Linear Algebra HW 8

gjd9961

November 2021

1 Problem 8.1

Let $A \in \mathbb{R}^{n\times m}$. The Singular Values Decomposition (SVD) tells us that there exists two orthogonal matrices $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{m\times m}$ and a matrix $\Sigma \in \mathbb{R}^{n\times m}$ such that $\Sigma_{1,1} \geq \Sigma_{2,2} \geq \cdots \geq 0$ and $\Sigma_{i,j} = 0$ for $i \neq j$

$$A = U\Sigma V.$$

The columns $u_1, \dots, u_n$ of U (respectively the columns $v_1, \dots, v_m$ of V) are called the left (resp. right) singular vectors of A. The non-negative numbers $\sigma_i \stackrel{\text{def}}{=} \Sigma_{i,i}$ are the singular values of A. Moreover we also know that

$$r \stackrel{\text{def}}{=} rank(A) = \#\{i \,|\, \Sigma_{i,i} \neq 0\}$$

a) Let $\widetilde{U} = \begin{pmatrix} | & & | \\ u_1 & \cdots & u_r \\ | & & | \end{pmatrix} \in \mathbb{R}^{n\times r}$, $\widetilde{V} = \begin{pmatrix} | & & | \\ v_1 & \cdots & v_r \\ | & & | \end{pmatrix} \in \mathbb{R}^{m\times r}$ and $\widetilde{\Sigma} = (\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r\times r}$. Show that $A = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T$.

Let x be a vector in the input space, $x \in \mathbb{R}^m$ and x be some linear combination of the orthonormal vectors of V, $x = \alpha_1 v_1 + \cdots + \alpha_m v_m$. Let y be a vector in the output space, $\mathbb{R}^n$, such that $Ax = y$ and therefore $U\Sigma V^T x = y$. Therefore, $y \in Im(A)$ and $y \in Im(U\Sigma V^T x)$. Let $r = Rank(A)$. Lets approach the problem one step at a time. What does $V^T x$ equal?

$$V^T x = \begin{pmatrix} -- & v_1 & -- \\ -- & \dots & -- \\ -- & v_m & -- \end{pmatrix} \begin{pmatrix} | \\ x \\ | \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ | \\ \alpha_m \end{pmatrix} \tag{1}$$

This is because matrix vector multiplication is the dot product of the rows of the matrix and the vector column, and since the vector x is a linear combination of the orthonormal columns of V we have:

$$\langle v_i, x\rangle = \langle v_i, \alpha_1 v_1 + \cdots + v_m\alpha_m\rangle = \langle v_i, v_1\alpha_1\rangle + \cdots + \langle v_i, v_m\alpha_m\rangle$$

Where we have for any $\langle v_i, \alpha_j v_j\rangle = \alpha_j$ if $v_i = v_j$, otherwise if $v_i \neq v_j$ then we have $\langle v_i, \alpha_j v_j\rangle = 0$ Now that we know what $V^T x$ is, lets figure out what $\Sigma V^T x$ is. Remember, Σ is a $\mathbb{R}^{n\times m}$ matrix, whose diagonals are the singular values of the matrix A and that r of them will be non zero.

$$\Sigma V^T x = \begin{pmatrix} \sigma_{1,1} & \dots & \dots & \dots & 0_{1,m} \\ 0 & \sigma_{2,2} & \dots & \dots & 0 \\ \vdots & & \ddots & \sigma_{r,r} & 0 \\ 0_{n,1} & \dots & \dots & \dots & 0_{n,m} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ | \\ \alpha_m \end{pmatrix} = \begin{pmatrix} \alpha_1 \times \sigma_1 \\ \vdots \\ \alpha_r \times \sigma_r \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

So the resulting output vector of $\Sigma V^T x$ will be a vector with its first r entries equal to $\alpha_i \times \sigma_i$ for $i \in \{1, \dots, r\}$ and the rest of the values (values $r+1$ through n) will be equal to 0. Lastly, lets see what happens when we do the whole thing: $U\Sigma V^T x$

$$U\Sigma V^T x = \begin{pmatrix} | & & | \\ u_1 & \cdots & u_n \\ | & & | \end{pmatrix} \begin{pmatrix} \alpha_1 \times \sigma_1 \\ \vdots \\ \alpha_r \times \sigma_r \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \sum_{i=1}^{r} \alpha_i \times \sigma_i \times u_i + \sum_{j=r+1}^{n} 0 \times u_j = \sum_{i=1}^{r} \alpha_i \times \sigma_i \times u_i = y$$

The process is the same for $\widetilde{U}\widetilde{\Sigma}\widetilde{V}^T x$, Therefore:

$$\begin{aligned} Ax &= Ax \\ U\Sigma V^T x &= \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T x \\ \sum_{i=1}^{r} \alpha_i \times \sigma_i \times u_i + \sum_{j=r+1}^{n} 0 \times u_j &= \sum_{i=1}^{r} \alpha_i \times \sigma_i \times u_i \\ \sum_{i=1}^{r} \alpha_i \times \sigma_i \times u_i &= \sum_{i=1}^{r} \alpha_i \times \sigma_i \times u_i \end{aligned} \tag{2}$$

That shows the equivalence for matrix vector multiplication, but we can show this without needing the vector. In general, any entry i, j in the matrix product of $U\Sigma V^T$ is equal to the entry in the matrix product of $\widetilde{U}\widetilde{\Sigma}\widetilde{V}^T$. We can show this as follows:

$$\begin{aligned} (U\Sigma V^T)_{i,j} &= (\widetilde{U}\widetilde{\Sigma}\widetilde{V}^T)_{i,j} \\ \sum_{l=1}^{n} u_{i,l} \times \sigma_i \times v_{l,j}^T &= \sum_{l=1}^{r} u_{i,l} \times \sigma_i \times v_{l,j}^T \\ \sum_{l=1}^{r} u_{i,l} \times \sigma_i \times v_{l,j}^T + \sum_{l=r+1}^{n} 0 \times u_j &= \sum_{l=1}^{r} u_{i,l} \times \sigma_i \times v_{l,j}^T \\ \sum_{l=1}^{r} u_{i,l} \times \sigma_i \times v_{l,j}^T &= \sum_{l=1}^{r} u_{i,l} \times \sigma_i \times v_{l,j}^T \quad \square \end{aligned} \tag{3}$$

This further proves their equivalence.

b) Give orthonormal bases of $Ker(A)$ and $Im(A)$ in terms of the singular vectors $u_1, \ldots, u_n$ and $v_1, \ldots, v_m$.

Firstly for the basis of Im(A). Let x be a vector in the input space, $x \in \mathbb{R}^m$ and x be some linear combination of the orthonormal vectors of V, $x = \alpha_1 v_1 + \cdots + \alpha_m v_m$. Let y be a vector in the output space, $\mathbb{R}^n$, such that $Ax = y$ and therefore $U\Sigma V^T x = y$. Therefore, $y \in Im(A)$ and $y \in Im(U\Sigma V^T x)$. Let $r = Rank(A)$.

Lets approach the problem one step at a time. What does $V^T x$ equal?

$$V^T x = \begin{pmatrix} -- & v_1 & -- \\ -- & \ldots & -- \\ -- & v_m & -- \end{pmatrix} \begin{pmatrix} | \\ x \\ | \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ | \\ \alpha_m \end{pmatrix} \tag{4}$$

This is because matrix vector multiplication is the dot product of the rows of the matrix and the vector column, and since the vector x is a linear combination of the orthonormal columns of V we have:

$$\langle v_i, x \rangle = \langle v_i, \alpha_1 v_1 + \cdots + v_m \alpha_m \rangle = \langle v_i, v_1 \alpha_1 \rangle + \cdots + \langle v_i, v_m \alpha_m \rangle$$

Where we have for any $\langle v_i, \alpha_j v_j \rangle = \alpha_j$ if $v_i = v_j$, otherwise if $v_i \neq v_j$ then we have $\langle v_i, \alpha_j v_j \rangle = 0$
Now that we know what $V^T x$ is, lets figure out what $\Sigma V^T x$ is. Remember, Σ is a $\mathbb{R}^{n \times m}$ matrix, whose diagonals are the singular values of the matrix A and that r of them will be non zero.

$$\Sigma V^T x = \begin{pmatrix} \sigma_{1,1} & \ldots & \ldots & \ldots & 0_{1,m} \\ 0 & \sigma_{2,2} & \ldots & \ldots & 0 \\ \vdots & & \ddots & \sigma_{r,r} & 0 \\ 0_{n,1} & \ldots & \ldots & \ldots & 0_{n,m} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ | \\ \alpha_m \end{pmatrix} = \begin{pmatrix} \alpha_1 \times \sigma_1 \\ \vdots \\ \alpha_r \times \sigma_r \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

So the resulting output vector of $\Sigma V^T x$ will be a vector with its first r entries equal to $\alpha_i \times \sigma_i$ for $i \in \{1, \ldots, r\}$ and the rest of the values (values $r+1$ through n) will be equal to 0. Lastly, lets see what happens when we do the whole thing: $U\Sigma V^T x$

$$U\Sigma V^T x = \begin{pmatrix} | & & | \\ u_1 & \cdots & u_n \\ | & & | \end{pmatrix} \begin{pmatrix} \alpha_1 \times \sigma_1 \\ \vdots \\ \alpha_r \times \sigma_r \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \sum_{i=1}^{r} \alpha_i \times \sigma_i \times u_i + \sum_{j=r+1}^{n} 0 \times u_j = \sum_{i=1}^{r} \alpha_i \times \sigma_i \times u_i = y$$

To recap what we have shown is that $y \in Im(A)$ is a linear combination of the first r columns of the orthonormal matrix U. That is to say:

$$\begin{aligned} y &= Ax \qquad \text{By definition} \\ y &= U\Sigma V^T x \qquad \text{By SVD} \\ y &= \sum_{i=1}^{r} \sigma_i \times \alpha_i \times u_i \quad \square \end{aligned} \tag{5}$$

Therefore, the basis of Im(A) can be expressed by the orthonormal basis $\{u_1, \ldots, u_r\}$

Now we can repeat this exact process, but we need to add an additional assumption. Let $r = Rank(A)$ and $r < m < n$, such that $dim(Ker(A)) \geq 1$. Let $x \in \mathbb{R}^m$ and $x \in Ker(A)$ such that $Ax = 0$ and $U\Sigma V^T x = 0$. Let $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$ and x be the linear combination of $x = \alpha_1 v_1 + \cdots + \alpha_m v_m$ where $\alpha_i = 0$ for $\alpha_1 \ldots \alpha_r$ and α_i non zero for $\alpha_{r+1}, \ldots, \alpha_m$

Lets approach the problem one step at a time. What does $V^T x$ equal?

$$V^T x = \begin{pmatrix} -- & v_1 & -- \\ -- & \ldots & -- \\ -- & v_m & -- \end{pmatrix} \begin{pmatrix} | \\ x \\ | \end{pmatrix} = \begin{pmatrix} 0 \\ | \\ \alpha_{r+1} \\ | \\ \alpha_m \end{pmatrix} \tag{6}$$

This is because matrix vector multiplication is the dot product of the rows of the matrix and the vector column, and since the vector x is a linear combination of the orthonormal columns of V we have:

$$\langle v_i, x \rangle = \langle v_i, \alpha_1 v_1 + \ldots v_m \alpha_m \rangle = \langle v_i, v_1 \alpha_1 \rangle + \cdots + \langle v_i, v_m \alpha_m \rangle$$

Where we have for any $\langle v_i, \alpha_j v_j \rangle = \alpha_j$ if $v_i = v_j$, otherwise if $v_i \neq v_j$ then we have $\langle v_i, \alpha_j v_j \rangle = 0$

In other words, the first r entries of $V^T x$ are equal to 0, and the entries from $r+1$ to m are equal to their respective α values. Now that we know what $V^T x$ is, lets figure out what $\Sigma V^T x$ is.

Remember, Σ is a $\mathbb{R}^{n \times m}$ matrix, whose diagonals are the singular values of the matrix A and that r of them will be non zero.

$$\Sigma V^T x = \begin{pmatrix} \sigma_{1,1} & \ldots & \ldots & \ldots & 0_{1,m} \\ 0 & \sigma_{2,2} & \ldots & \ldots & 0 \\ \vdots & & \ddots & \sigma_{r,r} & 0 \\ 0_{n,1} & \ldots & \ldots & \ldots & 0_{n,m} \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ \alpha_{r+1} \\ \vdots \\ \alpha_m \end{pmatrix} = \sum_{i=1}^{r} \sigma_i \times 0 + \sum_{j=r+1}^{m} 0 \times \alpha_j = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

So the resulting output vector of $\Sigma V^T x$ will be a vector with all of its entries equal to 0.
. Lastly, lets see what happens when we do the whole thing: $U\Sigma V^T x$

$$U\Sigma V^T x = \begin{pmatrix} | & & | \\ u_1 & \cdots & u_n \\ | & & | \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} = \sum_{i=1}^{n} 0 \times u_i = 0$$

To recap what we have shown is that $x \in Ker(A)$ is a linear combination of the $r+1$ through m columns of the orthogonal matrix V. Therefore, the basis for Ker(A) can be expressed by the orthonormal basis $\{v_{r+1}, \ldots, v_m\}$

2 Problem 8.2

For any two matrices $A, B \in \mathbb{R}^{n \times m}$ we define the *Frobenius inner-product* as

$$\langle A, B \rangle_F = Trace(AB).$$

We showed in the midterm that it verifies the points of the definition 2.1 of Lecture 4 for the square matrix case (one can also check that it works for rectangular matrices). Show that the induced norm $\|A\|_F = \sqrt{Trace(A^T A)}$ can be computed as a function of the singular values $\sigma_1, \ldots, \sigma_{\min(n,m)}$ of A as

$$\|A\|_F = \sqrt{\sum_{i=1}^{\min(n,m)} \sigma_i^2}.$$

To answer this question, we will need to use a fact we proved in past home-works: that $Trace(BC) = Trace(CB)$. Let the SVD of A equal $A = U\Sigma V^T$ where U and V are orthogonal matrices in $\mathbb{R}^{n \times n}$ and $\mathbb{R}^{m \times m}$ respectively, and Σ be the diagonal matrix of singular values $\sigma_1, \ldots, \sigma_{\min(n,m)}$ of A. Also, let B,C be two rectangular matrices, such that $B = V\Sigma^T$ and $C = \Sigma V^T$.

$$\begin{aligned}
\langle A, A \rangle &= \sqrt{Trace(A^T A)} \text{ By definition of Problem Statement} \\
\langle A, A \rangle &= \sqrt{Trace((U\Sigma V^T)^T (U\Sigma V^T))} \text{ By SVD} \\
\langle A, A \rangle &= \sqrt{Trace(V\Sigma^T U^T U\Sigma V^T)} \text{ Distribute Transpose} \\
\langle A, A \rangle &= \sqrt{Trace(V\Sigma^T \Sigma V^T)} \text{ as } U^T U = Id_n \\
\langle A, A \rangle &= \sqrt{Trace(BC)} \text{ Substitute Identities} \\
\langle A, A \rangle &= \sqrt{Trace(CB)} \text{ using } Trace(BC) = Trace(CB) \\
\langle A, A \rangle &= \sqrt{Trace(\Sigma V^T V \Sigma^T)} \text{ Substitute Identities} \\
\langle A, A \rangle &= \sqrt{Trace(\Sigma \Sigma^T)} \text{ as } V^T V = Id_n \\
\langle A, A \rangle &= \sqrt{\sum_{i=1}^{\min(n,m)} \sigma_i^2} \\
\|A\|_F &= \sqrt{\sum_{i=1}^{\min(n,m)} \sigma_i^2} \quad \square
\end{aligned} \tag{7}$$

3 Problem 8.3

a) Show that $\mathcal{H}(1)$ is true.

$\mathcal{H}(1)$ is true by the definition of the adjacency matrix of graph G. Taking $k = 1$ we want to prove that A^1 represents the number of ways to go from i to j in a single step. And, it just so happens, by definition, the adjacency matrix's entries of $A_{i,j}$ describe the number of ways to go from i to j in one step, therefore it holds true for when $k = 1$. We can illustrate this in the following way. Consider a the following graph and its related adjacency matrix:

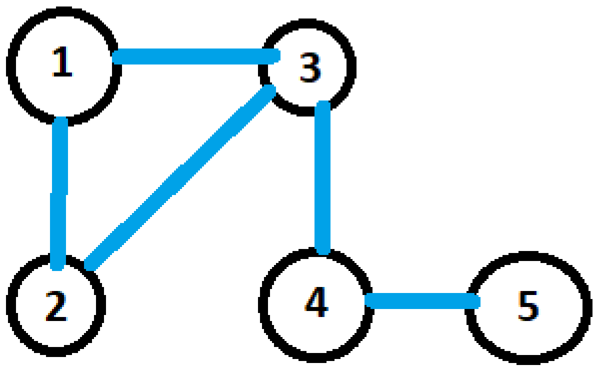


Figure 1: Graph G

Its adjacency matrix, $A^1_{i,j} = A_{i,j}$ is the number of ways we can go from i to j (or vica versa if we look at $A_{j,i}$ as the adjacency matrix is symmetric) in 1 step.

$$Adjacency\ Matrix\ of\ G = A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

b) Show that if $\mathcal{H}(k)$ is true for some k, then $\mathcal{H}(k+1)$ is also true.

Knowing what we know if part 1, that $\mathcal{H}(1)$ holds, we can show that $\mathcal{H}(1+1) = \mathcal{H}(2)$ is equal to $A^1 \times A^{k-1}$ where A is the adjacency matrix, and $k = 2$, and that $A^2_{i,j}$ represents the amount of paths of i to j with length 2. As all adjacency matrices are square symmetric, proving any arbitrary example will hold for all adjacency matrices. We will demonstrate with our graph G and our adjacency matrix A, and then generalize our solution to all adjacency matrices.

$$A^{k-1} \times A = \begin{pmatrix} 2 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 & 0 \\ 1 & 1 & 3 & 0 & 1 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Consulting with the graph in Figure 1, we can see that $A^2_{i,j}$ accurately describes all the paths from i to j in length k (2) steps. Since we have shown how $\mathcal{H}(k)$ where k=1 and $\mathcal{H}(k+1)$ where $k+1 = 2$ holds, we can generalize our solution for any positive integer k.

We have shown that our solution works for our base case, $\mathcal{H}(k)$ where $k = 1$, and that it works for any $\mathcal{H}(k+1)$ where k is a positive integer, then we know that any $\mathcal{H}(k)$ can be expressed as $\mathcal{H}(k) = A^k$, it follows that it can be expressed as $A^k_{ij} = \sum_{l=1}^{n} A^{k-1}_{il} * A_{lj}$ where A^{k-1}_{il} represents the number of ways to get from i to l in $k-1$ steps and A_{lj} is the number of ways to get from l to i in 1 step. We can do this for any positive integer k and it will hold. Therefore:

$$A^{k+1}_{ij} = \sum_{l=1}^{n} A^k_{il} * A_{lj} \quad \square$$

4 Problem 8.4

The goal of this problem is to use spectral clustering techniques on real data. The file `adjacency.txt` contains the adjacency matrix of a graph taken from a social network. This graphs has $n = 328$ nodes (that corresponds to users). An edge between user i and user j means that i and j are "friends" in the social network. The notebook `friends.ipynb` contains functions to read the adjacency matrix as well as instructions/questions.

While we focused in the lectures (and in the notes) on the graph Laplacian

$$L = D - A,$$

where A is the adjacency matrix of the graph, and $D = (\deg(1), \ldots, \deg(n))$ is the degree matrix, we will use here the "normalized Laplacian" (instead of L)

$$L_{\text{norm}} = D^{-1/2} L D^{-1/2} =_n -D^{-1/2} A D^{-1/2},$$

where $D^{-1/2} = (\deg(1)^{-1/2}, \ldots, \deg(n)^{-1/2})$. The reason for using a different Laplacian is that then "unnormalized Laplacian" L does not perform well when the degrees in the graph are very broadly distributed, i.e. very heterogeneous. In such situations, the normalized Laplacian L_{norm} is supposed to lead to a more consistent clustering.

It is intended that you code in Python and use the provided Jupyter Notebook. Please only submit a pdf version of your notebook (right-click → 'print' → 'Save as pdf').

In [31]:
```
%matplotlib inline
import matplotlib
import numpy as np
import matplotlib.pyplot as plt

from sklearn.cluster import KMeans
```

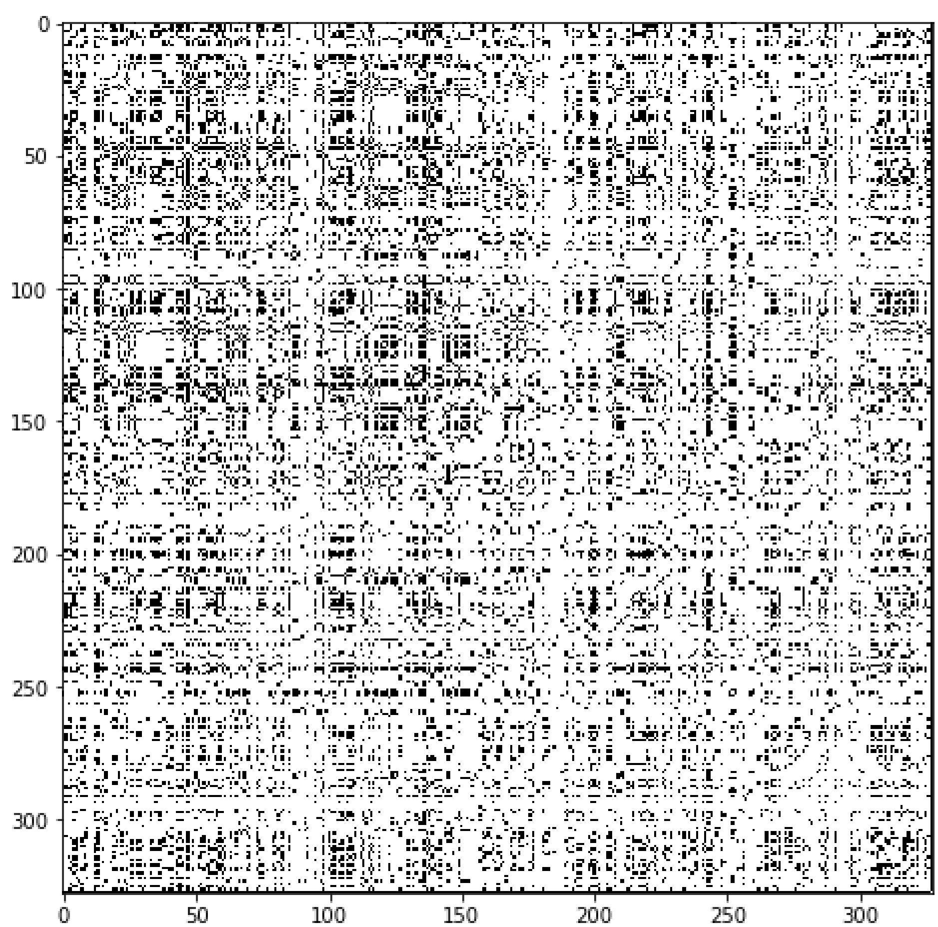
In [32]:
```
# Reads the adjacency matrix from file
A = np.loadtxt('adjacency.txt')
print(f'There are {A.shape[0]} nodes in the graph.')
matrix =A
```

```
There are 328 nodes in the graph.
```

As you can see above, the adjacency matrix is relatively large (328x328): there are 328 persons in the graph. In order to visualize this adjacency matrix, it is convenient to use the 'imshow' function. This plots the 328x328 image where the pixel (i,j) is black if and only if A[i,j]=1.

In [33]:
```
plt.figure(figsize=(8,8))
plt.imshow(A,aspect='equal',cmap='Greys',  interpolation='none')
```

Out[33]: <matplotlib.image.AxesImage at 0x1b5ae0aad60>



(a) Construct in the cell below the degree matrix:

$$D_{i,i} = \deg(i) \qquad \text{and} \qquad D_{i,j} = 0 \text{ if } i \neq j,$$

the Laplacian matrix:

$$L = D - A$$

and the normalized Laplacian matrix:

$$L_{\text{norm}} = D^{-1/2} L D^{-1/2}.$$

In [58]:

```
matrix = np.matrix(matrix)

d = np.zeros(len(matrix))

column_sum = matrix.sum(axis=0)

#Iterate through and change column sum
for j in range(0, len(matrix)):
    d[j] = column_sum[0,j]

#Initialize diagonal matrix
D = np.diag(d)
d = [1/(x**.5) for x in d]

#Get square roots
D_sqrt = np.diag(d)

#Calculate the Laplacian
L = D - A

#Get the L normalized
Lnorm = D_sqrt@L@D_sqrt
```

In [59]:

```
type(Lnorm)
```

Out[59]:

```
numpy.ndarray
```

(b) Using the command 'linalg.eigh' from numpy, compute the eigenvalues and the eigenvectors of L_{norm}.

In [48]:

```
# Get eigenvectors and eigenvalues
eig_val, eig_vec = np.linalg.eigh(Lnorm)
```

In [49]:
```
Lnorm
```

Out[49]:
```
array([[ 1.        ,  0.        ,  0.        , ...,  0.        ,
         0.        , -0.01064251],
       [ 0.        ,  1.        ,  0.        , ...,  0.        ,
         0.        , -0.00606998],
       [ 0.        ,  0.        ,  1.        , ...,  0.        ,
        -0.01628656, -0.00685914],
       ...,
       [ 0.        ,  0.        ,  0.        , ...,  1.        ,
         0.        , -0.00680698],
       [ 0.        ,  0.        , -0.01628656, ...,  0.        ,
         1.        , -0.00726126],
       [-0.01064251, -0.00606998, -0.00685914, ..., -0.00680698,
        -0.00726126,  1.        ]])
```

(c) We would like to cluster the nodes (i.e. the users) in 3 groups. Using the eigenvectors of L_{norm}, assign to each node a point in $\mathbb{R}^2$, exactly as explained in last lecture (also in 'Algorithm 1' of the notes) where you replace L by L_{norm}. Plot these points using the 'scatter' function of matplotlib.

In [50]:
```
# Reduce dimentionality to R2
E = np.zeros((328,2))
```

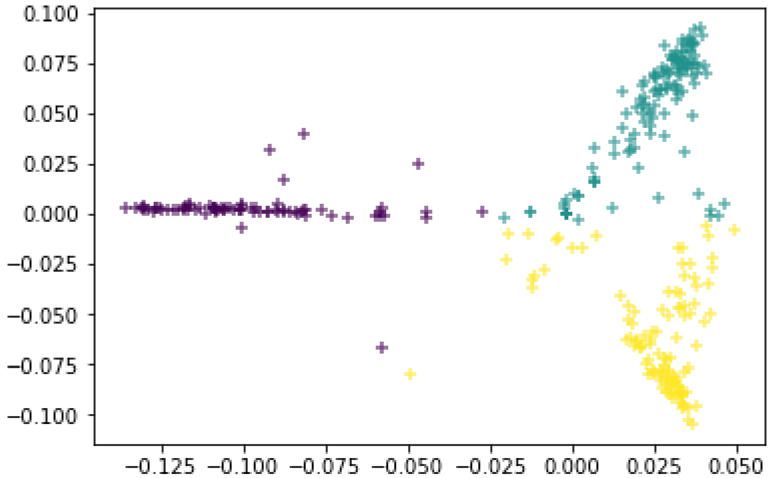
In [51]:
```
#Loop through and assign coordinate values to E
for i in range(len(E)):
    E[i,0] = eig_vec[:,1][i]
    E[i,1] = eig_vec[:,2][i]

E = np.array(E)
```

(d) Using the K-means algorithm (use the built-in function from scikit-learn), cluster the embeddings in $\mathbb{R}^2$ of the nodes in 3 groups.

In [60]:
```
# fit k means
kmeans = KMeans(n_clusters=3, random_state=0).fit(E)
labels=kmeans.labels_
```

In [61]:
```
#Plot scatter plot
plt.scatter(E[:,0], E[:,1], alpha=0.7, marker='+', c = labels)
```

Out[61]: `<matplotlib.collections.PathCollection at 0x1b5af89b5e0>`

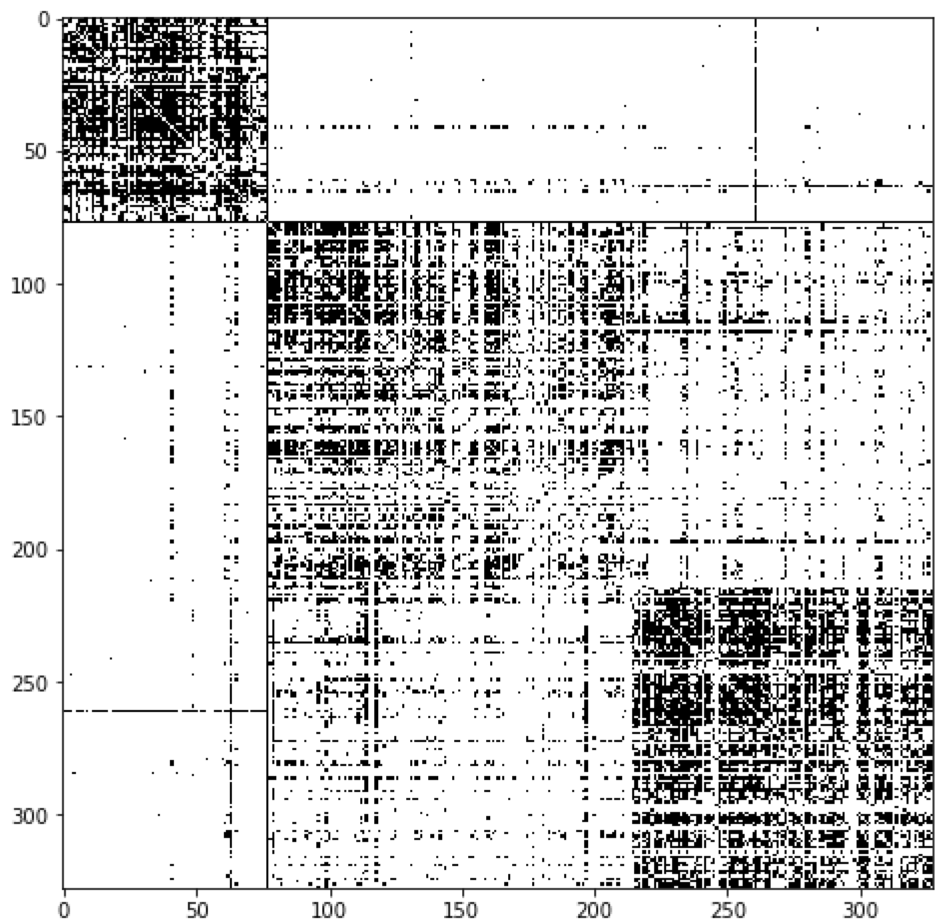


(e) Re-order the adjacency matrix according to the clusters computed in the previous question. That is, reorder the columns and rows of A to obtain a new adjacency matrix (that represents of course the same graph) such that the n_1 nodes of the first cluster correspond to the first n_1 rows/columns, the n_2 nodes of the second cluster correspond to the next n_2 rows/columns, and the n_3 nodes of the third cluster correspond to the last n_3 rows/columns. Plot the reordered adjacency matrix using 'imshow'.

In [62]:
```
## Get the index of each point in each cluster
zero_cluster = np.where(labels==0)
one_cluster = np.where(labels==1)
two_cluster = np.where(labels==2)

#Combine the seperated clusters into one groups
new_list = zero_cluster[0].tolist() + one_cluster[0].tolist() + two_cluster[0

#New matrix A
new_matrix = np.zeros((328,328))


#Take original matrix indices and assign to the regrouped values from new lis
for i in range(0,328):
    for j in range(0,328):
        new_matrix[i,j] = A[new_list[i],new_list[j]]
```

In [63]:
```
#Plot the figure
plt.figure(figsize=(8,8))
plt.imshow(new_matrix,aspect='equal',cmap='Greys',  interpolation='none')
```

Out[63]: <matplotlib.image.AxesImage at 0x1b5adfff5b0>



In [ ]:

In [ ]:

5 Problem 8.5 (Extra Credit)

Let G be a connected graph with n nodes. Define $L \in \mathbb{R}^{n \times n}$ the associate Laplacian matrix and $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ its eigenvalues. Let G' be a graph constructed from G by simply adding an edge. Similarly denote by λ_2' its second smallest eigenvalue. Show that $\lambda_2' \geq \lambda_2$.

In general, if we add an edge connecting two nodes, i and j, in Graph G, (meaning we connect nodes i and j to each other with a single edge), the corresponding entries of the associated Laplacian matrix, L, will decrease by 1 at L_{ij}, L_{ji}, and increase at L_{ii} and L_{jj} by one, as the entries are connected and the Lapliacian is defined as $L = D - A$ where D is the Degree matrix and A is the Adjacency matrix of Graph G'. We call the Laplacian matrix of graph G', L'.

We know from past homeworks that for any square symmetric matrices $A \in \mathbb{R}^{n \times n}$ that the $Trace(A)$ is equal to the sum of its eigenvalues. That is to say: $Trace(A) = Trade(D)$ where D is the diagonal matrix of $A's$ eigenvalues. When we added an edge, we increased the entries of the diagonals, that is to say $Trace(L') > Trace(L)$ and since L and L' are square symmetric matrices, it follows that $Trace(Eigenvalues\ of\ L') > Trace(Eigenvalues\ of\ L)$.

Furthermore, we know that $\lambda_2' \geq \lambda_2$ as when we add an edge it increases the graph's overall connectivity and in turn increase the graph's algebraic connectivity value of (λ_2). If we successfully added an edge between two unconnected nodes, it follows that $\lambda_2' > \lambda_2$, and in the case that the graph G is already fully connected, adding an edge does not increase connectivity thus its algebraic connectivity value remains the same, which would mean $\lambda_2' = \lambda_2$. So in general, when adding an edge to a graph, it follows that $\lambda_2' \geq \lambda_2$.