COMPSCI 514: Problem Set 4

Due: 12/01 by 11:59pm in Gradescope. Challenge Problems due 12/04 by 11:59pm.

Instructions:

- You are allowed 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 should separately submit the core competency problems from any challenge problems you choose to complete. These do not necessarily need to be submitted with the same groups.
- 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.

Core Competency Problems

1. Eigendecomposition and SVD Practice (10 points)

1. (2 points) For any $\mathbf{X} \in \mathbb{R}^{n \times d}$ prove that the eigenvalues of $\mathbf{X}^T\mathbf{X}$ are real and non-negative.
2. (2 points) Let $\mathbf{A} \in \mathbb{R}^{d \times d}$ be symmetric. Prove that any eigenvalue λ of $\mathbf{A}$ must be real. **Hint:** For the eigenvector $\mathbf{x}$ corresponding to λ, consider the quantity $\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}$.
3. (2 points) Prove that if $\mathbf{A}$ can be written as $\mathbf{U}\mathbf{S}\mathbf{U}^T$ where $\mathbf{U}$ has orthonormal columns and $\mathbf{S}$ is diagonal, then every column of $\mathbf{U}$ is an eigenvector of $\mathbf{A}$ and the diagonal entries of $\mathbf{S}$ are the corresponding eigenvalues. That is, all decompositions of this form are indeed eigendecompositions.
4. (2 points) Prove that any symmetric $\mathbf{A} \in \mathbb{R}^{d \times d}$ can be written in its SVD as $\mathbf{A} = \mathbf{V}\mathbf{\Sigma}\bar{\mathbf{V}}^T$ where $\mathbf{V} \in \mathbb{R}^{d \times d}$ and $\bar{\mathbf{V}} \in \mathbb{R}^{d \times d}$ are identical up to sign flips on their columns. That is, letting $\mathbf{v}_i$ and $\bar{\mathbf{v}}_i$ be the i^{th} columns of $\mathbf{V}$ and $\bar{\mathbf{V}}$ respectively, we either have $\mathbf{v}_i = \bar{\mathbf{v}}_i$ or $\mathbf{v}_i = -\bar{\mathbf{v}}_i$. **Hint:** Start by writing $\mathbf{A}$ in its eigendecomposition and than transforming this into a valid singular value decomposition.
5. (2 points) Let $\mathbf{A} \in \mathbb{R}^{d \times d}$ be a symmetric matrix. Consider the matrix $\mathbf{B} = \mathbf{A}^3 + 2\mathbf{I}$. Give a formula relating the eigenvalues of $\mathbf{B}$ to those of $\mathbf{A}$.

2. Eigendecomposition, SVD, and Matrix Inversion (6 points)

1. (2 points) Let $\mathbf{A} \in \mathbb{R}^{d \times d}$ be a non-singular symmetric matrix with eigendecomposition $\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$. Let $\mathbf{\Lambda}^{-1}$ be the diagonal matrix with diagonal entries equal to $1/\lambda_1, \ldots, 1/\lambda_d$. Show that $\mathbf{A}^{-1} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}^T$. **Hint:** To prove that $\mathbf{B} = \mathbf{A}^{-1}$ for some matrix $\mathbf{B}$ it suffices to show that $\mathbf{A}\mathbf{B} = \mathbf{B}\mathbf{A} = \mathbf{I}$.

2. (2 points) Consider any $\mathbf{A} \in \mathbb{R}^{n \times d}$ with SVD $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. One of the most classic data fitting methods, least squares regression is: given a vector $\mathbf{y} \in \mathbb{R}^n$, find:

$$\mathbf{b}_* \in \arg\min_{\mathbf{b} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{b} - \mathbf{y}\|_2^2. \tag{1}$$

The rows of $\mathbf{A}$ represent d-dimensional data points, the entries of $\mathbf{y}$ represent observations at these points, and $\mathbf{A}\mathbf{b}_*$ is the 'line of best fit', which attempts to fit these observations as closely as possible with a linear function of the rows. Prove that $\mathbf{b}_* = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{y}$ satisfies equation (1) above. Avoid using any calculus in your proof. **Hint:** Try plugging in $\mathbf{b}_*$ and see what you get. The solution will involve a projection matrix.

3. (2 points) Argue via part (2) that if $\mathbf{x} \in \mathbb{R}^d$ is in the row span of $\mathbf{A}$ then $\mathbf{B} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T$ inverts the action of $\mathbf{A}$ on $\mathbf{x}$. I.e., $\mathbf{B}\mathbf{A}\mathbf{x} = \mathbf{x}$. Similarly show that if $\mathbf{x} \in \mathbb{R}^n$ is in the column span of $\mathbf{A}$, $\mathbf{A}$ inverts the action of $\mathbf{B}$ on $\mathbf{x}$. I.e., $\mathbf{A}\mathbf{B}\mathbf{x} = \mathbf{x}$.

3. Spectral Graph Theory Practice (8 points)

1. (2 points) We class we saw that for any graph Laplacian $\mathbf{L}$, $\lambda_n(\mathbf{L}) = 0$ and $\mathbf{v}_n = \frac{1}{\sqrt{n}} \cdot \mathbf{1}$. Prove that for any disconnected graph, there is a second eigenvector $\mathbf{v}_{n-1}$ which is orthogonal to $\mathbf{v}_n$ and has corresponding eigenvalue $\lambda_{n-1}(\mathbf{L}) = 0$. **Hint:** Pick two arbitrary connected components S_1 and S_2 of the graph and let $\mathbf{v}_{n-1}$ have the same value on each vertex in one component, the same value on each vertex in the other component, and the same value on all vertices outside the two components.

2. (2 points) Prove that for any connected graph with Laplacian $\mathbf{L}$, $\lambda_{n-1}(\mathbf{L}) > 0$. **Hint:** Show that for any $\mathbf{v}$ which is not a scaling of the all ones vector that $\mathbf{v}^T\mathbf{L}\mathbf{v} > 0$. Use the identity $\mathbf{v}^T\mathbf{L}\mathbf{v} = \sum_{(i,j)\in E}(\mathbf{v}(i) - \mathbf{v}(j))^2$.

3. (2 points) Consider an unweighted undirected graph with adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. 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.) **Hint:** Apply Courant-Fischer.

4. (2 points) Consider an unweighted undirected graph with adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. Prove that $\lambda_1(\mathbf{A}) \leq d_{max}$, where d_{max} is the maximum degree of a vertex in the graph. **Hint:** Let $\mathbf{v}_1$ be the top eigenvector of $\mathbf{A}$ and let $i = \text{argmax}_{j\in[n]} |\mathbf{v}_1(j)|$. Prove that we cannot have $[\mathbf{A}\mathbf{v}_1](i) > d_{max} \cdot \mathbf{v}_1(i)$.

4. Three Community Stochastic Block Model (10 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 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. (2 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 orthogonal eigenvectors.

3. (2 points) Consider computing $\mathbf{v}_{n-1}$ and $\mathbf{v}_{n-2}$, the second and third smallest eigenvectors of $\mathbf{L}$. Then represent node i with the embedding $\mathbf{x}_i = [\mathbf{v}_{n-1}(i), \mathbf{v}_{n-2}(i)]$. Partition the nodes by applying k-means clustering to this embedded data set with $k = 3$. 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 1500 node graph from $G_{n,3}(p, q)$ with $p = .2$ and $q = .04$ and partition it with the above spectral clustering algorithm applied to $\mathbf{L}$. Plot the adjacency matrix $\mathbf{A}$, the spectral embedding (i.e., $x_i = [\mathbf{v}_{n-1}(i), \mathbf{v}_{n-2}(i)]$ for all i), and the output of the k-means algorithm. How well does the algorithm perform?

Challenge Problems

C1. Location Recovery via Low-Rank Approximation 🌶

Suppose you are given all pairs distances between a set of n points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n \in \mathbb{R}^d$, with $n > d$. Formally, you are given an $n \times n$ matrix $\mathbf{D}$ with $\mathbf{D}_{i,j} = \|\mathbf{p}_i - \mathbf{p}_j\|_2^2$. 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. Let $\mathbf{N}$ be $n \times n$ matrix with every row equal to $[\|\mathbf{p}_1\|_2^2, \|\mathbf{p}_2\|_2^2, \ldots, \|\mathbf{p}_n\|_2^2]$. Prove that $\mathbf{D} = \mathbf{N} + \mathbf{N}^T - 2\mathbf{P}\mathbf{P}^T$. **Hint:** Expand out $\|\mathbf{p}_i - \mathbf{p}_j\|_2^2$ as a dot product.

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

3. Since we can only recover the points up to translations, assume without loss of generality that the points have zero mean. I.e., that $\sum_{i=1}^n \mathbf{p}_i = \mathbf{0}$. Under this assumption, show that:

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

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

5. 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 $\|\mathbf{p}_i - \mathbf{p}_j\|_2$ 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. 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? **Hint:** Do our cities lie on a 2-dimensional plane?

C2. Location Recovery via Matrix Completion 🌶🌶

The problem of location recovery studied in C1 is closely related to both *triangulation in surveying/mapping* and *matrix completion*. Consider the setting of C1, but assume that for the U.S. cities dataset we actually only know the distance from every city to three other reference cities. I.e., we know just three columns $\mathbf{D}$. **Note:** You'll want to complete C1 before tackling this problem.

1. 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.

2. 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.

[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. 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) of C1.

C3. Top Eigenvalue Approximation Via Krylov Subspace Methods 🌶🌶

In this problem we will give an analysis of a Krylov subspace method for approximating the largest eigenvalue of a symmetric matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$. Krylov subspace methods improve upon the power method and are the dominant approach in practice for eigenvalue/eigenvector approximation, implemented e.g., in the `eigs` methods in SciPy and Matlab.

For simplicity we will assume throughout this problem that $\mathbf{A}$ has all non-negative eigenvalues, denoted by $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d \geq 0$. Let $\gamma = \frac{\lambda_1-\lambda_2}{\lambda_1}$ be the eigenvalue gap.

1. Let $\mathbf{x} \in \mathbb{R}^d$ be a random starting vector. For some integer $t > 0$, consider the Krylov matrix $\mathbf{K} = [\mathbf{x}, \mathbf{A}\mathbf{x}, \mathbf{A}^2\mathbf{x}, \ldots, \mathbf{A}^t\mathbf{x}]$. That is, $\mathbf{K} \in \mathbb{R}^{d\times t+1}$ has i^{th} column equal to $\mathbf{A}^{i-1}\mathbf{x}$. How long does it take to compute $\mathbf{K}$? **Hint:** Like in power method, you should avoid explicitly computing $\mathbf{A}^i$ for any i.

2. Assume that $\mathbf{K}$ has full column rank and let $\mathbf{Q} \in \mathbb{R}^{d\times t+1}$ be an orthonormal basis for the column span of $\mathbf{K}$. $\mathbf{Q}$ can be computed in $O(dt^2)$ time. Let $\tilde{\lambda}_1 = \lambda_1(\mathbf{Q}^T\mathbf{A}\mathbf{Q})$. Argue that:
$$\tilde{\lambda}_1 = \max_{\mathbf{v}\in\mathbb{R}^d:\|\mathbf{v}\|_2=1 \text{ and } \mathbf{v}\in span(\mathbf{Q})} \mathbf{v}^T\mathbf{A}\mathbf{v} = \max_{\mathbf{v}\in\mathbb{R}^d:\mathbf{v}\in span(\mathbf{Q})} \frac{\mathbf{v}^T\mathbf{A}\mathbf{v}}{\|\mathbf{v}\|_2^2},$$
where $span(\mathbf{Q})$ denotes the column span of $\mathbf{Q}$. Conclude that $\tilde{\lambda}_1 \leq \lambda_1(\mathbf{A})$. **Hint:** Use Courant-Fischer.

3. Let $p : \mathbb{R} \to \mathbb{R}$ be any degree t polynomial. Argue that $p(\mathbf{A})\mathbf{x} \in span(\mathbf{Q})$. Here, if $p(x) = c_0 + c_1x + \ldots + c_tx^t$, we define $p(\mathbf{A}) = c_0 \cdot \mathbf{I} + c_1 \cdot \mathbf{A} + \ldots + c_t \cdot \mathbf{A}^t$. Conclude that
$$\tilde{\lambda}_1 \geq \max_{\text{degree t polynomials p}} \frac{\mathbf{x}^T p(\mathbf{A})\mathbf{A}p(\mathbf{A})\mathbf{x}}{\|p(\mathbf{A})\mathbf{x}\|_2^2}.$$

By parts (2) and (3), to show that $(1-\epsilon)\lambda_1(\mathbf{A}) \leq \tilde{\lambda}_1 \leq \lambda_1(\mathbf{A})$, we just need to show that there exists some degree t polynomial p such that $\frac{\mathbf{x}^T p(\mathbf{A})\mathbf{A}p(\mathbf{A})\mathbf{x}}{\|p(\mathbf{A})\mathbf{x}\|_2^2} \geq (1-\epsilon)\lambda_1(\mathbf{A})$. We will do this below.

4. Write the random starting vector $\mathbf{x} \in \mathbb{R}^d$ in the eigenvector basis as $\mathbf{x} = c_1\mathbf{v}_1 + \ldots + c_d\mathbf{v}_d$ (where $\mathbf{v}_1, \ldots, \mathbf{v}_d$ are the eigenvectors of $\mathbf{A}$). Argue that $\frac{\mathbf{x}^T p(\mathbf{A})\mathbf{A}p(\mathbf{A})\mathbf{x}}{\|p(\mathbf{A})\mathbf{x}\|_2^2} \geq \lambda_1 \cdot \frac{c_1^2 \cdot p(\lambda_1)^2}{\sum_{i=1}^d c_i^2 p(\lambda_i)^2}$.

5. Recall from the power method analysis shown in class that if $\mathbf{x}$ is chosen to have random Gaussian entries, then with very high probability, $\max_{j\in[d]}\left|\frac{c_j}{c_1}\right| \leq cd^2\log d$ for some constant c. Assuming this bound holds, argue that there exists a polynomial p with degree $O(\sqrt{1/\gamma} \cdot \log(d/\epsilon))$ such that $\frac{\mathbf{x}^T p(\mathbf{A})\mathbf{A}p(\mathbf{A})\mathbf{x}}{\|p(\mathbf{A})\mathbf{x}\|_2^2} \geq (1-\epsilon)\lambda_1$. This establishes that $(1-\epsilon)\lambda_1(\mathbf{A}) \leq \tilde{\lambda}_1 \leq \lambda_1(\mathbf{A})$ when $t = O(\sqrt{1/\gamma} \cdot \log(d/\epsilon))$.

   To do so, use part (4) along with the following Lemma, which is obtained by considering the Chebyshev polynomials , which are a family of polynomials that grow as quickly as possible outside a certain interval:

Lemma 1. *For any $\gamma, \delta \in (0,1)$, there is a degree $O\left(\sqrt{1/\gamma} \cdot \log(1/\delta)\right)$ polynomial $\hat{p}$ such that $\hat{p}(1) = 1$ and $|\hat{p}(x)| \leq \delta$ for any $x < 1 - \gamma$.*

6. How does the iteration bound in part (5) compare to the bound shown in class for power method? When do you expect the Krylov subspace method to significantly outperform the power method? **Note:** In class we showed that power method outputs an approximate eigenvector with $\tilde{\mathbf{v}}_1$ with $\|\tilde{\mathbf{v}}_1 - \mathbf{v}_1\|_2 \leq \epsilon$. We could have also shown that $\tilde{\lambda}_1 = \tilde{\mathbf{v}}_1^T \mathbf{A} \tilde{\mathbf{v}}_1 \geq (1-\epsilon)\lambda_1$ using essentially the same analysis and achieving the same iteration bound.