MATH444 Midterm

Jiasen Zhang

1 Problem 1

Before processing the dataset ”cities.mat”, notice that the ratings of different attributes have different orders of magnitudes. For example, the average rating of ”climate” is about 500 while the average rating of ”housing” is about 8000. So instead of centering the data, I normalize the data so that the mean value is 0 and the standard variance is 1.

Computing the first two principal components of the normalized data and plot the absolute values of them in Figure 1, we can clearly indentify the largest components. The first two singular values are 33.5 and 20.0, that means the first PCA vector plays much more important role than the second one. So we focus on the first PCA vector.

According to Figure 1, the two largest components of the first PCA vetor correspond to the attributes ”health” and ”arts”. So we can conclude that health and arts are the predominant factor when differentiating the quality of life in the data.

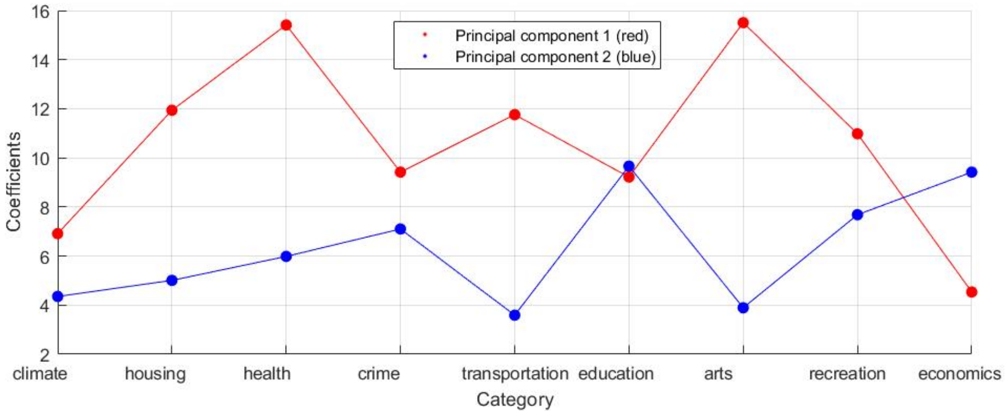


Figure 1: Absolute value of PCA components

2 Problem 2

To check if there are any natural clusters, I plot the first two principal components of the transpose of the data. As shown in Figure 2, there is a natural cluster: most data points tend to cluster together on the left. In fact, the data points corresponding to cities with lower ratings are closer to the left dense area. The data point at the rightmost corresponds to New York City, which has the highest average rating.

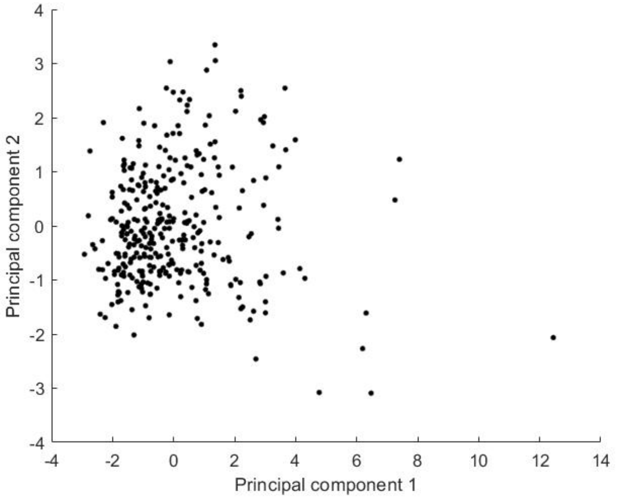
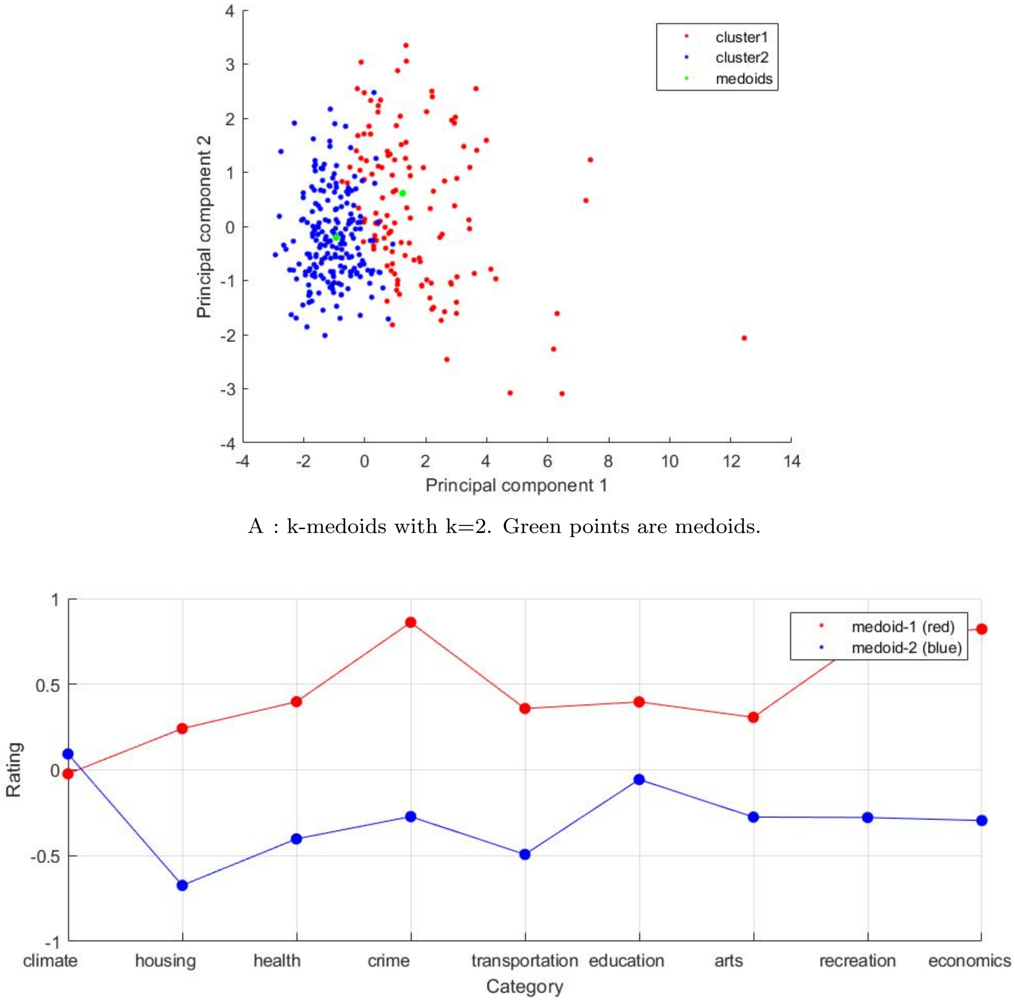


Figure 2: PCA of cities.mat

Running k-medoids with k=2, the results are:

	color	medoid city
cluster 1	red	Phoenix, AZ
cluster 2	blue	Evansville, IN-KY

As shown in Figure 3 A, when k=2, there are two clusters: red and blue. As we mentioned above, the red cluster should have higher ratings generally because it's farther from the dense area. In Figure 3 B, as we expected, the rating of medoid 1 corresponding to "Phoenix, AZ" has higher overall rating than the rating of medoid 2 corresponding to the less famous "Evansville, IN-KY".



A : k-medoids with k=2. Green points are medoids.

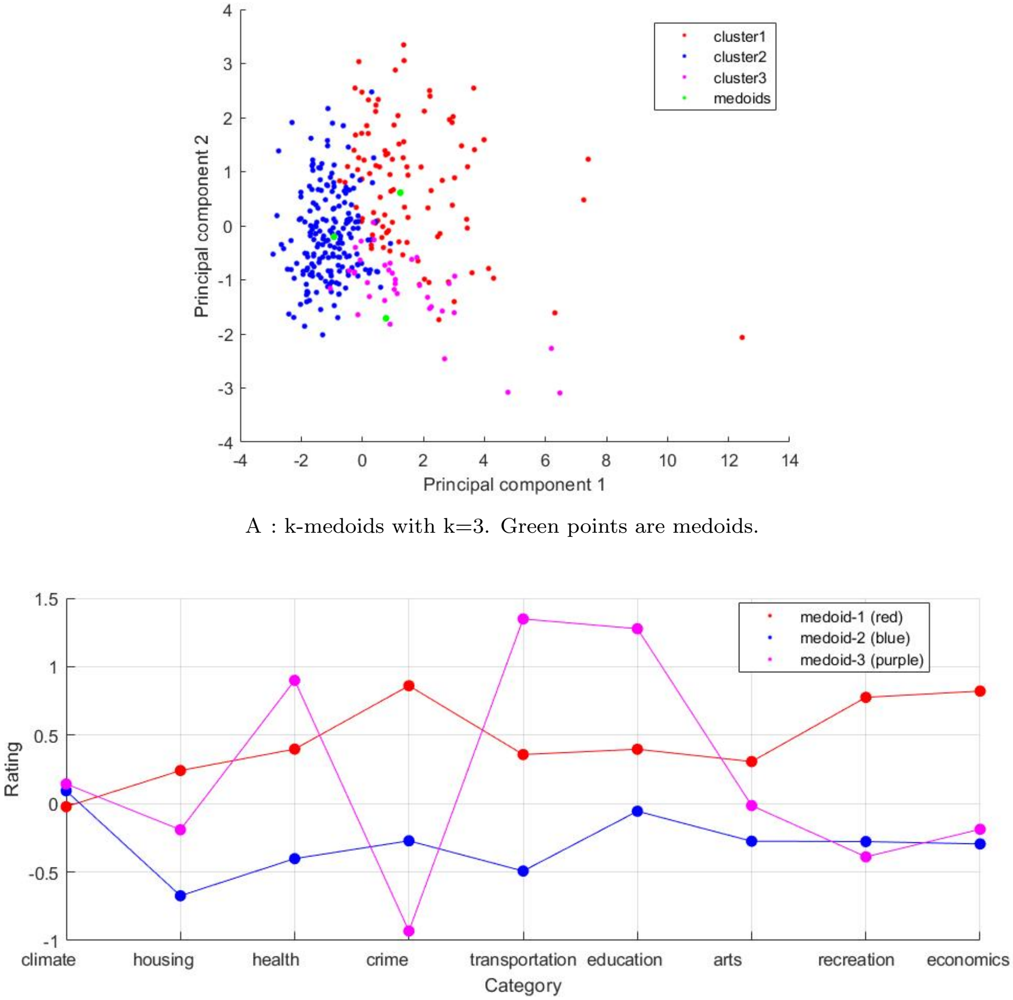
B : ratings of the medoids

Figure 3

Running k-medoids with k=3, the results are:

	color	medoid city
cluster 1	red	Phoenix, AZ
cluster 2	blue	Evansville, IN-KY
cluster 3	purple	Harrisburg-Lebanon-Carlisle, PA

As shown in Figure 4 A, when k=3, the third purple cluster is between the blue and red clusters. In Figure 4 B, most attribute ratings of purple cluster is lower than red cluster and higher than blue cluster. Generally the ratings of purple cluster is between the red and blue clusters, which is corresponding to the result in Figure 4 A.



A : k-medoids with k=3. Green points are medoids.

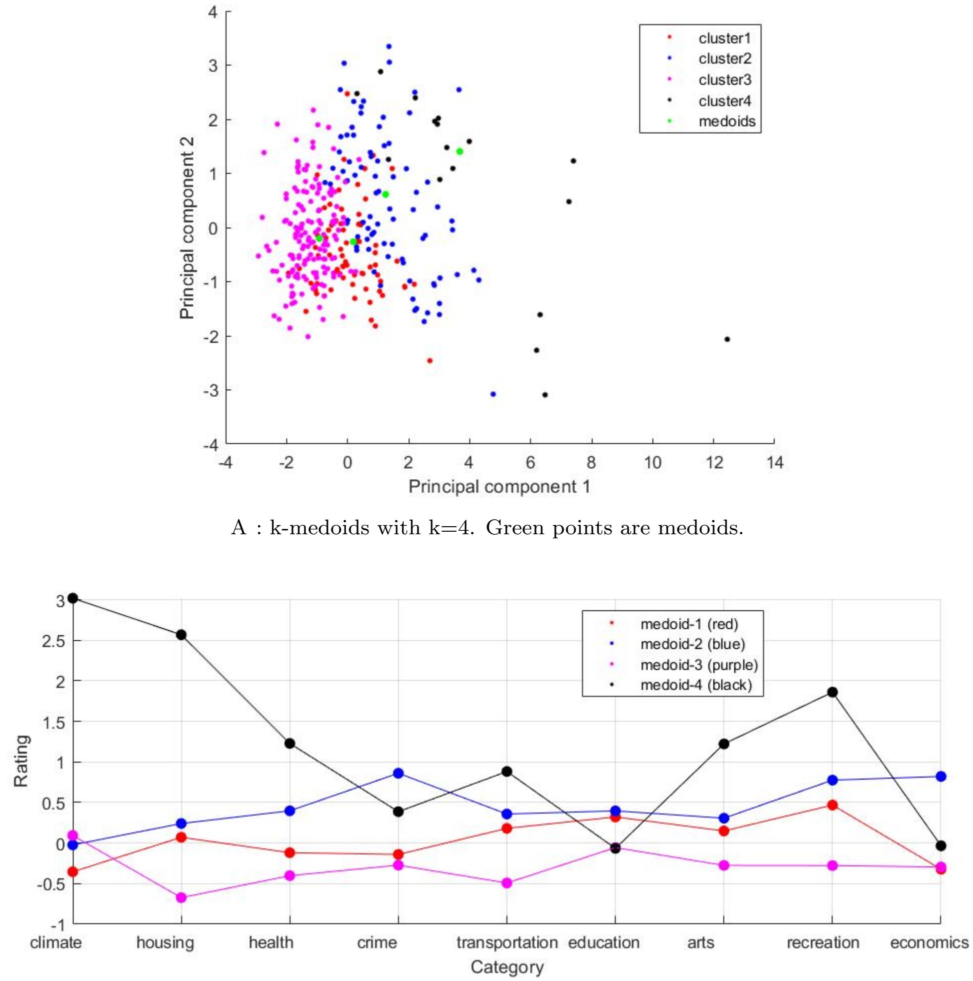
B : ratings of the medoids

Figure 4

Running k-medoids with k=4, the results are:

	color	medoid city
cluster 1	red	Kenosha, WI
cluster 2	blue	Phoenix, AZ
cluster 3	purple	Evansville, IN-KY
cluster 4	black	San Diego, CA

With the results above, in Figure 5 A we can easily know the four clusters from low to high ratings are purple, red, blue and black, and it's proved by Figure 5 B. The result is reasonable: we know "Kenosha, WI" and "Evansville, IN-KY" are not famous and therefore shouldn't have high ratings. Both "San Diego, CA" and "Phoenix, AZ" are famous and generally Dan Diego is more popular than Phoenix mainly for better climate.



A : k-medoids with k=4. Green points are medoids.

B : ratings of the medoids

Figure 5

3 Problem 3

When setting the rank as 2, for the most frequent result, the two cities that are closest to the feature vectors are Pheonix and Janesville. I think they are representative. As shown in Figure 6, Pheonix is better than Janesville in all attributes, it can represent the big cities with high quality of life. Janesville is a small city in the Wisconsin State with bad climate. It represents small and remote cities with low quality of life.

	color	city
city 1	red	Phoenix, AZ
city 2	blue	Janesville-Beloit, WI

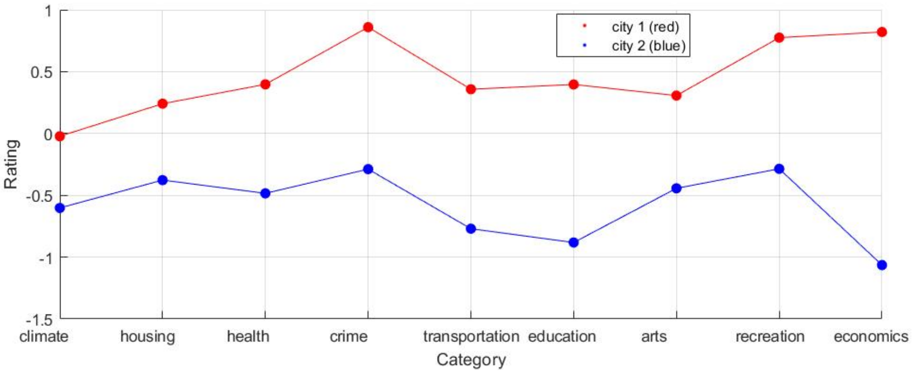


Figure 6: NMF: rank = 2

When the rank increases, the situation will be complicated. The result cities will represent different levels of quality of life. For example, when the rank is 3, the three cities are Phoenix, Sheboygan and Birmingham. According to Figure 7, the cities from high to low levels are Phoenix, Birmingham and Sheboygan.

	color	city
city 1	red	Phoenix, AZ
city 2	blue	Sheboygan, WI
city 3	purple	Birmingham, AL

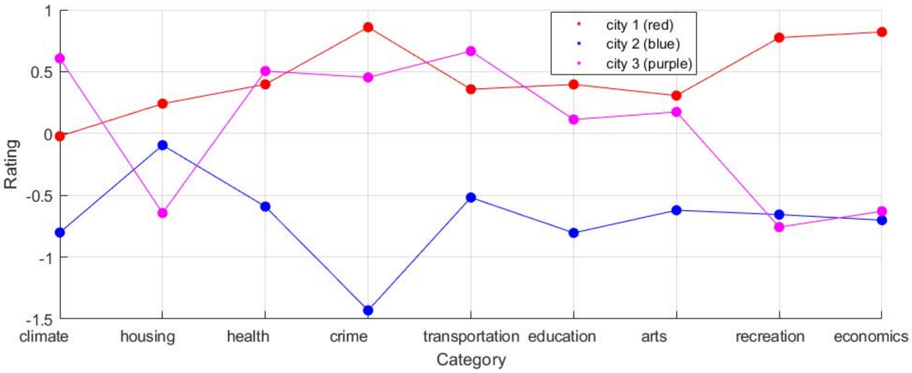
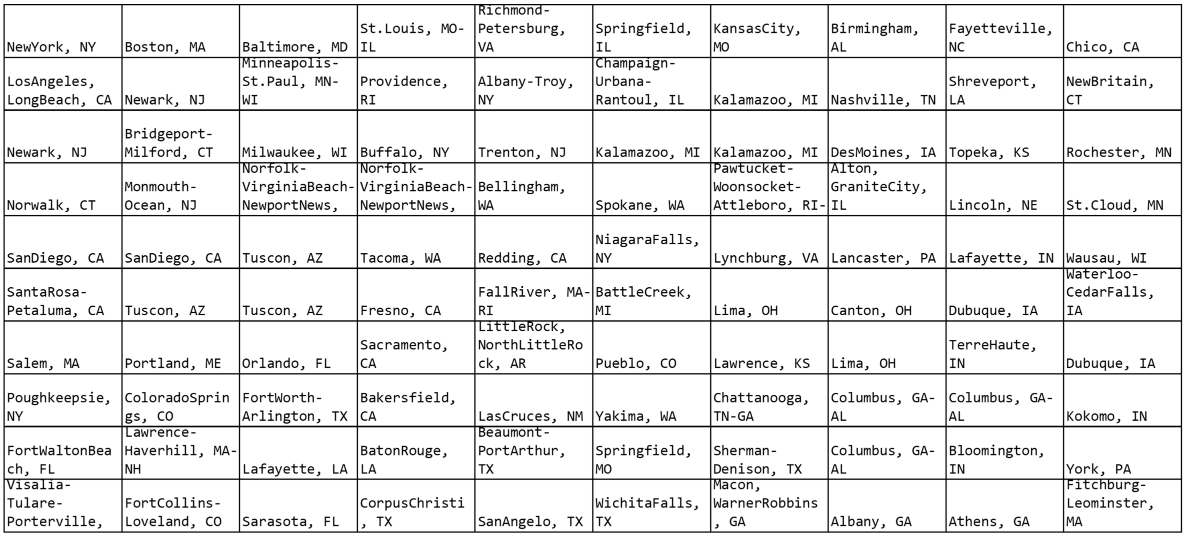


Figure 7: NMF: rank = 3

4 Problem 4

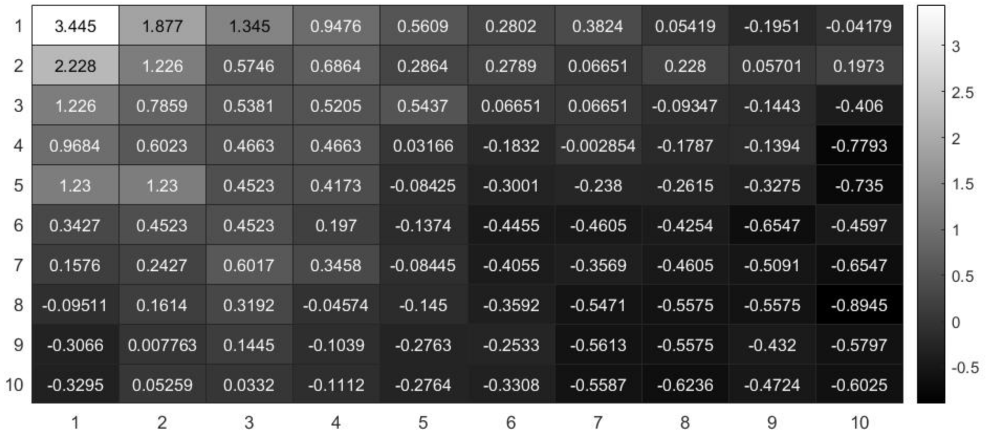
I run SOM algorithm with a 10 by 10 lattice and get 100 prototypes, then put the name of the cities that are closest to the prototypes in a 10 by 10 lattice. In addition, to help illustrate I compute the average ratings of the cities and put them in a lattice. The average rating can partly reflect the overall quality of a city.

The result in Figure 8 looks reasonable. As we can see, the cities on the upper left are big cities with higher ratings and quality of life. The lower right cities are less famous with lower quality of life. From upper left to lower right, the average ratings of cities decrease.



NewYork, NY	Boston, MA	Baltimore, MD	St.Louis, MO-IL	Richmond-Petersburg, VA	Springfield, IL	KansasCity, MO	Birmingham, AL	Fayetteville, NC	Chico, CA
LosAngeles, LongBeach, CA	Newark, NJ	Minneapolis-St.Paul, MN-WI	Providence, RI	Albany-Troy, NY	Champaign-Urbana-Rantoul, IL	Kalamazoo, MI	Nashville, TN	Shreveport, LA	NewBritain, CT
Newark, NJ	Bridgeport-Milford, CT	Milwaukee, WI	Buffalo, NY	Trenton, NJ	Kalamazoo, MI	Kalamazoo, MI	DesMoines, IA	Topeka, KS	Rochester, MN
Norwalk, CT	Monmouth-Ocean, NJ	Norfolk-VirginiaBeach-NewportNews,	Norfolk-VirginiaBeach-NewportNews,	Bellingham, WA	Spokane, WA	Pawtucket-Woonsocket-Attleboro, RI-	Alton, GraniteCity, IL	Lincoln, NE	St.Cloud, MN
SanDiego, CA	SanDiego, CA	Tuscon, AZ	Tacoma, WA	Redding, CA	NiagaraFalls, NY	Lynchburg, VA	Lancaster, PA	Lafayette, IN	Wausau, WI
SantaRosa-Petaluma, CA	Tuscon, AZ	Tuscon, AZ	Fresno, CA	FallRiver, MA-RI	BattleCreek, MI	Lima, OH	Canton, OH	Dubuque, IA	Waterloo-CedarFalls, IA
Salem, MA	Portland, ME	Orlando, FL	Sacramento, CA	LittleRock, NorthLittleRock, AR	Pueblo, CO	Lawrence, KS	Lima, OH	TerreHaute, IN	Dubuque, IA
Poughkeepsie, NY	ColoradoSprings, CO	FortWorth-Arlington, TX	Bakersfield, CA	LasCruces, NM	Yakima, WA	Chattanooga, TN-GA	Columbus, GA-AL	Columbus, GA-AL	Kokomo, IN
FortWaltonBeach, FL	Lawrence-Haverhill, MA-NH	Lafayette, LA	BatonRouge, LA	Beaumont-PortArthur, TX	Springfield, MO	Sherman-Denison, TX	Columbus, GA-AL	Bloomington, IN	York, PA
Visalia-Tulare-Porterville,	FortCollins-Loveland, CO	Sarasota, FL	CorpusChristi, TX	SanAngelo, TX	WichitaFalls, TX	Macon, WarnerRobbins, GA	Albany, GA	Athens, GA	Fitchburg-Leominster, MA

A : name of the cities



	1	2	3	4	5	6	7	8	9	10
1	3.445	1.877	1.345	0.9476	0.5609	0.2802	0.3824	0.05419	-0.1951	-0.04179
2	2.228	1.226	0.5746	0.6864	0.2864	0.2789	0.06651	0.228	0.05701	0.1973
3	1.226	0.7859	0.5381	0.5205	0.5437	0.06651	0.06651	-0.09347	-0.1443	-0.406
4	0.9684	0.6023	0.4663	0.4663	0.03166	-0.1832	-0.002854	-0.1787	-0.1394	-0.7793
5	1.23	1.23	0.4523	0.4173	-0.08425	-0.3001	-0.238	-0.2615	-0.3275	-0.735
6	0.3427	0.4523	0.4523	0.197	-0.1374	-0.4455	-0.4605	-0.4254	-0.6547	-0.4597
7	0.1576	0.2427	0.6017	0.3458	-0.08445	-0.4055	-0.3569	-0.4605	-0.5091	-0.6547
8	-0.09511	0.1614	0.3192	-0.04574	-0.145	-0.3592	-0.5471	-0.5575	-0.5575	-0.8945
9	-0.3066	0.007763	0.1445	-0.1039	-0.2763	-0.2533	-0.5613	-0.5575	-0.432	-0.5797
10	-0.3295	0.05259	0.0332	-0.1112	-0.2764	-0.3308	-0.5587	-0.6236	-0.4724	-0.6025

B : average ratings

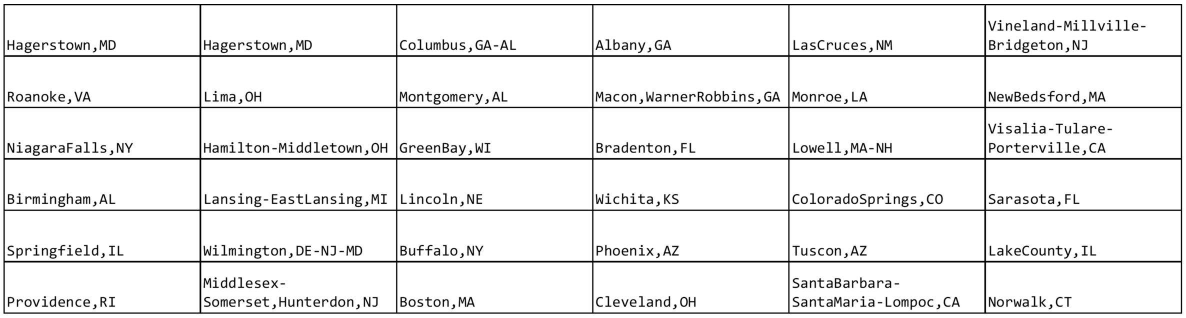
Figure 8

5 Problem 5

I run SOM algorithm with a 6 by 6 lattice and get 36 prototypes, then put the name of the cities that are closest to the prototypes in a 6 by 6 lattice.

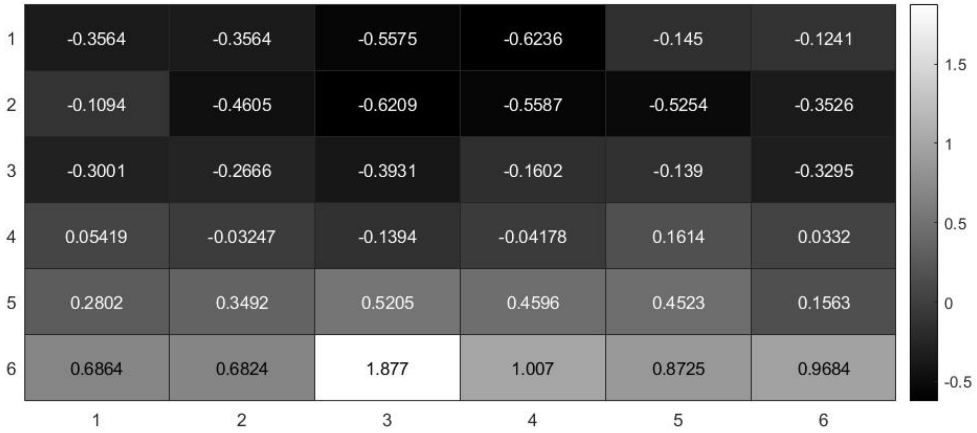
The result in Figure 9 looks reasonable too. This time the better cities gather at the edge rather than corner. Cities with similar quality are still closer to each other.

When running SOM algorithm, I found that when the size of lattice is smaller (less nodes), there will be less prototypes. As a result, the dissimilarity of the prototypes and their closest cities will be larger, and the map will be organized better. That's reasonable becasue it's easier to cluster less data.



Hagerstown,MD	Hagerstown,MD	Columbus,GA-AL	Albany,GA	LasCruces,NM	Vineland-Millville-Bridgeton,NJ
Roanoke,VA	Lima,OH	Montgomery,AL	Macon,WarnerRobbins,GA	Monroe,LA	NewBedsford,MA
NiagaraFalls,NY	Hamilton-Middletown,OH	GreenBay,WI	Bradenton,FL	Lowell,MA-NH	Visalia-Tulare-Porterville,CA
Birmingham,AL	Lansing-EastLansing,MI	Lincoln,NE	Wichita,KS	ColoradoSprings,CO	Sarasota,FL
Springfield,IL	Wilmington,DE-NJ-MD	Buffalo,NY	Phoenix,AZ	Tuscon,AZ	LakeCounty,IL
Providence,RI	Middlesex-Somerset,Hunterdon,NJ	Boston,MA	Cleveland,OH	SantaBarbara-SantaMaria-Lompoc,CA	Norwalk,CT

A : name of the cities



	1	2	3	4	5	6
1	-0.3564	-0.3564	-0.5575	-0.6236	-0.145	-0.1241
2	-0.1094	-0.4605	-0.6209	-0.5587	-0.5254	-0.3526
3	-0.3001	-0.2666	-0.3931	-0.1602	-0.139	-0.3295
4	0.05419	-0.03247	-0.1394	-0.04178	0.1614	0.0332
5	0.2802	0.3492	0.5205	0.4596	0.4523	0.1563
6	0.6864	0.6824	1.877	1.007	0.8725	0.9684

B : average ratings

Figure 9

6 Matlab Code

The code of PCA, k-medoids and SOM are completly the same as the previous homeworks.

6.1 NMF function

```
function [W, H] = NMF(X, k)
%Jiasen Zhang: NMF

% input:
% k = rank of W and H
% X = nonnegative matrix

% output: W[n,k], H[k,p]

tau = 0.001; % tolerance
tmax = 100000;
[n,p]=size(X);

% initialization
W = abs(rand(n,k));
H = abs(rand(k,p));
for j=1:k
    W(:,j)=W(:,j)/norm(W(:,j),inf);
end

% iteration
for t=1:tmax
    Xc = W*H;
    Hn = (W'*X)./(W'*Xc).*H; % new H
    Xc = W*Hn; % new Xc
    Wn = (X*Hn')./(Xc*Hn').*W; % new W
    for j=1:k
        Wn(:,j)=Wn(:,j)/norm(Wn(:,j),inf);
    end
    dWH = norm(Wn-W,'fro')/norm(W,'fro')+norm(Hn-H,'fro')/norm(H,'fro');
    W=Wn;
    H=Hn;
    fprintf('%d, %e \n',t,dWH);
    if dWH<tau
        break;
    end
end
end
```

6.2 Problem 1

```
clear all;clc;

load cities;
X = ratings;
% % % normalize X
stdX = std(X,0,1);
rateBar = mean(X,1);
for i=1:9
    X(:,i)=(X(:,i)-rateBar(i))/stdX(i);
end

[u,d,v] = svd(X);
diag(d) % singular values
d=diag(d.^2);
proportion=d/sum(d) % importance of PCA components
```

```
z = u(:,1:2)'*X; % PCA components

% plot PCA components
absz=abs(z);
figure();hold on;grid on;
scatter((1:9),absz(1,:),'r','k.','SizeData',500);
scatter((1:9),absz(2,:),'b','k.','SizeData',500);
plot(absz(1,:),'r');
plot(absz(2,:),'b');
xlabel('Category');
ylabel('Coefficients');
set(gca,'xticklabel',categories(1:9,:));
legend('Principal component 1 (red)','Principal component 2 (blue)');
```

6.3 Problem 2

```
clear all;clc;

load cities;
X = ratings;
%normalize X
stdX = std(X,0,1);
rateBar = mean(X,1);
for i=1:9
    X(:,i)=(X(:,i)-rateBar(i))/stdX(i);
end
X=X';
[n,p]=size(X);

% Get distance matrix D
D = zeros(p,p);
for i=1:p
    for j=1:i
        D(i,j)=norm(X(:,i)-X(:,j),2);
        D(j,i)=D(i,j);
    end
end

k = 4;
tau = 1e-6;
ninit=20;
[I_assign, I_bar] = my_k_medoids(k, D, tau, ninit);

[u,d,v] = svd(X);
z = u(:,1:2)'*X;

figure();hold on;
% plot the two clusters
scatter(z(1,I_assign==1),z(2,I_assign==1),'r','k.','SizeData',100);
scatter(z(1,I_assign==2),z(2,I_assign==2),'b','k.','SizeData',100);
scatter(z(1,I_assign==3),z(2,I_assign==3),'m','k.','SizeData',100);
scatter(z(1,I_assign==4),z(2,I_assign==4),'k','k.','SizeData',100);
for L=1:k
    scatter(z(1,I_bar(L)),z(2,I_bar(L)),'g','k.','SizeData',200);
end
legend('cluster1','cluster2','cluster3','cluster4','medoids');
xlabel('Principal component 1');
ylabel('Principal component 2');

% plot the average rating of two clusters
figure();hold on;grid on;
scatter((1:9),X(:,I_bar(1)),'r','k.','SizeData',500);
scatter((1:9),X(:,I_bar(2)),'b','k.','SizeData',500);
scatter((1:9),X(:,I_bar(3)),'m','k.','SizeData',500);
scatter((1:9),X(:,I_bar(4)),'k','k.','SizeData',500);
```

```
plot(X(:,I_bar(1)),'r');
plot(X(:,I_bar(2)),'b');
plot(X(:,I_bar(3)),'m');
plot(X(:,I_bar(4)),'k');
legend('medoid-1 (red)','medoid-2 (blue)','medoid-3 (purple)','medoid-4 (black)');
xlabel('Category');
ylabel('Rating');
set(gca,'xticklabel',categories(1:9,:));

% show the medoid cities
for L=1:k
    fprintf(names(I_bar(L),:));
    fprintf('\n');
end
```

6.4 Problem 3

```
clear all;clc;

load cities;
X = ratings;
%normalize X
stdX = std(X,0,1);
rateBar = mean(X,1);
for i=1:9
    X(:,i)=(X(:,i)-rateBar(i))/stdX(i);
end
X=X';
[n,p]=size(X);

k=3;
[W,H] = NMF(X,k);

%normalize W
stdW = std(W,0,2);
meanW = mean(W,2);
for i=1:9
    W(i,:)=(W(i,:)-meanW(i))/stdW(i);
end


% W contains feature vectors
% find closest cities
num = zeros(k,1);
for t=1:k
    num(t)=1;
    for tt=2:p
        Xt = X(:,tt);% city to be compared
        Xnum = X(:,num(t));% current chosen city
        if norm(Xt-W(:,t))<norm(Xnum-W(:,t))
            num(t) = tt;
        end
    end
end
names(num(:),:) % show the cities

figure();hold on;grid on;
scatter((1:9),X(:,num(1)),'r','k.','SizeData',500);
scatter((1:9),X(:,num(2)),'b','k.','SizeData',500);
scatter((1:9),X(:,num(3)),'m','k.','SizeData',500);
%scatter((1:9),X(:,num(4)),'k','k.','SizeData',500);
plot(X(:,num(1)),'r');
plot(X(:,num(2)),'b');
plot(X(:,num(3)),'m');
%plot(X(:,num(4)),'k');
```

```
%legend('city 1 (red)','city 2 (blue)');
legend('city 1 (red)','city 2 (blue)','city 3 (purple)');
%legend('city 1 (red)','city 2 (blue)','city 3 (purple)','city 4 (black)');
xlabel('Category');
ylabel('Rating');
set(gca,'xticklabel',categories(1:9,:));
```

6.5 Problem 4 and 5

```
clear all;clc;

load cities;
X = ratings;
%normalize X
stdX = std(X,0,1);
rateBar = mean(X,1);
for i=1:9
    X(:,i)=(X(:,i)-rateBar(i))/stdX(i);
end
X=X';
[n,p]=size(X);

% run SOM
K=36;
N = round(sqrt(K));
Tmax = 600*K;
T0 = 2000;
M = SOM(K, X, Tmax, T0);

name0 = char(K,43);
rate0 = zeros(N,N);
for i=1:N
    for j=1:N
        L = N*(i-1)+j;
        % find data closest to M(:,L)
        numForL = 1;
        for tt=1:p
            if norm((M(:,L)-X(:,tt)),2)<norm((M(:,L)-X(:,numForL)),2)
                numForL = tt;
            end
        end
        name0(L,1:43)=names(numForL,:);
        rate0(i,j)=mean(X(:,numForL));
    end
end
name0
%contourf(rate0')
heatmap(rate0');colormap(gray);
```