CS369G: Algorithmic Techniques for Big Data Spring 2015-2016

Problem Set 1

Prof. Moses Charikar *Due: March 19, 2016,* ***1:29pm***

Policy: You are permitted to discuss and collaborate on the homework but you must write up your solutions on your own. Furthermore, you need to cite your collaborators and/or any sources that you consulted. All homework submissions are subject to the Stanford Honor Code.

Submission: homework is due before class on Tuesday March 19, 2016. Email your solution to `cs369g.stanford@gmail.com` using the subject, `[PS1_2016]SUNetID` (e.g `[PS1_2016]jdoe`). Late submissions will not be accepted. We have included some graded problems as well as ungraded ones. Submit only the **graded portion of the homework**. We will provide solutions to the ungraded portion but we encourage you to try to solve them on your own.

Length of submissions: include as much of the calculations that show that you understand everything that is going through the answer. As a rule of thumb after you have solved the problem, try to identify what are the main steps taken and critical points of a proof and include them. Unnecessary long answers to questions will be penalized. The points next to each question are indicative of the hardness/length of the proof.

Order Statistics as Estimators [ungraded]

In class we often use the idealized assumption that our hash function $h : [m] \to [0, 1]$ assigns to each element $x \in [m]$ independently a value $h(x) \sim U[0, 1]$ uniformly distributed in $[0, 1]$. In this question, you explore the properties of such idealized hash functions and how to use them to obtain estimators.

(a) Given $U_1, \ldots, U_k$ random variables independently and uniformly distributed in $[0, 1]$, let $U_{(t)}$ denote the *t-th order statistic*, i.e., the t-th smallest element among the n random variables. Find the cumulative density function (CDF) of $U_{(t)}$.

(b) Compute the expectation $\mu_t = \mathbb{E}[U_{(t)}]$, variance $v_t = \mathrm{Var}[U_{(t)}]$.

(c) Let $r_k(t) := v_t/\mu_t^2$, given a constant $C > 1$ find the value $t_k(C)$ such that $r_k(t) = 1/C$. Show that $t_k(C) \leq C$ for all $k \geq 1$ and that $\Pr\left(|U_{(t_k(C))} - \mu_{t_k(C)}| \geq \epsilon\mu_{t_k(C)}\right) \leq \dfrac{1}{\epsilon^2 C}$.

(d) For each of the values $k \in \{30, 100, 1000, 10000\}$ plot the ratio $r_k(t)$ as a function of $t \in [30]$. Give the values of $r_k(20)$ for the same range of k. What do you observe?

(e) Consider that we have t independent hash functions $h^{(1)}, \ldots, h^{(t)}$ and we output the mean of the minimum values of our hash functions over the stream. Let $\bar{U}_{t,(1)}$ be the resulting estimator. Show that $\Pr\left(|\bar{U}_{t,(1)} - \mu_1| \geq \epsilon\mu_1\right) \leq \dfrac{1}{\epsilon^2} \cdot \dfrac{r_k(1)}{t}$.

(f) Let p be a large prime number such that $p \approx m^{(1+\gamma)}$ for some $\gamma > 0$. We assume that we need $2 \log p$ bits to store each hash function and $\log m$ bits to store each element of $x \in [m]$.

Let $\bar{b}_t$ be the number of bits that we need to implement the *mean estimator* $\bar{U}_{t,(1)}$ and $b_{(t)}$ the number of bits needed to implement the estimator that outputs the t-th smallest value of a single hash function over our stream. Compute the ratio $B(t,\gamma) = \bar{b}_t / b_{(t)}$ as a function of γ and t and find the limit as $t \to \infty$.

(g) Based on the previous results, which one of the two estimators would you use and why?

Comments: in order to calculate the number of bits assume that in order to know the minimum of the hash function you need to store both the hash function as well as the current element for which the minimum is attained.

1 k-wise independence and hash functions [20 points]

In many applications, we often require the construction of an independent set of random variables. However, in many of those cases we do not require full independence but a lot less.

Definition 1. *Let S be a finite set, and let $X_1, \ldots, X_n$ be a collection of random variables assuming values in S. The collection of random variables is* k-independent *if for all* distinct *indices $i_1, \ldots, i_k \in [n]$ and elements (not necessarily distinct) $s_1, \ldots, s_k \in S$,*

$$\Pr\left(X_{i_1} = s_1, \ldots, X_{i_k} = s_k\right) = \prod_{\ell=1}^{k} \Pr(X_{i_\ell} = s_\ell)$$

(a) **[ungraded]** Given a collection $\{X_i\}_{i\in[n]}$ of k-wise independent random variable, show that given functions $f_i : S \to \mathbb{R}$, the collection of random variables $\{f_i(X_i)\}_{i\in[n]}$ is k-wise independent.

(b) **[ungraded]** Let $\mathcal{D}$ be a product distribution on S^n (each coordinate is an independent random variable) and $\mathcal{D}'$ is a k-wise independent distribution with the same marginals as $\mathcal{D}$. Show that for any polynomial function $f : S^n \to \mathbb{R}$ of degree at most k, $\mathbb{E}_{\mathcal{D}}[f(X_1, \ldots, X_n)] = \mathbb{E}_{\mathcal{D}'}[f(X_1, \ldots, X_n)]$

In essence, k-wise independence allows us to analyze polynomial functions as if the random variables where independent. This is typically exploited to calculate moments and then apply Markov's inequality. In algorithmic applications we further require that those random variables are efficiently represented, i.e., using a small number of bits. A typical application of this idea is *hashing*. Let p be a large prime number and consider $\mathbb{F}_p = \{0, 1, \ldots, p-1\}$ the field of natural numbers mod p. For simplicity we will assume that we care only about hash functions $h : \mathbb{F}_p \to \mathbb{F}_p$.

Definition 2. *A family $\mathcal{H}$ of hash functions is called k-wise independent if when picking a uniform random element $h \in \mathcal{H}$, then for any k distinct elements $x_1, \ldots, x_k$ of $\mathbb{F}_p$ and any elements (not necessarily distinct) $s_1, \ldots, s_k \in \mathbb{F}_p$,* $\Pr\left(h(x_1) = s_1, \ldots, h(x_k) = s_k\right) = \dfrac{1}{p^k}$.

In class we saw one construction of such a family of hash functions. For any $k \geq 1$ define the family of hash functions $\mathcal{H}_k = \left\{ h_{\vec{a}} \,\middle|\, h_{\vec{a}}(x) = \sum_{i=0}^{k-1} a_i x^i, \forall \vec{a} \in \mathbb{F}_p^k \right\}$.

(c) **[20 points]** Show that $\mathcal{H}_k$ is k-wise independent.

Comments: Here the randomness is only in selecting a random element of $\mathcal{H}$.

2 Estimating F_3 using Complex Numbers [40 points]

In class we saw a particular way of producing estimates of frequency moments $F_k = \sum_{i=1}^{n} f_i^k$ and we briefly explored whether different estimators are possible. In this problem, you will see how one can use the field of complex numbers to achieve this. Let $\mathcal{R}_k = \{x \in \mathbb{C} | x^k = 1\}$ be the set of k-roots of unity. For simplicity we will focus on the case of $k = 3$.The proposed basic estimator works as follows:

1. For each $i \in [m]$ we pick independently a uniform random element $x_i \in \mathcal{R}_3$.
2. We form the random variable $Z = \sum_{i=1}^{n} f_i x_i$, by adding x_i to Z each time we come across element $i \in [m]$.
3. We estimate F_3 as Re Z^3.

One can think of the mapping $i \mapsto x_i$ as hash function, that instead of mapping to the 2-roots of unity $\{-1, +1\}$ (in the original AMS scheme) maps to the 3-roots. You will analyze properties of this estimator:

(a) **[10 points]** Show that for any element $i \in [m]$, $\mathbb{E}[x_i^j] = \mathbb{E}[\bar{x}_i{}^j] = \begin{cases} 0 & \text{if } 1 \le j < 3 \\ 1 & \text{if } j = 3 \end{cases}$.

(b) **[10 points]** Show that $\mathbb{E}[\text{Re } Z^3] = F_3$. *Hint: compute first* $\mathbb{E}[Z^3]$.

(c) **[20 points]** Show that $\text{Var}[\text{Re } Z^3] = O(F_2^3)$. *Hint: use the multinomial expansion*

3 Sketching for Faster Updates [40 points]

Recall that the AMS sketch from class for F_2 moment estimation can be thought of as picking a random $m \times n$ matrix Π with entries $\pm 1/\sqrt{m}$ for $m = O(1/\epsilon^{-2})$, and estimating $\|f\|_2^2$ as $\|\Pi f\|_2^2$. One can show that with at least $2/3$ probability,

$$(1 - \epsilon)\|f\|_2^2 \le \|\Pi f\|_2^2 \le (1 + \epsilon)\|f\|_2^2 \tag{1}$$

In this problem you will explore a different way of estimating F_2. Imagine picking $\Pi \in \{\pm 1, 0\}^{m \times n}$ differently, for each $i \in \{1, \ldots, n\}$ we pick a row $h_i \in [m]$ uniformly at random and set $\Pi_{h_i, i} = \pm 1$ (the sign is chosen uniformly at random from $\{-1, 1\}$), and all other entries of the i-th column are set to 0. This Π has the advantage that in turnstile streams (where each update vector Δx can contain a non-zero integer only at one location) we can process updates in constant time. Show that using this Π still satisfies the conditions of equation (1) with $2/3$ probability for $m = O(\epsilon^{-2})$.