COMPSCI 514: Problem Set 3

Due: 11/8 by 11:59pm in Gradescope.

Instructions:

- You are allowed to work on this problem set in a group of up to three members.
- You should choose your group from within your own class (either online or in-person).
- You may talk to members of other groups at a high level about the problems but **not work through the solutions in detail together**.
- Each group should **submit a single solution set**: one member should upload a pdf to Gradescope, marking the other members as part of their group in Gradescope.
- You must show your work/derive any answers as part of the solutions to receive full credit.

Core Competency Problems

1. LSH and s-curves (8 points)

1. (2 points) When we computed the hit rate in lecture we assumed that we were combining MinHash with a fully random hash function g that was collision free. If g was not collision free how would this change the probability of false positives and false negatives? Specifically would these quantities become large or smaller? Explain your answer.
2. (2 points) Consider the s-curve $f(s) = 1 - (1 - s^r)^t$. Prove that for fixed strictly positive integers r and t, $f(s)$ is an increasing function in s on the range $[0, 1]$, and that $f(0) = 0$ and $f(1) = 1$.
3. Suppose we want the s-curve to provide as sharp a cut-off as possible. I.e., for some $s^* \in [0, 1]$, $\epsilon \in [0, s^*]$ and $\gamma \in [0, 1]$ we want

$$f(s^* - \epsilon) \leq \gamma \text{ and } f(s^*) \geq 1 - \gamma \ .$$

   (a) (2 points) Find an expression for t in terms of r, s^*, and γ such that $f(s^*) \geq 1 - \gamma$. **Hint:** You may want to use the inequality that $1 - x \leq \exp(-x)$.

   (b) (2 points) Use this expression to find an expression for r such that $f(s^* - \epsilon) \leq \gamma$. **Hint:** You may want to use the inequality that $(1 - (s^* - \epsilon)^r)^t \geq 1 - (s^* - \epsilon)^r t$.

2. k-Means Clustering (12 points)

Consider n data points $x_1, \ldots, x_n \in \mathbb{R}^d$ and $\mathbf{X} \in \mathbb{R}^{n \times d}$ with these data points as its rows. Consider any clustering $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ of these data points. I.e., each C_j is a set (a cluster) of points

and each data point x_i is assigned to one of these sets. Let $\mu_j = \frac{1}{|C_j|}\sum_{x\in C_j} x$ be the centroid (i.e., the mean) of cluster C_j. The k-means clustering objective is to minimize

$$cost(\mathcal{C}, \mathbf{X}) = \sum_{j=1}^{k}\sum_{x\in C_j} \|x-\mu_j\|_2^2 \ .$$

1. (2 points) Prove that $cost(\mathcal{C}, \mathbf{X}) = \sum_{j=1}^{k} \frac{1}{2|C_j|}\sum_{x\in C_j}\sum_{y\in C_j}\|x-y\|_2^2$.

   Hint: Prove that both expressions the cost can be written as $\sum_{j=1}^{k}\left[\left(\sum_{x\in C_j}\|x\|_2^2\right) - |C_j|\cdot\|\mu_j\|_2^2\right]$. This will require some vector algebra. It will be helpful to use that for any vector z, $\|z\|_2^2 = \sum_{i=1}^{d} z(i)^2 = \langle z, z\rangle$, as well as to use the linearity of inner product.

2. (2 points) Suppose that $\mathbf{\Pi} \in \mathbb{R}^{m\times d}$ is a random projection matrix with each entry chosen independently as $\mathcal{N}(0, 1/m)$. For each x_i in the dataset, let $\tilde{x}_i = \mathbf{\Pi}x_i$ and let $\tilde{\mathbf{X}} \in \mathbb{R}^{n\times m}$ contain the compressed data points as its rows. By part (1), for $m = O(\frac{\log n}{\epsilon^2})$, with high probability, we have, for all clusterings $\mathcal{C}$,

   $$(1-\epsilon)\cdot cost(\mathcal{C}, \mathbf{X}) \le cost(\mathcal{C}, \tilde{\mathbf{X}}) \le (1+\epsilon)\cdot cost(\mathcal{C}, \mathbf{X}).$$

   Assuming that the above bound holds, prove that if we compute the optimal clustering on the compressed data, $\tilde{\mathcal{C}} = \arg\min cost(\mathcal{C}, \tilde{\mathbf{X}})$ then it is near-optimal for the original data, i.e., that: $cost(\tilde{\mathcal{C}}, \mathbf{X}) \le \frac{1+\epsilon}{1-\epsilon}\cdot \min cost(\mathcal{C}, \mathbf{X})$.

The next few questions focus on the connection between k-means clustering and low-rank matrix approximation/PCA.

3. (2 points) Let $\mathbf{X}_{\mathcal{C}}$ be the $n\times d$ matrix whose i^{th} row is equal to μ_j if x_i is assigned to cluster C_j in $\mathcal{C}$. Verify that the k-means cost function can be written as $cost(\mathcal{C}, \mathbf{X}) = \|\mathbf{X} - \mathbf{X}_{\mathcal{C}}\|_F^2$.

4. (2 points) Use (3) to prove that for any clustering $\mathcal{C}$, $cost(\mathcal{C}, \mathbf{X}) \ge \min_{\mathbf{B}:\text{rank}(\mathbf{B})\le k}\|\mathbf{X}-\mathbf{B}\|_F^2$. **Hint:** What is the rank of $\mathbf{X}_{\mathcal{C}}$?

5. (2 points) Show that we can write $\mathbf{X}_{\mathcal{C}} = \mathbf{V}\mathbf{V}^T\mathbf{X}$ where $\mathbf{V} \in \mathbb{R}^{n\times k}$ has orthonormal columns.

6. (2 points) Explain in a few sentences what parts (3)-(5) mean. How is k-means clustering similar to PCA? How is it different?

3. Alternative Approaches to Dimensionality Reduction (12 points)

The Johnson-Lindenstrauss lemma gives that, for any set of points $\vec{x}_1, \ldots, \vec{x}_n$, letting $\mathbf{\Pi} \in \mathbb{R}^{m\times d}$ be a random projection matrix with each entry chosen independently from $\mathcal{N}(0, \frac{1}{m})$ and $m = O\left(\frac{\log(n/\delta)}{\epsilon^2}\right)$, with probability $\ge 1-\delta$, for all $i, j \in [n]$:

$$(1-\epsilon)\|\vec{x}_i - \vec{x}_j\|_2^2 \le \|\mathbf{\Pi}\vec{x}_i - \mathbf{\Pi}\vec{x}_j\|_2^2 \le (1+\epsilon)\|\vec{x}_i - \vec{x}_j\|_2^2.$$

1. (2 points) Given $\mathbf{\Pi} \in \mathbb{R}^{m\times d}$ distributed as above, what is the runtime of computing the compressed vectors $\tilde{x}_i = \mathbf{\Pi}\vec{x}_i$ for all $\vec{x}_i$? Assume a single basic arithmetic operation (addition, multiplication, etc.) on real numbers takes $O(1)$ time.

2. (2 points) One very fast alternative to using a random projection matrix $\mathbf{\Pi}$ is to just perform dimensionality reduction by sampling. Let $\mathbf{W} \in \mathbb{R}^{m \times d}$ be a matrix that samples m rows of a vector uniformly at random and re-weights them. Specifically, the j^{th} row of $\mathbf{W}$ is equal to $\sqrt{d/m} \cdot \vec{e}_i$ with probability $1/d$ where $\vec{e}_i$ is the i^{th} standard basis vector for any $i \in [d]$. Show that if we let $\tilde{x}_i = \mathbf{W}\vec{x}_i$ then $\mathbb{E}[\|\tilde{x}_i - \tilde{x}_j\|_2^2] = \|\vec{x}_i - \vec{x}_j\|_2^2$. That is, this simple dimensionality reduction method preserves the distance between vectors in expectation.

3. (2 points) When might the above sampling method perform poorly in comparison to random projection even though it preserves distances in expectation?

Now suppose that the entries of $\mathbf{\Pi}$ are chosen independently and uniformly in the set $\{-1/\sqrt{m}, 1/\sqrt{m}\}$.

4. (2 points) Prove that for any $x \in \mathbb{R}^d$, $\mathbb{E}[\|\mathbf{\Pi}x\|_2^2] = \|x\|_2^2$.

5. (2 points) Prove that for any $x \in \{-1, 1\}^d$, $\mathbb{E}[\mathbf{y}(i)^4] \leq 3d^2/m^2$ where $\mathbf{y} = \mathbf{\Pi}x$.

6. (2 points) Prove that if we set $m = \frac{c}{\epsilon^2 \delta}$ for a large enough constant c, then for any $x \in \{-1, 1\}^d$ we will have
$$(1 - \epsilon)\|x\|_2^2 \leq \|\mathbf{\Pi}x\|_2^2 \leq (1 + \epsilon)\|x\|_2^2$$
with probability at least $1 - \delta$.

Challenge Problems (Complete 1 of 2)

C1. Sparse Random Projection (10 points) 🌶🌶

Let $\mathbf{C} \in \mathbb{R}^{m \times d}$ be a sparse random projection matrix distributed as follows: each column of $\mathbf{C}$ is independently chosen to have a single non-zero entry in a uniform random position, set to ± 1 each with probability $1/2$.

1. (2 points) How long (in big-O notation) does it take to compute $\tilde{\mathbf{x}} = \mathbf{C}x$ for a given vector $x \in \mathbb{R}^d$. How does this compare to the runtime of computing $\mathbf{\Pi}x$ where $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$ is a dense Gaussian random projection matrix? **Note:** Assume that your algorithm has knowledge of where the non-zero entries of $\mathbf{C}$ are.

2. (2 points) Prove that $\mathbb{E}[\|\mathbf{C}x\|_2^2] = \|x\|_2^2$. **Hint:** Use linearity of expectation and variance.

3. (2 points) Prove that $\text{Var}[\|\mathbf{C}x\|_2^2] \leq \frac{c\|x\|_2^4}{m}$ for some constant c. **Note:** This subquestion is a bit difficult. You might want to complete parts (4) and (5) first and then return to it. Also note that the entries of $\mathbf{C}x$ are not independent. So you cannot directly apply linearity of variance.

4. (2 points) Prove that if we set $m = \frac{c}{\epsilon^2 \delta}$ for a large enough constant c, then we will have $(1 - \epsilon)\|x\|_2^2 \leq \|\mathbf{C}x\|_2^2 \leq (1 + \epsilon)\|x\|_2^2$ with probability at least $1 - \delta$.

5. (2 points) How large must we set m (in big-O notation) to ensure that $\mathbf{C}$ gives an ϵ-distortion embedding for n data points with probability at least $1 - \delta$.

C2. Structured Random Projection (10 points) 🌶🌶🌶

We will now see how to make the sampling approach to dimensionality reduction from Problem 3 work by using a preprocessing step. Let $\mathbf{F} \in \mathbb{R}^{d \times d}$ be a "*Hadamard matrix*". You only need to know three properties of $\mathbf{F}$ to solve this problem:

(i) All entries of $\mathbf{F}$ are either $-1/\sqrt{d}$ or $1/\sqrt{d}$.

(ii) The columns of $\mathbf{F}$ are orthonormal. I.e., $\mathbf{F}^T\mathbf{F} = \mathbf{I}$ where $\mathbf{I}$ is the $d \times d$ identity matrix.

(iii) For any vector $\vec{x}$, $\mathbf{F}\vec{x}$ can be computed in $O(d \log d)$ time using the *fast Hadamard transform*, which is a variant of the fast Fourier transform.

Using these properties, solve the following problems:

1. (2 points) Let $\mathbf{S} \in \mathbb{R}^{d \times d}$ be a diagonal matrix with each entry $\mathbf{S}_{i,i}$ set independently to 1 with probability 1/2 and -1 with probability 1/2. Let $z = \mathbf{FS}x_i - \mathbf{FS}x_j$. Show that $\|z\|_2^2 = \|x_i - x_j\|_2^2$. **Hint:** Use that for any vector y, $y^T y = \langle y, y \rangle = \|y\|_2^2$.

2. (2 points) Show that, with probability $\geq 1 - \delta$, $\max_{k \in [d]} |z(k)| \leq \frac{c \log(d/\delta) \|x_i - x_j\|_2}{\sqrt{d}}$ for some constant c. **Hint:** Use Bernstein's inequality to bound each $|z(k)|$ and then a union bound to bound the maximum. In applying Bernstein's inequality, you might want to use that the maximum magnitude of an entry in a vector is bounded by its Euclidean norm.

Together, parts (1) and (2) show that the linear transformation $\mathbf{FS}$ exactly preserves the norm of $x_i - x_j$, while also spreading out its entries to be nearly uniform in size: so that the maximum is bounded by $O\left(\frac{\log(d/\delta)}{\sqrt{d}}\right)$ times the norm. By spreading out the entries, we can ensure that random sampling approximately preserves the norm of this vector. We analyze this fact formally below.

3. (2 points) Let $\mathbf{W} \in \mathbb{R}^{m \times d}$ be a sampling matrix as in Problem 3. Let $\tilde{x}_i = \mathbf{WFS}x_i$. Show that $\mathbb{E}[\|\tilde{x}_i - \tilde{x}_j\|_2^2] = \|x_i - x_j\|_2^2$.

4. (2 points) Show that for $m = O\left(\frac{\log(d/\delta)^4 \cdot \log(n/\delta)}{\epsilon^2}\right)$, with probability $\geq 1 - \delta$, for all $i, j \in [n]$,

$$(1 - \epsilon)\|x_i - x_j\|_2^2 \leq \|\tilde{x}_i - \tilde{x}_j\|_2^2 \leq (1 + \epsilon)\|x_i - x_j\|_2^2.$$

   Hint: Use Bernstein's inequality again, with the entry upper bound from part (2).

5. (2 points) How long does it take to compute the compressed vector $\mathbf{WFS}x_i$ for any x_i, assuming that $m < d$? How does this compare to the runtime of computing $\mathbf{\Pi}x_i$ where $\mathbf{\Pi} \in \mathbb{R}^{m \times d}$ is a dense Gaussian random projection matrix?