CSCI 1952Q: Algorithmic Aspects of Machine Learning (Spring 2024)

Written Assignment 4

Due at 1:00pm ET, Friday, April 12

1. (6 points) Consider the problem of counting the number of distinct elements in a data stream. Let $1 \le a_1, \dots, a_n \le m$ denote the first n elements in the data stream. Let d denote the number of distinct elements in $(a_1, \dots, a_n)$.

   Consider the following algorithm. Algorithm 1 hashes every element in the stream, maintains the t smallest distinct hash values, and then uses the t-th smallest hash value to estimate d.

   We write $[n]$ for $\{1, \dots, n\}$.

   Algorithm 1: Estimating the number of distinct elements.

   Input : $m \ge 10, n \ge 1, 0 < \epsilon < 1$, and a stream of n numbers $1 \le a_1, \dots, a_n \le m$.
   Output: an estimation of the number of distinct elements in $(a_1, \dots, a_n)$.

   ```
   Let M = m^3 and t = 40/ε^2.
   Suppose h is a hash function that maps [m] to [M] uniformly at random.
   Initialize S ← ∅.
   for i = 1 to n do
       if h(a_i) ∉ S then
           if |S| < t then
               S ← S ∪ {h(a_i)}.
           else
               If h(a_i) is smaller than the largest number in S, replace that number with h(a_i).
   if |S| < t then
       return |S|.
   else
       Let v be the largest number in S.
       return d̃ = tM/v.
   ```

   (1) Assume that h can be stored for free and $h(x)$ can be evaluated in $O(1)$ time. Show that implementing Algorithm 1 using (balanced) binary search trees requires $O(\frac{\log m}{\epsilon^2})$ bits of space, and each iteration of the for loop runs in time $O(\log m \log(1/\epsilon))$.

   Next, we will prove one side of the correctness of Algorithm 1: $\Pr[\widetilde{d} > (1+\epsilon)d] \le \frac{1}{10}$ (over the randomness in h).

   The event $\frac{tM}{v} = \widetilde{d} > (1+\epsilon)d$ happens iff $v < \frac{tM}{(1+\epsilon)d}$. In other words, for $\widetilde{d} > (1+\epsilon)d$ to happen, there must be at least t hash values that are less than $\frac{tM}{(1+\epsilon)d} \le (1 - \frac{\epsilon}{2})\frac{tM}{d}$.

Because the output of Algorithm 1 does not depend on the order of the elements, we can assume w.l.o.g. that $a_1, \ldots, a_d$ are the d distinct elements. Let $X_i \in \{0,1\}$ be the indicator random variable for the event $h(a_i) \le (1-\frac{\epsilon}{2})\frac{tM}{d}$. Let $Y = \sum_{i=1}^{d} X_i$.

That is, Y is the total number of hash values that are below the threshold, and we want to upper bound $\Pr[Y \ge t]$.

(2) Prove that $\mathbb{E}[X_i] \le (1-\frac{\epsilon}{2})\frac{t}{d}$ and $\mathbb{E}[Y] \le (1-\frac{\epsilon}{2})t$.

(3) Prove that $\mathrm{var}[X_i] \le (1-\frac{\epsilon}{2})\frac{t}{d} \le \frac{t}{d}$ and $\mathrm{var}[Y] \le t$.

(Hint: You can use the following facts without proving them.)

- For two random variables X_1 and X_2, we have $\mathbb{E}[X_1 + X_2] = \mathbb{E}[X_1] + \mathbb{E}[X_2]$.
- For two independent random variables X_1 and X_2, $\mathrm{var}[X_1 + X_2] = \mathrm{var}[X_1] + \mathrm{var}[X_2]$.

Consequently, by Chebyshev's inequality, Parts (3) and (4), and the choice of $t = \frac{40}{\epsilon^2}$,

$$\Pr[\widetilde{d} > (1+\epsilon)d] \;\le\; \Pr[Y \ge t] \;\le\; \Pr\Big[|Y - \mathbb{E}[Y]\,| > \frac{\epsilon t}{2}\Big] \le \frac{\mathrm{var}[Y]}{(\frac{\epsilon t}{2})^2} \le \frac{4}{\epsilon^2 t} \le \frac{1}{10}\,.$$

2. (6 points) In this question, we will study the uniqueness of PageRank and how to compute it via iterative methods. Consider an unweighted directed graph $G = (V, E)$ with $|V| = n$. We define the transition matrix $M \in \mathbb{R}^{n \times n}$ of G as

$$M_{i,j} = \begin{cases} \frac{1}{d(j)} & \text{if there is an edge from } j \text{ to } i, \\ 0 & \text{otherwise,} \end{cases}$$

where $d(j)$ is the outgoing degree of node j.

Let $0 < \alpha < 1$ be the teleport probability. Let $\mathbf{1} \in \mathbb{R}^n$ be the all ones vector.

(1) Prove that $(I - (1-\alpha)M)$ is a strictly column diagonally dominant matrix.
(A is strictly column diagonally dominant iff $|A_{j,j}| > \sum_{i \ne j} |A_{i,j}|$ for all j.)

(2) Prove that there is a unique vector $r^\star \in \mathbb{R}^n$ such that $r^\star = \alpha\frac{\mathbf{1}}{n} + (1-\alpha)Mr^\star$.
(You can use the following facts without proving them: all strictly column diagonally dominant matrices are invertible; the inverse of an invertible real matrix is a real matrix.)

(3) For a vector x, we define the ℓ_1-norm of x as $\|x\|_1 = \sum_i |x_i|$.
For a matrix A, we define the ℓ_1-norm of A as $\|A\|_1 = \max_{x \ne 0} \frac{\|Ax\|_1}{\|x\|_1}$.
Prove that $\|A\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |A_{i,j}|$.

(4) The PageRank vector $r^\star \in \mathbb{R}^n$ can be approximated as follows:
   - Start with any nonnegative vector $r_0 \in \mathbb{R}^n$ with $\|r_0\|_1 = 1$.
   - For $i = 1, \ldots, t$, iteratively compute $r_i = \alpha\frac{\mathbf{1}}{n} + (1-\alpha)Mr_{i-1}$.

   Prove that for some $t = O\left(\frac{\log(1/\epsilon)}{\alpha}\right)$, after t iterations we have $\|r_t - r^\star\|_1 \le \epsilon$.
   (Hint: You may find the inequality $\|Ax\|_1 \le \|A\|_1\|x\|_1$ useful.)

3. (2 bonus points) These extra-credit questions are related to Q1 and Q2.

   (1) Prove that any deterministic algorithm for computing a (1 ± 0.1)-approximation of the number of distinct elements in an n-element data stream must use $\Omega(n)$ bits of space.

   (Hint: Suppose Alice has the first $\frac{n}{2}$ numbers and Bob has the last $\frac{n}{2}$ numbers. The following claim might be useful.)

   Claim 1. *For all $n \geq 1$, there exists a set of bit strings $S \subseteq \{0,1\}^n$ such that, for some $1 \leq b \leq n$:*

   - *Every bit string in S has exactly b ones.*
   - *Any two strings in S have at most $\frac{b}{10}$ overlapping ones. Formally, for any $a_1, a_2 \in S$ where $a_1 \neq a_2$, we have $|\{1 \leq i \leq n : a_1(i) = a_2(i) = 1\}| \leq \frac{b}{10}$.*
   - *$|S| \geq 2^{cn}$ for some universal constant $c > 0$.*

   (2) Consider a linear system $Ax = b$ with $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$. Suppose A is a (symmetric) positive definite matrix. Let $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ denote the eigenvalues of A.

   Let $x^\star$ be the solution to $Ax = b$. Note that for any $\alpha > 0$, we have $\alpha A x^\star = \alpha b$, which is equivalent to $x^\star = (I - \alpha A)x^\star + \alpha b$.

   Consider the following iterative method for solving $Ax = b$.

   - Start with $x_0 = 0$.
   - For $i = 1, \ldots, t$, iteratively compute $x_i = (I - \alpha A)x_{i-1} + \alpha b$.

   Prove that one can choose the value of α such that after $t = O(\kappa \log(1/\epsilon))$ iterations, we have $\|x_t - x^\star\|_2 \leq \epsilon \|x^\star\|_2$, where $\kappa = \frac{\lambda_n}{\lambda_1}$ is the condition number of A.