HW 7

Due: October 30th, 2025

Problem 1: Johnson-Lindenstrauss for k-means clustering

In this problem we analyze a common algorithmic application of the Johnson-Lindenstrauss lemma.

Consider the k-means clustering problem. Given points $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, the goal is to partition the points into k disjoint sets (clusters) $C_1, \ldots, C_k$ so as to minimize the cost:

$$\text{Cost}(\mathbf{x}_1, \ldots, \mathbf{x}_n, C_1, \ldots, C_k) = \sum_{i=1}^{k} \sum_{j \in C_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|_2^2,$$

where $\boldsymbol{\mu}_i = \frac{1}{|C_i|} \sum_{j \in C_i} \mathbf{x}_j$ is the mean of the points in cluster C_i. In other words, we want to find k clusters that have as little internal variance as possible.

Part A. An Alternative Formulation of the k-means Cost

Prove that:

$$\text{Cost}(\mathbf{x}_1, \ldots, \mathbf{x}_n, C_1, \ldots, C_k) = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\substack{j,\ell \in C_i \\ j<\ell}} \|\mathbf{x}_j - \mathbf{x}_\ell\|_2^2.$$

Part B. Preservation of the Cost Function under Dimensionality Reduction

Suppose we embed $\mathbf{x}_1, \ldots, \mathbf{x}_n$ into $O(\log n/\varepsilon^2)$ dimensional vectors $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$ using the Johnson-Lindenstrauss construction from class. Recall that, with high probability, the Johnson-Lindenstrauss lemma ensures that:

$$(1-\varepsilon)\|\mathbf{x}_j - \mathbf{x}_k\|_2^2 \leq \|\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_k\|_2^2 \leq (1+\varepsilon)\|\mathbf{x}_j - \mathbf{x}_k\|_2^2,$$

for all j, k. Leverage this fact and part (a) to show that, for *all clusterings simultaneously*:

$$(1-\varepsilon)\text{Cost}(\mathbf{x}_1, \ldots, \mathbf{x}_n, C_1, \ldots, C_k) \leq \text{Cost}(\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n, C_1, \ldots, C_k) \leq (1+\varepsilon)\text{Cost}(\mathbf{x}_1, \ldots, \mathbf{x}_n, C_1, \ldots, C_k)$$

with high probability.

Part C. Guarantee for the Approximation Algorithm

Suppose we find a set of clusters $\tilde{C}_1, \ldots, \tilde{C}_k$ such that:

$$\text{Cost}(\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n, \tilde{C}_1, \ldots, \tilde{C}_k) \leq \gamma \cdot \text{Cost}(\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n, \tilde{C}_1^*, \ldots, \tilde{C}_k^*),$$

where $\tilde{C}_1^*, \ldots, \tilde{C}_k^*$ is the optimal partition for $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$—i.e., we find a γ-approximation to the optimal clustering for the dimensionally reduced points. Show that, for $\varepsilon \leq \frac{1}{2}$,

$$\text{Cost}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \tilde{C}_1, \ldots, \tilde{C}_k) \leq (1 + O(\varepsilon))\gamma \cdot \text{Cost}(\mathbf{x}_1, \ldots, \mathbf{x}_n, C_1^*, \ldots, C_k^*),$$

where $C_1^*, \ldots, C_k^*$ is the optimal partition for $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

In other words, we can compute an approximate clustering for our (possibly high dimensional) original points using just the dimensionality reduced data. This approach will succeed in solving k-means clustering using $\log \frac{n}{\varepsilon^2} < d$ dimensions.

Problem 2: Minimum Distortion ℓ_2 Embedding via SDP

The goal of this problem is to demonstrate how semidefinite programming (SDP) can be used to find a "minimum-distortion embedding of an arbitrary finite metric space into Euclidean space (ℓ_2)." (We explain what this means below.)

Let V be an arbitrary set of n points with pairwise distances $d(u, v) \geq 0$ that are symmetric ($d(u, v) = d(v, u)$), obey the *triangle inequality* (for u, v, and w, $d(u, v) \leq d(u, w) + d(v, w)$), and satisfy $d(u, v) = 0$ iff $u = v$. An *embedding* is a map $f : V \to \mathbb{R}^k$ for some k. We are interested in embeddings where the distance between points $u, v \in V$ is given by the Euclidean norm $\|f(u) - f(v)\|_2$ (we refer to this as an "embedding into ℓ_2").

The **distortion** of the embedding f is the smallest number $C \geq 1$ for which there exists a scaling factor $\alpha > 0$ such that for all distinct $u, v \in V$:

$$\alpha \cdot d(u, v) \leq \|f(u) - f(v)\|_2 \leq \alpha \cdot C \cdot d(u, v)$$

By scaling the entire embedding, we can set the minimum squared distance $\|f(u) - f(v)\|_2^2$ over all pairs to be at least $d(u, v)^2$. This allows us to simplify the condition to finding an embedding that minimizes C subject to:

$$d(u, v)^2 \leq \|f(u) - f(v)\|_2^2 \leq C \cdot d(u, v)^2 \quad \forall u, v \in V$$

You will formulate an algorithm to find a set of vectors $\{f(v) \in \mathbb{R}^n : v \in V\}$ attaining this minimum C.

Part A. Embedding and the Gram Matrix

The vectors $\{f(u)\}_{u \in V}$ in the target space $\mathbb{R}^k$ can be described by their **Gram matrix** G, which is an $n \times n$ matrix defined by the inner products $G_{uv} = f(u) \cdot f(v)$.

1. Show that the squared Euclidean distance $\|f(u) - f(v)\|_2^2$ can be expressed as a linear function of the entries of the Gram matrix G.

2. What is the necessary and sufficient condition on the matrix G for it to be a valid Gram matrix of some set of vectors? (You do not need to prove this, just state the condition).

Part B. The Feasibility Problem as an SDP

Instead of finding the vectors attaining the minimum distortion C directly, consider the following **decision problem**: Given a fixed value $C' \geq 1$, can we determine if there exists an ℓ_2 embedding with distortion at most C'?

Formulate this decision problem as a **semidefinite program feasibility problem**. Clearly define the variables of your program and state the constraints. Your constraints should be in terms of the variable matrix G and the given distortion C'.

Part C. Algorithm for Finding Minimum Distortion Embedding

Using your SDP formulation from Part (b) as a subroutine, describe a complete algorithm to find a set of vectors $\{f(v) \in \mathbb{R}^n : v \in V\}$ attaining the minimum possible distortion, C_{opt}, for embedding the metric space (V, d) into ℓ_2. Your algorithm should be able to find vectors attaining C_{opt} to within an arbitrary precision $\epsilon > 0$. Justify why your algorithm is correct.