Unsupervised Machine Learning Projects
Project 1

Principal component analysis and factor analysis

Deadline: April 24th 2015 (midnight the latest).

Exercise 1: Basic principal component analysis (PCA)

Instructions

1. Create artificial data as shown in Figure 1 (take 1000 sample points) assuming that they come from a bivariate Gaussian distribution with zero mean and covariance matrix to be estimated from the figure.
   In the report: Explain how you estimated the covariance matrix and generated the data. Make a scatter plot as in the figure.

2. Write a function that computes PCA for data of any dimension (that is, not for only two-dimensional data, but not for large-dimensional data either), the dimensions arrange in different rows .
   In the report: Explain what the code does, and how that is related to the underlying theory of PCA (not more than half a page). Include the function in a separate file.

3. Using the function previously written, do PCA on the four data sets generated before.
   In the report: Plot the principal component (PC) directions on top of the scatter plots, denoting the first and second with different colours.

4. For the data corresponding to the top-left scatter plot in Figure 1, project the data on each of the two PC directions.
   In the report: Plot histograms of the projections on each PC direction using 20 bins. Compute the variances of the projections on each PC direction, and explain their relation to the covariance matrix of the data.

5. Create Gaussian-distributed artificial data with variance 3 along the PC direction $\mathbf{v}_1 = \sqrt{1/2}[-1, 1]^T$ and variance equal to 1 along the PC direction $\mathbf{v}_2 = \sqrt{1/2}[1, 1]^T$ (take 1000 sample points). Compute the population covariance matrix (that is, the covariance matrix used to generate the data) and sample covariance matrix (that is, the covariance matrix from the data itself), their eigenvalues and eigenvectors.
   In the report: Show a scatter plot of the data, including the eigenvector of the population and sample covariance matrix scaled so that their norm is equal to the standard deviation of the data in the direction of the eigenvectors. Compare the eigenvalues and eigenvector and explain the differences (less than half a page).

6. Reduce the dimension of this data to 1 so that the reconstruction error is minimized (section 4.3 of lecture notes).
   In the report: Compute the average reconstruction error, the proportion of variance explained, and explain how are they related (less than half a page). Show a scatter plot for the 50 first data points with both the original points and their approximations, similar to Figure 2.

Hints:

1. The standard deviation in any direction can be roughly estimated as a sixth of the spread of the data cloud in that direction. This is known as the 3-σ rule.

2. The direction of the cloud can be controlled using the inverse process of PCA, that is, by rotating a diagonal covariance matrix.

3. For doing PCA, you will need to compute eigenvalue decompositions. To that end, you can use, though not exclusively, `eig` (Matlab/Octave), `eigen` (R), or `numpy.linalg.eig` (Python).

4. Always remember to centre the variables before PCA and FA.

5. Note that different functions may or may not sort eigenvalues (and their corresponding eigenvectors) in ascending or descending order. You must take this fact into account when choosing the PCA components.

6. When computing variances and standard deviations, bear in mind the difference between unbiased estimators for populations and samples, and always check which one your function (or the function you are using) are actually employing in the calculations.

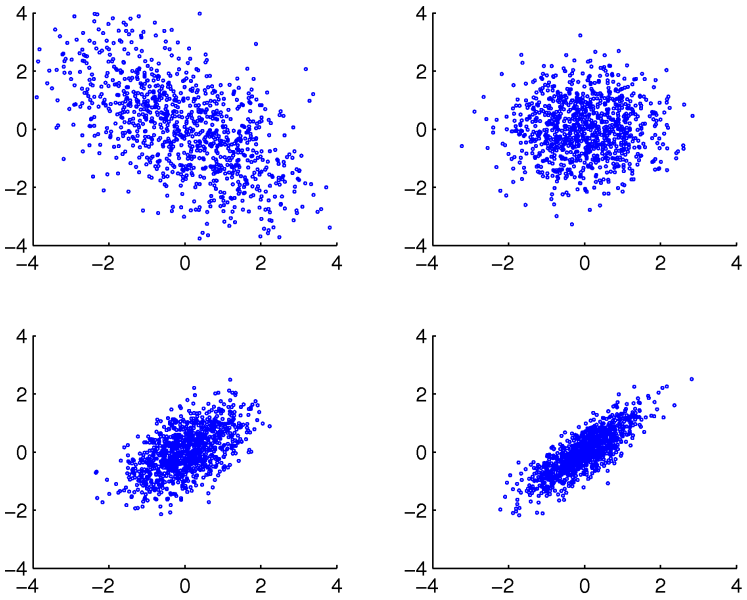

Figure 1: Scatter plots for exercise 1. They are of the same type as the scatter plot in figure 4.1 on page 30 of the lecture notes. The data is Gaussian with different covariance matrices. Every scatter plot shows 1000 data points.

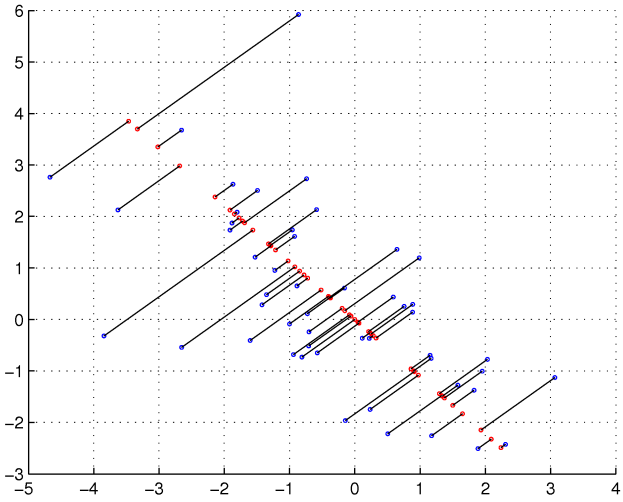

Figure 2: Sample solution for exercise 1 question 6. Original data in blue, approximation in red. The black lines show the correspondence.

Exercise 2: Principal component analysis and factor analysis

Instructions

1. Reproduce figure 4.2 on page 34 of the lecture notes, without the numbers as shown below in figure 3).
   In the report: Show the reproduced figure. Explain how to reproduce the figure in your own words, and the computation of the length of the red lines (less than half a page).

2. Compute the proportion of variance explained for the data matrix shown in equation 4.28 of the lecture notes as a function of the number of PCs (see section 4.3.3 on page 33 of the lecture notes).
   In the report: Show a plot of the proportion of variance explained as a function of the number of PCs.

3. Implement the quartimax rotation for factor analysis (FA) (section 5.5 of the lecture notes) using gradient optimization. Notice that this is constrained optimization because the matrix $\mathbf{U}$ must be orthonormal.
   In the report: Explain what the code does, and how that is related to the underlying theory (not more than half a page). Include the function in a separate file.

4. Test your implementation using equation 5.12 and compare it with equation 5.13.
   In the report: Show the output of your code and Explain how and why it may possibly differ from equation 5.13. Plot the value of J_{rot} during the optimization process.

5. Reproduce figure 5.1 on page 39 of the lecture notes using $\mathbf{A}$ before and after the quartimax rotation.
   In the report: Show both figures and explain them in your own words. Discuss why the latter figure may differ from figure 5.1 in the lecture notes (all text less than half a page).

Hints:

1. Always remember to centre the variables before PCA and FA.

2. Recall 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.

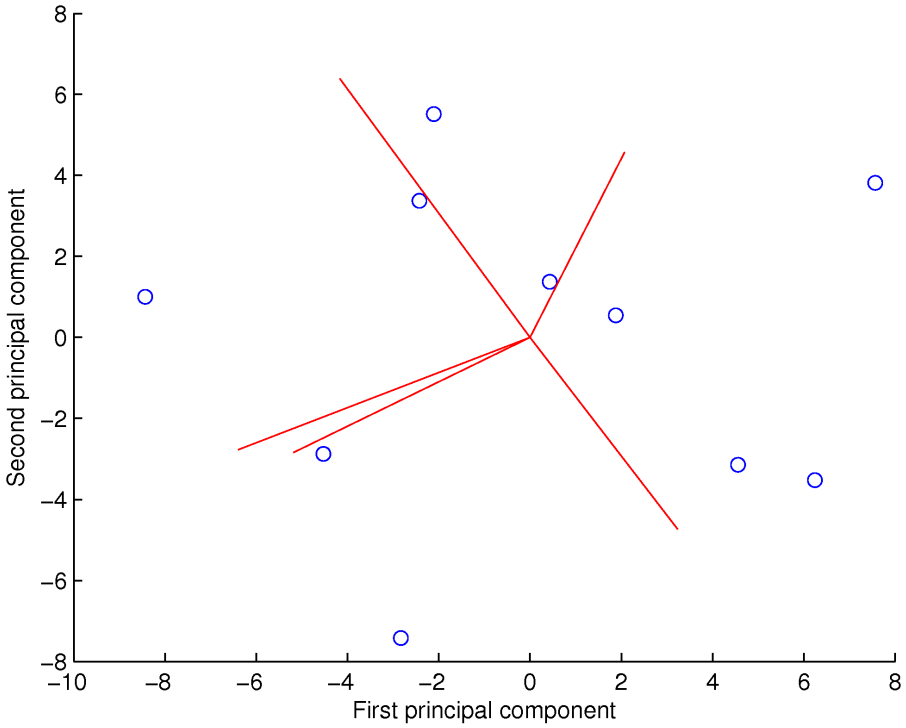

Figure 3: Reproduction of figure 4.2 on page 34 of the lecture notes.

Exercise 3: Denoising and compression by PCA and FA

For this exercise, you will use images of handwritten digits from the MNIST database. The file `digits.txt` contains data in the form of a matrix of size 784×100. Each column represents the image of a handwritten digit. Each image has a size of 28×28 pixels. The file `digitsnoisy.txt` contains the same digits with additional noise, and the file `noisesamp.txt` contains noise samples chosen from the same distribution than the noise added to the images.

1. Load the file `digitsnoisy.txt`, compute the mean in each pixel (rows) and remove it from all images, thereby centring the data.
   In the report: Visualize the first 16 digits before and after the centring, as well as the mean, using the provided functions `visual.m` (Matlab/Octave), `visual.R` (R), and `visual.py` (Python).

2. Perform PCA on the data using the function you developed.
   In the report: Plot the proportion of variance explained as a function of the number of PCs. Show the first 20 principal component directions using the provided function `visual`.

3. Project all 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). Load the file `digits.txt` containing the original digits without the added noise, and compute the reconstruction error with respect to both the noisy digits and the original ones. **In the report:** For the first 16 digits, show figures with the reconstructions and a figure with the reconstruction errors for both the original digits and the noisy digits as a function of the number of PCs (using the number of PCs indicated before). Explain how projecting on subspaces can be used for compression and denoising.

4. Project all digit on a 1, 2, 4, 8, 16, 32, 64, and 128 dimensional subspace spanned by the first factor loadings. To that end, load the file `noisesamp.txt`, compute the noise covariance matrix, subtract it from the covariance matrix of the noisy digits and compute the eigenvalues and eigenvectors (section 5.4.1 in the lecture noes). **In the report:** For the first 16 digits, show figures with the reconstructions and a figure with the reconstruction errors for both the original digits and the noisy digits as a function of the number of common factors. Compare the results with the ones obtained using PCA.

Hints:

1. Recall that when reconstructing using PCA, the mean subtracted before doing PCA must be added back after projecting onto the PCs.