Devin Trejo

ECE 3522: Stochastic Processes in Signals and Systems

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19122

I. Problem Statement

In this assignment we introduce Principal Component Analysis which will allow us to take a set of data and transform it so appears uncorrelated. Using some data built into MatLab we can successfully apply principal component analysis in a study that tracks 349 different cities' climate, housing, health, crime, transportation, education, arts, recreation, and economics statuses.

To begin applying principal component analysis we first must find our whitening transform matrix. The transform matrix is constructed from the Eigen vectors and values of our data's covariance matrix. The constructed is shown below:

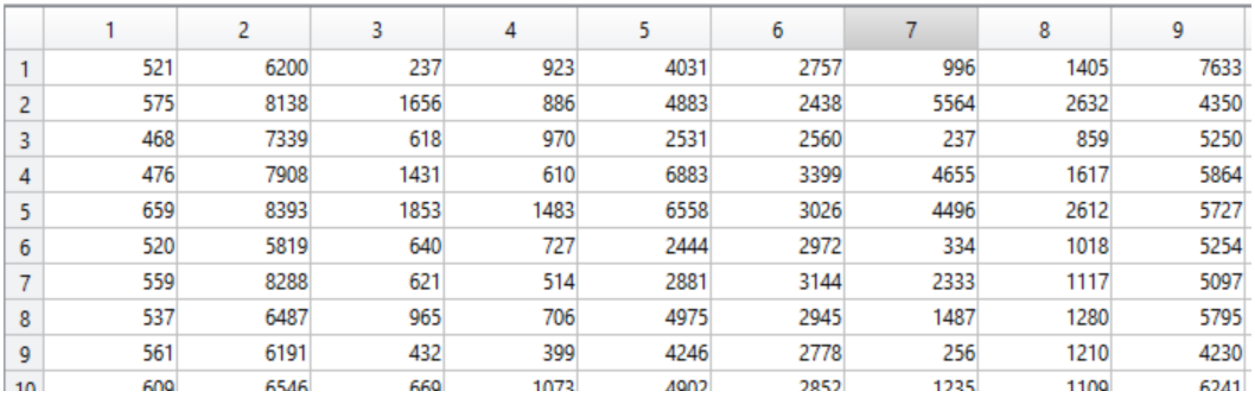
$$Whitening\ Transform = (\sqrt{EigenValue})(EigvenVector^{T})$$

The transformed data will appear uncorrelated. A test we perform to see if this is true is to compute the covariance matrix of our transformed data. If the data is truly uncorrelated the off-diagonal will be all zeros.

Lastly we will want to see which two cities are the most closely related. We will apply a Euclidean distance to determine which ratings of two cities are the most closely related. We perform this step for our un-transformed

II. Approach and Results

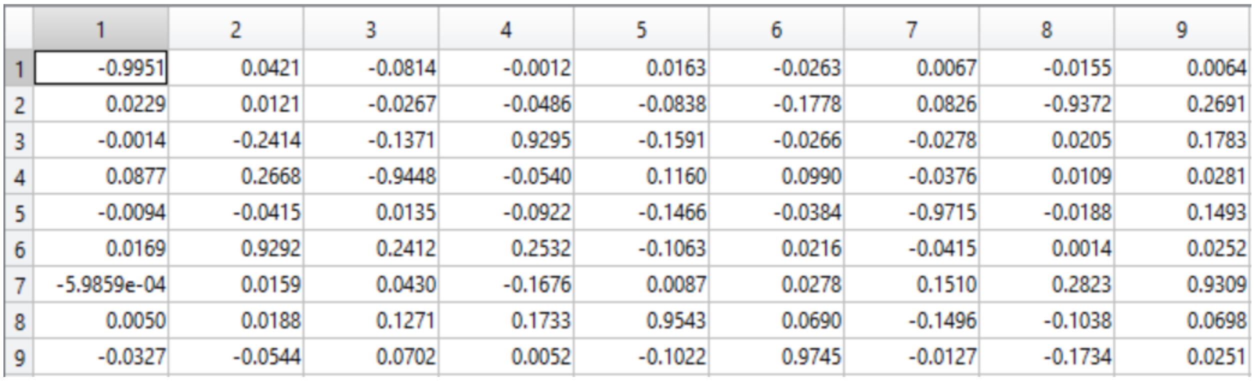
First we will showcase a sample of the raw ratings matrix.



	1	2	3	4	5	6	7	8	9
1	521	6200	237	923	4031	2757	996	1405	7633
2	575	8138	1656	886	4883	2438	5564	2632	4350
3	468	7339	618	970	2531	2560	237	859	5250
4	476	7908	1431	610	6883	3399	4655	1617	5864
5	659	8393	1853	1483	6558	3026	4496	2612	5727
6	520	5819	640	727	2444	2972	334	1018	5254
7	559	8288	621	514	2881	3144	2333	1117	5097
8	537	6487	965	706	4975	2945	1487	1280	5795
9	561	6191	432	399	4246	2778	256	1210	4230
10	609	6546	669	1073	4902	2852	1235	1109	6241

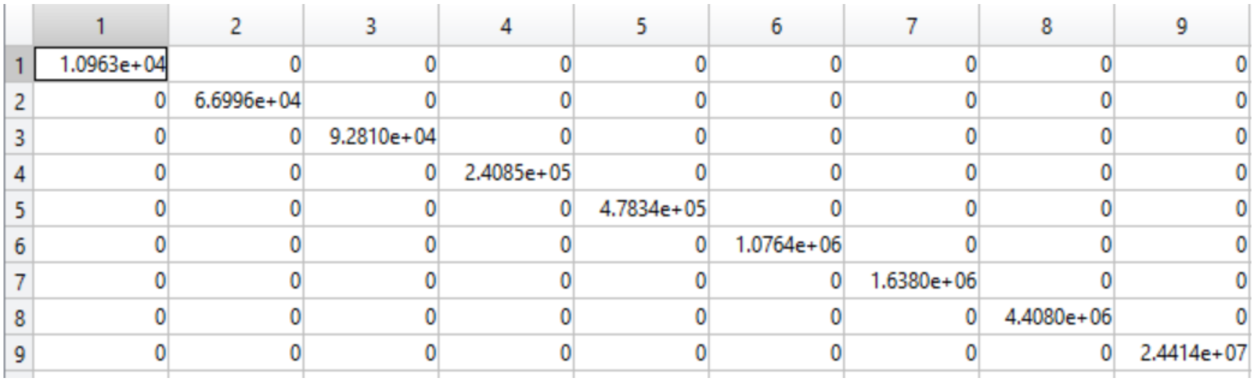
Figure 1: Sample of the Ratings Matrix (Note it is 349 Rows Long)

Now we can find the covariance of the ratings matrix which will lead us to finding the Eigen Vectors and values. We display them below:



	1	2	3	4	5	6	7	8	9
1	-0.9951	0.0421	-0.0814	-0.0012	0.0163	-0.0263	0.0067	-0.0155	0.0064
2	0.0229	0.0121	-0.0267	-0.0486	-0.0838	-0.1778	0.0826	-0.9372	0.2691
3	-0.0014	-0.2414	-0.1371	0.9295	-0.1591	-0.0266	-0.0278	0.0205	0.1783
4	0.0877	0.2668	-0.9448	-0.0540	0.1160	0.0990	-0.0376	0.0109	0.0281
5	-0.0094	-0.0415	0.0135	-0.0922	-0.1466	-0.0384	-0.9715	-0.0188	0.1493
6	0.0169	0.9292	0.2412	0.2532	-0.1063	0.0216	-0.0415	0.0014	0.0252
7	-5.9859e-04	0.0159	0.0430	-0.1676	0.0087	0.0278	0.1510	0.2823	0.9309
8	0.0050	0.0188	0.1271	0.1733	0.9543	0.0690	-0.1496	-0.1038	0.0698
9	-0.0327	-0.0544	0.0702	0.0052	-0.1022	0.9745	-0.0127	-0.1734	0.0251

Figure 2: Eigen Vector Matrix for Ratings Data



	1	2	3	4	5	6	7	8	9
1	1.0963e+04	0	0	0	0	0	0	0	0
2	0	6.6996e+04	0	0	0	0	0	0	0
3	0	0	9.2810e+04	0	0	0	0	0	0
4	0	0	0	2.4085e+05	0	0	0	0	0
5	0	0	0	0	4.7834e+05	0	0	0	0
6	0	0	0	0	0	1.0764e+06	0	0	0
7	0	0	0	0	0	0	1.6380e+06	0	0
8	0	0	0	0	0	0	0	4.4080e+06	0
9	0	0	0	0	0	0	0	0	2.4414e+07

Figure 3: Eigen Value Matrix for Ratings Data

From the Eigen value matrix we see that the largest value occurs in the 9th column which tells us that the 9th column has the most variance in it. You can see this is true if you reference Figure 1. Referencing the corresponding categories matrix loaded in with the cities data we see how the 9th column corresponds to the economic status of each city. Since this Eigen value is the largest we know the Eigen vector corresponding to this Eigen value has the most weight in the transform. The second largest Eigen Value occurs in the 8th column which correspond to a city's recreation. If we remove the two rating criteria with the largest Eigen values we will observe how much weight they have on the variance of the ratings data set (climate, housing, health). If we look at the Eigen values for these columns we see how they

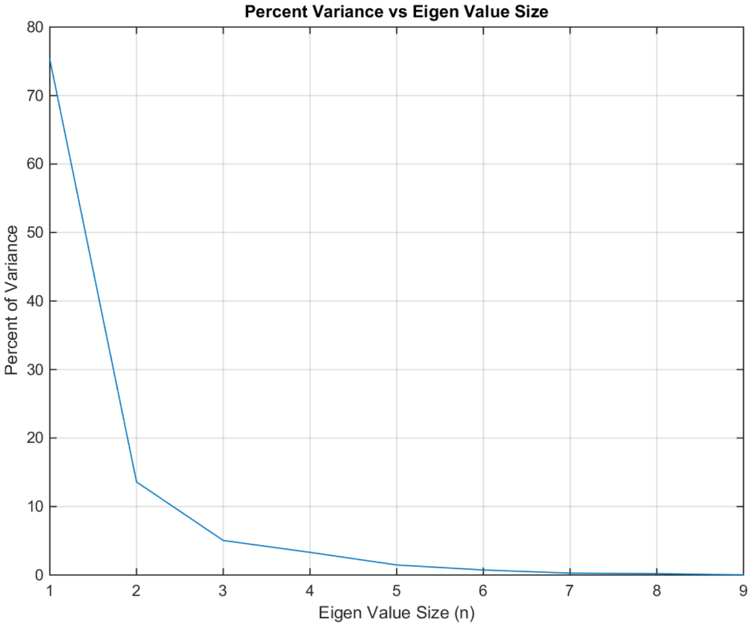
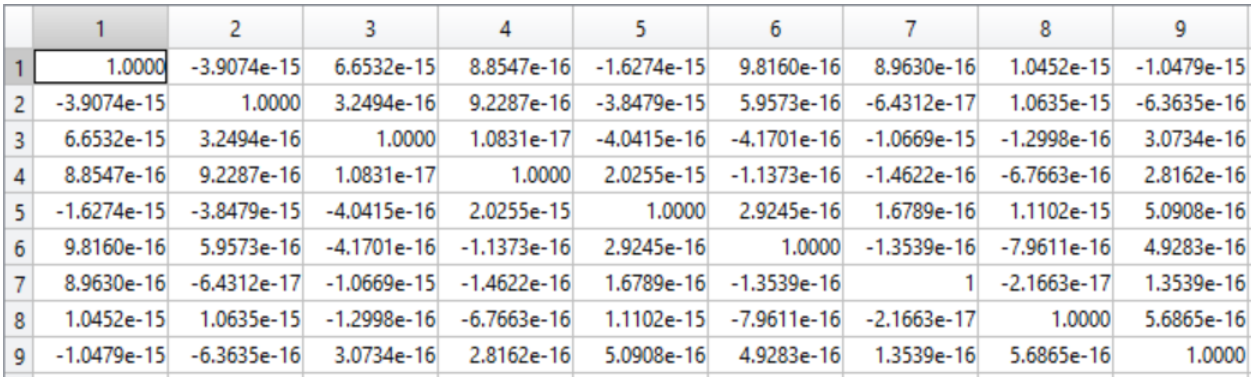


Figure 4: Variance v Eigen Value Size

The above plot showcases how the largest Eigen values effect the variance of the data. We can now successfully compute our transformation matrix. We call our new matrix Y1. If we compute the covariance matrix of Y1 we observe an identity matrix (with rounding errors).



	1	2	3	4	5	6	7	8	9
1	1.0000	-3.9074e-15	6.6532e-15	8.8547e-16	-1.6274e-15	9.8160e-16	8.9630e-16	1.0452e-15	-1.0479e-15
2	-3.9074e-15	1.0000	3.2494e-16	9.2287e-16	-3.8479e-15	5.9573e-16	-6.4312e-17	1.0635e-15	-6.3635e-16
3	6.6532e-15	3.2494e-16	1.0000	1.0831e-17	-4.0415e-16	-4.1701e-16	-1.0669e-15	-1.2998e-16	3.0734e-16
4	8.8547e-16	9.2287e-16	1.0831e-17	1.0000	2.0255e-15	-1.1373e-16	-1.4622e-16	-6.7663e-16	2.8162e-16
5	-1.6274e-15	-3.8479e-15	-4.0415e-16	2.0255e-15	1.0000	2.9245e-16	1.6789e-16	1.1102e-15	5.0908e-16
6	9.8160e-16	5.9573e-16	-4.1701e-16	-1.1373e-16	2.9245e-16	1.0000	-1.3539e-16	-7.9611e-16	4.9283e-16
7	8.9630e-16	-6.4312e-17	-1.0669e-15	-1.4622e-16	1.6789e-16	-1.3539e-16	1	-2.1663e-17	1.3539e-16
8	1.0452e-15	1.0635e-15	-1.2998e-16	-6.7663e-16	1.1102e-15	-7.9611e-16	-2.1663e-17	1.0000	5.6865e-16
9	-1.0479e-15	-6.3635e-16	3.0734e-16	2.8162e-16	5.0908e-16	4.9283e-16	1.3539e-16	5.6865e-16	1.0000

Figure 5: Covariance of Transformed Ratings Matrix

Now that we have our original data and our transformed data we will find the two cities that share the closest ratings. In our first case we use our raw untransformed data. In our second case we use our transformed data which will allow us to find the two cities who are the most similar across all the different categorizes. Lastly, we will limit our ratings data to be only the first three categorizes (climate, housing, and health).

Raw Data:

The closest cities: Johnstown, PA & Elkhart-Goshen, IN

The third closest city: Altoona, PA

Transformed Data:

The closest cities: Hamilton-Middletown, OH & Grand Rapids, MI

The third closest city: Lewiston-Auburn, ME

Transformed Data limited to first 3 criteria:

The closest cities: St. Joseph, MO & Champaign-Urbana-Rantoul, IL

The third closest city: Medford, OR

Figure 6: Closest cities Result

In our first scenario we see that the three closest cities are two from PA and on from IN. Since we found these distance measures from the un-transformed data set we note that these three cities are most similar in regard to economic status only. When we switch to using the transformed data set we see how we end up with different results. The two closest cities if were to weight all categories equally are a city from OH and one from MI. Lastly if we only concentrate on finding the three closest cities in terms of the first three categorizes our results change once again.

III. MATLAB CODE

```
%% Program script
clear; clc; close all

% Load Cities data set
load cities

% Compute Covariance Matrix
CovC = cov(ratings)

% Find Eigen Value/Vectors
[eigVec, eigVal] = eig(CovC)

% Tranformed Data (Whitening Transform)
V1 = (eigVal^-(1/2))*eigVec';

% Display Covariance Matrix (Note: Its an identity)
Y = (V1*ratings')';
Ycov = cov(Y)

% Plot percent of variance accounted for by the first n eigenvalues
n = 1:size(ratings,2);
sortEigVal = sort(sort(eigVal, 1, 'descend'), 2, 'descend');
for i = n
    varPer(i) = (sortEigVal(1,i))/sum(sortEigVal(1,:), 2)*100;
end

figure();
plot(n, varPer);
xlabel('Eigen Value Size (n)');
ylabel('Percent of Variance');
title('Percent Variance vs Eigen Value Size');
grid on;


% Find the two cities that are closet together
fprintf('Raw Data:\n');
findThreeSimilar(names, ratings, size(ratings,2));
fprintf('Transformed Data:\n');
findThreeSimilar(names, Y, size(Y,2));
fprintf('Transformed Data limited to first 3 criteria:\n');
findThreeSimilar(names, Y, 3);
```

1 Main Program Script. We first load in our cities data set and compute its covariance. From the covariance matrix we are able to find a whitening transform that will uncorrelated the data. To test the hypothesis we compute the covariance of our transformed data to the console. For each Eigen value we compute the correspond variance it has on the data. After we find our transformed data we will now find the closet cities using the ‘findThreeSimilar’ function.

```
 function findThreeSimilar(labels, data, criterialL)

% Limit the scope of the data
dataLimited = data(:,1:criterialL);

% Find the two closest cities
for i = 1:size(dataLimited, 1)
    for j = 1:size(dataLimited,1)
        for k = 1:size(dataLimited, 2)
            temp(k) = (dataLimited(i,k)-dataLimited(j,k))^2;
        end
        disData(i,j) = sqrt(sum(temp));
    end
end
clear temp

% Remove the zeros along the main diagonal
for i = 1:size(disData,1)
    for j = 1:size(disData,2)
        if(i == j)
            disData(i,j) = inf;
        end
    end
end

% Find minimumn distance
minDis = min(min(disData));
[minR, minC] = find(disData==minDis);
fprintf(['The closest cities: ' labels(minR(1),:) ' & ' labels(minC(1),:) '\n']);
```

2 We write a ‘findThreeSimilar’ function to find the three cities that share the closet ratings. First we go through and limit the scope of our data if necessary. Then we use a Euclidean distance measure to find the cities that have the closest ratings to one another. Since the minimum Euclidean distance will always occur when comparing a rating to itself (along the main diagonal) we set these distances equal to infinity. Now we can find the true ratings which are closest together. We print out the two closest cities to the console.

```
% Begin finding the third closest city
for i = 1:length(dataLimited(1,:))
    disData3(i) = (dataLimited(minR(1),i)+dataLimited(minC(1),i))/2;
end

for i = 1:size(dataLimited,1)
    for j = 1:size(dataLimited, 2)
        temp(j) = (dataLimited(i,j)-disData3(j))^2;
    end
    dis2C(i) = sqrt(sum(temp));
    if((i==minC(1)) || (i==minR(1)) )
       dis2C(i) = inf;
    end
end

% Fianlly we can find the third closest state
minc2z = find(dis2C==min(dis2C));
fprintf(['The third closest city: ' labels(minc2z,:) '\n\n']);

end
```

3 The second half of the function finds a third city which shares the closest ratings to the first two. We perform a similar procedure to what we went through for the two cities by first finding a Euclidean distance. Then we take out the main diagonal. Finally using the find function in MatLab we are able to find the city which shares similar ratings to the first two.

IV. Conclusions

What is the important take away from the results is how your results change depending on the weights of your data. We observed how the economic status has the most variance thus if we only use our un-transformed data the two cities that share the closest economic status will appear the most similar. These will closely related in terms of economic status but that may be it. If you want to find the two cities that share the most characteristics across all categorizes you need to transform the data. In our second trial we could say the cities of Hamilton-Middletown, OH & Grand Rapids, MI share more in common than Johnstown, PA & Elkhart-Goshen, IN do. Principal component analysis is useful in simplifying the process of comparing two cities. If we did not apply principal component analysis we would need to weight in the covariance matrix in our distance calculations.