Problem 8.1.

a) (i) Symmetry. For all $A, B \in \mathbb{R}^{n\times m}$ $\langle A,B\rangle_F = Tr(A^T B) = Tr((A^T B)^T)$

$= Tr(B^T A) = \langle B,A\rangle_F.$

(ii) Linearity. Let $A, B, C \in \mathbb{R}^{n\times m}$, $\alpha \in \mathbb{R}$:

$\langle \alpha A + B, C\rangle_F = Tr((\alpha A+B)^T C)$ by linearity of the trace.

$= Tr(\alpha A^T C + B^T C) = \alpha Tr(A^T C) + Tr(B^T C)$

$= \alpha \langle A, C\rangle_F + \langle B, C\rangle_F.$

(iii) Let $A \in \mathbb{R}^{n\times m}$.

$$\langle A,A\rangle_F = Tr(A^T A) = \sum_{i=1}^{m} (A^T A)_{i,i} = \sum_{i=1}^{m}\sum_{j=1}^{n} A^T_{i,j} A_{j,i}$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} A_{j,i}^2$$

Hence $\langle A,A\rangle_F \geq 0$ with equality if and only if $A_{i,j}^2 = 0$ for all i,j, ie $A=0$.

b) Let $A = U\Sigma V^T$ be the singular value decomposition of A.

$Tr(A^T A) = Tr(V\Sigma^T U^T U \Sigma V^T)$

$= Tr(V\Sigma^T\Sigma V^T)$ because U orthogonal.

$= Tr(V^T V \Sigma^T \Sigma)$ because $Tr(M_1 M_2) = Tr(M_2 M_1)$

$= Tr(\Sigma^T\Sigma)$ because V orthogonal.

$= \sum_{i=1}^{\min(n,m)} \sigma_i^2$

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

Problem 8.2 :

a) Let $A = U\Sigma V^T$ be the SVD of A.
A is $n\times n$, hence Σ is $n\times n$:

$$\Sigma = \begin{pmatrix} \sigma_1 & & (0) \\ & \ddots & \\ (0) & & \sigma_n \end{pmatrix}.$$

U and V^T are invertible (because they are orthogonal)
hence (from what we proved in HW3):
$\mathrm{rank}(A) = \mathrm{rank}(U\Sigma) = \mathrm{rank}(\Sigma)$

We conclude that:
A is invertible $\iff \mathrm{rank}(A) = n$
$\iff \mathrm{rank}(\Sigma) = n$
$\iff$ all the σ_i are non-zero.

b) A is invertible, hence all the σ_i are non-zero : Σ is invertible.

$$A^{-1} = (U\Sigma V^T)^{-1} = (V^T)^{-1}\Sigma^{-1}U^{-1}$$
$$= V\Sigma^{-1}U^T \quad \left(\begin{array}{l}\text{because } U \text{ and } V \text{ are} \\ \text{orthogonal matrices.}\end{array}\right)$$

Hence the singular values of A^{-1} are the diagonal entries of Σ^{-1} : $\sigma_1^{-1}, \dots \sigma_n^{-1}$

The largest singular value of A^{-1} is therefore σ_n^{-1} because $0 < \sigma_n \leq \dots \leq \sigma_1$

We conclude that $\sigma_1(A)\,\sigma_1(A^{-1}) = \frac{\sigma_1}{\sigma_n} \geq 1$.

Problem 8.3

a) There is a path of length 1 from i to j if and only if there is an edge between i and j that is $A_{i,j} = 1$. If not, $A_{i,j} = 0$.
Hence for all i,j, $A_{i,j}$ is the number of path of length 1 from i to j. $\mathcal{H}(1)$ is true.

b) Assume that $\mathcal{H}(k)$ is true for some $k \geq 1$.

Let $i,j \in \{1,2,\dots n\}$.

$$(A^{k+1})_{i,j} = \sum_{l=1}^{n} (A^k)_{i,l}\, A_{l,j} = \sum_{\substack{l=1 \\ l\sim j}}^{n} (A^k)_{i,l}$$

Let $N(j) = \{ l \in \{1 \dots n\} \mid l \sim j \}$ be the set of the neighbours of j.

We can divide the paths of length $k+1$ from i to j in $\# N(j)$ groups; according to the last node $l \in N(j)$ that they "visit" before arriving at j.

For a given neighbor $l \in N(j)$, there are $(A^k)_{i,l}$ paths of length k from i to l (since we assumed that $\mathcal{H}(k)$ holds), hence there are $(A^k)_{i,l}$

paths of length $k+1$ whose second last node is ℓ.

Hence the total number of paths of length $k+1$ from i to j is

$$\sum_{\ell \in N(j)} (A^k)_{i,\ell} = \sum_{\substack{\ell=1 \\ \ell \sim j}}^{n} (A^k)_{i,\ell} = (A^{k+1})_{i,j}$$

We conclude that $\mathcal{H}(k+1)$ is true.

Problem 8.5

Let $M \in \mathbb{R}^{n \times m}$, let $M = U \Sigma V^T$ be the SVD of M.

- (Positive definiteness).

If $\|M\|_* = 0$, then $\sigma_1 = \dots = \sigma_{\min(n,m)} = 0$ hence $\Sigma = 0$ and $M = 0$.

- (Homogeneity)

Let $\lambda \in \mathbb{R}$, let $\delta = \begin{cases} 1 & \text{if } \lambda \geq 0 \\ -1 & \text{if } \lambda < 0 \end{cases}$

By definition of δ: $\lambda = |\lambda| . \delta$.

$$\lambda M = U \cdot |\lambda| \Sigma \cdot (\delta V)^T$$

The matrices U and δV are orthogonal (indeed $(\delta V)^T \delta V = \delta^2 V^T V = Id$), and $|\lambda|\Sigma$ has non-negative entries, ~~outside~~ and zeros outside the diagonal.

Hence $|\lambda| \sigma_1, \dots |\lambda| \sigma_{\min(n,m)}$ are the singular values of λM.

We get $\|\lambda M\|_* = \sum_{i=1}^{\min(n,m)} |\lambda| \sigma_i = |\lambda| \cdot \|M\|_*$

- (Triangular inequality)

We are going to show that

$$\|M\|_* = \max_{\substack{Q \in \mathbb{R}^{n\times m} \\ \sigma_{max}(Q) \leq 1}} Tr(M Q^T)$$

where $\sigma_{max}(Q)$ denotes the maximal singular value of Q.

Let $U\Sigma V^T$ be the SVD of A and define

$Q = U J V^T$

where $J \in \mathbb{R}^{n\times m}$, $J_{i,j} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{otherwise.} \end{cases}$

The largest ~~eigen~~ singular value of Q is 1 : $\sigma_{max}(Q) = 1$.

Compute

$$Tr(MQ^T) = Tr(U\Sigma V^T V J^T U^T) = Tr(U \Sigma J^T U^T)$$

$$= Tr(U^T U \Sigma J^T) = Tr(\Sigma J^T) = \sum_{i=1}^{\min(n,m)} \sigma_i$$

because U and V are orthogonal matrices and $Tr(AB) = Tr(BA)$ for all matrices A, B.

Hence we get that

$$\|M\|_* \leqslant \max_{\substack{Q \in \mathbb{R}^{n \times m} \\ \sigma_{max}(Q) \leq 1}} Tr(MQ^T)$$

Let us prove the converse inequality:
let $Q \in \mathbb{R}^{n \times m}$ such that $\sigma_{max}(Q) \leq 1$

$$Tr(MQ^T) = Tr(M^T Q) = Tr(V \Sigma^T U^T Q)$$
$$= Tr(\Sigma^T U^T Q V)$$
$$= \sum_{i=1}^{\min(n,m)} \sigma_i (U^T Q V)_{i,i}$$

Let $v_1, \dots v_m$ be the columns of V
$u_1 \dots u_n$ ——— of U

We have, for $i \in \{1, \dots \min(n,m)\}$, $(U^T Q V)_{i,i} = u_i^T Q v_i$

Hence we get by Cauchy-Schwarz that

$$|(U^T Q V)_{ii}| = |\langle u_i, Q v_i \rangle| \leqslant \|Q v_i\| \|u_i\| = \|Q v_i\|$$

$\|Q v_i\|^2 = v_i^T Q^T Q v_i$ is less than the largest eigenvalue of $Q^T Q$ (because $\|v_i\| = 1$) which is the square of the largest singular value of Q which is less than 1.
Hence $\|Q v_i\|^2 \leq 1$

We conclude that $(U^T Q V)_{i,i} \leq 1$ and that

$$\mathrm{Tr}(MQ^T) \leq \sum_{i=1}^{\min(n,m)} \sigma_i = \|M\|_*$$

This proves that $\max\limits_{\substack{Q \in \mathbb{R}^{n \times m} \\ \sigma_{max}(Q) \leq 1}} \mathrm{Tr}(MQ^T) \leq \|M\|_*$

Consequently: $\boxed{\|M\|_* = \max\limits_{\substack{Q \in \mathbb{R}^{n \times m} \\ \sigma_{max}(Q) \leq 1}} \mathrm{Tr}(MQ^T)}$

Let now $A, B \in \mathbb{R}^{n \times m}$

$$\|A+B\|_* = \max_{\substack{Q \in \mathbb{R}^{n \times m} \\ \sigma_{max}(Q) \leq 1}} \left(\mathrm{Tr}(AQ^T) + \mathrm{Tr}(BQ^T) \right)$$

$$\leq \max_{\substack{Q \in \mathbb{R}^{n \times m} \\ \sigma_{max}(Q) \leq 1}} \mathrm{Tr}(AQ^T) + \max_{\substack{Q \in \mathbb{R}^{n \times m} \\ \sigma_{max}(Q) \leq 1}} \mathrm{Tr}(BQ^T) = \|A\|_* + \|B\|_*$$

In [2]:
```
%matplotlib inline
import scipy
import matplotlib
import numpy as np
import matplotlib.pyplot as plot
from sklearn.cluster import KMeans
```

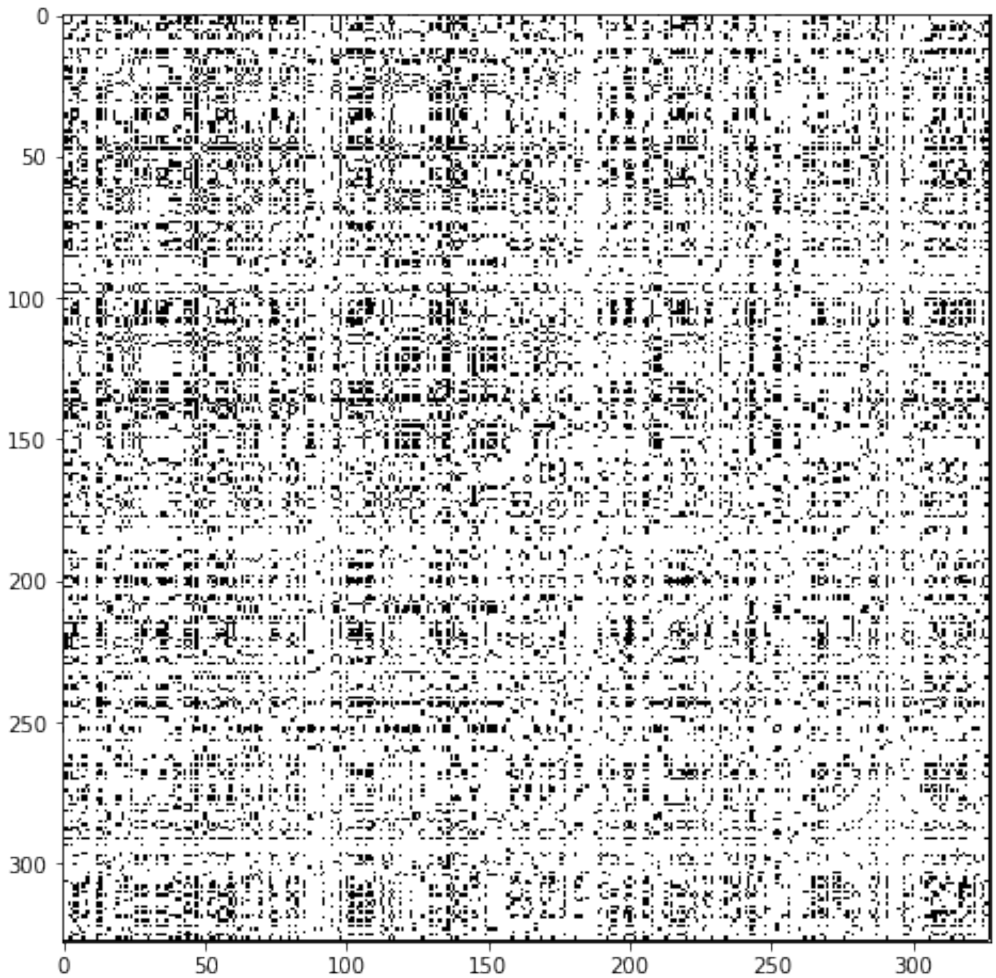
In [3]:
```
# Reads the adjacency matrix from file
A=np.loadtxt('adjacency.txt')
print("There are",A.shape[0],"nodes.")
```

```
There are 328 nodes.
```

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 [4]:
```
plot.figure(figsize=(8,8))
plot.imshow(A,aspect='equal',cmap='Greys',  interpolation='none')
```

Out[4]: <matplotlib.image.AxesImage at 0x127ad6040>



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 [5]:
```
N=A.shape[0]
d = np.sum(A,axis=0)
D=np.diag(d)
D2=np.diag(1/np.sqrt(d))
L= np.identity(N) - D2 @ A @ D2
```

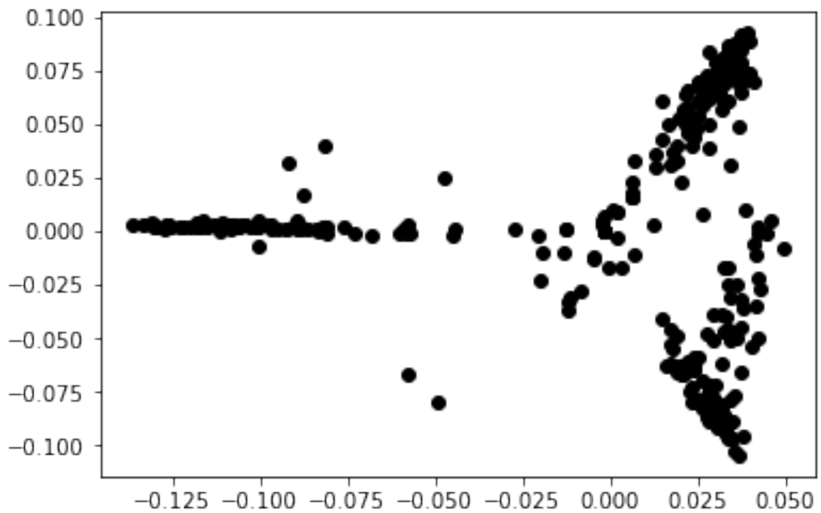
b) Using the command 'linalg.eigh' from numpy, compute the eigenvalues and the eigenvectors of L_{norm}.

In [6]:
```python
v,w = np.linalg.eigh(L)
```

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 R^2, exactly as in 'Algorithm 1' of the notes where you replace L by L_{norm}. Plot these points using the 'scatter' function of matplotlib.

In [7]:
```python
X=w[:,1:3]
plot.scatter(X[:,0],X[:,1], color='black')
```

Out[7]: <matplotlib.collections.PathCollection at 0x127dbed30>

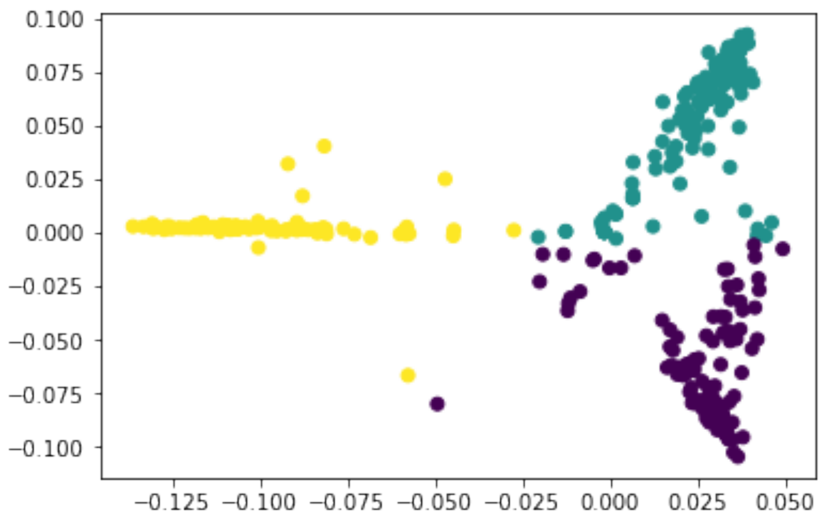


d) Using the K-means algorithm (use the built-in function from scikit-learn)

In [8]:
```python
# Replace ??? by the matrix of the points computed in (c)
kmeans = KMeans(n_clusters=3, random_state=0).fit(X)
labels=kmeans.labels_
# labels contains the membership of each node
```

In [9]:
```python
# Color the points accorind to their labels, to check that kmeans work
plot.scatter(X[:,0],X[:,1],c=labels)
```

Out[9]:
```
<matplotlib.collections.PathCollection at 0x127b228b0>
```

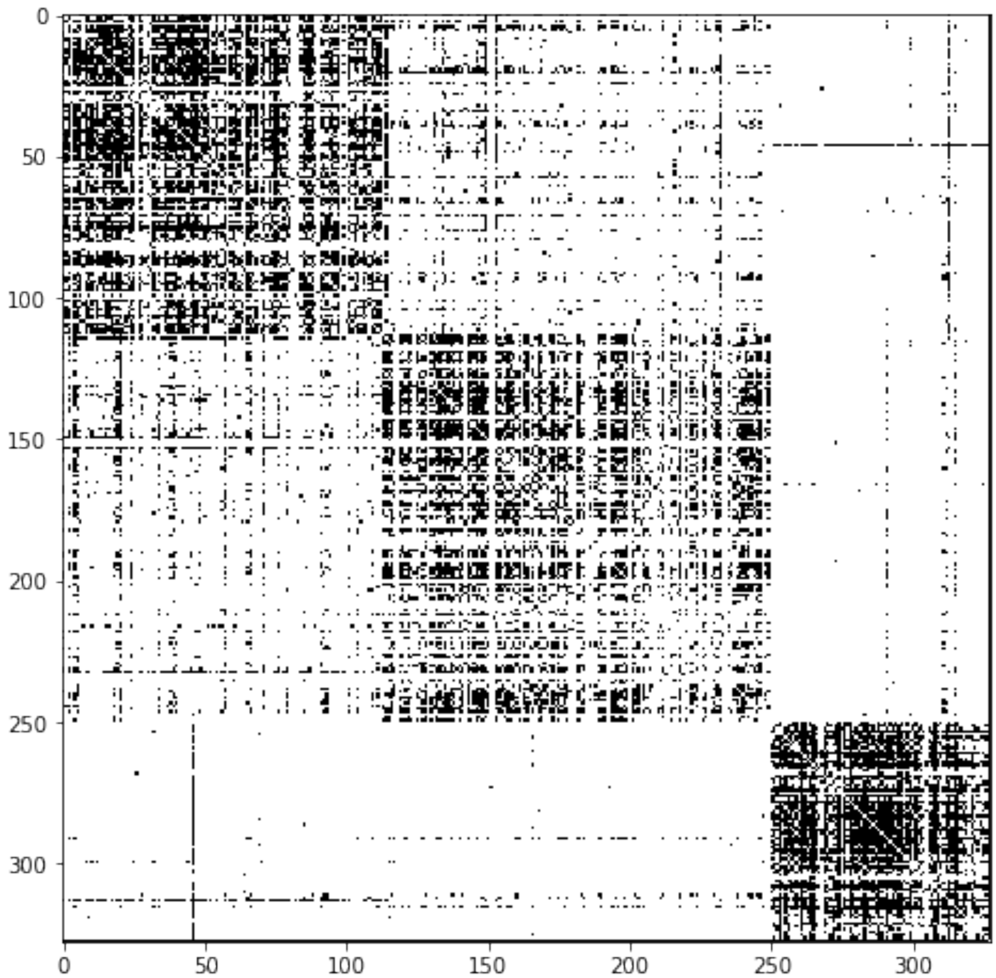


e) Re-order the adjacency matrix according to the clusters computed in the previous question. That is reorder the columns and rows of A to obtain a new adjacency matrix (that represent of course the same graph) such that the n_1 nodes of the first cluster correspond to the first n_1 rows/columns, the 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 [10]:
```python
nodes = np.arange(N)
cluster1 = nodes[labels==0]
cluster2 = nodes[labels==1]
cluster3 = nodes[labels==2]
permutation = np.concatenate((cluster1, cluster2, cluster3))
rA=np.zeros((N,N))
for i in range(N):
    for j in range(N):
        rA[i,j]= A[permutation[i],permutation[j]]
```

In [11]:
```
plot.figure(figsize=(8,8))
plot.imshow(rA,aspect='equal',cmap='Greys',  interpolation='none')
```

Out[11]: <matplotlib.image.AxesImage at 0x127ba3d60>



In [ ]: