Unsupervised Machine Learning - Project 1

Principal component analysis and factor analysis

Deadline: Tuesday 1st of April (midnight the latest).

Instructions

General description

The project consist of several exercises, in which students are asked to implement unsupervised-machine-learning methods from the course Unsupervised Machine Learning using Matlab or R. Using the implemented algorithms, students are also asked to answer a number of questions and write a report presenting and discussing the results. These skills are extremely important to understand the benefits and limitations of existing algorithms, and how to customize them in real applications. The course also teaches scientific reporting skills, fundamental for communicating the results obtained with existing methods and the rationale behind improvements and breakthroughs.

Instructions for solving the exercises

1. Each exercise can be solved by students working individually or in pairs (not more than two students). When working in pairs, only one report and group of source codes per exercise per pair is enough.

2. Results mostly consist of a figure or a short program output. In both cases, add a short paragraph explaining what is being shown and why the results are the answer to the question.

3. Each figure in the reports must be accompanied by the corresponding source code for reproducing the results.

4. All questions must be answered.

5. Whenever the exercises ask you to discuss a result, the result must be discussed. If you have no idea what to discuss, please state it in the report or ask me for some hints before submitting.

6. Whenever the exercises ask you to test whether an algorithm works, always give an example which shows that the algorithm indeed works (or not).

7. Source codes must include comments reflecting the connection between the code lines and the math underlying the implemented method.

Instructions for writing the report

1. Reports must be written in English. Aim to write in either American or British spelling, but avoid using both in the same report, such an inconsistence is generally considered bad practice.

2. For each exercise, students are required a separate report as a single pdf containing both results and figures. For each project, there must be as many reports as exercises.

3. Reports must include a title page with the following information:

a) Project title

b) Exercise number

c) Name, student ID number, affiliation (for example, Department of Computer Science) and academic level (for example, Bachelor student, Master student or PhD student), of each student involved in the project.

4. All figures must be numbered and include a short caption summarizing the results shown.

5. Figures should be positioned immediately after they are first mentioned in the text.

6. All axes in figures must have names and labels.

7. All abbreviations and acronyms must be defined the first time they are used in the text.

8. Because algorithms in machine learning are applied in a wide variety of disciplines, reports should be written clearly and simply so that they are accessible to a wide audience. Avoid the use of specialized terms (jargon) or explain them concisely (but not didactically) when unavoidable.

Completion of the project

1. The project will be completed once all exercises are submitted to the following email address: hugo.eyherabide@helsinki.fi.

2. All exercises must be submitted before the deadline indicated above. Failure to do so will yield a reduction of 5% per day of the points corresponding to the missing exercises.

3. Each exercise must be submitted as a single zip file containing both the pdf file with the report and the source codes of the implementation.

4. Exercises can be submitted individually after completion. Feedback may be given immediately after together with a chance to correct mistakes if desired, but grades will be given after the deadline and submission of all exercises.

5. Each exercise in the project gives you an equal number of points. They will, at the end, determine your grade for the computer project.

6. Grades are based solely on the reports: All your results should be stated in the report. Codes are only taken into account if problems are detected in your results. In order to get a good grade, reports must be clear and enjoyable to read.

Exercise 1: Basic principal component analysis (PCA)

*In this exercise, you will need to compute eigenvalue decompositions. In Matlab, this is done with the command **eig**, and in R, with the command **eigen**. Note that in Matlab, eigenvalues (and their corresponding eigenvectors) are sorted in ascending order, whereas in R, they are sorted in descending order.*

1. Create artificial data as shown in Figure 1 (take 1000 sample points).The data needs to be just qualitatively similar. Make a scatter plot as in the figure, and explain how you created the data. Especially, say how you control the "direction" of the data, and its spread.

2. Write an implementation of principal component analysis capable of handling data of any dimension (that is, not for only two-dimensional data, but not for large-dimensional data either).

3. Do PCA for two of the four scatter plots constructed in exercise 1 question 1. Plot the principal components (PC) directions on the top of the scatter plots. The figures should look similar to figure 4.1 on page 30 of the lecture notes (but show both principal component directions).

4. For one of the four scatter plots, project the data on each of the two PC directions, and make a histogram (20 bins worked fine for me). Compute the variance of the projections to each PC direction and say how the variances relate to the covariance matrix of the data.

5. Create artificial data which has the vectors

$$\mathbf{v}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\end{bmatrix}^T \quad \mathbf{v}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix}-1 & 1\end{bmatrix}^T$$

   as PC directions with PCs having variance 1 and 3, respectively (take 1000 sample points). Make a scatter plot of the data and compute its covariance matrix. What are the eigenvectors and eigenvalues of the covariance matrix?

6. Reduce the dimension of this data to 1 so that the reconstruction error is minimized (sections 4.2-4.3 of lecture notes). For the 50 first data points, make a scatter plot which shows both the original points and their approximation (like in Figure 2). How large is the average reconstruction error (take the average over all data points, not only the 50 first ones)? How large is the proportion of variance explained? What is the relation between average reconstruction error and proportion of variance explained?

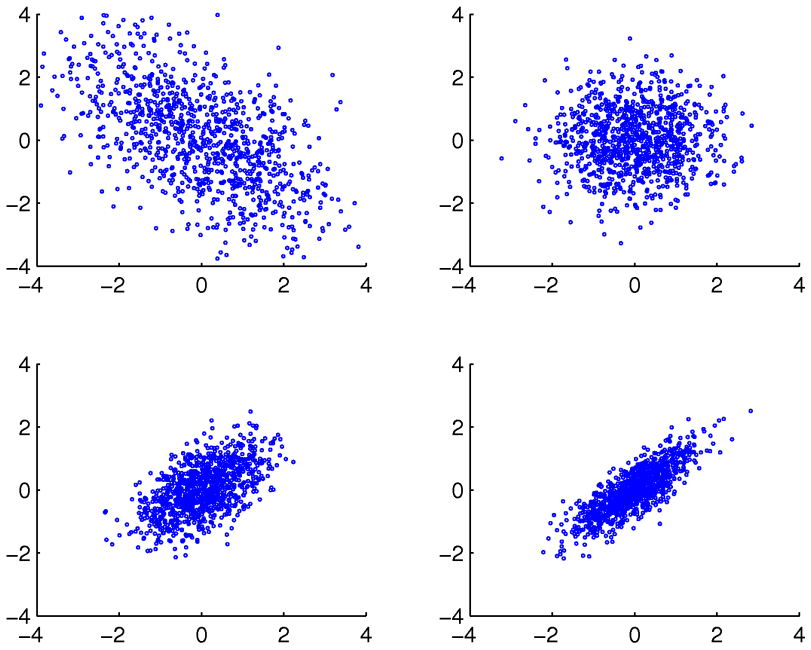

Figure 1: Scatter plots for exercise 1. They are of the same type as the scatter plot in figure 4.1 on page 30 of the lecture notes. The data is Gaussian with different covariance matrices. Every scatter plot shows 1000 data points.

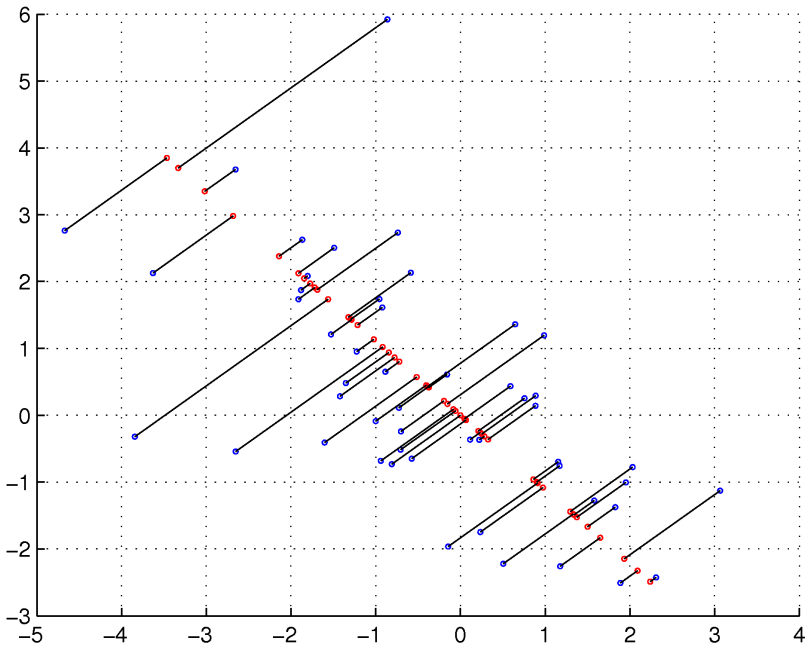

Figure 2: Sample solution for exercise 1 question 6. Original data in blue, approximation in red. The black lines show the correspondence.

Exercise 2: Principal component analysis and factor analysis

1. Reproduce figure 4.2 on page 34 of the lecture notes (numbers are not necessary and can be replaced by circles, as shown below in figure 3). Explain the figure in your own words. What decides the length of the red lines?

2. Make a plot which shows the proportion of variance explained as a function of the number of principal components (see section 4.3.3 on page 33 of the lecture notes).

3. Implement the quartimax rotation for factor analysis (FA), which is explained in section 5.5 on page 38 of the lecture notes. You can use gradient optimization and the results obtained in the exercise session of the course Unsupervised Machine Learning to derive the update rule for the quartimax rotation. Notice that this is constrained optimization because the matrix $\mathbf{U}$ must be orthonormal.

4. Test your implementation: does the algorithm bring you from equation 5.12 to equation 5.13 in the lecture notes? Plot the value of J_{rot} during the optimization process. Notice that J_{rot} in equation 5.14 with G defined by equation 5.15 is invariant under a change of sign of all the elements of $\mathbf{U}$. Therefore, your implementation may yield $\mathbf{A}$ or $-\mathbf{A}$, being both correct.

5. Now you are ready to reproduce figure 5.1 on page 39 of the lecture notes. Visualize $\mathbf{A}$ before and after the quartimax rotation. Explain the figures in your own words. Notice that if your implementation yields $-\mathbf{A}$ (instead of $\mathbf{A}$), the new coordinates will be flipped compared to those in the lecture notes. That figure is correct.

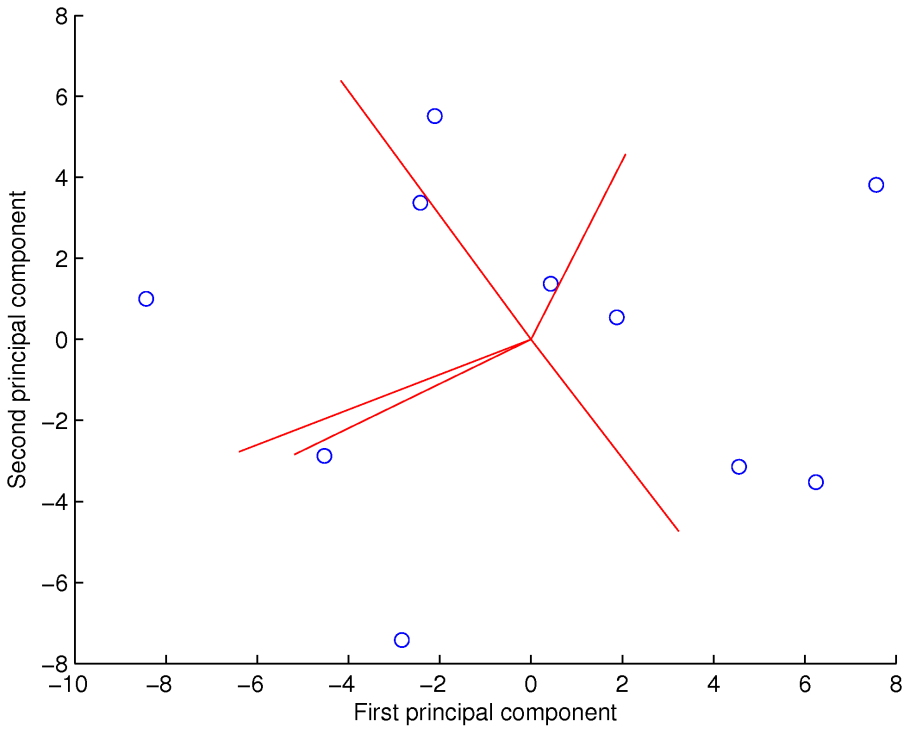

Figure 3: Reproduction of figure 4.2 on page 34 of the lecture notes.

Exercise 3: Denoising and compression by PCA and FA

For this exercise, you will use images of handwritten digits from the MNIST database. You can download the database from the homepage. The file `digits.txt` contains data in the form of a matrix of size 784×1000 in which each column represents an image of a handwritten digits of 28×28 pixels.

1. Load the data. For preprocessing, remove the average from the images, and normalize each image to unit norm. Then, center the data by removing the mean of each of the 784 random variables. Choose 100 digits and visualize them before and after the preprocessing. Visualize the mean of each of the 784 random variables. You can use the provided functions `visual.m` (for Matlab) and `visual.R` (for R). For example, for visualizing the first 100 digits, run `visual(digits(:,1:100),1,10)` in Matlab or `visual(digits,10)` in R.

2. Perform PCA on the data. Show the proportion of variance explained in function of the number of principal components. Show the first 20 principal component directions. How much variance do they explain?

3. Choose 10 digits you like. Project each digit on a 1, 2, 4, 8, 16, 32, 64, and 128 dimensional subspace spanned by the first principal component directions (see section 5.6 in lecture notes). Show the approximations and the original digits. Compare and discuss the result. For each dimension, what is the average reconstruction error? Explain how projecting on subspaces can be used for compression.

4. Use the computed mean and principal component directions to clean up the 100 numbers in the file `noisyDigits.txt`. Visualize the digits before and after cleaning up.