BioMath - PCA Exercises - August 27, 2015

1. **Plotting eigenvectors**

   Load the data from "pca_example.mat", as found on the wiki. This will create a variable called X. Each column of X represents the outcome of a single trial of an experiment; each trial produces two numbers, so there are two rows.

   • We're first going to do PCA without subtracting the mean, to see what happens. Compute the covariance matrix $C = XX^T/N$, where N is the number of trials, and find its eigenvalues and eigenvectors. Plot a 2-dimensional scatterplot of the outcome of all the trials of the experiment, with the first row of X on the horizontal axis, and its second row on the vertical axis. For each of the eigenvectors, add to your plot a line that starts at the origin, goes in the direction of the eigenvector, and is of length equal to the square root of the corresponding eigenvalue. (This length represents the standard deviation of the data along that eigenvector, i.e., the square root of the variance.) Do these lines lie in the directions of the major axes of the data cloud?

   • Now let's redo that exercise after subtracting the mean. Subtract from each row of X the mean of the values on that row. Find the covariance matrix for your mean-subtracted data, and now plot the eigenvector directions as you do above. Do the lines do a better or worse job than they did before, in terms of characterizing the data cloud?

   • Confirm that V is an orthonormal matrix, namely, each column is unit length and the columns are orthogonal to each other. What is VV^T? What kind of mapping is the one that changes basis to the eigenvector basis, i.e., going to $\mathbf{x}_{new} = V^{-1}\mathbf{x}$?

   • Compute X_{new}, namely the original data expressed in the eigenvector basis. Redo the exercise of plotting the data, computing the covariance matrix and plotting the eigenvectors, but now for the X_{new} data. Comment on what you find.

2. **Reducing dimensions**.

   Download the pca_probs_2_and_3.zip file from the wiki, and unzip it. You'll find two .mat files in there. Load the sixdim.mat data file into Matlab. This will create a variable called X. Once again, each column of X is the outcome of one trial of an experiment. Here, each experiment generates 6 numbers, so each outcome is a point in 6-dimensional space. There are 500 trials in the data set.

   • To try to visualize this, make a scatterplot of dimension 1 vs dimension 2 (this should have 500 points in it); another scatterplot of dimension 3 vs dimension 4; and another scatterplot of dimension 5 versus dimension 6. Do you see any obvious patterns? Visualizing things in 6 dimensions is hard. We'll see whether PCA can help us cut down the number of dimensions.

• Recall that if C_{ij} is the element in the i^{th} row and j^{th} column of the covariance matrix, then $C_{ij} = \langle x_i x_j \rangle$ (after making the data have mean zero in each of the six dimensions). Compute the covariance matrix for this data, and find its eigenvectors and eigenvalues. Sort the eigenvalues from largest to smallest, and make a plot of their sorted values. This is called a "scree plot". What fraction of the total variance lies in the first, second, third, and fourth eigendirections? If you consider the cumulative total of the first three eigendirections, what fraction of the total variance lies in that three-dimensional space?

• Consequently, if you knew the eigenvectors, how many numbers do you need to completely describe each of the original 6-dimensional data points? Why?

• Make a scatterplot of the data in the eigenvector space. That is, project each point onto each eigenvector, and now plot projection onto eig 1 versus projection onto eig 2 (this should have 500 data points); in another plot, show projection onto eig 3 vs projection onto eig 4; and in a final plot, projection onto eig 5 versus projection onto eig 6.

• If you wanted to do dimensionality reduction, which three dimensions would you drop?

3. **Reducing dimensions under noisy conditions**

• Consider a matrix formed in Matlab as $X = randn(2, n)$. This represents n trials where each trial produces 2 uncorrelated numbers (each entry in X is independent of all the others, so they are all uncorrelated). Compute the covariance matrix, and make a plot of the value of the magnitude of the off-diagonal entry in the covariance matrix as a function of n, for n running from 5 to 500. Run your code many times, to see the plot many times. What do you observe? What does this tell you about measured covariance in data sets composed of small n?

Now load the pantsdata.mat file. Again, this will create a variable called X. This time it is a cell. This data is real data, taken from the pants sizes of $n = 13$ Princeton students, postdocs, and faculty. Their names remain anonymous. Compute the covariance matrix for this data set, find its eigenvectors and eigenvalues.

• Make a scree plot showing the variance accounted for in each of the eigendirections.

• You want to drop some dimensions. Which ones should you drop? How many? Is there a clear criterion to make this decision?

• Permute the data randomly along each dimension. For example, for Cankle, permute the data randomly among the different individuals. (The function randperm.m will come in useful.) Then do a separate random permutation for each of the other dimensions. Since this randomizes the data among all the individuals, any correlations should be lost. Compute the covariance matrix for this new randomized data set. Did you get a diagonal matrix? Any non-zero off-diagonals here are due to the finite data size. This randomized data set gives

us an idea of what we should expect a scree plot to look like if there were no correlations among the different dimensions, given the noise induced by the finite data size. Make a scree plot of both the original data and the randomized data. Does this help you decide how many dimensions to keep?

4. **PCA and finding clusters**. Often we are wondering whether our big data set is composed of separate clusters of points. Because it can be difficult to discern that in many dimensions, one approach is to first do PCA to reduce the number of dimensions. For example, we might say "I'm going to rotate to the eigenvector basis, and look only at the number of dimensions necessary to capture 90% of the total variance." Perhaps this will drop us down to one or two dimensions. We can dthen o a scatterplot of the reduced data there, and perhaps more easily discern visibly separate clusters.

   • Load bsausage.mat. This will create a variable X which is 2-by-500, representing 500 trials of an experiment that produces two numbers on each trial. Make a scatterplot of dimension 1 vs dimension 2 (this should have 500 points in it); another scatterplot of dimension 3 vs dimension 4; and another scatterplot of dimension 5 versus dimension 6. Do you see any obvious patterns? Is the data set composed of separate clusters of points? Use "axis equal" after each plot so that the aspect ratio of vertical versus horizontal on the screen is 1.

   • Compute the covariance matrix for this data, and find its eigenvectors and eigenvalues. Sort the eigenvalues from largest to smallest, and make a plot of their sorted values. (Our "scree plot" again!). How many dimensions do you need to capture 90% of the total variance?

   • Project the data onto the eigenvector with the largest eigenvalue, and then plot a histogram of the values that you get from this projection. Is this histogram unimodal or multimodal (i.e., single-peaked or multi-peaked)? Based on that result, would you call the data clustered or not?

   • Now project the data onto the eigenvector with the *second* largest eigenvalue, and then plot a histogram of the values that you get from this projection. Is this histogram unimodal or multimodal (i.e., single-peaked or multi-peaked)? Based on that result, would you call the data clustered or not? What does this tell you about whether it was a good idea or not to only take the number of dimensions needed to capture 90% of the variance?

   • Make a scatterplot of the data in the space of the first two eigenvectors. That is, project each original data point onto the eigenvector with the largest eigenvalue, then project it onto the eigenvector with the second largest eigenvalue, and then plot one versus the other. If you do this for all 500 points, you'll get a scatterplot of with 500 points. Use "axis equal" after the plot again, so that the aspect ratio of vertical versus horizontal on the screen is 1. Looking at this data, by eye: if this were your original data set, which would be the first eigendirection? Which would be the second? Even though the first eigendirection captures

most of the variance, does projecting onto it and ignoring the second reveal the clustered nature of the data?