UML computer project 1

Juha-Antti Isojärvi
013455341
Department of Mathematics and Statistics
Master student

Mikko Sysikaski
013573016
Department of Computer Science
Master student

1 Exercise set 1

1.1 Exercise 1

First we were given plots of twodimensional data, and were asked to create similar data artificially. The distribution of the data in the plots looked similar to the twodimensional gaussian or multinormal distributions. Therefore we decided to generate points from a gaussian distribution.

A random vector $X = (X_1, X_2, \ldots, X_n)$ is standard normally distributed, denoted $X \sim N_n(0, I)$, if its components X_i are independent and normally distributed with zero mean and unit variance. It is true, that the mean vector $E(X) = 0$ and covariance matrix $Cov(X) = I_n$.

Now if a random vector

$$X = AU + \mu$$

is defined for some $U \sim N_n(0, I)$, and fixed $A \in \mathbb{R}^{m\times n}$ and $\mu \in \mathbb{R}^n$, then it is true that $E(X) = \mu$ and $Cov(X) = AA^T$.

A random vector X is multinormally distributed, if it has the same distribution as the random vector

$$X = AU + \mu.$$

Then it has mean $E(X) = \mu$ and covariance $Cov(X) = AA^T$. Denote $Cov(X) \doteq \Sigma$. The multinormal distribution is denoted $X \sim N_n(\mu, \Sigma)$.

It also holds, that if $X \sim N_n(\mu, \Sigma)$, and $B \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ are fixed, then

$$BX + b \sim N_m(B\mu + b, B\Sigma B^T).$$

Now, one way of simulating gaussian twodimensional data, is to first simulate twodimensional standard normally distributed, i.e. white, data, and then affinely transform this data by multiplying with some matrix $A \in \mathbb{R}^{2\times 2}$. The affinely transformed data has then a gaussian distribution with zero mean and covariance AA^T.

So, we generated two random samples from a normal distribution and put these together to form twodimensional white data. If one would plot this data, one would see a spherical point cloud centered at zero. Then we dilated this cloud in the y- and x-axis direction with multiplying by a dilation matrix D and then rotated the dilated data cloud with multiplying by a rotation matrix R_θ. The result was a sample from the distribution $N_2(0, AA^T)$, where $A \doteq R_\theta D$.

The resulting data cloud is centered at zero, and has elliptical shape. The main axis of this data cloud has angle θ measured form the x-axis. The spread of the cloud is determined by the dilation matrix D. Our resulting four generated artificial data can be seen plotted in Figure 1.

This was a simple method for generating artificial twodimensional gaussian data, with easy control of the direction and elongation of the data cloud. Of course there are other methods as well. In the MASS-package of R there is a function rmvnorm, which takes as inputs the mean vector and the covariance matrix and generates gaussian data. We didn't feel the need to get further into the details of this method. Manipulation of the direction and the elongation of the data cloud is always achieved by manipulation of the covariance matrix, as in the method we described.

Another way of manipulating the direction and elongation of the data cloud is by 'inverse' eigenvalue decomposition of the covariance matrix. About this method more in section 1.4.

1.2 Exercise 2

Next, we were given the task to perform principal component analysis on two of the generated artificial data. We chose the data seen in Figure 1 in the upper left and lower left frame.

In order to do principal component analysis, we first need the covariance matrix of the data. In the context of PCA one doesn't have prior knowledge of the underlying distribution, and therefore we used the standard method of computing a sample covariance matrix. In R this can be done with the function cov.

Next we did eigenvalue decomposition of the covariance matrix. In the lecture notes it is proven that the first principal component is the eigenvector corresponding to the largest eigenvalue, the second PC the eigenvector corresponding to the second largest eigenvector, and so on. In this case of twodimensional data there are only two principal components. The directions of the principal components of the point sets are shown in Figure 2.

1.3 Exercise 3

In the third exercise of this set, we needed to project one of the artificial data we created on each of its principal components. Histograms of the projected data can be seen in Figure 3. The variance of the projected data should be equal to the eigenvalue of the corresponding eigenvector of the covariance matrix of the original data. This was verified after computing the variance of the projected data and the eigenvalue decomposition of the covariance matrix:

```
> var
[1] 4.020849 1.002429
> eig$values
[1] 4.020849 1.002429
```

1.4 Exercise 4

1.5 Exercise 5

2 Exercise 2

2.1

The task was to do primary component analysis on the matrix

$$X = \begin{pmatrix} 5 & 3 & 0 & 1 & -1 & -3 & 5 & 0 & -4 & -4 \\ -2 & -1 & 0 & 0 & 1 & 4 & -3 & 1 & 5 & 3 \\ 0 & 1 & 4 & -1 & 0 & 5 & 5 & -5 & -3 & -3 \\ 0 & 2 & 3 & 0 & -1 & 3 & 3 & -7 & -2 & 0 \\ 3 & 4 & -2 & 1 & 3 & -3 & -3 & 2 & 0 & 0 \end{pmatrix}.$$

The Figure 6 displays the data and the original variables projected to the first two principal components.

2.2

The amount of variance explained as a function of the number of principal components used is displayed in Figure 7. It can be seen that the projection to the first two components in Figure 6 convey about 90.6% of the information of the data.

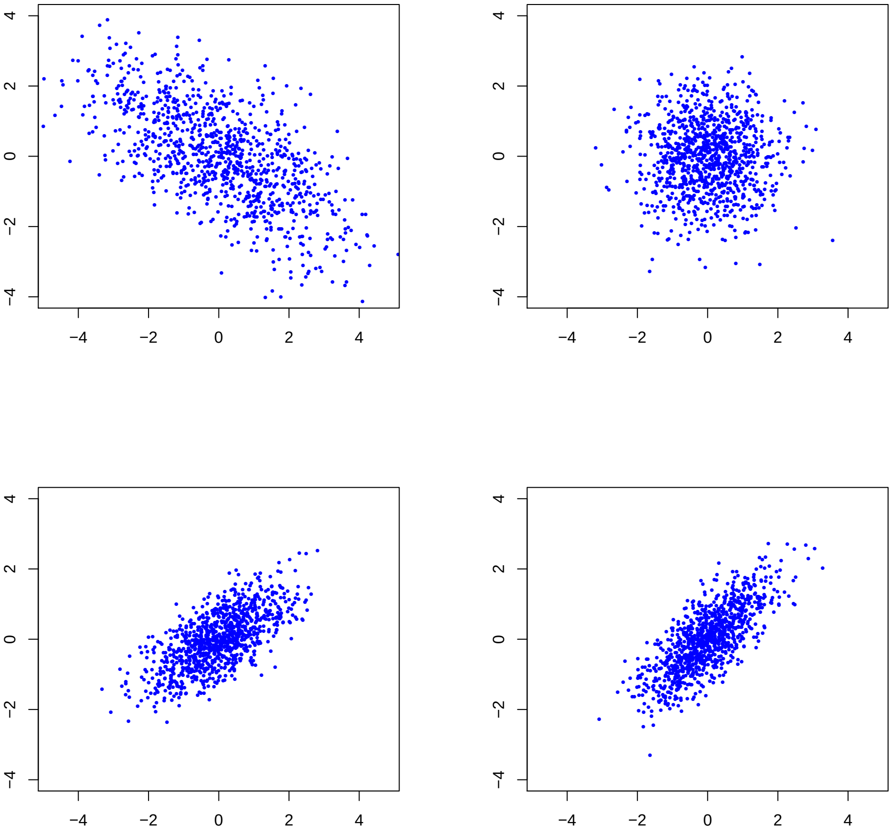

Figure 1: The scatter plots of the data generated in task 1.1.

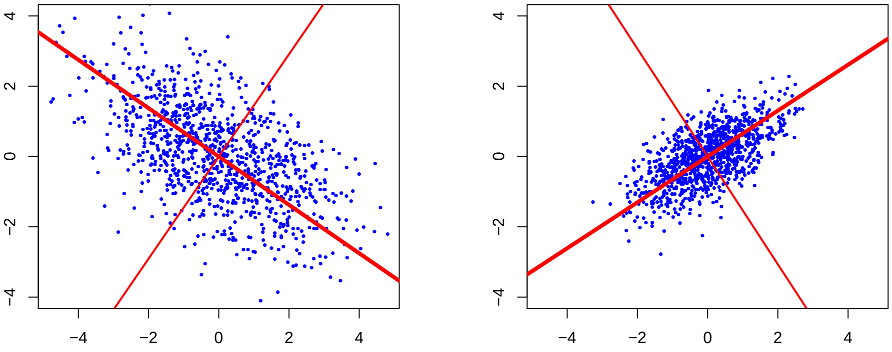

Figure 2: Principal components of the first and the third point set. The first PC is the bolder line.

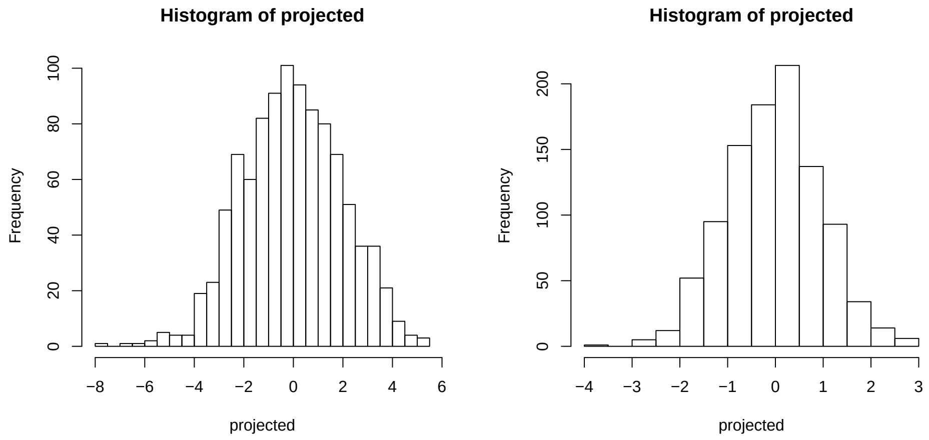

Figure 3: Histograms of the 1-dimensional data obtained by projecting the points of the first point set on its principal components.

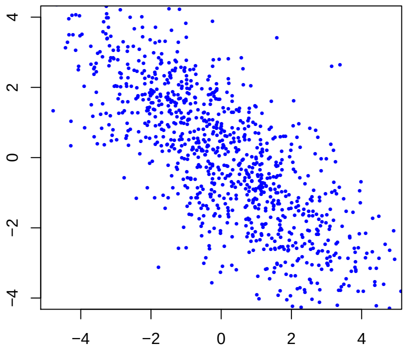

Figure 4:

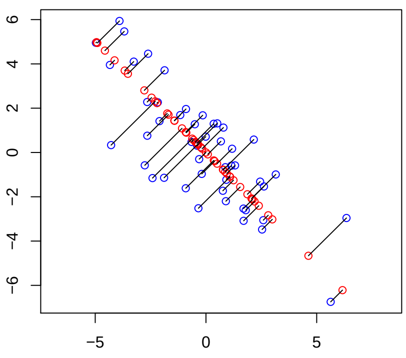

Figure 5:

2.3

2.4

The quartimax applied to the first two principal components of $\mathbf{X}$. Figure 8 shows the projections of the variables on the rotated components. Note that the figure is the same as Figure 6, only rotated. The rotation doesn't change the subspace spanned by the principal components, so they explain the same amount of variance before and after rotation.

3 Exercise 3

3.1

3.2

3.3

3.4

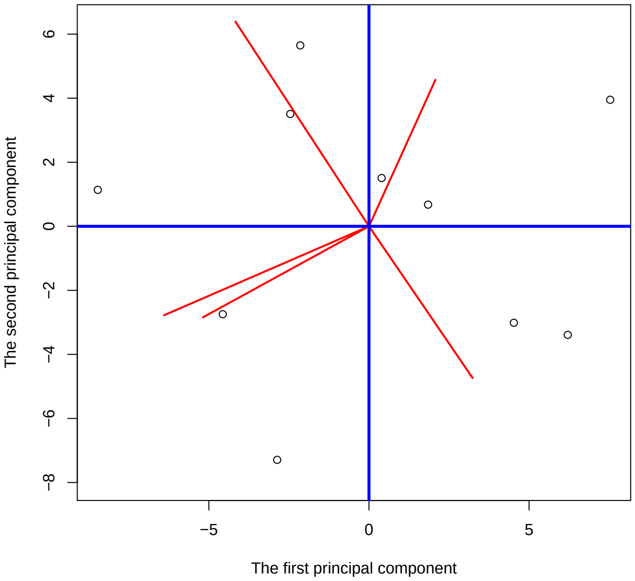

Figure 6: Reproduction of Figure 4.2 on the lecture notes. The blue points are the data points projected to the first two principal components. The red lines are the projections of the original coordinate axes.

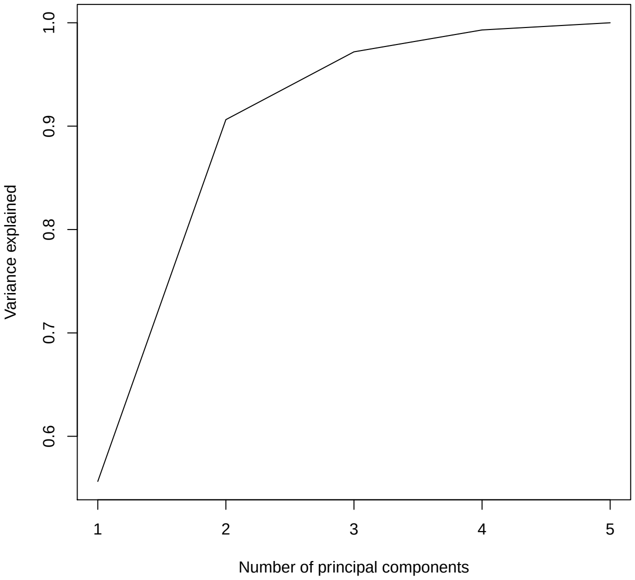

Figure 7: The proprotion of the variance explained by using only some of the principal components.

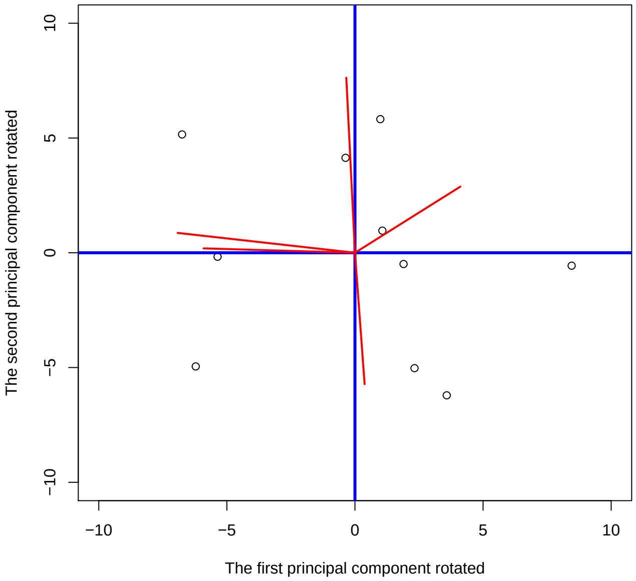

Figure 8: The projection to principal components after rotating them using the quartimax algorithm to have the original variables as close to the new coordinate axes as possible.

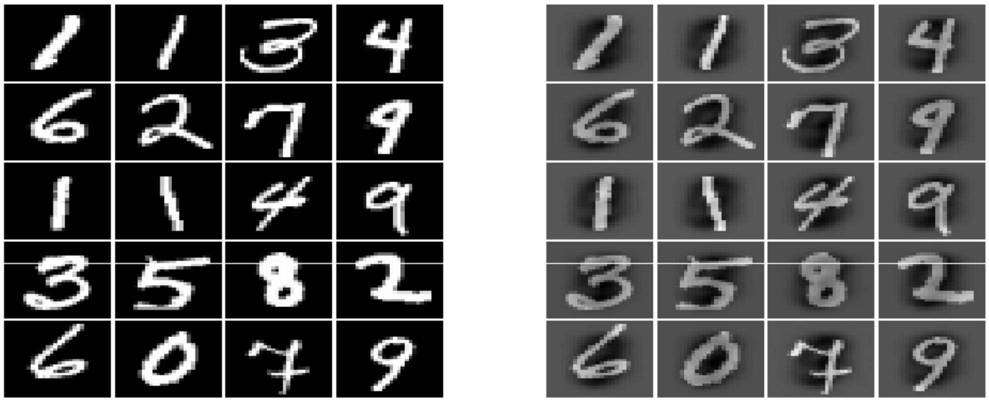

Figure 9:

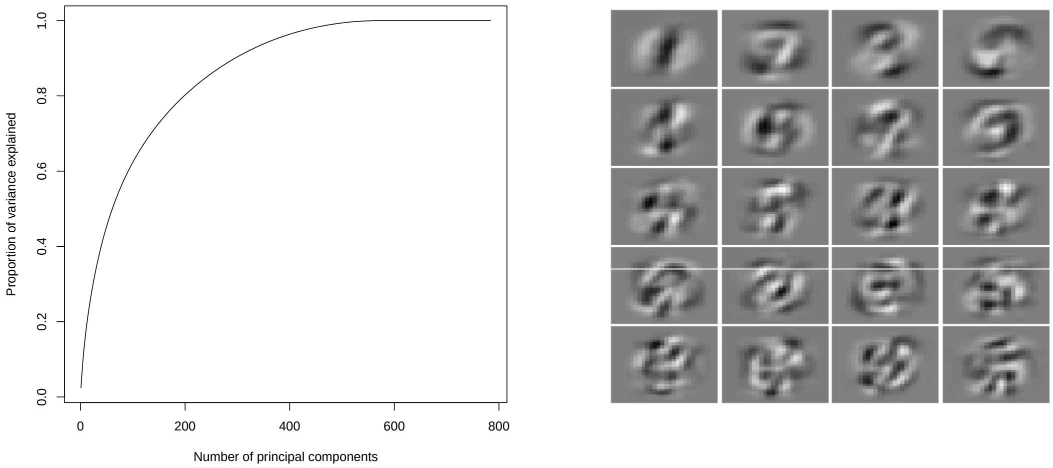

Figure 10:

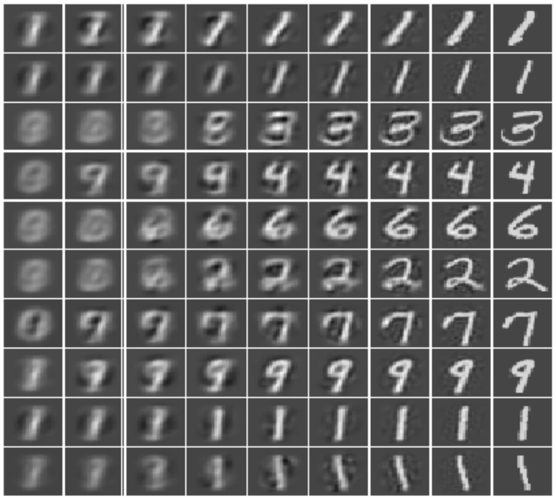

Figure 11:

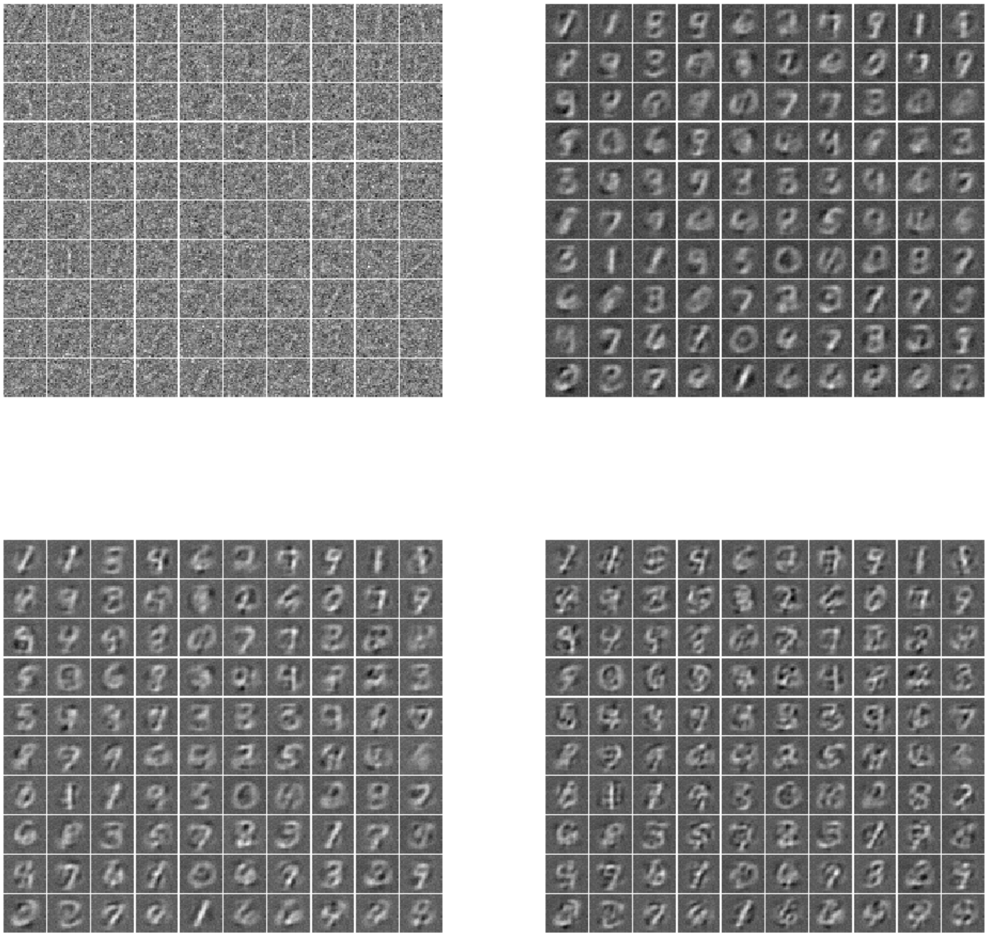

Figure 12: