Unsupervised Learning HW3

Due: Mon Nov 19, 2018 at 11:59pm

All homeworks (including this one) should be typesetted properly in pdf format. Late homeworks or handwritten solutions will not be accepted. You must include your name and UNI in your homework submission. To receive credit, a typesetted copy of the homework pdf must be uploaded to Gradescope by the due date. You must show your work to receive full credit. Discussing possible solutions for homework questions is encouraged on piazza and with your peers, but everyone must write their own individual solutions. You must cite all external references you used (including the names of individuals you discussed the solutions with) to complete the homework.

1 **[readings]** Read any two papers of your choice from the following list, summarize their main results, discuss their significance and provide a short proof sketch of their technical results.

- "Nearest neighbor preserving embeddings" by Indyk and Naor.
- "Optimality of the Johnson-Lindenstrauss lemma" by Larseni and Nelson.
- "Distance Preserving Embeddings for General n-Dimensional Manifolds" by Verma.
- "Some theory for ordinal embedding" by Arias-Castro.
- "Learning Sparsely Used Overcomplete Dictionaries via Alternating Minimization" by Agarwal, Anandkumar, Jain, and Netrapalli.
- "Poincare Embeddings for Learning Hierarchical Representations" by Nickel and Kiela.

2 **[JL for Infinite Size Sets]** We saw in class that a (scaled) linear projection onto a random subspace can approximately preserve interpoint distances between a set of finite number of points (cf. Johnson-Lindenstrauss JL-Lemma). One can prove JL-Lemma by applying the following key technical result.

Lemma: Pick any $0 < \epsilon < 1/2$, fix any unit vector $w \in \mathbb{R}^D$, and let $\phi : \mathbb{R}^D \to \mathbb{R}^d$ (where $d < D$), be a random subspace map. Then,

$$\mathbf{Pr}_{\phi}\Big[\|\Phi(w)\|^2 < (1-\epsilon)\frac{d}{D} \text{ or } \|\Phi(w)\|^2 > (1+\epsilon)\frac{d}{D}\Big] \leq 3e^{-d\epsilon^2/4}.$$

Recall that JL-Lemma only guarantees approximate preservation of distances among *finite* pair of points. In some applications one is interested in preserving interpoint distances amongst *infinite* pairs of points. Unfortunately one cannot get a JL-type result (with significant compression) for an arbitrary infinite-size set of points. But if the infinite-size set has some sort of *structure* progress can be made.

Consider a fixed but unknown k-dimensional affine space $S \subset \mathbb{R}^D$. Show that: For any $0 < \epsilon < 1/2$ there exists a linear map $f : \mathbb{R}^D \to \mathbb{R}^d$ (with $d = O(k/\epsilon^2)$), such that for all $u, v \in S$

$$(1-\epsilon)\|u-v\| \leq \|f(u)-f(v)\| \leq (1+\epsilon)\|u-v\|.$$

(Hint: Consider a finite cover of an appropriate subset of S and apply regular JL-Lemma to it. Using that, argue that it implies preservation of interpoint distances between all points in S)

3 **[Cheeger's Inequality]** In the following, let $G = (V, E)$ be a d-regular graph with n vertices (undirected and unweighted). For simplicity, let $V = [n]$. Let $S \subset V$ be a subset of vertices. Recall that:

- $E(S, \overline{S}) = \{(i, j) \in E : i \in S, j \in \overline{S}\}$ is the set of edges between S and its complement
- $\mathrm{vol}(S) = \sum_{i \in S} d_i$ is the sum of degrees of the vertices in S
- the *edge expansion* $\phi(S)$ *of* S is defined as:

$$\phi(S) = \frac{|E(S, \overline{S})|}{\mathrm{vol}(S)}.$$

  (c.f. quantity in Normalized Cut objective).

- the *edge expansion* ϕ of a graph is:

$$\phi := \min_{S \subset V} \max\left\{\phi(S), \phi(\overline{S})\right\} = \min_{S : |S| \leq \frac{|V|}{2}} \phi(S),$$

  which gives a measure of how much of a bottleneck there is in the graph; this is also the discrete analog to the *Cheeger isoperimetric constant* from Riemannian geometry.

Let $L = D - A$ be the Laplacian, with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$. The *Rayleigh quotient* for L is the function:

$$R(x) = \frac{x^T L x}{\|x\|^2}.$$

In this problem, we will prove *Cheeger's inequality* for d-regular graphs,[1]

$$\frac{\lambda_2}{2d} \leq \phi \leq \sqrt{\frac{2\lambda_2}{d}}. \tag{1}$$

[First Inequality: Relaxation] To prove the first inequality, notice that ϕ is the solution to the following optimization problem:

$$\phi = \min_{x \in \left\{\mathbf{1}_S : |S| \leq \frac{|V|}{2}\right\}} \frac{1}{d} \cdot \frac{x^T L x}{\|x\|^2},$$

where $\mathbf{1}_S \in \mathbb{R}^V$ is the indicator function on $S \subset V$. We can try *relaxing* the problem to all functions in $\mathbb{R}^V$, obtaining a lower bound:

$$\min_{x \in \mathbb{R}^V} \frac{1}{d} \cdot R(x) \leq \phi.$$

[1] Usually, Cheeger's inequality is written in terms of the *normalized Laplacian*, $\mathcal{L} = D^{-1/2} L D^{-1/2}$. For d-regular graphs, this is just $\frac{1}{d} L$. Thus, if $\mu_1 \leq \cdots \mu_n$ are the eigenvalues of $\mathcal{L}$, then Cheeger's inequality just states:

$$\frac{\mu_2}{2} \leq \phi \leq \sqrt{2\mu_2}.$$

The minimizer of the Rayleigh quotient is the smallest eigenvalue of L. But, we know that 0 is always an eigenvector of L with eigenvector $\mathbf{1}_V$, so this bound is quite trivial: $0 \leq \phi$. We can do better by first projecting the functions in $\left\{\mathbf{1}_S : |S| \leq \frac{|V|}{2}\right\}$ to the orthogonal complement of $\mathrm{span}(\mathbf{1}_V)$, before applying the relaxation step.

(i) Let $P : \mathbb{R}^V \to \mathbb{R}^V$ denote the projection to $\mathrm{span}(\mathbf{1}_V)^\perp$. Prove that after applying P, the squared-lengths of functions in our objective set do not shrink by more than half:

$$\min_{x \in \left\{\mathbf{1}_S : |S| \leq \frac{|V|}{2}\right\}} \frac{\|Px\|^2}{\|x\|^2} \geq \frac{1}{2}.$$

Modify the relaxation step to deduce that $\dfrac{\lambda_2}{2d} \leq \phi$.

[Second Inequality: Randomized Rounding] Let $v \in \mathbb{R}^V$ be the minimizer of your modified relaxation problem. Now, we'll round v into a discrete approximation; randomized rounding will help us prove an upper bound on the quality of our approximate solution.

As before, we can think of v as a graph embedding into $\mathbb{R}^1$. Without loss of generality, index the vertices so that the coordinates of v satisfy $v_1 \leq \cdots \leq v_n$. A natural bipartition of V is:

$$\{i : i \leq k\} \sqcup \{i : i > k\},$$

for some choice of $k \in [n-1]$. The smaller subset S_k becomes the approximation. Below, we'll provide a way to construct a clever probability distribution $\mathcal{D}$ over choice of k. You'll show the following:

$$\Pr_{k \sim \mathcal{D}}\left[\phi(S_k) \leq \sqrt{\frac{2\lambda_2}{d}}\right] > 0. \tag{2}$$

Before proving this bound, let's show that this gives us a constructive approximation to the min edge expansion problem.

(ii) Give a polynomial-time algorithm whose input is a d-regular graph and deterministically outputs a proper subset $S \subset V$ such that:

$$\phi(S) \leq \sqrt{\frac{2\lambda_2}{d}}.$$

Assuming that there exists $\mathcal{D}$ such that Equation 2 holds, prove the correctness of your algorithm, and give a coarse time complexity to your algorithm.

As a remark, there is nothing special about the probability distribution that we'll construct. A more general view on this technique is: (1) construct a probability distribution $\mathcal{D}'$ over subsets of V, then (2) prove that $\Pr_{S \sim \mathcal{D}'}[\phi(S) \leq \gamma] > \delta \geq 0$, which implies $\phi \leq \gamma$. The strength of the bound depends on the distribution and what we can show using it.

Now, we prove the bound by showing something slightly stronger. Let $u \in \mathbb{R}^V$ satisfy:

- $u_1 \leq \cdots \leq u_n$,
- $u_{\lfloor \frac{n}{2} \rfloor} = 0$,
- $u_1^2 + u_n^2 = 1$.

We'll construct a distribution $\mathcal{D}$ from a vector u where:

$$\Pr_{k\sim\mathcal{D}}\left[\phi(S_k) \leq \sqrt{\frac{2R(u)}{d}}\right] > 0.$$

Define the probability distribution function over $[n-1]$ by:

$$p(k) = |u_{k+1}^2 - u_k^2|.$$

Check for yourself that the third condition ensures that p is a probability distribution. For intuition, we want the probability of choosing k to depend on how far apart the vertices k and $k+1$ are in the embedding; the larger the gap $|u_{k+1}-u_k|$ is, the more likely choosing k should be. For our distribution though, instead of having the probability of choosing k proportional to $|u_{k+1} - u_k|$, we have it proportional to $|u_{k+1}^2 - u_k^2|$.

Of course, the minimizer v might not actually satisfy those above properties. But we can construct u satisfying those properties from v by translating v and scaling by some constants, $u = \alpha(v + \beta\mathbf{1})$.

(iii) Prove that if $u = \alpha(v + \beta\mathbf{1})$, then $R(u) \leq R(v)$. Also show that $R(v) = \lambda_2$.

(iv) If $(i, j) \in E$ is an edge (assume $i < j$), show that

$$\Pr_{k\sim\mathcal{D}}[i \leq k \text{ and } j > k] \leq |u_i - u_j|(|u_i| + |u_j|).$$

Use this to upper bound the expected edges cut by the choice of k, in terms of u:

$$\mathbb{E}_{k\sim\mathcal{D}}\left[|E(S_k, \overline{S}_k)|\right].$$

(v) Compute the probability that the vertex i is contained in S_k. Use this to compute $\mathbb{E}_{k\sim\mathcal{D}}[\text{vol}(S_k)]$.

(vi) Using (iv) and (v), prove that:

$$\frac{\mathbb{E}\left[|E(S_k, \overline{S}_k)\right]}{\mathbb{E}\left[\text{vol}(S_k)\right]} \leq \sqrt{\frac{2R(u)}{d}}.$$

(vii) Let X and Y be two positive random variables. Prove that:

$$\Pr\left[\frac{X}{Y} \leq \frac{\mathbb{E}X}{\mathbb{E}Y}\right] > 0.$$

(viii) Use (iii), (vi) and (vii) to deduce Equation 2.

4 **[Non-linear dimension reduction using kernel PCA]** Given a (mean centered) data $D \times n$ data matrix X. A typical way of doing PCA is to analyze the eigenvectors/values of the $D \times D$ covariance matrix XX^T. If $D \gg n$, then computing the covariance (and thus PCA is computationally prohibitive).

A natural question to wonder is what if instead we compute the eigenvectors/values of the much smaller inner product $n \times n$ matrix $X^\mathsf{T}X$?

(i) Let (λ_i, v_i) be the eigenvalue/vector pairs of XX^T, and (μ_i, u_i) be the eigenvalue/vector pairs of $X^\mathsf{T}X$. Show that one can write (λ_i, v_i) purely in terms of (μ_i, u_i) and possibly the original data, thus significantly improving the computational effort!

(ii) Say we want to project the given data matrix X into the $k < \min\{D, n\}$ dimensional PCA subspace. Using only (μ_i, u_i) and possibly the original data (cf. Part (i)), derive an expression for the k-dimensional PCA projection of the data matrix X.

(iii) Say we want to project a *new* datapoint $x \in \mathbb{R}^D$ (that is not present in the input data matrix) into the $k < \min\{D, n\}$ dimensional PCA subspace. Using only (μ_i, u_i) and possibly the original data and the new datapoint (cf. Part (i)), derive an expression for the k-dimensional PCA projection of x.

(iv) Another advantage of computing PCA using the inner product matrix $X^\mathsf{T}X$ is that we can kernelize all calculations for finding the k-dimensional PCA projection of both the input data matrix as well as of new datapoints (cf. Parts (ii) & (iii)). Let $K(x_i, x_j)$ be a kernel function that efficiently computes the inner product $\langle \phi(x_i), \phi(x_j) \rangle$ in a (possibly nonlinear) feature space ϕ. Derive kernelized expressions for k-dimensional PCA projection in the feature space ϕ of a datapoint x_i from the input data matrix and for a new datapoint x.

(v) Practically analyze your kernalized-PCA result on a simple dataset in $\mathbb{R}^2$ where points are distributed on a circle. What do 1D and 2D PCA projections of the circle data yield when you apply linear, quadratic and rbf kernels?

(no code submission required)

5 **[The cocktail party problem]** The *cocktail party effect* is the ability of an individual to focus on a specific human voice while filtering out other voices or background noise. Here we will explore how to algorithmically achieve this.

Download the datafile `mixed_sound_dataset.zip`. This zipfile contains three separate datasets, each containing audio recordings of multiple individuals speaking simultaneously (`.wav` format) from two microphones.

Your goal is to separate the audio signal of each individual. In doing so you'll find the discussion in the following paper helpful: "Independent Component Analysis: A Tutorial" by Hyvrinen and Oj.

(Note: it is not necessary to use ideas, algorithm from this paper. You can come up with your own novel algorithm, extend the ideas from the suggested paper, or use ideas from some other papers. Please make sure to cite all your sources)

(i) Propose your detailed algorithm for separating audio signals and implement the model in a scientific programming language of your choice.

You must provide the detailed step by step algorithm (with explanation) in the homework pdf and the code on Courseworks to receive full credit.

(ii) Use your algorithm to separate the audio signals for all three datasets and submit the results (as a playable `.wav` file) on Courseworks.