òscar Celma Herrada. Music recommendation and discovery in the long tail. Technical report, Universitat Pompeu Fabra, 2009.

[39] M. T. CHAO. A general purpose unequal probability sampling plan. *Biometrika*, 69(3):653–656, 12 1982.

[40] Surajit Chaudhuri, Rajeev Motwani, and Vivek Narasayya. Random sampling for histogram construction: How much is enough? In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, SIGMOD '98, pages 436–447, New York, NY, USA, 1998. ACM.

[41] You-Wei Cheah et al. Provenance quality assessment methodology and framework. *J. Data and Information Quality*, 5(3):9:1–9:20, 2015.

[42] Flavio Chierichetti, Anirban Dasgupta, Ravi Kumar, and Silvio Lattanzi. On reconstructing a hidden permutation. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 28:604–617, 2014.

[43] Fernando Chirigati, Harish Doraiswamy, Theodoros Damoulas, and Juliana Freire. Data polygamy: The many-many relationships among urban spatio-temporal data sets. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pages 1011–1025, New York, NY, USA, 2016.

[44] Fernando Chirigati et al. Data polygamy: The many-many relationships among urban spatio-temporal data sets. In *SIGMOD*, 2016.

[45] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a python package). *Neurocomputing*, 307:72 – 77, 2018.

[46] Yeounoh Chung et al. Estimating the impact of unknown unknowns on aggregate query results. In *SIGMOD*, pages 861–876, 2016.

[47] Edith Cohen, Graham Cormode, and Nick Duffield. Don't let the negatives bring you down: sampling from streams of signed updates. *ACM SIGMETRICS Performance Evaluation Review*, 40(1):343–354, 2012.

[48] Andrew Crotty, Alex Galakatos, Emanuel Zgraggen, Carsten Binnig, and Tim Kraska. Vizdom: Interactive analytics through pen and touch. *Proceedings of the VLDB Endowment*, 8(12):2024–2027, 2015.

[49] Andrew Crotty, Alex Galakatos, Emanuel Zgraggen, Carsten Binnig, and Tim Kraska. The case for interactive data exploration accelerators (ideas). In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, page 11. ACM, 2016.

[50] Yahoo! Research Webscope Datasets. Yahoo! Answers browsing behavior version 1.0. `http://webscope.sandbox.yahoo.com`, (Accessed on) September 2016.

[51] Lorenzo De Stefani, Alessandro Epasto, Matteo Riondato, and Eli Upfal. Trièst: Counting local and global triangles in fully dynamic streams with fixed memory size. *ACM TKDD'17*, 11(4):43, 2017.

[52] Kevin L Delucchi. The use and misuse of chi-square: Lewis and burke revisited. *Psychological Bulletin*, 94(1):166, 1983.

[53] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, December 2006.

[54] Jean-Claude Deville and Yves Tillé. Efficient balanced sampling: The cube method. *Biometrika*, 91(4):893–912, 2004.

[55] Persi Diaconis and Arun Ram. *Analysis of systematic scan Metropolis algorithms using Iwahori-Hecke algebra techniques*. Department of Statistics, Stanford University, 2000.

[56] Evanthia Dimara, Anastasia Bezerianos, and Pierre Dragicevic. The attraction effect in information visualization. *IEEE Trans. Vis. Comput. Graph.*, 23(1), 2016.

[57] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[58] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toni Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems*, pages 2350–2358, 2015.

[59] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

[60] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 117–126. ACM, 2015.

[61] Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th conference on World Wide Web*, pages 613–622. ACM, 2001.

[62] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, volume 3876, pages 265–284. Springer, 2006.

[63] Kacha Dzhaparidze and JH Van Zanten. On bernstein-type inequalities for martingales. *Stochastic processes and their applications*, 93(1):109–117, 2001.

[64] Jean-Pierre Eckmann and Elisha Moses. Curvature of co-links uncovers hidden thematic layers in the World Wide Web. *Proceedings of the National Academy of Sciences*, 99(9):5825–5829, 2002.

[65] Bradley Efron and Trevor Hastie. *Computer Age Statistical Inference*, volume 5. Cambridge University Press, 2016.

[66] H. Ehsan, M. A. Sharaf, and P. K. Chrysanthis. Muve: Efficient multi-objective view recommendation for visual data exploration. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 731–742, May 2016.

[67] Alessandro Epasto, Silvio Lattanzi, Vahab Mirrokni, Ismail Oner Sebe, Ahmed Taei, and Sunita Verma. Ego-net community mining applied to friend suggestion. *Proceedings of the VLDB Endowment*, 9(4):324–335, 2015.

[68] Alessandro Epasto, Silvio Lattanzi, and Mauro Sozio. Efficient densest subgraph computation in evolving graph. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 300–310. ACM, 2015.

[69] Dhivya Eswaran, Christos Faloutsos, Sudipto Guha, and Nina Mishra. Spotlight: Detecting anomalies in streaming graphs. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1378–1386, 2018.

[70] Adir Even et al. Dual assessment of data quality in customer databases. *J. Data and Information Quality*, 1(3), 2009.

[71] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Comparing top k lists. *SIAM Journal on Discrete Mathematics*, 17(1):134–160, 2003.

[72] Wenfei Fan et al. Discovering conditional functional dependencies. *IEEE Trans. Knowl. Data Eng.*, 23(5):683–698, 2011.

[73] Vicky Fasen, Claudia Klüppelberg, and Annette Menzel. Quantifying extreme risks. In *Risk-A Multidisciplinary Introduction*, pages 151–181. Springer, 2014.

[74] Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

[75] Helmut Finner and Veronika Gontscharuk. Controlling the familywise error rate with plug-in estimator for the proportion of true null hypotheses. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1031–1048, 2009.

[76] Irene Finocchi, Marco Finocchi, and Emanuele G. Fusco. Clique counting in mapreduce: Algorithms and experiments. *JEA*, 20:1.7:1–1.7:20, October 2015.

[77] R.A. Fisher. *The design of experiments*. Oliver and Boyd, Edinburgh, Scotland, 1935.

[78] Michael A. Fligner and Joseph S. Verducci. Distance Based Ranking Models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3), 1986.

[79] Michael A. Fligner and Joseph S. Verducci. Multistage ranking models. *Journal of the American Statistical Association*, 83(403), 1988.

[80] Dean P Foster and Robert A Stine. α-investing: a procedure for sequential control of expected false discoveries. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(2):429–444, 2008.

[81] The R Foundation. The r project for statistical computing. `https://www.r-project.org/`.

[82] David A Freedman. On tail probabilities for martingales. *the Annals of Probability*, pages 100–118, 1975.

[83] Juliana Freire et al. Exploring what not to clean in urban data: A study using new york city taxi trips. *IEEE Data Eng. Bull.*, 39(2), 2016.

[84] Yulia Gavrilov, Yoav Benjamini, Sanat K Sarkar, et al. An adaptive step-down procedure with proven fdr control under independence. *The Annals of Statistics*, 37(2), 2009.

[85] Donald Geman, Christian d'Avignon, Daniel Q. Naiman, and Raimond L. Winslow. Classifying gene expression profiles from pairwise mRNA comparisons. *Statistical applications in genetics and molecular biology*, 2004.

[86] Rainer Gemulla, Wolfgang Lehner, and Peter J. Haas. Maintaining bounded-size sample synopses of evolving datasets. *The VLDB Journal*, 17(2):173–201, 2008.

[87] Alison L Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International statistical review*, 70(3):419–435, 2002.

[88] Nadine Gissibl, Claudia Klüppelberg, and Johanna Mager. Big data: Progress in automating extreme risk analysis. In *Berechenbarkeit der Welt?*, pages 171–189. Springer, 2017.

[89] Phillip Good. *Permutation tests: a practical guide to resampling methods for testing hypotheses*. Springer Science & Business Media, 2013.

[90] Priscilla E Greenwood and Michael S Nikulin. *A guide to chi-squared testing*, volume 280. John Wiley & Sons, 1996.

[91] Max Grazier G'Sell et al. Sequential selection procedures and false discovery rate control. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(2), 2016.

[92] Hua Guo, Steven Gomez, Caroline Ziemkiewicz, and David Laidlaw. A case study using visualization interaction logs and insight. *IEEE Trans. Vis. Comput. Graph.*, 2016.

[93] Wenge Guo. A note on adaptive bonferroni and holm procedures under dependence. *Biometrika*, 96(4):1012–1018, 2009.

[94] Hack your way to scientific glory. https://projects.fivethirtyeight.com/p-hacking/, 2018. Accessed: 2018-08-01.

[95] A. Hajnal and E. Szemerédi. Proof of a conjecture of P. Erdős. In *Combinatorial theory and its applications, II (Proc. Colloq., Balatonfüred, 1969)*, pages 601–623, 1970.

[96] Bronwyn H Hall, Adam B Jaffe, and Manuel Trajtenberg. The NBER patent citation data file: Lessons, insights and methodological tools. Technical report, National Bureau of Economic Research, 2001.

[97] Peter Hall and Christopher C Heyde. *Martingale limit theory and its application*. Academic press, 2014.

[98] Pat Hanrahan. Analytic database technologies for a new kind of user: the data enthusiast. In *SIGMOD*, 2012.

[99] Sariel Har-Peled and Micha Sharir. Relative (p, ε)-approximations in geometry. *Discrete & Computational Geometry*, 45(3):462–496, 2011.

[100] Moritz Hardt and Jonathan Ullman. Preventing false discovery in interactive data analysis is hard. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 454–463. IEEE, 2014.

[101] Taher H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th conference on World Wide Web*, pages 517–526. ACM, 2002.

[102] Taher H Haveliwala, Aristides Gionis, Dan Klein, and Piotr Indyk. Evaluating strategies for similarity search on the web. In *Proceedings of the 11th conference on World Wide Web*, pages 432–442. ACM, 2002.

[103] Yosef Hochberg. A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4):800–802, 1988.

[104] Yosef Hochberg and Yoav Benjamini. More powerful procedures for multiple significance testing. *Statistics in medicine*, 9(7):811–818, 1990.

[105] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. YAGO2: Exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th conference companion on World wide web*, pages 229–232. ACM, 2011.

[106] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[107] Gerhard Hommel. A stagewise rejective multiple test procedure based on a modified bonferroni test. *Biometrika*, 75(2):383–386, 1988.

[108] Yifan Huang, Haiyan Xu, Violeta Calian, and Jason C Hsu. To permute or not to permute. *Bioinformatics*, 22(18):2244–2248, 2006.

[109] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[110] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *IJF*, 22(4):679–688, 2006.

[111] Stratos Idreos et al. Overview of data exploration techniques. In *SIGMOD*, 2015.

[112] Amazon Inc. Amazon mechanical turk. `https://www.mturk.com`.

[113] John P. A. Ioannidis. Why most published research findings are false. *Plos Med*, 2(8), 2005.

[114] Shweta Jain and C. Seshadhri. A fast and provable method for estimating clique counts using turán's theorem. In *WWW '17*, pages 441–449, 2017.

[115] Harold Jeffreys. *The theory of probability*. OUP Oxford, 1998.

[116] Madhav Jha, C. Seshadhri, and Ali Pinar. A space-efficient streaming algorithm for estimating transitivity and triangle counts using the birthday paradox. *ACM Transactions on Knowledge Discovery from Data*, 9(3):15:1–15:21, 2015.

[117] Madhav Jha, C. Seshadhri, and Ali Pinar. A space-efficient streaming algorithm for estimating transitivity and triangle counts using the birthday paradox. *ACM TKDD*, 9(3):15:1–15:21, February 2015.

[118] Madhav Jha, C. Seshadri, and Ali Pinar. Path sampling: A fast and provable method for estimating 4-vertex subgraph counts. In *WWW '15*, pages 495–505, 2015.

[119] Yunlong Jiao and Jean-Philippe Vert. The Kendall and Mallows kernels for permutations. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1935–1944, 2015.

[120] M. I. Jordan. The era of big data. *ISBA Bulletin*, 18(2), 2011.

[121] Hossein Jowhari and Mohammad Ghodsi. New streaming algorithms for counting triangles in graphs. In *Computing and Combinatorics: 11th Annual International Conference*, COCOON '05, pages 710–716. Springer, 2005.

[122] Niranjan Kamat, Prasanth Jayachandran, Karthik Tunga, and Arnab Nandi. Distributed and interactive cube exploration. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 472–483. IEEE, 2014.

[123] T. Kanamori, T. Suzuki, and M. Sugiyama. f -divergence estimation and two-sample homogeneity test under semiparametric density-ratio models. *IEEE Transactions on Information Theory*, 58(2):708–720, Feb 2012.

[124] Daniel M. Kane, Kurt Mehlhorn, Thomas Sauerwald, and He Sun. Counting arbitrary subgraphs in data streams. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming*, volume 7392 of *Lecture Notes in Computer Science*, pages 598–609. Springer, 2012.

[125] Kimberly Keeton et al. Do you know your iq?: a research agenda for information quality in systems. *SIGMETRICS Performance Evaluation Review*, 37(3):26–31, 2009.

[126] Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the thirty-ninth ACM symposium on Theory of computing*, pages 95–103. ACM, 2007.

[127] Alicia Key, Bill Howe, Daniel Perry, and Cecilia Aragon. Vizdeck: self-organizing dashboards for visual analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 681–684. ACM, 2012.

[128] Rogier Kievit et al. Simpson's paradox in psychological science: a practical guide. *Frontiers in psychology*, 4, 2013.

[129] Albert Kim, Eric Blais, Aditya Parameswaran, Piotr Indyk, Sam Madden, and Ronitt Rubinfeld. Rapid sampling for visualizations with ordering guarantees. *Proceedings of the VLDB Endowment*, 8(5):521–532, 2015.

[130] Adam Kirsch, Michael Mitzenmacher, Andrea Pietracaprina, Geppino Pucci, Eli Upfal, and Fabio Vandin. An efficient rigorous approach for identifying statistically significant frequent itemsets. *Journal of the ACM (JACM)*, 59(3):12, 2012.

[131] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[132] Mihail N. Kolountzakis, Gary L. Miller, Richard Peng, and Charalampos E. Tsourakakis. Efficient triangle counting in large graphs via degree-based vertex partitioning. *Internet Mathematics*, 8(1–2):161–185, 2012.

[133] Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, July 2001.

[134] Aryeh Kontorovich. Concentration in unbounded metric spaces and algorithmic stability. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 28–36, 2014.

[135] Tim Kraska et al. Mlbase: A distributed machine-learning system. In *CIDR*, 2013.

[136] Ravi Kumar and Sergei Vassilvitskii. Generalized distances between rankings. In *Proceedings of the 19th conference on World wide web*, pages 571–580. ACM, 2010.

[137] Konstantin Kutzkov and Rasmus Pagh. On the streaming complexity of computing local clustering coefficients. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 677–686. ACM, 2013.

[138] Konstantin Kutzkov and Rasmus Pagh. Triangle counting in dynamic graph streams. In *Scandinavian Symposium and Workshops on Algorithm Theory*, SWAT '14, pages 306–318. Springer, 2014.

[139] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600. ACM, 2010.

[140] Matthieu Latapy. Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theoretical Computer Science*, 407(1):458–473, 2008.

[141] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1):2, 2007.

[142] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *journal of the Association for Information Science and Technology*, 58(7):1019–1031, 2007.

[143] M. Lichman. UCI machine learning repository, 2013.

[144] Yongsub Lim and U Kang. MASCOT: Memory-efficient and accurate sampling for counting local triangles in graph streams. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 685–694. ACM, 2015.

[145] Yongsub Lim and U Kang. Mascot: Memory-efficient and accurate sampling for counting local triangles in graph streams. In *KDD'15*, pages 685–694. ACM, 2015.

[146] Paul Liu, Austin R Benson, and Moses Charikar. Sampling methods for counting temporal motifs. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 294–302, 2019.

[147] Paul Liu, Austin R. Benson, and Moses Charikar. Sampling methods for counting temporal motifs. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, page 294–302, New York, NY, USA, 2019. Association for Computing Machinery.

[148] Zhicheng Liu et al. The Effects of Interactive Latency on Exploratory Visual Analysis. *IEEE Trans. Vis. Comput. Graph.*, 20(12), 2014.

[149] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. immens: Real-time visual querying of big data. In *Computer Graphics Forum*, volume 32, pages 421–430. Wiley Online Library, 2013.

[150] Maarten Löffler and Jeff M. Phillips. Shape fitting on point sets with probability distributions. *CoRR*, abs/0812.2967, 2008.

[151] Tyler Lu and Craig Boutilier. Learning Mallows models with pairwise preferences. In *Proceedings of the 28th International Conference on Machine Learning*, pages 145–152, 2011.

[152] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, Nov 2007.

[153] Colin L. Mallows. Non-null ranking models. i. *Biometrika*, 44(1/2):pp. 114–130, 1957.

[154] Madhusudan Manjunath, Kurt Mehlhorn, Konstantinos Panagiotou, and He Sun. Approximate counting of cycles in streams. In *European Symposium on Algorithms*, ESA '11, pages 677–688. Springer, 2011.

[155] Adam Marcus et al. Crowdsourced data management: Industry and academic perspectives. *Foundations and Trends in Databases*, 6(1-2):1–161, 2015.

[156] John H. McDonald. *Handbook of Biological Statistics*. Sparky House Publishing, Baltimore, Maryland, USA, second edition, 2009.

[157] Christopher Meek and Marina Meila. Recursive inversion models for permutations. In *Advances in Neural Information Processing Systems*, pages 631–639, 2014.

[158] Marina Meila, Kapil Phadnis, Arthur Patterson, and Jeff A Bilmes. Consensus ranking under the exponential model. *arXiv preprint*, 2012.

[159] Massimo Melucci. On rank correlation in information retrieval evaluation. In *ACM SIGIR Forum*, volume 41, pages 18–33. ACM, 2007.

[160] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[161] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[162] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *ACM SIGCOMM'07*, pages 29–42, 2007.

[163] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.

[164] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2nd edition, 2005.

[165] Michael Mitzenmacher and Eli Upfal. *Probability and Computing*. Cambridge university press, 2017.

[166] D. Moritz, C. Wang, G. L. Nelson, A H. Lin, M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. In *IEEE Trans. Visualization and Comp. Graphics (Proc. InfoVis)*, 2018.

[167] Jerzy Neyman and Elizabeth L Scott. Consistent estimates based on partially consistent observations. *Econometrica: Journal of the Econometric Society*, pages 1–32, 1948.

[168] Dang Tuan Nhon et al. Transforming scagnostics to reveal hidden features. *IEEE Trans. Vis. Comput. Graph.*, 20(12), 2014.

[169] Vincenzo Nicosia, John Tang, Cecilia Mascolo, Mirco Musolesi, Giovanni Russo, and Vito Latora. Graph metrics for temporal networks. In Petter Holme and Jari Saramäki, editors, *Temporal Networks*, pages 15–40. Springer, 2013.

[170] Luca Oneto, Alessandro Ghio, Davide Anguita, and Sandro Ridella. An improved analysis of the Rademacher data-dependent bound using its self bounding property. *Neural Networks*, 44:107–111, 2013.

[171] Rasmus Pagh and Charalampos E. Tsourakakis. Colorful triangle counting and a MapReduce implementation. *Information Processing Letters*, 112(7):277–281, March 2012.

[172] Rasmus Pagh and Charalampos E. Tsourakakis. Colorful triangle counting and a mapreduce implementation. *Inf. Process. Lett.*, pages 277–281, 2012.

[173] Kirill Paramonov, Dmitry Shemetov, and James Sharpnack. Estimating graphlet statistics via lifting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 587–595, 2019.

[174] Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, page 601–610, New York, NY, USA, 2017. Association for Computing Machinery.

[175] Ha-Myung Park and Chin-Wan Chung. An efficient MapReduce algorithm for counting triangles in a very large graph. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 539–548. ACM, 2013.

[176] Ha-Myung Park and Chin-Wan Chung. An efficient mapreduce algorithm for counting triangles in a very large graph. In *CIKM '13*, pages 539–548, 2013.

[177] Ha-Myung Park, Sung-Hyon Myaeng, and U. Kang. PTE: Enumerating trillion triangles on distributed systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM, 2016.

[178] Ha-Myung Park, Francesco Silvestri, U. Kang, and Rasmus Pagh. MapReduce triangle enumeration with guarantees. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1739–1748. ACM, 2014.

[179] A. Pavan, Kanat Tangwongsan, Srikanta Tirthapura, and Kun-Lung Wu. Counting and sampling triangles from a graph stream. *VLDB'13*, pages 1870–1881, 2013.

[180] Aduri Pavan, Kanat Tangwongsan, Srikanta Tirthapura, and Kun-Lung Wu. Counting and sampling triangles from a graph stream. *Proceedings of the VLDB Endowment*, 6(14):1870–1881, 2013.

[181] Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *ACL (1)*, pages 1341–1351, 2013.

[182] Leo Pipino et al. Data quality assessment. *Commun. ACM*, 45(4):211–218, 2002.

[183] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, volume 5, pages 2–4, 2005.

[184] Tiberiu Popoviciu. Sur les èquations algèbriques ayant toutes leurs racines rèelles. *Mathematica*, 9:129–145, 1935.

[185] Frank C Porter. Testing consistency of two histograms. *arXiv preprint arXiv:0804.0380*, 2008.

[186] Tao Qin, Xiubo Geng, and Tie-Yan Liu. A new probabilistic model for rank aggregation. In *Advances in neural information processing systems*, pages 1948–1956, 2010.

[187] X. Qin, Y. Luo, N. Tang, and G. Li. Deepeye: An automatic big data visualization framework. *Big Data Mining and Analytics*, 1(1):75–82, March 2018.

[188] Mahmudur Rahman, Mansurul Alam Bhuiyan, and Mohammad Al Hasan. Graft: An efficient graphlet counting method for large graph analysis. *IEEE TKDE*, 26(10):2466–2478, 2014.

[189] Payam Refaeilzadeh, Lei Tang, Huan Liu, and M. TAMER ÖZSU. *Cross-Validation*, pages 532–538. Springer US, Boston, MA, 2009.

[190] Matteo Riondato, Mert Akdere, Ugur Çetintemel, Stanley B. Zdonik, and Eli Upfal. The vc-dimension of queries and selectivity estimation through sampling. *CoRR*, abs/1101.5805, 2011.

[191] Matteo Riondato et al. Efficient discovery of association rules and frequent itemsets through sampling with tight performance guarantees. *TKDD*, 8(4), 2014.

[192] Matteo Riondato et al. Mining frequent itemsets through progressive sampling with rademacher averages. In *KDD*, 2015.

[193] Matteo Riondato et al. ABRA: approximating betweenness centrality in static and dynamic graphs with rademacher averages. *CoRR*, abs/1602.05866, 2016.

[194] Joseph P Romano, Azeem M Shaikh, and Michael Wolf. Control of the false discovery rate under dependence using the bootstrap and subsampling. *Test*, 17(3):417, 2008.

[195] Joseph P Romano and Michael Wolf. Exact and approximate stepdown methods for multiple hypothesis testing. *Journal of the American Statistical Association*, 100(469), 2005.

[196] Daniel Russo and James Zou. How much does your data exploration overfit? controlling bias via information usage. *arXiv preprint arXiv:1511.05219*, 2015.

[197] Donald G. Saari. Which is better: the Condorcet or Borda winner? *Social Choice and Welfare*, 26(1):107–129, 2006.

[198] Tetsuya Sakai. Evaluation with informational and navigational intents. In *Proceedings of the 21st conference on World Wide Web*, pages 499–508. ACM, 2012.

[199] Babak Salimi, Johannes Gehrke, and Dan Suciu. Hypdb: Detect, explain and resolve bias in olap. *arXiv preprint arXiv:1803.04562*, 2018.

[200] Seyed-Vahid Sanei-Mehri, Ahmet Erdem Sariyuce, and Srikanta Tirthapura. Butterfly counting in bipartite networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2150–2159, 2018.

[201] Sanat K Sarkar. On methods controlling the false discovery rate. *Sankhyā: The Indian Journal of Statistics, Series A (2008-)*, pages 135–168, 2008.

[202] Sanat K Sarkar, Wenge Guo, and Helmut Finner. On adaptive procedures controlling the familywise error rate. *Journal of Statistical Planning and Inference*, 142(1):65–78, 2012.

[203] William W. Schapire, Cohen Robert E., and Yoram Singer. Learning to order things. In *Advances in Neural Information Processing Systems*, volume 10, page 451. MIT Press, 1998.

[204] M Schemper. A survey of permutation tests for censored survival data. *Communications in Statistics-Theory and Methods*, 13(13):1655–1665, 1984.

[205] Erich Schubert, Remigius Wojdanowski, Arthur Zimek, and Hans-Peter Kriegel. On Evaluation of outlier rankings and outlier scores. In *SDM*, pages 1047–1058, 2012.

[206] Jinwook Seo and Ben Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2):96–113, July 2005.

[207] Juliet Popper Shaffer. Multiple hypothesis testing. *Annual review of psychology*, 46, 1995.

[208] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[209] Grace S. Shieh. A weighted Kendall's tau statistic. *Statistics & probability letters*, 39(1):17–24, 1998.

[210] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[211] Yedendra Babu Shrinivasan and Jarke J van Wijk. Supporting the analytical reasoning process in information visualization. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1237–1246. ACM, 2008.

[212] Zbyněk Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62(318), 1967.

[213] R John Simes. An improved bonferroni procedure for multiple tests of significance. *Biometrika*, 73(3):751–754, 1986.

[214] Matthew Skala. Hypergeometric tail inequalities: ending the insanity. *arXiv preprint*, 1311.5939, 2013.

[215] Johannes Sorz, Martin Fieder, Bernard Wallner, and Horst Seidler. High statistical noise limits conclusiveness of ranking results as a benchmarking tool for university management. *PeerJ PrePrints*, 3:e1162, 2015.

[216] Charles Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:88–103, 1904.

[217] Spurious correlations. http://tylervigen.com/old-version.html, 2018. Accessed: 2018-08-01.

[218] Thomas Steinke and Jonathan Ullman. Interactive fingerprinting codes and the hardness of preventing false discovery. In *Conference on Learning Theory*, pages 1588–1628, 2015.

[219] Jean M. Steppe and Kenneth W. Bauer. Improved feature screening in feedforward neural networks. *Neurocomputing*, 13(1):47 – 58, 1996.

[220] Melissa K Stern and James H Johnson. Just noticeable difference. *The Corsini Encyclopedia of Psychology*, pages 1–2, 2010.

[221] John D Storey, Jonathan E Taylor, and David Siegmund. Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(1):187–205, 2004.

[222] W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization*, volume 1905, pages 861–871. International Society for Optics and Photonics, 1993.

[223] Siddharth Suri and Sergei Vassilvitskii. Counting triangles and the curse of the last reducer. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 607–614. ACM, 2011.

[224] Jonathan Taylor and Robert J Tibshirani. Statistical learning and selective inference. *Proceedings of the National Academy of Sciences*, 112(25):7629–7634, 2015.

[225] The Koblenz Network Collection (KONECT). Last.fm song network dataset. `http://konect.uni-koblenz.de/networks/lastfm_song`, Accessed on September 2016.

[226] Charalampos E Tsourakakis, U. Kang, Gary L. Miller, and Christos Faloutsos. Doulion: counting triangles in massive graphs with a coin. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 837–846. ACM, 2009.

[227] Charalampos E Tsourakakis, Mihail N Kolountzakis, and Gary L Miller. Triangle sparsifiers. *Journal of Graph Algorithms and Applications*, 15(6):703–726, 2011.

[228] John W Tukey et al. Comparing individual means in the analysis of variance. *Biometrics*, 5(2):99–114, 1949.

[229] V. Vapnik et al. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.*, 16(2), 1971.

[230] Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, 2015.

[231] Manasi Vartak et al. SEEDB: efficient data-driven visualization recommendations to support visual analytics. *PVLDB*, 8(13), 2015.

[232] Manasi Vartak, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. Seedb: automatically generating query visualizations. *Proceedings of the VLDB Endowment*, 7(13):1581–1584, 2014.

[233] Sebastiano Vigna. A weighted correlation index for rankings with ties. In *Proceedings of the 24th Conference on World Wide Web*, pages 1166–1176, 2015.

[234] Jeffrey S. Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, 1985.

[235] Jiannan Wang et al. A sample-and-clean framework for fast and accurate query processing on dirty data. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, pages 469–480, 2014.

[236] Peter H Westfall, S Stanley Young, et al. *Resampling-based multiple testing: Examples and methods for p-value adjustment*, volume 279. John Wiley & Sons, 1993.

[237] Kanit Wongsuphasawat et al. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans. Vis. Comput. Graph.*, 22(1), 2016.

[238] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2648–2659. ACM, 2017.

[239] xkcd. Significant. https://xkcd.com/882/, 2018. Accessed: 2018-08-01.

[240] Emine Yilmaz and Javed A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 15th ACM conference on Information and knowledge management*, pages 102–111. ACM, 2006.

[241] Emine Yilmaz, Javed A. Aslam, and Stephen Robertson. A new rank correlation coefficient for information retrieval. In *Proceedings of the 31st annual ACM SIGIR conference*, pages 587–594. ACM, 2008.

[242] Emanuel Zgraggen, Alex Galakatos, Andrew Crotty, Jean-Daniel Fekete, and Tim Kraska. How progressive visualizations affect exploratory analysis. *IEEE Trans. Vis. Comput. Graph.*, 2016.

[243] Emanuel Zgraggen, Zheguang Zhao, Robert C. Zeleznik, and Tim Kraska. Investigating the effect of the multiple comparisons problem in visual analysis. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, Montreal, QC, Canada, April 21-26, 2018*, page 479, 2018.

[244] Zheguang Zhao, Lorenzo De Stefani, Emanuel Zgraggen, Carsten Binnig, Eli Upfal, and Tim Kraska. Controlling false discoveries during interactive data exploration. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, pages 527–540, 2017.

[245] Alain F. Zuur, Elena N. Ieno, and Chris S. Elphick. A protocol for data exploration to avoid common statistical problems. *Methods in Ecology and Evolution*, 1(1):3–14, 2010.