Prototype Exercise 3

Exercises 11, 12, 13
CSC872
Pattern Analysis and Machine Intelligence

https://bidal.sfsu.edu/~kazokada/csc872/FaceClassification_Data.zip

Download it!

Fast Prototyping Exercise

- **Fast Prototyping**
  - **Learn how to do a quick proof of concept by building a prototype**
  - **Correctness matters (no sloppy algorithm!)**
  - **Speed matters (no beautification!)**
  - **No perfect SE necessary**
  - **No copying of codes online.**
  - **Parameterization/Visualization/Experimentation**
    - Find out what are **free parameters** in your algorithm whose value must be hand-picked by you
    - Learn how to view internal variable's current values
    - Learn how to visualize your prototype's results in plots/images etc
    - Tweak the parameter values and study your prototype's behavior to understand the how algorithm works
- **Group Work**
  - **You are encouraged to freely exchange ideas and codes**
  - **Contributions to others are as valuable as making your own work**

Fast Prototyping Exercise

- **Please upload your matlab codes thru iLearn forum for my grading and your playing!**
  - **First two exercises: Due on midnight of the day (just what you did during the exercise)**
  - **Third last exercise: Due on midnight next day (complete version with some doc/screen shots of running the code)**
- **Your grade on FP exercise will be partly based on these submitted codes and what I observe during the in-class exercises.**
- **If received helps from others and/or used codes from others, please credit the person who helped you.**

Platforms

- MATLAB
  - MathWorks: http://www.mathworks.com/
  - http://en.wikipedia.org/wiki/MATLAB
- MATLAB @ SFSU
  - https://at.sfsu.edu/at-mathworks-matlab
- Various tutorials available online
  - https://matlabacademy.mathworks.com/?s_tid=acb_tut

Public Libraries

- OpenCV (Computer Vision)
  - http://www.intel.com/technology/computing/opencv/overview.htm
- ITK (Medical Imaging)
  - http://www.itk.org/
- WEKA (Machine Learning)
  - http://www.cs.waikato.ac.nz/~ml/weka/index.html

Classification of Facial Gender

- Learn an LDA classifier that classify input image to either female or male (binary classification)
- Smaller sized images are provided

Paper 3

- **Swets and Weng**
- **Using Discriminant Eigenfeatures for Image Retrieval**
- **PAMI, 18(8): (1996)**

- **A.M. Martinez, A.C. Kak,**
- ***PCA versus LDA*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2): 228-233 (2001)**
- http://www2.ece.ohio-state.edu/~aleix/

Data

- The same but reorganized data from the FP#1 for face recognition
- https://bidal.sfsu.edu/~kazokada/csc872/FaceClassification_Data.zip
- Images are organized in 2 folders
- Female: 54 32x32 8bit facial images
- Male: 45 32x32 8bit facial images

Linear Discriminant Analysis

- Supervised learning for classification
- Input: Grayscale Image
  - 8bit (0-255) grayscale images of 32x32 size
- LDA projection: linear transformation to a 1D space
- Threshold-based Classification: find the Bayes optimal threshold for the data after projection.
- Output: binary class labels
  - E.g., female (+1), male (-1)

LDA Setting

- Feature extraction (projection): $\boldsymbol{y} = \boldsymbol{w}^T \boldsymbol{x}$
- LDA: Find $\boldsymbol{w}$ such that within-class scatter ($\boldsymbol{S}_w$) is minimized and between-class scatter ($\boldsymbol{S}_b$) is maximized.
- LDA solution is given by solving a generalized eigen-value problem
  - $\boldsymbol{S}_b \boldsymbol{w} = \lambda \boldsymbol{S}_w \boldsymbol{w}$
  - [V D] = eig(S_b,S_w)

LDA Limitations

- You get only ***K-1*** non-zero eigen-vectors where K is the # of classes
- You need at least ***K+d*** samples to have non-singular S_w where ***d*** is the dimensionality of inputs
  - Singular matrix cannot be inverted!!!

PCA+LDA Solution

- Because S_w is singular you cannot solve the generalized eigenvalue problem
- Soln:
  - First Perform PCA on the entire data set
  - Find a subset with ***K*** top ***PCs*** that capture most of data variance
  - Project all the input data points to the PCA subspace
  - Compute the S_w and S_b with the projected datapoints in the PCA space (low-dimensional space)
  - Perform LDA
  - Compute the slope and intercept of the discriminant function from the LDA results

Formulae

$\boldsymbol{X} = \{\boldsymbol{X_1}, \boldsymbol{X_2}, \dots, \boldsymbol{X_k}, \dots, \boldsymbol{X_K}\}$

$M_k = |X_k|$: the number of sampels in k–th class

$\boldsymbol{\mu_k} = \frac{1}{M_k} \sum_{i=1}^{M_k} \boldsymbol{x_i}$: mean of k–th class

$\boldsymbol{\mu} = \frac{1}{|X|} \sum_{k=1}^{K} M_k \boldsymbol{\mu_k}$: mean of X

$$\boldsymbol{S_w} = \sum_{k=1}^{K} \sum_{i=1}^{M_k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^t$$

$$\boldsymbol{S_b} = \sum_{k=1}^{K} M_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^t$$

LDA: find w that maximizes $J(\boldsymbol{w}) = \frac{\boldsymbol{w^t S_b w}}{\boldsymbol{w^t S_w w}}$

Formulae

LDA: find w that maximizes $J(\boldsymbol{w}) = \frac{\boldsymbol{w^t S_b w}}{\boldsymbol{w^t S_w w}}$

$$\boldsymbol{S_b w}_m = \lambda_m \boldsymbol{S_w w}_m$$

$$(\boldsymbol{S_w^{-1} S_b}) \boldsymbol{w}_m = \lambda_m \boldsymbol{w}_m$$

Algorithm

1) Given labeled data X = *{Xp,Xn}:* female (*p*), male (*n*)
2) Do PCA $\boldsymbol{CV} = \boldsymbol{VD}$ $\boldsymbol{V} = \{\boldsymbol{PC_1}, \ldots, \boldsymbol{PC_N}\}$ $\boldsymbol{D} = \mathrm{diag}(ev_{\boldsymbol{1}}, \ldots, ev_N)$
3) Find top *K* PCs that cover 9X% of the variance
4) Form PCA model *F* $\boldsymbol{F} = \{\boldsymbol{PC_1}, \ldots, \boldsymbol{PC_K}\}^{\boldsymbol{t}}$
5) Project all data points to *FX={Pp,Pn}:*
6) **Compute S_w and S_b from {*Pp,Pn*}** $\boldsymbol{p}_i = F * (\boldsymbol{x}_i - \mu)$
7) **Solve a generalized eigen-value problem with S_w and S_b**
8) **This results in a single PC: *v***
9) Compute discriminant slope *w* = *v' * F*
10) Compute discriminant intercept ***b* = *v*'*(*meanPp*+*meanPn*)/*2***
11) **The result if *w**(*x*-μ)-*b* is positive then *(+1)* otherwise (-*1*), (check if the sign is right)**

Useful MATLAB Codes

For LDA

- Set *X* as a matrix with each row is a vectorized face
- *m* = mean(*X*): sampel mean of *X*
- *S* = cov(*X*): covariance matrix (mean removed)
- Scatter matrix = cov(*X*).*(*N-1*) *N*: # of samples in *X*
- [*V D*]=eig(*A*,*B*): generalized eigenvalue problem solver
- hist (projected *X1* and *X2*): create a histogram