Advanced Topics in Machine Learning (600.692)
Homework 3: Face Recognition with Varying Illumination

Instructor: René Vidal

Due Date: 03/07/2014, 11.59PM Eastern

READING MATERIAL: Chapter 2 and Appendix B.4 of GPCA book.

1. **Implementation of PCA, PPCA and model selection techniques.** Implement the following functions using at most 5 lines of MATLAB code per function.

Function `[mu,Ud,Y]=pca(X,d)`

Parameters

`X` $D \times N$ data matrix.
`d` Number of principal components.

Returned values

`mu` Mean of the data.
`Ud` Orthonormal basis for the subspace.
`Y` Low-dimensional representation (or principal components).

Description

Finds the d principal components of a set of points from the SVD of the data matrix X.

Function `[mu,Ud,sigma]=ppca(X,d)`

Parameters

`X` $D \times N$ data matrix.
`d` Number of principal components.

Returned values

`mu` Mean of the data.
`Ud` Basis for the subspace (does not need to be orthonormal).
`sigma` Standard deviation of the noise.

Description

Finds the parameters of the PPCA model $\boldsymbol{\mu}$ and $\Sigma = U_d U_d^\top + \sigma^2 I$.

Function `d=pca_model_selection(X,tau)`

Parameters

`X` $D \times N$ data matrix.
`tau` Threshold

Returned values

`d` Number of principal components.

Description

Finds the number of principal components for PCA as $\widehat{d} = \min_d \left\{ d : \sigma_{d+1}^2 < \tau \right\}$.

Function `d=ppca_model_selection(X,method)`

Parameters

`X` $D \times N$ data matrix.
`method` BIC, AIC, G-AIC

Returned values

`d` Number of principal components.

Description

Finds the number of principal components using different model selection methods.

2. **Face recognition using PCA and PPCA.** In this exercise you will use a small subset of the Yale B dataset[1], which contains photos of ten individuals under various illumination conditions. Specifically, you will use only images from the first three individuals under ten different illumination conditions.

   Download the file YaleB-Dataset.zip. This file contains the image database along with the MATLAB function `loadimage.m`. Decompress the file and type `help loadimage` at the MATLAB prompt to see how to use this function. The function operates as follows.

   Function `img=loadimage(individual,condition)`

   Parameters

`individual`	Number of the individual.
`condition`	Number of the image for that individual.

   Returned values

`img`	The pixel image loaded from the database.

   Description

   Read and resize an image from the dataset. The database (directory `images`) must be in the same directory as this file.

   (a) Apply PCA with $d = 2$ to all 10 images from individual 1. Plot the mean face $\boldsymbol{\mu}$ and the first two eigenfaces $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. What do you observe? Plot $\boldsymbol{\mu} + y_1\boldsymbol{u}_1$ for $y_1 = -\sigma_1 : 0.2\sigma_1 : \sigma_1$ and $\boldsymbol{\mu} + y_2\boldsymbol{u}_2$ for $y_2 = -\sigma_2 : 0.1\sigma_2 : \sigma_2$. What do the first two principal components capture? Repeat for individuals 2 and 3.

   (b) Apply PPCA with $d = 2$ to all 10 images from individual 1. Plot the mean face $\boldsymbol{\mu}$ and the first two eigenfaces $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. What differences do you observe between the eigenfaces of PCA and those of PPCA? Plot $\boldsymbol{\mu} + y_1\boldsymbol{u}_1$ for $y_1 = -1 : 0.2 : 1$ and $\boldsymbol{\mu} + y_2\boldsymbol{u}_2$ for $y_2 = -1 : 0.2 : 1$. What differences do you observe between the principal components of PCA and those of PPCA? Repeat for individuals 2 and 3.

   (c) Divide all the images in two sets: *Training Set* (images from individuals 1 to 3 and images 1-5) and *Test Set* (images from individuals 1-3 and images 6-10). Apply PCA to the *Training Set* using $d = 10$. Plot the mean face and the eigenfaces. Plot also the singular values of the data matrix. Project the *Test Set* onto the face subspace given by PCA, i.e., $Y_{test} = U_d^\top(X_{test} - \boldsymbol{\mu}\mathbf{1}^\top)$. Plot the projected faces, i.e., $\text{Proj}(X_{test}) = \boldsymbol{\mu}\mathbf{1}^\top + U_d Y_{test}$. Classify these faces by using 1-nearest-neighbor, that is, label an image $\boldsymbol{x}$ as corresponding to individual i if its projected image $\boldsymbol{y}$ is closest to a projected image $\boldsymbol{y}_j$ of individual i. Report the percentage of correctly classified face images for $d = 1, \dots, 10$. Which value of d gives the best recognition performance? Compare that with the results of doing model selection to determine the number of principal components for some threshold τ as well as with the estimates of BIC, AIC and G-AIC for PPCA.

Submission instructions. Please follow the same instructions as in HW1.

[1] `http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html`.