Data Mining Lab 2 - Hints

May 23, 2014

Disclaimer: If you have a tough time understanding PCA, then I recommend starting with Problem 3 (Spike sorting). This will take you through step by step on how to do PCA. The rest of this report serves as a guide to make the homework experience less burdensome and to save you time. The hints are provided for each problem individually.

1 Programming With PCA

This problem is more or less straight forward. The equations you need to implement the required functions are given in the class notes. The tricky part to remember is for the PCA reconstruction equation in the notes, the input Y into the PCA reconstruction function are the points projected onto the eigenvectors.

$$Y = \begin{bmatrix} \langle u_1, x_1 \rangle & \langle u_1, x_2 \rangle & \dots & \langle u_1, x_N \rangle \\ \vdots & \vdots & & \vdots \\ \langle u_M, x_1 \rangle & \langle u_M, x_2 \rangle & \dots & \langle u_M, x_N \rangle \end{bmatrix}.$$

Recall that the inner product $\langle u_1, x_1 \rangle$ can be written $u_1^T x_1$. The Z data in the PCAProjection function is some arbitrary data that you want to project. For the myPCA function use the functions 'sort' and 'diag' to help you implement the code.

2 Testing the PCA - Digit Dataset

- To extract the digit information use the labels Y as indices into the data matrix. Each row of the data matrix is a square image that has been vectorized. So to view the image you have to reshape that row into a square image.
- 1) To visualize the eigenvectors you need to use the 'reshape' and 'imshow' functions. Use the syntax imshow(imageName, []). Reshape the eigenvector as a square image, so the length of each side will be the squareroot of the length of the eigenvector.

- 2 & 3) For the reconstruction just use two images to reconstruct. The process of reconstruction involves computing the PCA component from your original data, projecting the desired data onto the eigenvectors, then reconstructing the image. The reconstructed image will always have the dimensionality equal to that of the original data.
- All digit dataset
  - 1) The results will not look like any specific numbers, so don't waste time trying to make them look like numbers.
  - 2) The different coordinates refers to using random eigenvectors instead of eigenvector 1:M. So for example you might form your projection matrix with eigenvectors 1, 2, 100, 200. Then do another synthesized image with eigenvectors from 1:M to see the real digits.

3 Eigenfaces - Face Recognition

- a & b) Use the functions you created in the previous problems to carry out the problem. Use the plot image function given to plot the images. Remember to subtract the mean from your data before you form the covariance matrix. The mean face will be the mean of the data samples, it should be a 2500x1 vector.
- c) The value of M indicates that you should reconstruct the image with the first 100 or 150 eigenvectors.
- d) If you do not have the knnclassify function in your matlab, you will have to do k-nearest neighbor from scratch. Some useful functions to do this are bsxfun, sort, mode, pdist2 and max. Look up the corresponding documentation to see what each function does. Compute the eigenvectors from Subset1YaleFaces.mat. Project the data Subset2YaleFaces onto the eigenvectors obtained from Subset1Yalefaces. There are 7 occurences of each digit in the data. So your k should not be more than 7 in your trials. For M I choose to project to some combination of eigenvectors from 1:1000. Anything beyond 1000 eigenvectors will be computationally expensive. The recognition error is the number of incorrectly classified samples / number of total samples.

4 Dimensionality Reduction

- b) To visualize the projected original image you have to use your reconstruction function.