Emilie Doyle

ECE 3522: Stochastic Process in Signals and Systems

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19121

# Problem Statement

The objective of this assignment is to learn more about principal component analysis. Principal component analysis is to be completed on the MATLAB data set that contains ratings for nine different components of cities. There are 392 cities and these components are; climate, housing, health, crime, transportation, education, arts, recreation and economics. The objective of principal components analysis is to take variables that are on different scales and to convert them all of to the same type of scale.

The first task is to compute the covariance matrix of the raw data. Then eigenvector and eigenvalue analysis of the covariance matrix must be computed. Next, we must determine the Whitening Transform of the data and compare it to the covariance of the transformed data. Then the percentage of variance was to be determined and plotted for the first nine eigenvalues from largest to smallest. Next, the eigenvectors for the two largest eigenvalues are to be analyzed. Lastly, the data was used to determine the three most similar cities. This was to be accomplished in three different ways. First the original data was used to find the three closest cities through the use of Euclidean distances. Next, the transformed data was then used to find the three closest cities using Euclidean distances again. Lastly, Euclidean distances was then used for solely the three most influential

# Approach and Results

I first started by loading in the data about the cities into MATLAB through the use of the load command. I then took the covariance of the matrix of the ratings of the different cities. Next, I used the eig function in MATLAB, which returns the eigenvalues and eigenvectors of the provided matrix. In order to find the Whitening Transform, I implemented the equation for the obtaining the matrix to then find the covariance of to obtain the Whitening Transform. I used the diag function in order to extract the eigenvalues from the matrix D, and then used sort in order to sort them in descending order. In order to find the percentage, I cycled through each value of the eigenvalues and divided it by the sum of all of the eigenvalues. I plotted the percent of the eigenvalues as well as the cumulative sum of the eigenvalues in order to check my results.

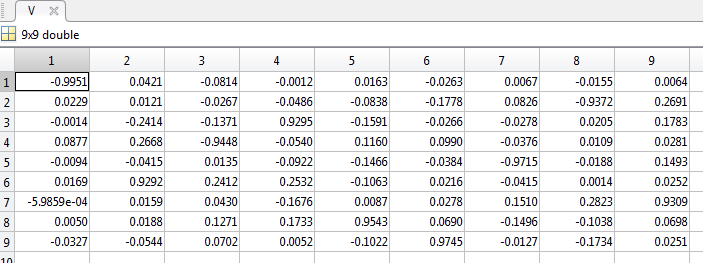


Figure : Eigenvectors

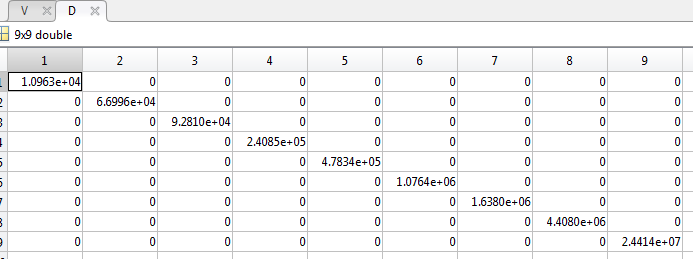


Figure : Eigenvalues

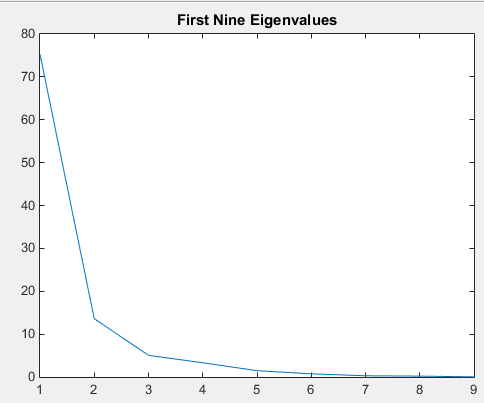


Figure : First Nine Eigenvalues

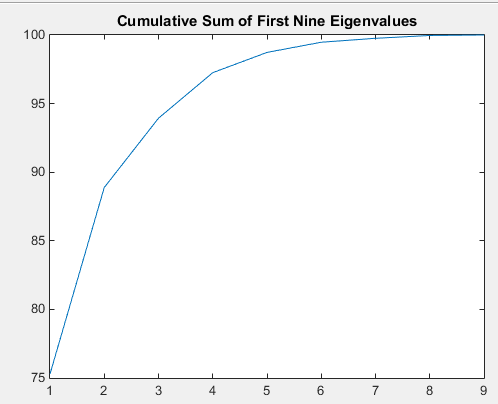


Figure : Cumulative Sum of First Nine Eigenvalues

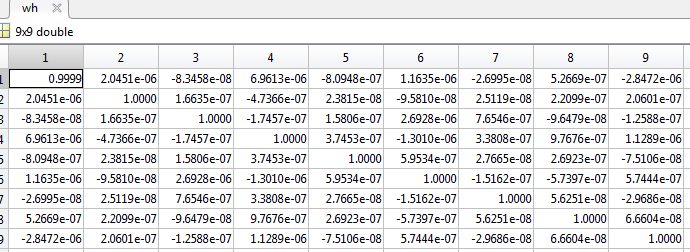


Figure : Whitening Transform Covariance

I coded for the nearest cities starting with the weighted values, even though the prompt asked for the original values first. I used the same basic form of code for all three sections that asked for the three closest cities. I used the pdist function in MATLAB to determine the Euclidean distance between. The pdist returns the distances in one long array, so I used squareform in order to convert it into an easily readable matrix. In order to eliminate concern that zeros may posses in determining the minimum value, I converted the zero values to be NaN values. I then used the min command on the resulting matrix. This will find the minimum values of each column and put them in an array. I then used the min command again to find the minimum value of all of those minimum values. The values represent the Euclidean distance between the points. When calling the min function, the returned values are the resulting matrix or array, and the index of the minimum value. The first min collects the minimum values, the second min determines what row of the matrix the minimum value lies in and then the third min determines the minimum of that specified row in order to find the exact index coordinates of it. The average of these values for the two cities was found by adding up the total ratings and then dividing by the two cities. The average was then concatenated with the rest of the rating matrix in order to determine the next closest city.

When the code is run for the original ratings two closest cities are city 93 and city 147. These cities correspond to Elkhart-Goshen, IN and Johnston, PA, respectively. The third closest is city 10, which is Amarillo, TX.

For the uncorrelated data, the two closest cities are city 47 and city 74. The cities that correspond to these numbers are Bremerton, WA and Columbus, OH. The third closest city is city 99, which corresponds to Fall River, MA-RI.

For the third set of data, in which solely the three most prominent eigenvalues were concerned, the closest two cities were city 176 (Little Rock, North Little Rock, AR) and city 191 (Minneapolis-St. Paul, MN-WI), and the third city is city 152, which corresponds to Kansas City, KS.

# MATLAB Code

%Stoch CA 12

clear;

load cities

C= cov(ratings);

[V,D]= eig(C);

W= ratings\*V\*diag(1./(diag(D)+1).^(0.5))\*V';

wh= cov(W); %wh looks not like identity matrix, but it. check in command window.

m= diag(D);

d=sort( m, 'descend');

for i = 1:9

decimal(i)=d(i)/sum(d);

percent(i)= decimal(i)\*100;

a= sum(percent);

end

% figure(1)

% plot(percent);

% title('First Nine Eigenvalues');

% figure(2)

% plot(cumsum(percent));

% title('Cumulative Sum of First Nine Eigenvalues');

%Highest Eigenvalues are the two last values, which correspond to

%recreation (8) and economics (9)

d2= sort(decimal);

F= d2';

A= ratings';

%%

%Weighted Values

f=d2\*(ratings');

E= pdist(f');

S= squareform(E);

S(~S)= nan;

[M, I] = min(S);

[CM, CI]= min(M);

%Index is row 47 of S, value is 0.0059.So it is the overall smallest, and the smallest

%value of row 47.

[TM, TI]= min (S(47,:));

%In row 47, the minimum value has an index of 74. Row 47, Column 74 of S

%So the closest two cities are city 47 and city 74...

%CAvg= (0.5)\*(((ratings(47,:).\*d2)+((ratings(74,:).\*d2))));

CAvg= (0.5)\*(((ratings(47,:))+((ratings(74,:)))));

TMP= [CAvg; ratings(1:46,:);ratings(48:73,:); ratings(75:329,:)];

f2=d2\*(TMP');

E2= pdist(f2');

S2= squareform(E2);

S2(~S2)= nan;

[M2, I2] = min(S2(1,:));

%The smallest distance in relation to the first column (CAvg) would then be

%the minimum of row 1 comparisons, which apparently is in City 99

%%

%Unweighted Values

EO= pdist(ratings);

SO= squareform(EO);

SO(~SO)= nan;

[MO, IO]=min(SO);

[CMO, CIO]= min(MO);

[TMO, TIO]=min(SO(93,:));

%In row 93 of the original matrix, the minimum value has an index of 147.

%Row 93, column 147 of S, so those are the two closest cities

COAvg=(0.5)\*(((ratings(93,:))+((ratings(147,:)))));

TMPO= [COAvg; ratings(1:92,:);ratings(94:146,:); ratings(148:329,:)];

EO2= pdist(TMPO);

SO2= squareform(EO2);

SO2(~SO2)= nan;

[MO2, IO2]= min(SO2(1,:));

%Smallest distance from average (1) is at 10

%%

%First three features

R= ratings(:, 7:9);

f3= d2(7:9)\*(R');

E3= pdist(f3');

S3= squareform(E3);

S3(~S3)=nan;

[M3,I3]= min(S3);

[CM3, CI3]= min(M3); %CI3= 176

[TM3, TI3]= min(S3(176,:)); %TI3= 191

CAvg3= (0.5)\*(((ratings(176,:))+((ratings(191,:)))));

TMP3= [CAvg3; ratings(1:175,:);ratings(177:190,:); ratings(192:329,:)];

f3=d2\*(TMP3');

E3= pdist(f3');

S3= squareform(E3);

S3(~S3)= nan;

[M3, I3] = min(S3(1,:));% smallest distance from average is 152

# Conclusions

Principal components analysis seems like an extremely helpful tool for large and various data analysis and applications. I can see how this could be extremely influential in deciding major contributing factors for data analysis regarding marketing decisions. The ability to compare unlike variables in a similar format is extremely powerful.

Note that the resulting matrix for the Whitening Transform should be an identity matrix, but it does not appear this way when you view the matrix as a variable. However, when you view it in the Command Window it appears as an identity matrix because the diagonal values are so close to one and the non-diagonal values are essentially zero.

It was interesting to note that the eigenvalues were already in ascending order, so it was necessary to switch them to descending order. The factors that appeared to have the most weight or influence were then the last two of the original ascending list, or the first two of the new descending list. These correspond to recreation and to economics. Economics definitely makes sense for a leading proponent for reasons that people would like a city. Strong economics generally means strong opportunity for jobs and general well being. This could also be tied to strong industry and high wages. Recreation seems like much less of a factor. It is important to note that the relative percentages are 75% to 5%, so really, economics seems to be more of the deciding factor.

I think it may be a bit difficult to evaluate the correctness of the closeness of the cities without having a real working knowledge of the cities it points towards. The only one I know a bit of is Johnstown, PA, which is right around where my mom grew up and my grandmother lives. Overall, the cities seem rather random, but it is a large set of cities. All three sets of close cities seem very different from each other, which is not exactly what I would expect. I would expect the original data results to be vastly different from the other two, but I would expect the other two to be quite similar, because the economics is weighted so highly. I guess that either goes to show error, or that the other factors that make up about 20% of the weight still have significant pull on the total value.