L-Diversity for Data Analysis: Data Swapping with Customized Clustering

Sam Boger Advisor: Roberto Tamassia
Department of Computer Science, Brown University

December 2020

Abstract

Data anonymization should support the analysts who intend to use the anonymized data. Releasing datasets that contain personal information requires anonymization that balances privacy concerns while preserving the utility of the data. This work shows how choosing anonymization techniques with the data analysts' requirements in mind improves effectiveness quantitatively, by minimizing the discrepancy between querying the original data versus the anonymized result, and qualitatively, by simplifying the workflow for querying the data.

1 Introduction

Data sharing offers immense opportunity as modern machine learning models and human analysis often depends on large training sets. However, valuable information is frequently also personal, and privacy concerns can prevent data owners from sharing their data. This work considers tabular data with a fixed set of attributes for every row in the table where every row describes one entity. The attributes either relate to the identity of the entity, called Quasi-Identifiers (QIs), or to a sensitive piece of information about that entity, called sensitive values. An examples of this situation are medical records where a patient's personal and demographic information would be QIs and the medical diagnoses would be sensitive values. The primary privacy risk considered in this line of research is that publishing this data could allow an adversary with access to the published data to associate an individual to their sensitive values. Several approaches have been offered in order to alleviate this privacy concern while still sharing useful data to the public. These approaches all involve trading off some utility of the data, usually by modifying the data itself before publishing it, with the strength of the privacy protections they offer.

Shifting perspectives to the intended consumers of the published data can shed light on how to choose among these different approaches and tradeoffs. In particular, imagine analysts know the schema of the original dataset and already know what kinds of queries they care most about issuing. Now, utility can be defined as minimizing the discrepancy the analysts will see between querying the original data and the anonymized data. Furthermore, changing the schema or data definitions when anonymizing the data would disrupt their workflow and should be avoided if possible.

Different prior works in this field address these design decisions separately which invites an opportunity to combine these techniques. In particular, allowing the analysts to easily customize the heuristics used during anonymization, as well as publishing anonymized data with identical schema and data definitions to the original dataset, will best meet analysts' requirements while still protecting individuals' privacy.

1.1 Unique Contributions

This project intends to combine the insights from these different works to describe a promising anonymization technique that adequately protects privacy, preserves utility, and produces an easy-to-use resulting dataset. In particular, the proposed method uses a customized distance function [9] to cluster the input data and performs data swapping [18] within these clusters to produce a K-anonymous, L-diverse [13], and θ-diverse anonymized dataset.

2 Related Work

2.1 Anonymization Definitions

K-anonymity [19] is one of the most influential works for defining privacy-preserving anonymization. Removing personally identifiable information from datasets can still leave QIs which can be combined with separate data sources by an adversary to re-identify an individual. If the original dataset has sensitive information about individuals, then this re-identification leaks the association of that individual to those sensitive values and thus the effort to remove the identifiers was insufficient to preserve privacy. K-anonymity ad-

QI: Age	QI: Education	Sensitive value: Occupation
20-30	Bachelors	Sales
20-30	Bachelors	Sales
20-30	Bachelors	Sales
30-40	Doctorate	Prof-specialty
30-40	Doctorate	Exec-managerial
30-40	Doctorate	Armed-Forces

Figure 1: 3-anonymous, not 2-diverse table. Note that all rows share their QIs with at least 2 other rows, making it 3-anonymous. However, there is one group that has only 1 distinct sensitive value, meaning it is not 2-diverse.

dresses this concern by ensuring that at least k rows have identical QIs so that any individual effectively blends in with at least $k-1$ other individuals. To create these homogeneous groups, the QIs are "generalized", or mapped onto a less granular space, e.g. an integer age may be mapped instead to a range of ages. See figure 1 for a table that has been generalized and is now 3-anonymous. Furthermore, certain rows can be suppressed (removed from the dataset entirely) to reduce the amount of generalization necessary. Now, individual re-identification from outside data sources can only imply membership in a group of size at least k. One weakness of this approach is that it is possible that all sensitive values for a K-anonymous group are the same, and identifying an individual within that group reveals their sensitive values regardless of the group's size. Many subsequent works expanded on this original definition and discussed improvements to this method.

L-diversity [13] is one such extension to this method which places a further constraint on the anonymized dataset that each group of records with identical QIs must also satisfy. L-diversity can be used to describe a couple different properties. Here, we call a group L-diverse if and only if every group contains at least L distinct sensitive values, and the table is L-diverse if and only if every record belong to a group that is L-diverse. This reference includes more definitions and clarifications of this property, but we will be using this basic form that simply counts distinct sensitive values for consistency with other cited works.

A related condition to L-diversity is θ-diversity as defined by a paper [8] on computing L-diverse clusters. Call the size of a group of records N, and the most frequent sensitive value in the group S. The group is θ-diverse if and only if $\frac{S}{N} \leq \theta$, and the table is θ-diverse if and only if every record belong to a group that is θ-diverse. See figure 2 for an example of a table that is not 0.5-θ-diverse. Intuitively this means that learning which group an individual belongs to will never reveal its associated sensitive value with probability greater than θ. Auxiliary information may exist that one or more sensitive value is highly unlikely for an individual, so a diverse set of sensitive values within each group protects against this process of elimination style of attack. The tunable parameters L and θ in this system control the likelihood of this leakage by varying the diversity of every group.

Further conditions have also been considered, including T-closeness [10] where each group of homogenized records must contain a similar (within a factor of T) distribution of sensitive values as the entire original table. This emphasizes privacy guarantees at the expense of increased difficulty in computing the anonymized table and decreased utility for analysis. For this project we do not consider T-closeness although similar ideas would apply in principle.

An alternative approach to sharing anonymized datasets is to use Differential Privacy [7] (DP) to preserve privacy. Instead of modifying the underlying data, DP allows analysts to query the full unmodified tables, but adds noise to the results returned by the query. This approach intends to make it very difficult for an analyst to determine whether any particular individual is present in the dataset or not. The design of this system requires the data owner to answer queries interactively, since the queries run over the raw sensitive data which the analyst cannot have direct access to, as well as allotting and managing a "privacy budget" either globally or per-analyst. Both BP and one-time anonymization make tradeoffs between usability, utility, and privacy.

2.2 Methods of Computing Anonymizations

In order to maintain better utility of the anonymized table, as well as more efficient computation of the anonymized dataset, many papers have explored clustering methods for constructing K-anonymity or L-diversity groups. By grouping together similar records, the correlations that exist in the original dataset can be better preserved while still providing the privacy guarantees of these anonymization methods [2,12,14]. Different clustering algorithms are described in these works (weighted feature C-means, density clustering) which all seek to collect similar records into groups that will then be generalized to produce the anonymized datasets. Most of these methods address K-anonymity only, as efficiently computing L-diverse and θ-diverse clusters presents further challenges. One algorithm for computing K-anonymous, L-diverse, and θ-Diverse clusters is effective in practice but the algorithm lacks complete definition and does not have theoretical guarantees of runtime or optimality in the general case [8].

QI: Age	QI: Education	Sensitive value: Occupation
20-30	Bachelors	Sales
20-30	Bachelors	Sales
20-30	Bachelors	Sales
20-30	Bachelors	Craft-repair
30-40	Doctorate	Sales
30-40	Doctorate	Armed-Forces
30-40	Doctorate	Prof-specialty
30-40	Doctorate	Sales

Figure 2: 3-anonymous, 2-diverse, not 0.5-θ-diverse table. Note that all rows share their QIs with at least 2 other rows, making it 3-anonymous. Both groups have at least 2 distinct sensitive values, meaning it is 2-diverse. However, the most common sensitive value in one group represents $\frac{3}{4}$ of the total group, meaning it is not 0.5-θ-diverse.

In addition to varying the clustering algorithms, other prior work has modified the distance metric between rows to better preserve utility for a particular use case [9]. Here, the data analysts who intend to use the anonymized data contribute training data regarding which rows they consider similar to teach the anonymization system a customized distance metric. Clustering according to this metric still produces a K-anonymous dataset, but the groups contain rows that are more similar according to this customized notion of distance.

Another paper [18] has an approach to preserving utility where instead of producing a dataset with groups of rows that have identical quasi-identifiers through generalization and suppression, all records in a given group are randomly swapped with their associated sensitive values. Even for an adversary with knowledge of which rows were in the same group, this ensures a similar guarantee as a θ-diverse table where re-identifying a certain row only reveals the correct sensitive value with probability $\leq \theta$. Figure 3 shows an example of data swapping to preserve privacy.

Similarly to shuffling within each group of records, another paper [20] proposes publishing the sets of QIs in a separate table than the sensitive values but with a join key specific to each group–which they call an anatomy. Therefore, similarly to data swapping [18], the QIs in the original dataset are published but it is not revealed exactly which sensitive value corresponds to which row.

2.3 Utility Analysis

There are multiple techniques for analyzing how well utility is preserved in the anonymized output of these algorithms as compared to the original data. Most common are table-wide statistical metrics (e.g. [18]) and the quality of machine learning model trained over the anonymized data (e.g. [15]). Earlier work [20] uses a test suite of queries to assess utility. Other efforts attempt to reconcile the different metrics but conclude that it is difficult to correlate the statistical measures of the anaonymized datasets to the effectiveness of models trained over those datasets [16].

The anatomy work [20] constructs counting queries by randomly choosing attribute values to select for at various configurations. They compute the relative error by running the same queries on a dataset with no anonymization versus L-diverse datasets produced with generalization or anatomy. Their results demonstrate that the relative error in answering these queries using an anatomy is less than the relative error when using a generalization.

To evaluate data swapping [18], the authors quantify the improved utility of a data swapped and generalized tables at the aggregate level by computing and comparing the correlation between QIs and sensitive values in the different anonymized datasets compared to the original dataset. Data swapping preserves correlation much better than generalization in their experiments.

2.4 Attacking Anonymized Data

Other works on the privacy properties of k-anonymity addresses privacy risks in different ways. The original papers make reasonable assumptions about the types of background knowledge an adversary might have when doing formal calculations about probabilities of a successful attack. They note that in principle it is impossible to account for arbitrary background knowledge of attackers.

Subsequent works advance the theory surrounding this topic by bringing in techniques from other areas of computer science. Information theory can provide a framework for considering the tension in anonymization between providing accurate information to the legitimate users of the anonymized dataset while withholding private information from adversaries with access to the same dataset [11]. With a more sophisticated model of this tradeoff, the problem can be viewed as an optimization of this tradeoff for average or worst-case information leakage to an adversary [5].

For any particular k-anonymous dataset, it is possible to launch a customized attack with background knowledge to attempt to recover information about the original dataset. One such example [17] was able to reidentify particular rooms in a

k-anonymous dataset unless the anonymization also employed suppression to remove outliers in the data.

3 Method

3.1 Clustering Algorithm

The algorithm used for clustering extends the algorithm in prior work [8] on computing clusters that satisfy K-anonymity, L-diversity, and θ-diversity, which together will be called *K,L,θ-Diversity*. At a high level, this algorithm works with the following steps:
Input: N points.
Output: K,L,θ-Diversity satisfying clusters containing a subset of the N points.

1. If there any unassigned points, take an arbitrary point and assign it to a new cluster with this point as the only element.

2. Find the closest unassigned point to the current cluster. If the new point has a sensitive value not present in the cluster, assign it to the cluster. Otherwise, assign it to a backup queue which will be emptied later. Repeat this step until the cluster has L distinct values, then go back to step 1. If there are no more unassigned points, go to the next step instead.

3. For each point in the backup queue, find the closest eligible cluster. An eligible cluster must have fewer than K elements and adding this new point must not violate the θ-diversity property. If no clusters are eligible, unassign this new point. Otherwise, add this point to this closest eligible cluster.

4. If any points are unassigned (i.e. there were no eligible cluster for a point in the previous step), go back to step 1 and attempt to form new clusters. Otherwise proceed.

5. Attempt to merge clusters until all are satisfied according to the privacy requirements. To do this, for each cluster with K, L, or θ diversity unsatisfied, evaluate merging it with each other cluster. Identify the closest cluster where, if merged, the result would satisfy all diversity requirements. If one exists, merge those clusters. Otherwise, merge it with the closest cluster that does not violate the θ diversity requirement if one exists. If none exist, this cluster cannot be included in the output.

6. Output all clusters that satisfy K,L,θ-Diversity.

Call the number of clusters at a given time C. Each time step 2 runs, it compares the distance from a cluster to each unassigned point. In total this is quadratic in the number of points that do not start new clusters in step 1, which is $N-C$. Now there are U unassigned points. Each of these is compared to each cluster, for a total of $U*C$ comparisons in step 3. In total so far, this runtime is $O((N-C)^2+U*C)$. If we assume that $N >> C$ and $U \approx N$ this can simplify to $O(N^2)$. Step 4 causes steps 1-3 to be repeated, say, M more times, for a total runtime so far of $O((1+M)*N^2)$. Step 5 merges clusters by comparing each one to the remaining clusters, for a quadratic number of comparison in the number of clusters at this point, C, which is upper bounded by N. The output step is linear in the number of clusters. The total runtime is therefore $O(M*N^2)$.

Each comparison is a call to the distance function. In this analysis, the distance function is assumed to run in constant time. However, this need not be the case, as some distances may depend on each of the points in the cluster. For the analysis we assume an efficient implementation that is approximately constant time.

In the above analysis, the biggest unknown to the runtime is how many repetitions of steps 1-3 are necessary, M. When M is low or even 0, the clustering is quadratic in the number of input points N, but in the worst case it is similar to N making the runtime $O(N^3)$ which would preclude using this algorithm except on tiny datasets. The privacy parameters affect this behavior since they impact how many clusters are created and which points are eligible to be added into the clusters.

Analyzing instead how well the algorithm performs in terms of the quality of output, the biggest unknown is how many clusters end up satisfying the privacy constraints and can be output instead of being omitted. The privacy parameters also influence this as they impact the composition of the clusters and also which clusters are eligible to be merged together in step 5. The properties of the clusters themselves are not guaranteed to be optimal in terms of minimizing distances within clusters or other global metrics since the algorithm takes a greedy or arbitrary approach in each step.

The benefit of this algorithm is that in practice it can run quickly, can output clusters that contain all the input points, and satisfies all three strict privacy constraints. Some other clustering methods run in $O(N^3)$ regardless of privacy parameters. The implementation of the clustering algorithm for this work and the experimental setup can be found on GitHub [1].

3.2 Data Swapping

Once the clusters are computed, the sensitive values in the cluster are then permuted randomly between members of the cluster. All the rows of all the clusters are then output as an anonymized table. This table has identical schema and data to the original dataset except for the permuted sensitive values within each cluster.

3.3 Comparing Tables

The clustering algorithm makes use of a distance function that takes two clusters, where a single point can be considered a singleton cluster, and returns a real number that represents

Input			Output		
QI: Age	QI: Education	Sensitive value: Occupation	QI: Age	QI: Education	Sensitive value: Occupation
21	Bachelors	Sales	21	Bachelors	Sales
22	Bachelors	Armed-Forces	22	Bachelors	Craft-repair
24	Doctorate	Sales	24	Doctorate	Armed-Forces
30	Bachelors	Craft-repair	30	Bachelors	Sales

Figure 3: An example of data swapping. The input is a 4-anonymous, 3-diverse, 0.5-θ-diverse cluster. The output has the same QIs, but with the sensitive values randomly permuted within the cluster. Note that the sensitive value may not change for a particular row. Note further that the QIs do not need to be generalized since the association with sensitive values are disguised in the output by this permutation of sensitive values rather than homogenizing the QIs.

how close the clusters are. Different distance functions can be substituted into the algorithm to change the composition of the clusters. Running the clustering and data swapping components with various distance functions produces different versions of the anonymized tables that can be compared.

Comparing the different tables can be done by issuing queries to each of the tables and comparing the results. To keep the comparison fair, the same queries should be run against each table. Rather than choosing static queries, the queries can be randomly generated according to a given template. Comparing the results of running many such random queries over each table can reveal the utility of the data-swapped tables, and in particular which types of queries work best with which tables. The precise tables, query template, and query randomization that instantiate this method are described in the following section.

4 Experiments and Evaluations

4.1 Dataset

The dataset that most frequently occurs in prior works on this topic is the UCI ADULT database [6] which will be used here as well. This dataset contains 32,561 census records with demographic, financial, and other personal information, but is publicly available information made available explicitly for use in research. The fields used as QIs and the sensitive value, and the number of possible values for each, are described in figure 4.

4.2 Privacy Parameters

The default anonymization will be performed by forming groups with the following configurations, which together satisfy K,L,θ-Diversity:

- K-anonymity with $K = 10$ meaning every group has at least 10 rows.
- L-diversity with $L = 5$ meaning every group has at least 5 distinct sensitive values.
- θ-diversity with $\theta = 0.3$ meaning no sensitive value accounts for more than $\frac{3}{10}$ of the sensitive values in each group.

4.3 Performance

Clustering is done with the algorithm described in section 3.1. On this dataset and with these choices of parameters, the adapted algorithm succeeds in running, outputs clusters containing all input rows, and satisfies these strict privacy constraints. The runtime to produce an anonymization was around 210 seconds. All experiments were run on a 4.0 GHz Intel i7-6700k processor with 16GB of RAM.

The clustering runtimes for this dataset seem to scale quadratically with the size of the input. This matches the analysis of the algorithm in section 3.1 since in these experiments there were no unassigned points in step 4, so $M = 0$. Figure 5 shows how runtime varies as the number of rows changes. This runtime chart was produced by measuring the runtime of the clustering algorithm on subsets of the full dataset which has 32,561 rows. Extrapolating this trendline into larger datasets indicates that this would take approximately 35 minutes to run on a dataset with 100,000 rows and 61 hours on 1,000,000 rows. To run on datasets on the order of millions of rows, therefore, algorithmic improvements would be necessary.

4.4 Distance Metrics

The clusters are compared based on a default distance function unless otherwise specified. The default is a linear combination of orthogonal components that correspond to each QI. For age and sex the algorithm computes the average of each weighted between 0 and 1, with the two sexes present in the dataset arbitrarily mapped to 1 and 0 respectively, and ages linearly mapped to [0,1]. These cluster averages are each then combined using weighted univariate distance as discussed in this prior work [3]. The other categorical dimensions are defined to be of distance 1 if there is any value in one cluster that is not present in the other and 0 otherwise. The total distance between two clusters is then a linear combination of each of

Attribute	Used as	Type	Distinct Values
Race	QI	Category	5
Education	QI	Category	16
Sex	QI	Category	2
Age	QI	Integer	74 [17,90]
Occupation	Sensitive value	Category	14

Figure 4: Attributes in the ADULT database.

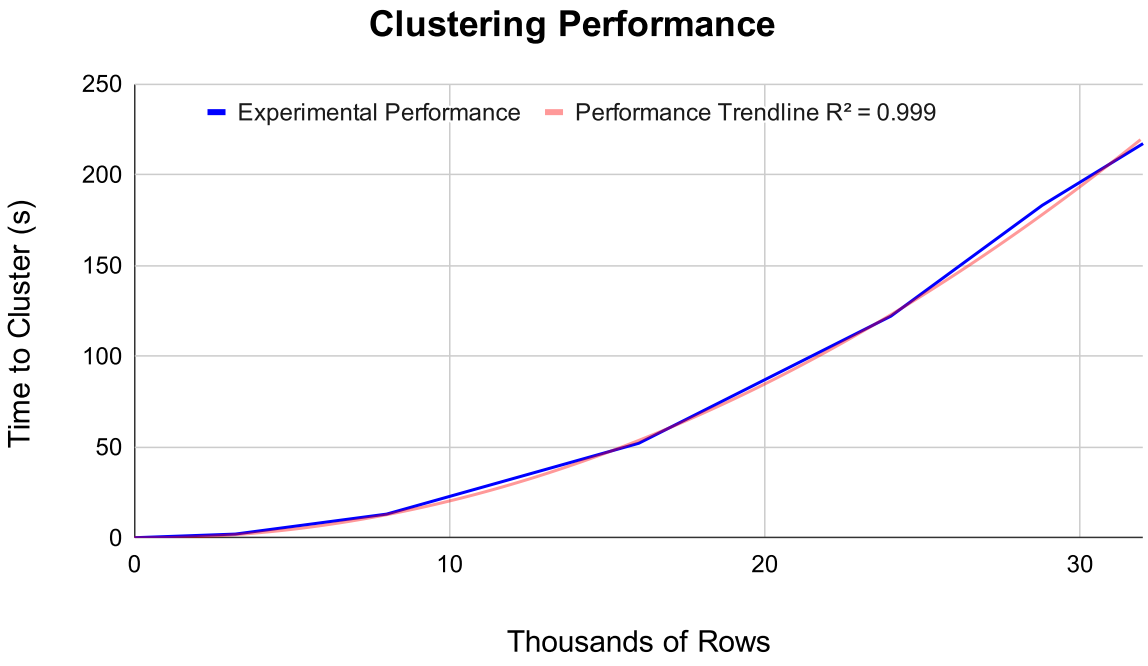

Figure 5: Clustering Performance

these distances.

The weights of the linear combination, which default to 1, impact how clustering behaves. For instance, weighting the age distance more heavily will produce clusters that have lower variance of age within clusters. These weights were varied in the different experiments below to demonstrate how this impacts the utility of the resulting tables. The table names and the weights used in their distance function are described in figure 6.

4.5 Simulations

To understand how the different distance metrics affect utility, the following experiments were run. Queries were generated and run on each of the datasets to compare their results. Every query returns the frequency count of occupations, which is the sensitive value in our dataset, restricted to a subset of the quasi-identifier's values. Specifically, for each quasi-identifier, a random subset of the possible values are included in the WHERE clause of the query. Building on prior evaluation methods [20], the number of values to include for each quasi-identifier is called the *selectivity* of that dimension. By default, each selectivity is set to include approximately half of the possible values of a quasi-identifier for each query, so of the 16 possible education values in the dataset, 8 distinct values are selected uniformly at random for each query. The queries have the format shown in 7.

This query format was chosen to be typical of how an analyst might use this dataset to learn about correlations in the data. Comparing histograms that cover different segments of the population can provide a compelling visual representation of the observed differences in those segments. For example, with this dataset, queries of this form could be used to measure how the distributions of occupations differs between older and younger people of a particular demographic, or with multiple datasets collected over time, these types queries could help measure whether certain occupations are become more diverse according to race or sex. One key difference between the mentioned prior work's experiments [20] and this experiment is that these queries return histograms of the sensitive values selected by the query rather than total counts which allows for more fine-grained differentiation of the quality of results.

A query, Q_i, is generated according to this procedure and executed across all datasets: the original unmodified table, the BaseTable, and each of the four test tables with their customized distance functions (see figure 6). Each query on a table results in a frequency count of occupations which is then compared to that of the original unmodified table using the χ^2 distance function.

One round of simulation consists of m such random queries, where $m = 1000$ for this experiment. The equations for relative error are below. The relative errors for each table in the experiment are shown in figure 8. A value less than 100% indicates that the custom distance function reduced the total error while a value greater than 100% indicates the opposite. Relative error is used since the magnitude of χ^2 varies significantly with how frequent the randomly selected attributes appear in the dataset, so the relative metric is more informative and stable between executions than an absolute metric.

$$\chi^2(X,Y) = \sum_{j=1}^{n} \frac{(X_j - Y_j)^2}{X_j + Y_j} \tag{1}$$

$$BaseError = \sum_{i=1}^{m} \chi^2(Q_i(BaseTable), Q_i(Original)) \tag{2}$$

$$TestError = \sum_{i=1}^{m} \chi^2(Q_i(TestTable), Q_i(Original)) \tag{3}$$

$$RelativeError = 100\% * \frac{TestError}{max(BaselineError, 1)} \tag{4}$$

4.6 Varying Selectivity

The selectivity for each quasi-identifier affects how well a particular dataset will perform in the simulation. For example the extreme case where every quasi-identifier has 100% selectivity results in any anonymized dataset performing perfectly, since the whole dataset is selected by the query, and the anonymized dataset has the same overall frequency count of sensitive values as the original. The other extreme where no rows are selected also results, vaccuously, in all datasets performing identically.

Varying one quasi-identifiers' selectivity at a time reveals the impact of the different distance metrics used in clustering. In the following charts, each of the four quasi-identifiers had their selectivity varied from the minimum to maximum possible values. For instance, with the education quasi-identifier, the simulation was run with default selectivity for the other quasi-identifiers, but with education selectivity varying from including only one random education value in the query to all 16, for a total of 16 rounds of simulations. At each selectivity, the chart shows the relative error for each table 9,10,11,12.

4.7 Evaluation

The basic simulation with default selectivities (figure 8) shows that custom distance metrics perform similarly to the BaseTable for random queries with default selectivities.

Varying selectivity, however, shows that the distance metric can be tailored to the types of queries that will be made. In the graphs, the tables which were produced with a distance function prioritizing the QI being varied are shown with the bolded line in the graph. At low selectivities for a QI (towards the left of the X-axis in these graphs), the dataset produced with a distance metric weighted towards that low-selectivity QI performed best among all tested tables. Additionally, for

Table	Race_Weight	Education_Weight	Sex_Weight	Age_Weight
BaseTable	1	1	1	1
RaceTable	5	1	1	1
EducationTable	1	5	1	1
SexTable	1	1	5	1
AgeTable	1	1	1	5

Figure 6: Custom Distance Tables

```
SELECT occupation, COUNT(*) as total
FROM adult
WHERE education IN ("E1", ..., "E8") AND
      sex in ("S1") AND
      race in ("R1", "R2", "R3") AND
      age > A1 AND
      age < A2
GROUP BY occupation;
```

Figure 7: Query Format used in Experiment. Values in the WHERE clause are distinct and selected uniformly at random from the attribute's distinct values.

Table	RelativeError
BaseTable	100.0%
RaceTable	95.3%
EducationTable	99.2%
SexTable	117.4%
AgeTable	94.8%

Figure 8: Default Sensitivities

all experiments where the selectivity was at or below 50%, the matching table outperformed the BaseTable. This aligns with the intuition behind custom distance function and clustering, as the sensitive values are swapped between rows with similar quasi-identifiers according to that distance function. Queries specific to one attribute have lower error when the clustering prioritizes that same attribute than when it does not.

Another observation of the charts is that the relative error of the EducationTable grows significantly as the education selectivity gets close to 100%. This pattern is somewhat present in the SexTable as well with respect to sex selectivity, although since there are only two possible selectivities for this QI the trend is less clear. This could be related to the fact that education and sex have the highest statistical correlation with the sensitive value of occupation compared to the other QIs. Further study is necessary to explain this observation more thoroughly. The main conclusion of the analysis, however, is concerned with low selectivity where all charts show similar behavior anyway.

5 Privacy Properties

The primary privacy goal of all the anonymization methods covered in this paper is to reduce the likelihood of an adversary associating an entity to its sensitive value. Prior work on K-anonymity, L-diversity, and θ-diversity assume that an adversary can identify an individual with a row within a group, but hides the sensitive value among the other members of the group.

In this data-swapping implementation, we further assume that although the output does not explicitly indicate which cluster a row belongs to, that the adversary may still learn this. In an extreme case where the clusters are disjoint across a particular attribute like age, and the adversary knows the age of their target, this assumption is plausible. In a typical case, the clusters being unlabelled in the output is an extra layer of obscurity, but the privacy properties hold even it it were revealed.

By ensuring that every cluster meets the requirements of K,L,θ-Diversity, this solution gains the privacy properties of each. Although these protections are not perfect individually nor in combination, as evaluated in this critique [4], they do offer non-negligible benefits. It is worth noting that clustering could improve resilience against attackers with background knowledge over arbitrary grouping. For example, if the adversary has to guess which sensitive value corresponds to a given row within a cluster, they can increase their odds of a successful guess by ruling out unlikely or implausible values. Say that using the dataset described in section 4.1 a data-swapped group has one young individual and the rest are older, and all the sensitive values but one correspond to an occupation requiring a lot of experience, the attacker may have a good chance at guessing that the young individual has the occupation that does not require much experience. However, given that the clusters prioritize members with similar quasi-identifiers including age, these implausible values are likely less frequent.

Another desired privacy property could be to prevent an adversary from learning whether an entity is present in the dataset at all, regardless of their sensitive value. This is a fundamental weakness of both the anatomy and data swapping approaches, since the full set of quasi-identifiers are present in the anonymized table, giving the adversary a very good chance of succeeding in such a membership test. However,

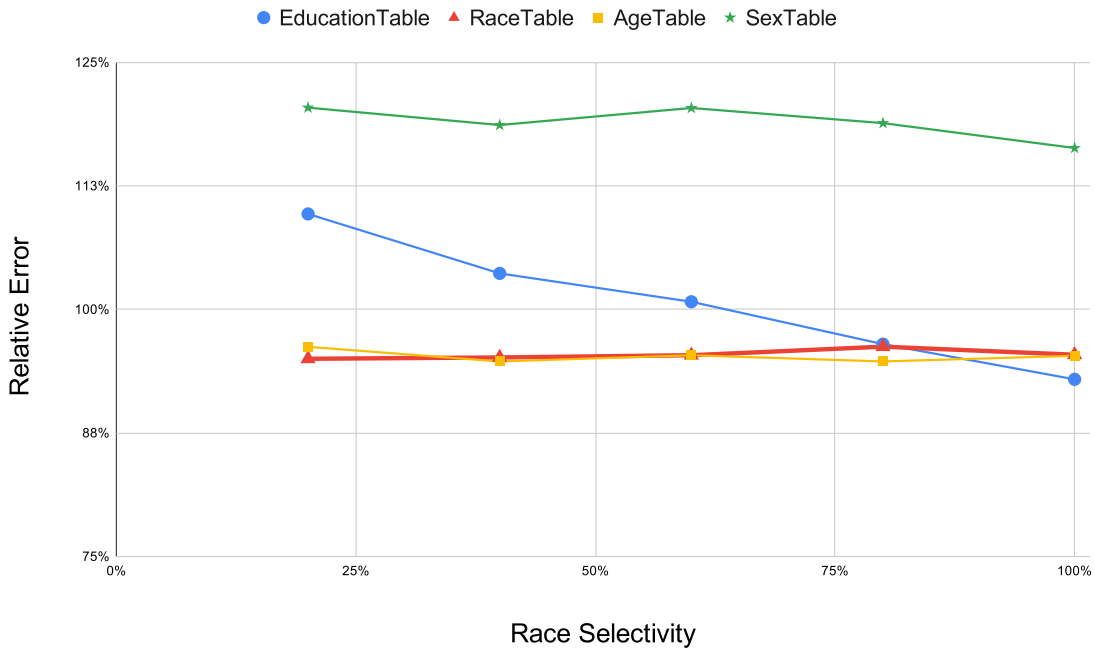

Figure 9: RaceTable Relative Error by Selectivity.
At each selectivity, one simulation with 1000 queries of the form in figure 7 are run over each table.

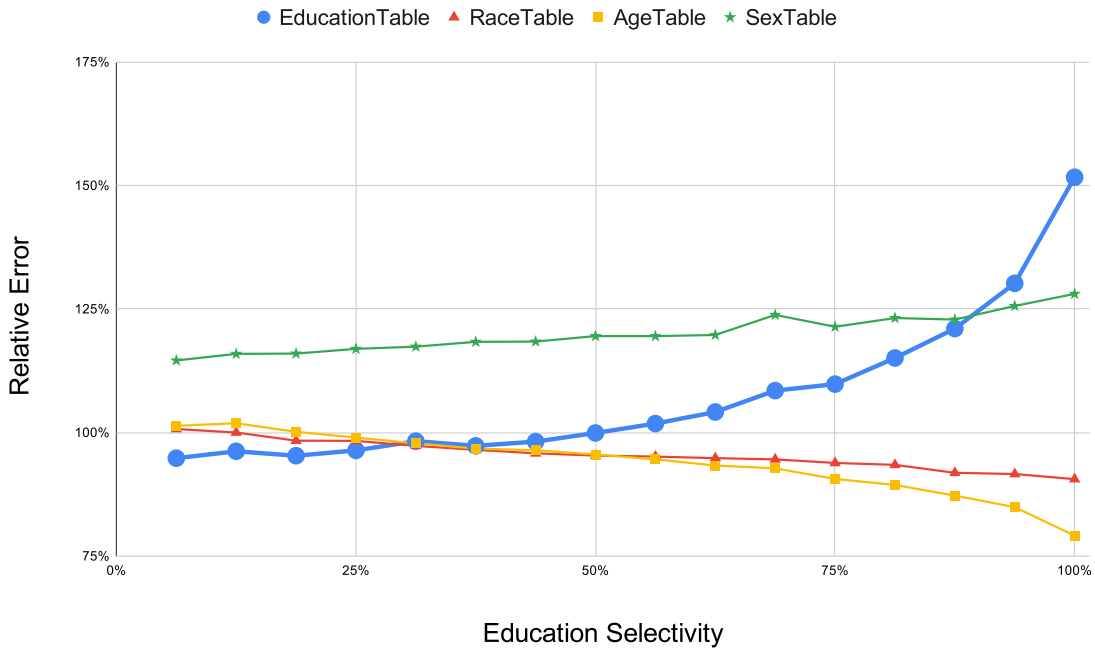

Figure 10: EducationTable Relative Error by Selectivity.
At each selectivity, one simulation with 1000 queries of the form in figure 7 are run over each table.

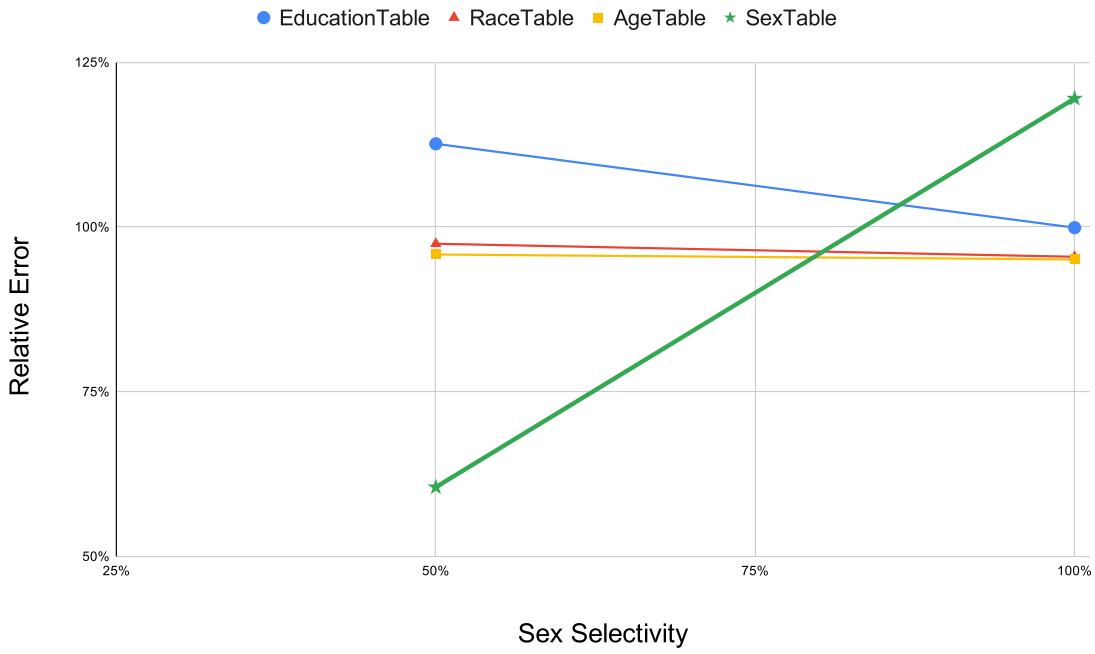

Figure 11: SexTable Relative Error by Selectivity.
At each selectivity, one simulation with 1000 queries of the form in figure 7 are run over each table.

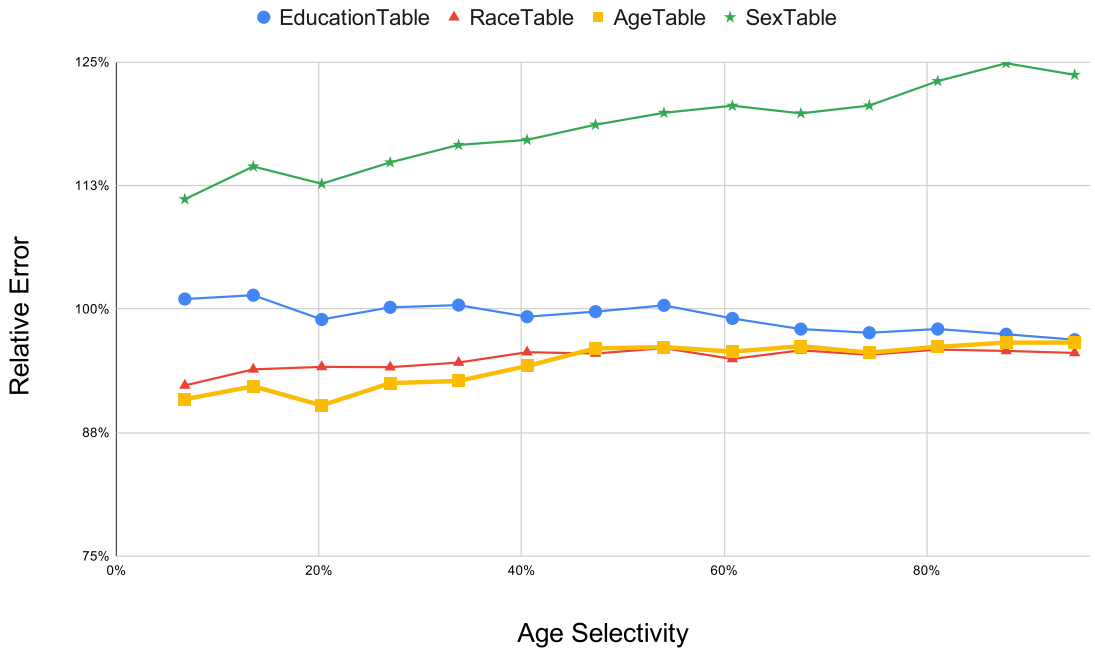

Figure 12: AgeTable Relative Error by Selectivity.
At each selectivity, one simulation with 1000 queries of the form in figure 7 are run over each table.

none of the alternative methods fully address membership testing either as even tables produced with generalization do not protect against this type of attack in principle or in practice.

6 Future Work

In terms of the cluster algorithm used, future work could improve how well it performs when the privacy constraints are such that the current version runs for an unacceptable duration or fails to include rows in its output. In a related direction, better understanding how to set privacy constraints that make the algorithm succeed depending on the structure of the underlying data would improve how easily this can be applied to new datasets.

In terms of the customized clustering, optimizing the distance functions for utility is an open research area. This work showed that varying it can improve certain kinds of utility, but exactly how much can be gained is unclear. It would also be interesting to apply this approach to different datasets to evaluate how generally these results hold.

For the usability benefits of this method, further research into real world case studies of when data anonymization is used, or is considered but rejected, would help to understand exactly which obstacles are the highest priority to tackle. Are current methods sufficient, do they fail in terms of difficult of analysis, or do they degrade quality too much?

7 Conclusion

Privacy preserving anonymization techniques allow data owners to make tradeoffs between the privacy risk of releasing data and the potential utility of the data they release. Practical applications of these techniques should consider how to minimize the disruption that these techniques cause for data analysts in terms of usability as well as impact on the quality of their results. Data swapping offers less disruption in terms of producing functional analysis than generalization-based methods since the output has an identical schema to the original dataset. Clustering, and in particular customized clustering, shows promise for improving the accuracy of certain kinds of queries over the anonymized table. As shown in this work, queries that are highly selective according to one attribute benefit from clustering that focuses on grouping rows with similar values in that attribute. Privacy techniques that are minimally disruptive in both process and quality have a greater chance of seeing adoption and improving privacy for all.

References

[1] Samuel Boger. L-Diversity for Data Analysis, 2020. https://github.com/SamBoger/Query-Accuracy-on-l-Diversity.

[2] Chuang-Cheng Chiu and Chieh-Yuan Tsai. A k-anonymity clustering method for effective data privacy preservation. In *International Conference on Advanced Data Mining and Applications*, pages 89–99. Springer, 2007.

[3] Josep Domingo-Ferrer and Josep Maria Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and data Engineering*, 14(1):189–201, 2002.

[4] Josep Domingo-Ferrer and Vicenç Torra. A critique of k-anonymity and some of its enhancements. In *Third International Conference on Availability, Reliability and Security*, pages 990–993. IEEE, 2008.

[5] Flávio du Pin Calmon and Nadia Fawaz. Privacy against statistical inference. In *50th annual Allerton conference on communication, control, and computing (Allerton)*, pages 1401–1408. IEEE, 2012.

[6] Dheeru Dua and Casey Graff. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, 2017. http://archive.ics.uci.edu/ml.

[7] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[8] Gaoming Yang, Jingzhao Li, Shunxiang Zhang, and Li Yu. An enhanced l-diversity privacy preservation. In *10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 1115–1120, 2013.

[9] Ruoxi Jia, Fisayo Caleb Sangogboye, Tianzhen Hong, Costas Spanos, and Mikkel Baun Kjærgaard. PAD: protecting anonymity in publishing building related datasets. In *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*, pages 1–10, 2017.

[10] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond *k*-anonymity and *l*-diversity. In *IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[11] Cheng Liu, Youliang Tian, Jinbo Xiong, Yanhua Lu, Qiuxian Li, and Changgen Peng. Towards attack and defense views to K-Anonymous using information theory approach. *IEEE Access*, 7:156025–156032, 2019.

[12] Jie Liu, Shou-Lin Yin, Hang Li, and Lin Teng. A density-based clustering method for k-anonymity privacy protection. *Journal of Information Hiding and Multimedia Signal Processing*, 8(1):12–18, 2017.

[13] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. *L*-Diversity: privacy beyond *k*-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es, 2007.

[14] Sang Ni, Mengbo Xie, and Quan Qian. Clustering Based K-anonymity Algorithm for Privacy Preservation. *IJ Network Security*, 19(6):1062–1071, 2017.

[15] Fisayo Caleb Sangogboye, Ruoxi Jia, Tianzhen Hong, Costas Spanos, and Mikkel Baun Kjærgaard. A framework for privacy-preserving data publishing with enhanced utility for cyber-physical systems. *ACM Transactions on Sensor Networks (TOSN)*, 14(3-4):1–22, 2018.

[16] Tanja Šarčević, David Molnar, and Rudolf Mayer. An Analysis of Different Notions of Effectiveness in k-Anonymity. In *International Conference on Privacy in Statistical Databases*, pages 121–135. Springer, 2020.

[17] Jens Hjort Schwee, Fisayo Caleb Sangogboye, and Mikkel Baun Kjærgaard. Evaluating practical privacy attacks for building data anonymized by standard methods. In *International Workshop on Security and Privacy for the Internet-of-Things*, 2019.

[18] Jordi Soria-Comas and Josep Domingo-Ferrer. Probabilistic *k*-anonymity through microaggregation and data swapping. In *IEEE International Conference on Fuzzy Systems*, pages 1–8. IEEE, 2012.

[19] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[20] Xiaokui Xiao and Yufei Tao. Anatomy: Simple and effective privacy preservation. In *Proceedings of the 32nd international conference on Very large data bases*, pages 139–150. VLDB Endowment, 2006.