PRINCETON UNIVERSITY FALL '19 COS 521:ADVANCED ALGORITHMS

Homework 3

Out: *Oct 14* Due: *Oct 28*

Instructions:

- Upload your solutions (to the non-extra-credit) as *a single* PDF file (one PDF total) to Mechanical TA. Please anonymize your submission (do not list your name in the PDF title or in the document itself). If you forget, it's OK.
- If you choose to do extra credit, upload your solution to the extra credits as a single PDF file to Mechanical TA. Please again anonymize your submission.
- You may collaborate with any classmates, textbooks, the Internet, etc. Please attach a brief "collaboration statement" listing any collaborators at the end of your PDF (if you forget, it's OK). You should write up your solutions individually.
- For each problem, you should aim to keep your writeup below one page. For some problems, this may be infeasible, and for some problems you may write significantly less than a page. This is not a hard constraint, but part of the assignment is figuring out how to easily convince the grader of correctness, and to do so concisely. "One page" is just a guideline: if your solution is longer because you chose to use figures (or large margins, display math, etc.) that's fine.
- Each problem is worth twenty points (even those with multiple subparts).

§1 The ℓ_1 distance between vectors $x, y \in \mathbb{R}^d$ is defined as $\|x - y\|_1 = \sum_{i=1}^{d} |x_i - y_i|$. Consider the Johnson-Lindenstrauss dimensionality reduction method described in lecture: $x \to \Pi x$ where each entry in $\Pi \in \mathbb{R}^{m \times d}$ equals

$$\Pi_{ij} = c \cdot g_{ij},$$

for some fixed scaling factor c and $g_{ij} \sim \mathcal{N}(0, 1)$. Describe an example (i.e., a set of points in $\mathbb{R}^d$) which shows that, for any choice of c, this method does *not* preserve ℓ_1 distances, even within a factor of 2. You may pick a single d for your example.

Hint: A simple example exists with just three vectors. You may want to use the fact that this choice of Π preserves ℓ_2 distances. You may also want to use some of the JL analysis given in lecture as a black box.

§2 A k-sparse vector is any vector with at most k nonzero entries. Let $\mathcal{S}_k$ be the set of all k-sparse vectors in $\mathbb{R}^d$. Show that, if Π is chosen to be a Johnson-Lindenstrauss embedding matrix (e.g. a scaled random Gaussian matrix) with $s = O(\frac{k \log d}{\epsilon^2})$ rows then, with high probability,

$$(1 - \epsilon)\|\Pi x\|_2 \leq \|x\|_2 \leq (1 + \epsilon)\|\Pi x\|_2$$

for all $x \in \mathcal{S}_k$, simultaneously.

Hint: You will want to use some result from the JL lecture as a black-box.

§3 A *matroid* on $[n]$ elements is a collection of sets that generalized the concept of linear independence for vectors. Specifically, a matroid $\mathcal{I}$ satisfies:

- **Non-trivial:** $\emptyset \in \mathcal{I}$.
- **Downwards-closed:** If $S \in \mathcal{I}$, then $T \in \mathcal{I}$ for all $T \subseteq S$.
- **Augmentation:** If $S, T \in \mathcal{I}$, and $|S| > |T|$, then there exists an $i \in S \setminus T$ such that $T \cup \{i\} \in \mathcal{I}$.[1]

Prove that the following collections are matroids:

(a) Sets of size at most k (that is, the elements are $[n]$, and $\mathcal{I} = \{X \subseteq [n] | \, |X| \leq k\}$).

(b) Acyclic subgraphs of any undirected graph $G = (V, E)$ (that is, the elements are E and $\mathcal{I} = \{X \subseteq E | X \text{ contains no cycles}\}$).

(c) Let $G = (L, R, E)$ be a bipartite graph. The elements are L, and $\mathcal{I} = \{X \subseteq L | \, |N(S)| \geq |S| \; \forall S \subseteq X\}$ ($N(S)$ are the neighbors of S: $\{x \in R | \, \exists y \in S, (x, y) \in E\}$). That is, $X \in \mathcal{I}$ if and only if all nodes in X can be simultaneously matched to R.

§4 Given weights $w_i \geq 0, i \in [n]$, and some collection of feasible sets $\mathcal{I}$, your goal is to find the max-weight feasible set: $\arg\max_{S \in \mathcal{I}} \{\sum_{i \in S} w_i\}$. Consider a greedy algorithm that first sorts the elements in decreasing order of w_i (i.e. picks a permutation σ such that $w_{\sigma(i)} \geq w_{\sigma(i+1)}$ for all i), then iteratively does the following (initializing $A = \emptyset$, $i = 1$, go until $i > n$): Check if $A \cup \{\sigma(i)\} \in \mathcal{I}$. If so, add $\sigma(i)$ to A. Update $i := i + 1$. Prove that this greedy algorithm finds the max-weight feasible set no matter what non-negative weights are input *if and only if* $\mathcal{I}$ is a matroid (that is, prove that the algorithm succeeds whenever $\mathcal{I}$ is a matroid. Also, if $\mathcal{I}$ is not a matroid, provide an instance of weights for which the algorithm fails).

§5 Given a data matrix $X \in \mathbb{R}^{n \times d}$ with n rows (data points) $x_1, \ldots, x_n \in \mathbb{R}^d$, the *k-means clustering problem* asks us to find a partition of our points into k disjoint sets (clusters) $\mathcal{C}_1, \ldots \mathcal{C}_k \subseteq \{1, \ldots, n\}$ with $\bigcup_{j=1}^{k} \mathcal{C}_j = \{1, \ldots, n\}$.

Let $c_j = \frac{1}{|\mathcal{C}_j|} \sum_{i \in \mathcal{C}_j} x_i$ be the centroid of cluster j. We want to choose our clusters to minimize the sum of squared distances from every point to its cluster centroid. I.e. we want to choose $\mathcal{C}_1, \ldots \mathcal{C}_k$ to minimize:

$$f_X(\mathcal{C}_1, \ldots \mathcal{C}_k) = \sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j} \|c_j - x_i\|_2^2.$$

There are a number of algorithms for solving the k-means clustering problem. They typically run more slowly for higher dimensional data points, i.e. when d is larger. In this problem we consider what sort of approximation we can achieve if we instead solve the problem using dimensionality reduced vectors in place of $x_1, \ldots, x_n$.

[1] Think of this as a generalization of linear independence: if I give you a set S of k linearly independent vectors, and T of $< k$ linearly independent vectors, then there is some vector in S not spanned by T.

Let $OPT_X = \min_{\mathcal{C}_1,\ldots\mathcal{C}_k} f_X(\mathcal{C}_1,\ldots\mathcal{C}_k)$.

Suppose that Π is a Johnson-Lindenstrauss map into $s = O(\log n/\epsilon^2)$ dimensions and that we select the optimal set of clusters for $\Pi x_1,\ldots,\Pi x_n$. Call these clusters them $\tilde{\mathcal{C}}_1,\ldots\tilde{\mathcal{C}}_k$. Show that they obtain objective value $f_X(\tilde{\mathcal{C}}_1,\ldots\tilde{\mathcal{C}}_k) \leq (1+\epsilon)OPT_X$, with high probability.

(Hint: reformulate the objective function to only involve ℓ_2 distances between data points.)

Extra Credit:

§1 (Extra credit, follows "Given a data matrix...") Instead, suppose we reduce our points to k dimensions using the SVD. I.e. let $V_k \in \mathbb{R}^{d\times k}$ have the first k right singular vectors of X. Show that, if $\tilde{\mathcal{C}}_1,\ldots\tilde{\mathcal{C}}_k$ are the optimal clusters for $V_k^T x_1,\ldots,V_k^T x_n$, then $f_X(\tilde{\mathcal{C}}_1,\ldots\tilde{\mathcal{C}}_k) \leq 2OPT_X$.

(Hint: show that for every set of clusters, there is an orthonormal matrix $C \in \mathbb{R}^{n\times k}$ such that $f_X(\mathcal{C}_1,\ldots\mathcal{C}_k) = \|X - CC^T X\|_F^2$. I.e. reformulate k-means as a k-rank approximation problem.)

§2 (Extra credit) Calculate the eigenvectors and eigenvalues of the (adjacency matrix of the) n-dimensional boolean hypercube, which is the graph with vertex set $\{-1,1\}^n$ and x, y are connected by an edge iff they differ in exactly one of the n locations. (Hint: Use symmetry extensively.)