COGS 185, Spring 2016

Assignment 4

Robust Principal Component Analysis

Due: May 24, 11:59 PM 2016, PDT (grace period: May 23, 3:00Am, 2016).

Late policy: Every 5% of the total points will be deducted for every extra day past due.

Submit your report to Ted.

Try to download the RPCA code at

http://perception.csl.illinois.edu/matrix-rank/sample_code.html#RPCA

and YaleB face dataset at (used the cropped dataset which contains 39 subjects):

http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

Alternatively, the dataset can be found here: http://pages.ucsd.edu/~ztu/courses/CroppedYale.zip

Divided faces from YaleB (cropped) into training and testing. The extended Yale Face Database B contains faces images of 39 human subjects under different illumination conditions. If you experience heavy computational burden, you are allowed to use a subset of the YaleB face dataset to reduce the computational complexity by reducing the number of human subjects (e.g. 10).

Note that you can reduce the image size to e.g. 32 x 32, or alternatively download a processed dataset at: http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html (there is a .mat 32x32 Data File that contains reduced sized images).

Task: Try to use two dimensionality reduction approaches: PCA and RPCA on the training data. Obtain the basis for the faces and use them as basis. Note that learning PCA and RPCA should be on your entire training dataset without the separation of individual subjects. The classification task (to classify each face into a human subject) below is based on the features after dimensionality reduction using either PCA or RPCA.

For each image, use the projection value of each face to the basis as feature for each component and train an multi-class SVM classifier (e.g. one vs. all) or any multi-class classifier you like to use. Compare the classification performances between PCA and RPCA. Report your results using different number principal components (basis functions).

Hint: After the A matrix is obtained using RPCA, try to use SVD or PCA to obtain the principal components. Even you are calling the builit-in pca function in matlab, it is much faster to use the SVD option. When projecting to the principal components to compute your features, remember to subtract the mean from the original data before the projection.

Write a report to explain the method and your experiments.

Please see below step-by-step guideline for doing PCA on face images:

Data (you don't need to report result on this dataset; it is provided to you to guide you through the basic pca procedure):
https://sites.google.com/site/ucsdcogs185spring2016/assignments/assignment4/Face_40by40_500.mat

This .mat file consists of a 1600x500 matrix, facemat, storing 500 faces of 100 celebrities, 5 faces from each. Each face is an image of size 40x40 and we have reshaped each image into a COLUMN vector of length 1600.

Matlab also has built-in function "pca()" to perform principal component analysis. To call "pca()", you can use facemat directly (no need to compute zero-mean matrix Z).

Try to use the 'SVD' option when calling the pca function.

[COEFF, SCORE] = pca(X, 'Algorithm', 'svd')

To understand how PCA works, you can perform pca using steps listed below. Note that computing eigen vectors on the convariance matrix to obtain principal components is a less preferred route as singular value decomposition (SVD) on the original space is much faster and effective.

1. Create a zero-mean matrix "Z" from "facemat".

Z = facemat - repmat(MeanFace, [1, size(facemat,2)]);

2. Calculate the covariance matrix.

C = Z*Z'/size(facemat,2);

3. Calculate the eigenvector and eigenvalues of C, using matlab function "eig()". [V, D] = eig(C);

V is a matrix with each column being one eigenvector.

D is a diagonal matrix and each diagonal element is the corresponding eigenvalue.

4. Sort the eigenvectors based on the magnitude of their corresponding eigenvalues.

[sv si] = sort(diag(D),'descend');

Vs = V(:,si);