Homework 2

Nate Mankovich
Math Modeling

February 24, 2018

Part A: Theory and Foundations

Problem 1.

Consider the optimization problem

$$\text{maximize } u^T Av$$

subject to the constraints

$$||u||_2 = ||v_2|| = 1$$

Show that a necessary condition for the solutions is given by the SVD equations.

$$\begin{aligned}
\mathcal{L} &= u^T Av = \mu(u^T u - 1) - \lambda(v^T v - 1) \\
&= \sum_{i=1}^{n} u_i \sum_{j=1}^{n} a_{ij} v_j - \mu(\sum_{i=12}^{n} u_i - 1) - \lambda(\sum_{i=1}^{n} v_i^2 - 1) \\
\text{Then for all } i \leq n,\ \frac{\partial \mathcal{L}}{\partial u_i} &= \sum_{j=1}^{n} a_{ij} v_j - \mu 2 u_i = 0 \\
\text{and for all } j \leq n,\ \frac{\partial \mathcal{L}}{\partial v_j} &= \sum_{i=1}^{n} a_{ij} u_i - \lambda 2 v_j = 0. \\
\frac{\partial \mathcal{L}}{\partial \mu} &= u^T u - 1 = 0 \iff ||u|| = 1 \\
\frac{\partial \mathcal{L}}{\partial \lambda} &= v^T v - 1 = 0 \iff ||v|| = 1
\end{aligned}$$

Now we can do some simplification of the partials with respect to u_i and v_j:

$$\sum_{j=1}^{n} a_{ij} v_j = \mu 2 u_i \iff Av = 2\mu u \iff A = u(2\mu)v^T$$

$$\sum_{i=1}^{n} a_{ij} u_i = \lambda 2 v_j \iff Au = 2\lambda v \iff A = v(2\lambda)u^T$$

Since $A = M^T M$ we have $A^T = A$. This gives us $u = v$. Also, notice this is indeed the vector version of the SVD for A where the singular values are 2μ. Also, notice u, v are both normal.

Problem 2.

Let $H = I - ee^T/n$ where e is the column vector of n ones. Let A be an $n \times n$ matrix where a_{ij} is the entry in the ith row, jth column. Also $A = -\frac{1}{2}\Delta$ where Δ is the squared distance matrix.

a) Show that $\frac{1}{n}ee^T$ is a projection and hence H is a complementary projection.

$$\left(\left(\frac{1}{n}ee^T\right)^2\right)_{ij} = \left(\frac{1}{n}e^T e\frac{1}{n}e^T e\right)_{ij} = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{n} = \frac{1}{n} = \left(\frac{1}{n}ee^T\right)_{ij} \implies \left(\frac{1}{n}ee^T\right)^2 = \frac{1}{n}ee^T$$

Therefore $\frac{1}{n}ee^T$ is a projection and $H = 1 - \frac{1}{n}ee^T$ is a complementary projection by definition.

b) Provide a geometric interpretation of the action of each of the projections in a).
This is a projection of a given vector onto span(e). It effectively reduces m patterns living in n dimensional space into m patterns living in 1 dimensional space.

c) What is the result of applying H to A from the left, i.e., $B = HA$. Show explicitly.

$$HA = (I - \frac{1}{n}ee^T)A = A - \frac{1}{n}ee^T A \implies (HA)_{ij} = a_{ij} - \overline{a_{.j}}$$

d) What is the result of applying H to A from the right, i.e., $B = AH$. Show explicitly.

$$AH = A(I - \frac{1}{n}ee^T) = A - A\frac{1}{n}ee^T \implies (AH)_{ij} = a_{ij} - \overline{a_{i.}}$$

e) Let $B = HAH$. Does multiplying HA by H from the right undo the result described in c)? Give a mathematical argument.
Suppose, by way of contradiction, multiplying HA by H from the right undo the result described in c). Then $HAH = I$. But Problem 3 (see below) shows that

$$HAHe = 0 \neq e \implies HAH \neq I$$

Problem 3.

Show that the vector of ones is an eigenvector of B. What is the associated eigenvalue?

$$\begin{aligned}
Be &= HAHe \\
&= H\begin{bmatrix} a_{11} - \overline{a_{.1}} & a_{12} - \overline{a_{.2}} & \dots & a_{1n} - \overline{a_{.n}} \\ a_{21} - \overline{a_{.1}} & a_{22} - \overline{a_{.2}} & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ a_{n1} - \overline{a_{.1}} & a_{n2} - \overline{a_{.2}} & \dots & a_{nn} - \overline{a_{.n}} \end{bmatrix} e \\
&= \begin{bmatrix} 1 - 1/n & \dots & 1 - 1/n \\ 1 - 1/n & \dots & . \\ . & \dots & . \\ . & \dots & . \\ . & \dots & . \\ 1 - 1/n & \dots & 1 - 1/n \end{bmatrix} \begin{bmatrix} \sum_{j=1}^{n}(a_{1j} - \overline{a_{.j}}) \\ \sum_{j=1}^{n}(a_{2j} - \overline{a_{.j}}) \\ . \\ . \\ . \\ \sum_{j=1}^{n}(a_{nj} - \overline{a_{.j}}) \end{bmatrix} \\
&= \begin{bmatrix} \sum_{k=1}^{n}(\sum_{j=1}^{n}(a_{kj} - \overline{a_{.j}}))(1 - 1/n)) \\ . \\ . \\ . \end{bmatrix} \\
&= \sum_{k=1}^{n}(\sum_{j=1}^{n}(a_{kj} - \overline{a_{.j}}))(1 - 1/n))e
\end{aligned}$$

So the associated eigenvalue is

$$\begin{aligned}
\lambda_e &= \sum_{k=1}^{n}\Big(\sum_{j=1}^{n}(a_{kj}-\overline{a_{\cdot j}})(1-1/n)\Big)\\
&= \sum_{k=1}^{n}\Big(\sum_{j=1}^{n}(a_{kj}-\overline{a_{\cdot j}}-1/n(a_{kj}-\overline{a_{\cdot j}}))\Big)\\
&= \sum_{k=1}^{n}\Big(\sum_{j=1}^{n}(a_{kj}-\frac{1}{n}a_{kj})\Big)+\sum_{k=1}^{n}\Big(\sum_{j=1}^{n}(\frac{1}{n}\overline{a_{\cdot j}}-\overline{a_{\cdot j}})\Big)\\
&= \sum_{k=1}^{n}\sum_{j=1}^{n}a_{kj}-\sum_{k=1}^{n}\sum_{j=1}^{n}\frac{1}{n}a_{kj}+\sum_{k=1}^{n}\sum_{j=1}^{n}\frac{1}{n}\overline{a_{\cdot j}}-\sum_{k=1}^{n}\sum_{j=1}^{n}\overline{a_{\cdot j}}\\
&= n^2\overline{a_{\cdot\cdot}}-n\overline{a_{\cdot\cdot}}+n\overline{a_{\cdot\cdot}}-n^2\overline{a_{\cdot\cdot}}\\
&= 0
\end{aligned}$$

Problem 4.

Show that the columns of $\tilde{V}$ where

$$B=\tilde{V}\tilde{V}^T$$

have a property

$$\tilde{V}^Te=0$$

where e is the vector of ones.

We know from Problem 3 that $\tilde{V}\tilde{V}^Te=0e=0$.

$$0=\langle 0,e\rangle=\langle\tilde{V}\tilde{V}^Te,e\rangle=\langle\tilde{V}^Te,\tilde{V}^Te\rangle=||\tilde{V}^Te||$$

We know $||x||=0 \iff x=0$. So we have proven $\tilde{V}^Te=0$.

Problem 5.

Using the result from Problem 4., show that the mean value of the configuration of points coming from MDS is the origin.

We know the rows of $\tilde{V}$ are the points coming from MDS. So the mean of the points coming from MDS would be the mean of the rows of $\tilde{V}$. This is the same as the mean of the columns of $\tilde{V}^T$. To calculate this mean we can simply do $\frac{1}{n}\tilde{V}^Te=\frac{1}{n}0=0$. Therefore the mean value of the configuration of points coming from MDS is the origin.

Part B: Computing

Problem 1.

Download the city data consisting of pairwise distances from Canvas. Make a map using MDS to determine a configuration of points in 2-dimensions. Verify that the mean of the configuration is zero. What can you conclude from the eigenvalues of B?

I used MATLAB to verify that the mean of the configuration is zero. Since there are 8 non-negative eigenvalues, the dimension of our reconstruction is 8. We can also see that 0 is an eigenvalue of B associated with the eigenvector $.2877e$ is what we expect from the theory and foundations portion of the homework. Since $B.2877e = 0 \iff Be = 0$. This is another way of saying $.2877e \in \text{span}\{e\} = E_0$.
Below is a map using MDS to determine a configuration of points in 2-dimensions (the first two columns of $\tilde{V}$).

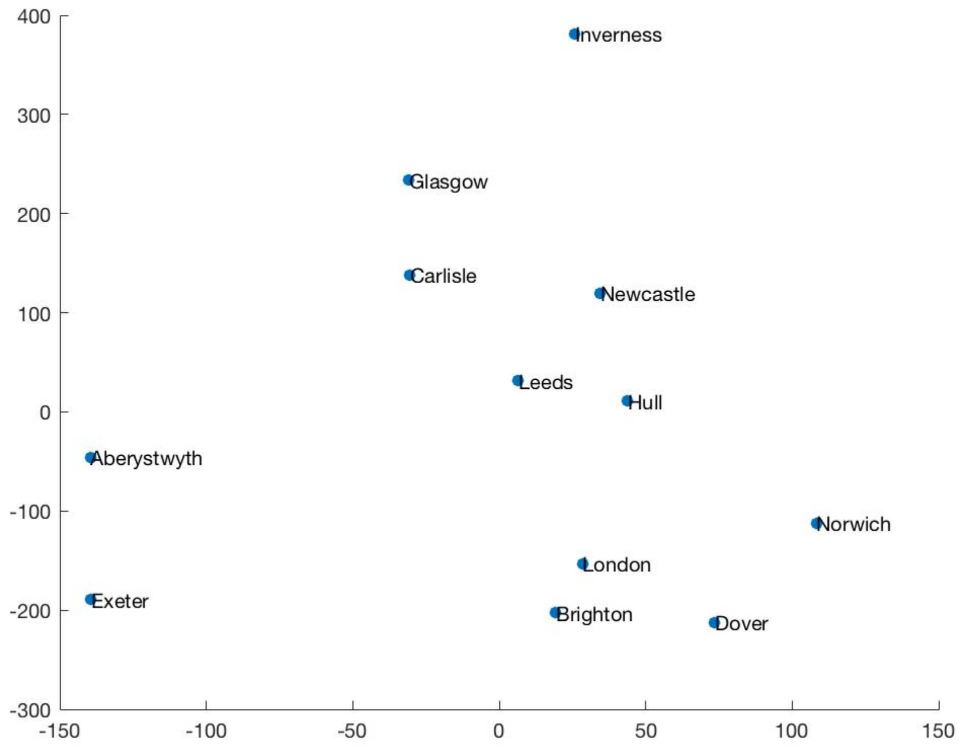

Figure 1: The MDS reconstruction of the configuration.

Below is the MATLAB code used for running the MDS algorithm.

```
M = [0 244 218 284 197 312 215 469 166 212 253 270;
    0 0 350 77 167 444 221 583 242 53 325 168;
    0 0 0 369 347 94 150 251 116 298 57 284;
    0 0 0 0 242 463 263 598 257 72 340 164;
    0 0 0 0 0 441 279 598 269 170 359 277;
    0 0 0 0 0 0 245 169 210 392 143 378;
    0 0 0 0 0 0 0 380 55 168 117 143;
    0 0 0 0 0 0 0 0 349 531 264 514;
    0 0 0 0 0 0 0 0 0 190 91 173;
    0 0 0 0 0 0 0 0 0 0 273 111;
    0 0 0 0 0 0 0 0 0 0 0 256;
    0 0 0 0 0 0 0 0 0 0 0 0]

D = M+M';

%make p the rank of D
p = rank(D);

%make A
A = (-1/2)*D.*D;

%make H
H = eye(12)-(1/p)*ones(12);

%compute B
B = H*A*H;

%eigenvectors and eigenvalues
[V,L] = eig(B);
dL = diag(L);

%throw away negative eigenvalues
i=1
for k=1:p
    if dL(k) >=0
        fixedL(i) = L(k,k);
        fixedV(:,i) = V(:,k);
        i = i+1
    end
end

%vsquiggle
SfixedL = sqrt(fixedL);
for i=1:9
    vSquiggle(:,i) = SfixedL(i)*fixedV(:,i);
end

%rotate and reflect vSquiggle to the proper orientaion
X = [-vSquiggle(:,2),vSquiggle(:,1)]

%scale X
scatter(X(:,1),X(:,2),'filled')
```

```
%label the points
text(X(1,1),X(1,2),'Aberystwyth');
text(X(2,1),X(2,2),'Brighton');
text(X(3,1),X(3,2),'Carlisle');
text(X(4,1),X(4,2),'Dover');
text(X(5,1),X(5,2),'Exeter');
text(X(6,1),X(6,2),'Glasgow');
text(X(7,1),X(7,2),'Hull');
text(X(8,1),X(8,2),'Inverness');
text(X(9,1),X(9,2),'Leeds');
text(X(10,1),X(10,2),'London');
text(X(11,1),X(11,2),'Newcastle');
text(X(12,1),X(12,2),'Norwich');


%Verify that the mean of the configuration is zero
mean(vSquiggle)
```

Problem 2.

Add a new UK city after the fact and put it on your map in red. Comment on the quality of your map both visually and quantitatvely, by comparing the original distance matrix with the one resulting from the one computed from your configuration.

We removed London from the distance matrix. Then we calculate classical MDS on the remaining eleven cities. Then we use distance-based triangulation to compute embedding coordinates for London. From examining the two output maps, it appears Landmark MDS produces results equally as accurate as classical MDS.
On closer examination we can see the coordinates for London using Landmark MDS are $34.1734, -167.0011$, whereas its coordinates using classical MDS are $28.4926, -153.0790$.

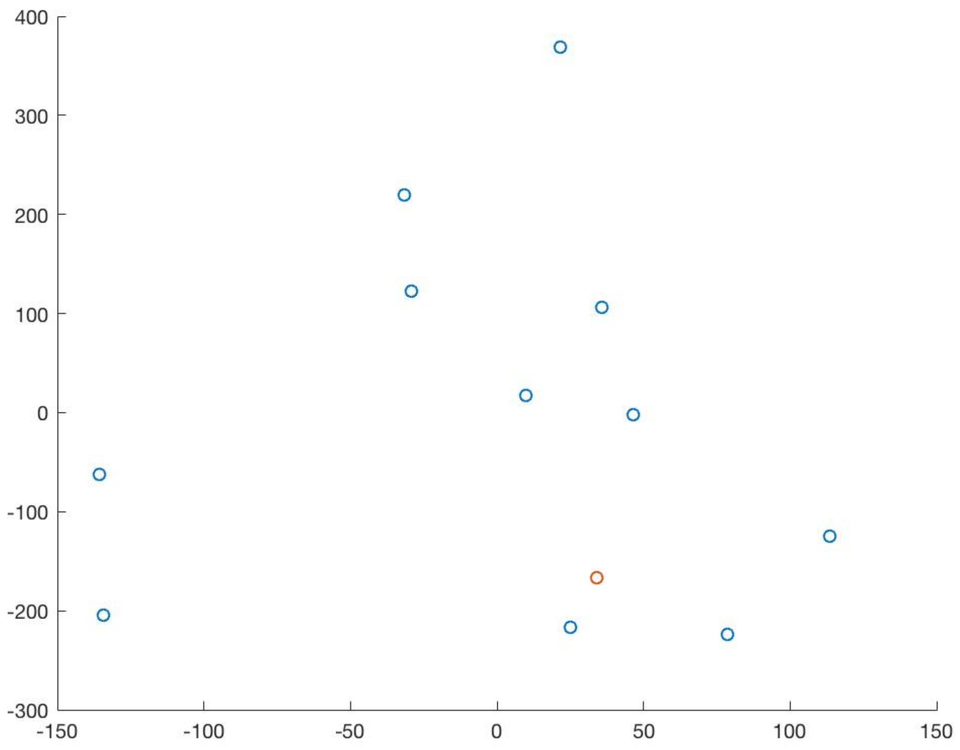

Figure 2: Adding London to a configuration using the Landmark MDS algorithm. The eleven landmark cities are in blue and London is in red.

```
%Aberystwyth, Brighton, Carlisle, Dover, Exeter, Glasgow,
%Hull, Inverness, Leeds, Newcastle upon Tyne, Norwich

M = [0 244 218 284 197 312 215 469 166 253 270;
    0 0 350 77 167 444 221 583 242 325 168;
    0 0 0 369 347 94 150 251 116 57 284;
    0 0 0 0 242 463 263 598 257 340 164;
    0 0 0 0 0 441 279 598 269 359 277;
    0 0 0 0 0 0 245 169 210 143 378;
    0 0 0 0 0 0 0 380 55 117 143;
    0 0 0 0 0 0 0 0 349 264 514;
    0 0 0 0 0 0 0 0 0 91 173;
    0 0 0 0 0 0 0 0 0 0 256;
    0 0 0 0 0 0 0 0 0 0 0]

D = M+M';

%new city is London
ya = [212; 53; 298; 72; 170; 392; 168; 531; 190; 273; 111]

%make p the rank of D
p = rank(D);

%make A
A = (-1/2)*D.*D;

%make H
H = eye(p)-(1/p)*ones(p);

%compute B
B = H*A*H;

%eigenvectors and eigenvalues
[V,L] = eig(B);
dL = diag(L);

%throw away negative eigenvalues
i=1
for k=1:p
    if dL(k) >=0
        fixedL(i) = L(k,k);
        fixedV(:,i) = V(:,k);
        i = i+1
    end
end

%vsquiggle
SfixedL = sqrt(fixedL);
for i=1:8
    vSquiggle(:,i) = SfixedL(i)*fixedV(:,i);
end

%calculate delta_a
da = ya.*ya;

%calculate delta_mu
dmu = mean(D.*D,2);

%make transpose of pseudoinverse of Vsquiggle (aka Lsharp)
for i=1:8
    Lsharp(i,:) = fixedV(:,i)'/SfixedL(i);
end

%reconstruct ya
```

```
xa = (-1/2)*Lsharp*(da-dmu);

%test mutha fluka
for i=1:11
    X(:,i) = Lsharp*(B(:,i)-dmu);
end

%plot of data (in proper orientation)
scatter(-vSquiggle(:,2),-vSquiggle(:,1));
hold
scatter(-xa(2),-xa(1));
```