**Technische Universität München**
**Fakultät für Informatik**
**Prof. Nassir Navab**

**WS 15/16**
**Exercise Sheet 2**
**October 27, 2015**

# Exercises in Machine learning in Medical Imaging

## Exercise 1 Subspace Methods

In this exercise you will familiarise with subspace methods. These methods allow to obtain a lower dimensional representation of the dataset at hand that retains desirable properties. For example, using PCA it is possible to project the data into a linear subspace, through an orthonormal basis, while retaining the most important modes of variation of the data itself. In other words, when applied to images for example, PCA is capable of finding relationships between neighbouring pixels that vary together and therefore it can fuse their content in order to preserve the main modes of variation of the data even after projection into the subspace.

When data belongs to different classes PCA does not constitute a good choice as a subspace method. Even data that is linearly separable could be projected into a subspace where this desirable characteristic is lost. It is possible, in this case, to use LDA (Linear Discriminant Analysis) which can be used to find a linear orthonormal base that projects the data into a subspace where data-points belonging to the same class are clustered together and data-points that belong to different classes are pushed as far apart as possible.

In this exercise you are required to implement the two classes defined in the files myLDA.m and myPCA.m (Figure 1). For you comfort and amusement you are provided with three other functions that show how to use myLDA and myPCA and showcase their main capabilities. These files are compareMnistPCALDA.m, doFacesLDA.m (pictures of people wearing glasses belong to class=1) and doFacesPCA.m. You need to download additional data, from the course website, that comes in form of compressed file which should be unpacked in a folder called "Data" inside the exercise folder (Eg. "EX3"). It is very important that you go through these functions and you understand what they do and how the function. If you have problems running the function compareMnistPCALDA.m try to decrease the number of data-points that you use for training and testing.

## Exercise 2 PCA

The implementation of PCA requires you to complete the functions "myPCA" which is the class constructor and the other member functions as specified in the following.

The constructor realises the core of the method by implementing the following steps:

a) Compute the mean $m$ of the data-points and store it.

b) De-mean the dataset $D$ containing $n$ data-points.

c) Compute the SVD (singular value decomposition, MATLAB implementation) $D - m/\sqrt{n-1}$ and obtain the matrices $U$, $S$ and $V$.

d) The matrix $U$ can be ignored, the matrix $S$ contains the eigenvalues which should be squared and stored in a vector as variances associated to the eigenvectors that are contained in $V$. The sign of some eigenvectors is changed as prescribed by the relative convention (this part of the code is provided by us).

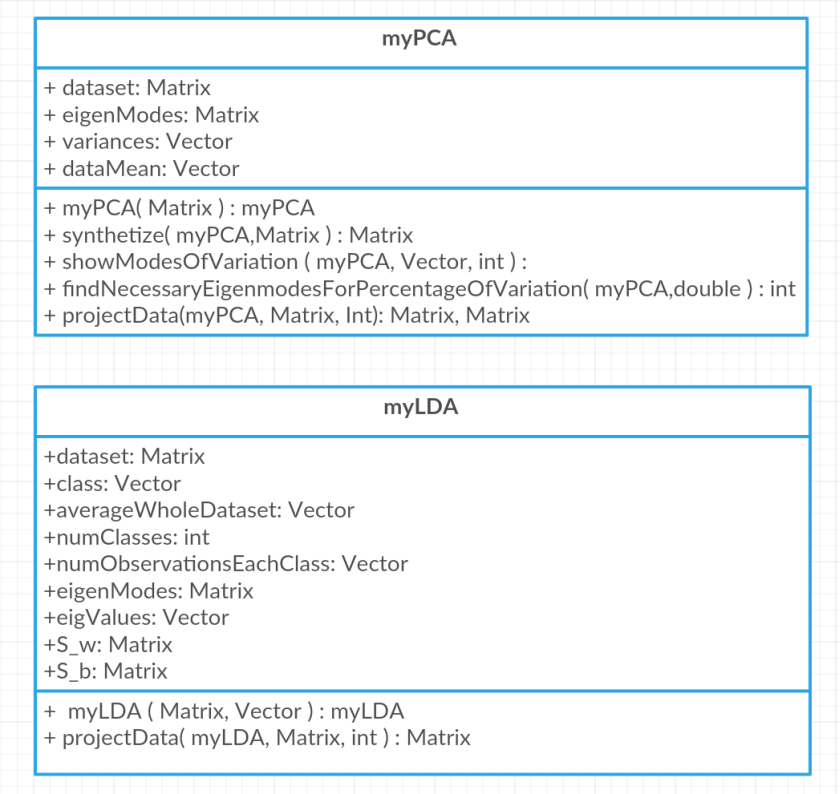


Figure 1: Class diagram for myPCA and myLDA.

The other member functions of the class myPCA are:

a) "projectData" which, given a dataset $D$ having $n$ data-points in $d$ dimension, and the number of desired output dimensions $d'$, projects $D$ through the eigenvectors in the $d'$ dimensional subspace spanned by the first $d'$ eigenvectors $E'$ discovered through PCA. The function returns "Dnew" which are the data-points as they appear in the subspace (they have a dimensionality $d'$) and "weights" which were used to get such projection. These solutions are sought by using the equations:

$$D_{new} = (D - m)^T * E'; \text{and} w = E^T * (D - m); \tag{1}$$

b) "synthetize" which, given a set of weights $w$ having $d'$ rows and $n$ columns, reconstruct a $d$ dimensional set of $n$ signals by selecting $d'$ eigenvectors $E'$ and computing:

$$D_{legal} = m + E' * w; \tag{2}$$

c) "findNecessaryEigenmodesForPercentageOfVariation" which uses the variances, which can be obtained as the squared eigenvalues, to discover how many eigenvectors are necessary to preserve a certain percentage of the data variance (supplied as an argument).

d) "showModesOfVariation" is a function that shows the first $p$ modes of variation. You are not required to implement this function as it is provided only for your amusement. Please refer to the comments in the code to discover its usage.

## Exercise 3 LDA

The implementation of LDA requires you to complete the functions "myLDA" which is the class constructor and the function "projectData".

The constructor of the class implements the core of this subspace method:

a) Compute the d-dimensional mean vector for the whole dataset.

b) Divide the dataset in sub-datasets according to the classes

c) Compute the d-dimensional mean vectors $m_i$ for the different classes from the dataset, and store the number of data-points that we have for each class $N_i$.

d) Compute the scatter matrices (between-class and within-class scatter matrix):

$$S_b = \sum_{i=1}^{classes} N_i * (m_i - m)(m_i - m)^T \tag{3}$$

$$S_w = \sum_{i=1}^{classes} S_i \text{ and } S_i = \sum_{x \in class_i} (x - m_i)(x - m_i)^T. \tag{4}$$

where $m_i$ is the within class mean vector that you computed before, and $m_{all}$ is the mean vector of the whole dataset.

e) regularise the matrix $S_w$ by summing to it the identity matrix scaled by a constant and compute its inverse obtaining $S_w^{-1}$

f) compute v=$S_w^{-1} * S_b$.

g) Compute the eigenvectors $[e_1, e_2, ..., e_d]$ and corresponding eigenvalues $[?_1, ?_2, ..., ?_d]$ for the matrix $v$. This solution is sought using SVD in the exercise. The eigenvectors are the new orthogonal base that you can use to project the data into the subspace.

The function "projectData" takes as input argument the object itself, the dataset $D$ that you want to project, and the number of eigenvectors $k$ you want to use for this projection. The function returns $D_new$ having as many columns ad $D$ and $k$ rows. The projection happens by first selecting the first $k$ eigenvectors and by multiplying $D^T$ with the new, reduced, set of eigenvectors.

$$D_{new} = D^T * E'. \tag{5}$$