COMPSCI 514: Problem Set 4

Due: December 1, 11:59pm in Gradescope.

Instructions:

- You are allowed to, and highly encouraged to, work on this problem set in a group of up to three members.
- 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 may talk to members of other groups at a high level about the problems but **not work through the solutions in detail together**.
- You must show your work/derive any answers as part of the solutions to receive full credit.

1. Optimal Low-Rank Approximation – From Scratch (10 points)

In class we used the Courant-Fischer theorem to argue that the best low-rank approximation to any matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ is given by $\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T$ where $\mathbf{V}_k \in \mathbb{R}^{d \times k}$ contains the top k eigenvectors of $\mathbf{X}^T\mathbf{X}$ (i.e., the top k right singular vectors of $\mathbf{X}$). Here you will prove this from scratch, using just the basic properties of projection matrices and eigenvectors.

1. (2 points) Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be any matrix and $\mathbf{M} \in \mathbb{R}^{n \times d}$ be any rank-k matrix with SVD $\mathbf{M} = \mathbf{Q}\mathbf{D}\mathbf{Z}^T$ for orthonormal $\mathbf{Q} \in \mathbb{R}^{n \times k}$, $\mathbf{Z} \in \mathbb{R}^{d \times k}$, and diagonal $\mathbf{D} \in \mathbb{R}^{k \times k}$. Prove that $\|\mathbf{X} - \mathbf{M}\|_F^2 = \|\mathbf{X}\mathbf{Z}\mathbf{Z}^T - \mathbf{M}\|_F^2 + \|\mathbf{X} - \mathbf{X}\mathbf{Z}\mathbf{Z}^T\|_F^2$.
2. (2 point) Use part (1) to argue that if $\mathbf{M} = \underset{\mathbf{B}:\text{rank}(\mathbf{B}) \leq k}{\arg\min} \|\mathbf{X} - \mathbf{B}\|_F^2$ then $\mathbf{X}\mathbf{Z}\mathbf{Z}^T = \mathbf{M}$.
3. (2 points) Using a similar argument as above, one can show that, if $\mathbf{M} = \arg\min_{\mathbf{B}:\text{rank}(\mathbf{B}) \leq k} \|\mathbf{X} - \mathbf{B}\|_F^2$, then $\mathbf{Q}\mathbf{Q}^T\mathbf{X} = \mathbf{M}$. Use this and part (2) to show that: $\mathbf{X}^T\mathbf{X}\mathbf{Z} = \mathbf{Z}\mathbf{D}^2$. **Hint:** It may be helpful to prove as an intermediate step that $\mathbf{X}\mathbf{Z} = \mathbf{Q}\mathbf{D}$ and/or $\mathbf{Q}^T\mathbf{X} = \mathbf{D}\mathbf{Z}^T$.
4. (2 points) Use part (3) to argue that each column of $\mathbf{Z}$ is an eigenvector of $\mathbf{X}^T\mathbf{X}$.
5. (2 points) Complete the proof, showing that the best low-rank approximation of $\mathbf{X}$ is given by $\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T$ where $\mathbf{V}_k$ contains the top k eigenvectors of $\mathbf{X}^T\mathbf{X}$.

2. Recovering Locations from Distances (14 points + 6 points bonus)

Suppose you are given all pairs distances between a set of n points $\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n \in \mathbb{R}^d$, with $n > d$. Formally, you are given an $n \times n$ matrix $\mathbf{D}$ with $\mathbf{D}_{i,j} = \|\vec{p}_i - \vec{p}_j\|_2^2$. You would like to recover the location of the original points, up to possible translations, rotations, and reflections, which will not change the pairwise distances.[1] Let $\mathbf{P} \in \mathbb{R}^{n \times d}$ be the matrix with the n points as rows.

1. (2 points) Let $\mathbf{N}$ be $n \times n$ matrix with every row equal to $[\|\vec{p}_1\|_2^2, \|\vec{p}_2\|_2^2, \ldots, \|\vec{p}_n\|_2^2]$. Prove that $\mathbf{D} = \mathbf{N} + \mathbf{N}^T - 2\mathbf{P}\mathbf{P}^T$. **Hint:** Expand out $\|\vec{p}_i - \vec{p}_j\|_2^2$ as a dot product.

2. (2 points) Give an upper bound on rank($\mathbf{D}$).

3. (4 points) Show that:

$$(\mathbf{P}\mathbf{P}^T)_{i,j} = -\frac{1}{2}\left[\mathbf{D}_{i,j} - \frac{1}{n}\sum_{k=1}^{n}\mathbf{D}_{i,k} - \frac{1}{n}\sum_{k=1}^{n}\mathbf{D}_{j,k} + \frac{1}{n^2}\sum_{\ell=1}^{n}\sum_{k=1}^{n}\mathbf{D}_{k,\ell}\right].$$

   Hint: Since we can only recover the points up to translations anyways, you can assume without loss of generality that the points are have zero mean. I.e., $\sum_{i=1}^{n}\vec{p}_i = \vec{0}$.

4. (2 points) Describe an algorithm that, given $\mathbf{D}$, uses the formula above to recover $\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n \in \mathbb{R}^d$ up to rotation and translation. **Hint:** Even if you haven't figured out part (3), you can use the given formula to solve this part.

5. (2 points) Run your algorithm on the U.S. cities dataset provided in `UScities.txt` and plot the output. The distances in the file are Euclidean distances $\|\vec{p}_i - \vec{p}_j\|_2$ so you need to square them to obtain $\mathbf{D}$. Does the output make sense? Plot the estimated city locations and identify a few cities in your plot. Submit your code with the problem set.

6. (2 points) Plot the spectrum of the distance matrix $\mathbf{D}$ from part (5). Is the rank of $\mathbf{D}$ what was predicted in part (2)? What might be an explanation for any deviations?

7. **Bonus:** (6 points – quite challenging!) The problem of location recovery is closely related to both *triangulation in surveying/mapping* and *matrix completion*. Let's assume that for the U.S. cities dataset we actually only know the distance from every city to three other reference cities. That is, we know just three columns $\mathbf{D}$.

   (a) (2 points) Describe an algorithm that recovers the full distance matrix $\mathbf{D}$ using just these three columns. **Hint:** Given three columns of $\mathbf{D}$, think about how to find four vectors that span all columns of $\mathbf{D}$, using the ideas of parts (1)-(3). Then think about how to recover all the columns of $\mathbf{D}$ from this span.

   (b) (2 points) Describe the geometric intuition, perhaps using a picture, behind why we can recover all distances, and in turn city locations, given just the distances with three reference cities. This intuition doesn't have to exactly align with your algorithm above.

   (c) (2 points) Implement your algorithm and use it to recover the distance matrix $\mathbf{D}$ for the U.S. cities dataset. There will be some error due to approximation errors. Let $\tilde{\mathbf{D}}$ represent your recovered distance matrix. What is $\frac{\|\mathbf{D}-\tilde{\mathbf{D}}\|_F}{\|\mathbf{D}\|_F}$? Did you algorithm work well? Use your recovered matrix $\tilde{\mathbf{D}}$ to recover approximate positions of the U.S. cities. How do your results look in comparison to those of part (4).

[1] Formally, you want to recover the points up to a translation plus multiplication by an orthogonal matrix, which performs a unitary transformation https://en.wikipedia.org/wiki/Unitary_transformation

3. The Many Meanings of Graph Spectra (12 points)

In class we used the eigenvectors of the adjacency matrix and Laplacian to partition a graph across a small but well-balanced cut. The eigenvectors and eigenvalues of these matrices can tell us about a lot of other properties of the underlying graph as well. For all problems below, consider unweighted, undirected graphs with adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and Laplacian $\mathbf{L} \in \mathbb{R}^{n \times n}$.

1. (2 points) All Laplacian matrices have $\lambda_n(\mathbf{L}) = 0$. Prove that $\lambda_{n-k}(\mathbf{L}) = 0$ if and only if the graph has at least $k+1$ connected components. I.e., the number of 0 eigenvalues in $\mathbf{L}$ tells us the number of connected components it has. **Hint:** First prove the 'if' part of this statement. Then prove the 'only if', which is a bit harder.

2. (2 points) Show that $\lambda_1(\mathbf{A}) \geq c - 1$ where c is the size of the largest clique in the graph (i.e., the largest set of nodes that are all connected to each other.)

3. The top eigenvectors of $\mathbf{A}$ can also be used to find large cliques in the graph, which can correspond to anomalies, such as auto-generated accounts on social media sites. Consider a basic random graph model: a graph G on n nodes has each edge added with probability $p < 1$, independently. Then a random subset of $\sqrt{n}$ nodes $S \subset V$ is selected and all connections between the nodes in S are added, creating a clique of size $\sqrt{n}$. Assume for simplicity that you also add self-loops to all nodes in S.

   (a) (2 points) What is $\mathbb{E}[\mathbf{A}]$ for this random graph? For simplicity, order the nodes so that the subset of $\sqrt{n}$ nodes appear first. What is $\text{rank}(\mathbb{E}[\mathbf{A}])$?

   (b) (2 points) Argue that (up to a scaling factor) the top eigenvector $\vec{v}_1 \in \mathbb{R}^n$ of $\mathbb{E}[\mathbf{A}]$ has $\vec{v}_1(i) = \alpha$ for $i \in S$ and $\vec{v}_1(i) = 1$ for $i \notin S$ where $\alpha > 1$. I.e., the entries of $\vec{v}_1$ corresponding to the nodes in S are larger than the entries corresponding to other nodes. **Hint:** First argue that $\vec{v}_1(i)$ must have a single value for all $i \in S$ and a single value for all $i \notin S$. You may also use that by the *Perron-Frobenius theorem*, $\alpha \geq 0$.

   (c) (4 points) Generate a graph G according to the prescribed model with $n = 900$ (and so $|S| = 30$) and $p = 1/8$. Compute the top eigenvector of $\mathbf{A}$ and look at its 30 largest entries. What fraction of nodes in S do you recover by looking at these entries?

4. Stochastic Block Model Generalized (12 points)

In class we applied spectral methods to partition a graph into two large subsets of vertices with relatively few connections between them. We discussed how spectral clustering can be used to partition a graph into $k > 2$ pieces by combining a rank-k spectral embedding with e.g., k-means clustering. In this problem we will consider this method applied to the stochastic block model with a larger number of communities.

Let $G_{n,3}(p, q)$ be the distribution over random graphs where n is divided into three subsets X, Y, Z each with $n/3$ nodes in them (assume for simplicity that n is divisible by 3). Node i, j are connected with probability p if they are in the same subset (X, Y, or Z) and with probability $q < p$ if they are in different subsets. Connections are all made independently.

1. (2 points) Consider drawing a random graph $G \sim G_{n,3}(p, q)$. Let $\mathbf{A}$ be its adjacency matrix and $\mathbf{L}$ be its Laplacian, with nodes sorted by community id. What is $\mathbb{E}[\mathbf{A}]$? What is $\mathbb{E}[\mathbf{L}]$?

2. (4 points) What are the top three eigenvectors and eigenvalues of $\mathbb{E}[\mathbf{A}]$? What are the bottom three eigenvectors and eigenvalues of $\mathbb{E}[\mathbf{L}]$? **Note:** the eigendecompositions of $\mathbb{E}[\mathbf{A}]$ and $\mathbb{E}[\mathbf{L}]$ are not unique. Just describe one valid set of eigenvectors.

3. (2 points) Consider computing $\vec{v}_{n-1}$ and $\vec{v}_{n-2}$, the second and third smallest eigenvectors of $\mathbf{L}$. Then represent node i with the embedding $\vec{x}_i = [\vec{v}_{n-1}(i), \vec{v}_{n-2}(i)]$. Partition the nodes by applying k-means clustering to this embedded data set. Assume that you can find the optimal clustering efficiently. If $\mathbf{A}, \mathbf{L}$ were exactly equal to their expectations, describe how this method would perform in recovering the communities X, Y, and Z. **Note:** You don't need to actually implement the method to answer this question. Just describe how it should work in theory.

4. (4 points) Generate a 1200 node graph from $G_{n,3}(p, q)$ with $p = .1$ and $q = .02$ and partition it with the above spectral clustering algorithm applied to $\mathbf{L}$. Plot the adjacency matrix $\mathbf{A}$, the spectral embedding (i.e., $x_i = [\vec{v}_{n-1}(i), \vec{v}_{n-2}(i)]$ for all i), and the output of the k-means algorithm. How well does the algorithm perform?