1st Exercise on Medical Image Processing

LU 183.630 - 2016SS

21.4.2016

Questions to: Markus Krenn: `markus.krenn@meduniwien.ac.at`,
Markus Holzer: `markus.holzer@meduniwien.ac.at`

1 Submission guidelines

- One submission per team, per mail to `markus.krenn@meduniwien.ac.at`
- Deadline: May 25th, 12pm.

Submission content:

- Executable code as `.zip`, filename: `Abgabe-PCA-XX.zip`, where `XX` holds your group number.
- A script `RUN.m`, which computes all results and plots without user interaction.
- The code has to be documented including information on which exercise is solved in which function or file.
- A PDF report that contains explanation and interpretation of all exercises (approximately 3 - 5 pages).
- The reports front page contains group number and team member names.
- The report has to be short and precise, (do not include code).
- The report can be written in german or english.
- Questions regarding the exercise / feedback to your code or parts of your report: during laboratory sessions or per mail.

2 Assignment

Aim of the first lab exercise is to get familiar with Matlab, the implementation of principal component analysis (PCA) and the investigation of its characteristics. Parts of the code (data, plot functions) are provided so that students can focus on the main topics of the assignment.

2.1 Helper functions

`plot2DPCA.m` and `plot3DPCA.m` can be used to plot data points, Eigenvectores & Eigenvalues, ellipses & ellispoidal and reconstructed data. `plotDEMO.m` demonstrates their usage , `open plot2DPCA.m` shows the respective documentation.

Useful Matlab functions for this assignment: `clear`, `close all`, `load`, `plot` (use the parameter `'.'` to plot dots), `axis equal`, `bar`, `imagesc`, `repmat`, `mean`, `var`, `std`, `sort`, `mvnrnd` and `eig`. Documentation and example usages of built-in Matlab functions can be accessed with the command `doc functionname`.

2.2 Fragestellung

Maximum number of points per subtask are denoted in brackets (30 in total).

In this exercise, data points are given in a $d \times n$-Matrix $\mathbf{D}$, where n denotes the number of points and d their dimensionality, i.e. $2 \times n$ for a set of 2D points and und $3 \times n$ for points in 3D.

1. **Covariance matrix**

   (a) Implement a function `ourCov.m`, that computes the covariance matrix $\mathbf{C}$ of a data matrix $\mathbf{D}$. Using the matlab built-in function `cov` for implementation is prohibited but it can be used to compare your results (`cov` takes a $n \times d$ matrix as input). **(2 points)**

   (b) Compute $\mathbf{C}$ for all matrices in `daten.mat`. Visualize the data utilizing `plot` in separate figures and set the axis to equal scale (`axis equal`). Interprate the resulting covariance matrices $\mathbf{C}$ of all datasets, Which position in $\mathbf{C}$ holds which information? **(2 points)**

2. **PCA** – Implement a function `pca.m`, that computes the PCA for a data matrix $\mathbf{D}$. The computation has to be independent from the dimensionality of the data. The function should return the Eigenvalues in descending order as well as the corresponding normalized Eigenvectors (sorted according to the Eigenvalues!). You can use the Matlab function `eig` to compute Eigenvectors and values. **(2 points)**

(a) use `plot2DPCA.m` to plot results for all matrices of `daten.mat`. **(1 point)**

(b) What do Eigenvectors represent, where can one see that information in the plot? **(1.5 points)**

(c) What do Eigenvalues represent, where can one see that information in the plot? Is there a relation to the total variance of the data? **(1.5 points)**

(d) How does a missing substraction of mean values (on $\mathbf{D}$) affect the computation? **(1 point)**

3. **Subspace projection**

   (a) Perform PCA on the data in `data3`. Project the data in `data3` to the main vector (plot). What's the datas dimensionality now? Reconstruct the projection and plot the result using `plot2DPCA.m`. Describe the effect of projection and reconstruction regarding the data points. What's the average error of reconstructed compared to original data points. **(3 points)**

   (b) Perform a similar investigation using the side vector (not the main vector!). Which vector would you use to obtain a reconstruction of the data with minimal error using as little vectors as possible (1 vector in this case)? **(1 Punkt)**

4. **Investigation in 3D**

   (a) Perform PCA and plot the data and Eigenvectors of the data in `daten3d.mat`. Describe the relation between covariance matrix, Eigenvalues, Eigenvectors and the ellipsoidal of standard deviations. **(2 Punkt)**

   (b) Project the data in the subspace constructed by the two first Eigenvectors. What is the dimensionality of the data now? Reconstruct the points in the original space and plot the result. What type of information has been lost? **(1 point)**

5. **Shape modeling**

   (a) Perform PCA on the shape data stored in `shape.mat` – the matrix `aligned` is composed as follows `nPoints x nDimensions x nShapes`. Implement a function `generateShape`, that takes a parameter vector `b` as input and computes new shapes, where the length of `b` indicates the number of Eigenvectors to be considered for shape generation. **(4 Punkt)**

(b) Implement the function `plotShape`, that plots shapes (in blue) and visualize and interprate individual *modes*. (d.h. `b` $= 0$ except of one entry) in the range of $\pm 3\lambda$, where λ denotes the standard deviation of a specific mode. The function should also plot the mean shape in red. Describe and interprate that task in your report. **(4 Punkt)**

(c) Set `b=randn(1,nEigenvectors).*stddeviations`. Restrict the number of Eigenvectors with the length of `b` similar to the tasks with 2D and 3D data points. Plot the resulting shapes and interprate. Finally restrict the number of eigenvectors so that the shape model captures 100%, 95%, 90% and 80% of the total variance **(4 Punkt)**.