School of Computer Science and Technology
University of Science and Technology of China

Marked Exercises for
Algorithms for Big Data
2022 Spring

Due 27 March 2022 at 23:59

Exercise 1 *10 points*
Let $\sum_{i=1}^{r} \sigma_i u_i v_i^T$ be the SVD of A, where $A \in \mathbb{R}^{n\times d}$. Show that $|u_1^T A| = \sigma_1$ and $|u_1^T A| = \max_{\|u\|=1} \|u^T A\|$, where $\|x\| = \sqrt{\sum_{i=1}^{d} x_i^2}$ for a vector $x \in \mathbb{R}^d$.

Exercise 2 *20 points*
Let $\sum_{i=1}^{r} \sigma_i u_i v_i^T$ be the SVD of a rank r matrix A. Let $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$ be a rank k-approximation to A for some $k < r$. Express the following quantities in terms of the singular values $\{\sigma_i, 1 \leq i \leq r\}$.

(a) $\|A_k\|_F^2$

(b) $\|A_k\|_2^2$

(c) $\|A - A_k\|_F^2$

(d) $\|A - A_k\|_2^2$

Exercise 3 *15 points*
Let $k < d$. Let $U \in \mathbb{R}^{d\times k}$ be a random matrix such that its (i,j)-th entry is denoted as u_{ij}, where $\{u_{ij}\}$ are independent random variables such that

$$u_{ij} = \begin{cases} 1 & \text{with probability } \frac{1}{2}, \\ -1 & \text{with probability } \frac{1}{2} \end{cases}$$

Now we use matrix U as a random projection matrix. That is, for a (row) vector $a \in \mathbb{R}^d$, we map it to

$$f(a) = \frac{1}{\sqrt{k}} aU$$

For each j such that $1 \leq j \leq k$, define $b_j = [f(a)]_j$, i.e., b_j is the j-th entry of $f(a)$.

- What is the expectation $\mathrm{E}[b_j]$?
- What is $\mathrm{E}[b_j^2]$?
- What is $\mathrm{E}[\|f(a)\|^2]$?

Exercise 4 *15 points*
In the class, we have seen an algorithm, denoted by $\mathcal{A}$, for the (c,r)-ANN problem with success probability at least 0.6. That is, upon a queried vertex x such that there exists a point a^* in the set $\mathcal{P}$ with $d(x, a^*) \leq r$, the algorithm $\mathcal{A}$ outputs some $a \in \mathcal{P}$ with $d(x,a) \leq c \cdot r$ with probability at least 0.6.
Let $\delta \in (0,1)$. Using the above $\mathcal{A}$ as a subroutine, give a new algorithm $\mathcal{B}$ with success probability at least $1-\delta$. That is, for the above query vertex x, the algorithm $\mathcal{B}$ outputs some $a \in \mathcal{P}$ with $d(x,a) \leq c \cdot r$ with probability at least $1-\delta$. Your algorithm should use as little query time as possible. Explain the correctness of your algorithm and state its query time, assuming the query time of $\mathcal{A}$ is $T_{\mathcal{A}}$.

Exercise 5 *20 points*
Let $\alpha \in (0,1]$. Suppose we change the (basic) Morris algorithm to the following:

(a) Initialize $X \leftarrow 0$

(b) For each update, increment X by 1 with probability $\frac{1}{(1+\alpha)^X}$

(c) For a query, output $\tilde{n} = \frac{(1+\alpha)^X - 1}{\alpha}$.

Let X_n denote X in the above algorithm after n updates. Let $\tilde{n} = \frac{(1+\alpha)^{X_n} - 1}{\alpha}$.

- Calculate $\mathrm{E}[\tilde{n}]$ and upper bound $\mathrm{Var}[\tilde{n}]$.
- Let $\epsilon, \delta \in (0, 1)$. Based upon the above algorithm, give a new algorithm such that with probability at least $1 - \delta$, it outputs an estimator $\tilde{n}$ such that $|\tilde{n} - n| \leq \epsilon n$. Explain the correctness and the space complexity (i.e., the number of used bits) of your algorithm. It suffices to give an algorithm with space complexity that is a polynomial function of $1/\delta$.

Exercise 6 *20 points*
Consider a stream of m integers $a_1, a_2, \ldots, a_m$ such that each $a_i \in [n] = \{1, 2, \ldots, n\}$. We would like to estimate the *median* of these numbers using small space. Formally, let $S = \{a_1, a_2, \ldots, a_m\}$, and define $\mathrm{rank}(b) = |\{a \in S : a \leq b\}|$. For simplicity, suppose elements in S are distinct, and m is known to the algorithm. Given $\varepsilon, \delta \in (0, 1)$, our goal is to find a number b such that

$$\Pr[|\mathrm{rank}(b) - \frac{m}{2}| > \varepsilon m] < \delta. \tag{1}$$

Consider the following algorithm:

- Maintain t uniform samples from S (e.g., by using Reservoir sampling)
- Output the median of these t samples

Choose the smallest possible t so that inequality (1) holds. Give an explanation of the correctness of the resulting algorithm and state its space complexity.

Hint: You can partition S into 3 groups: $S_L = \{a \in S : \mathrm{rank}(a) \leq m/2 - \varepsilon m\}$, $S_M = \{a \in S : m/2 - \varepsilon m \leq \mathrm{rank}(a) \leq m/2 + \varepsilon m\}$, and $S_H = \{a \in S : \mathrm{rank}(a) \geq m/2 + \varepsilon m\}$. Note that if less than $t/2$ elements from both S_L and S_H are present in the sample, then the median of the samples is a "good" estimator.
