COURSE 02457

Non-Linear Signal Processing: Exercise 2

This exercise is based on C.M. Bishop: *Pattern Recognition and Machine Learning*, sections 1.4, 2.3.0-2.3.4 and appendix C. The objective of the exercise is to become familiar with the 2D normal distribution, the notion of covariance, and using projections on eigenvectors as features.

Print and comment on the figures produced by the software main2a.m to main2e.m as outlined below at the four **Checkpoints**.

Multivariate normal Distribution

Let $\mathbf{x}$ be a d-dimensional variable, i.e. $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T$. The probability of the variable $\mathbf{x}$ lying in a region, $\mathcal{A}$, which is a subspace of $\mathbf{R}^d$ is given by

$$p(\mathbf{x} \in \mathcal{A}) = \int_{\mathcal{A}} p(\mathbf{x}) d\mathbf{x}, \tag{1}$$

where $p(\mathbf{x})$ is the probability density function of the variable $\mathbf{x}$.

In one dimension, the normal probability density function is given by

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{2}$$

where μ and σ^2 are the mean and variance respectively. In d dimensions, the general multivariate normal probability density function is given by

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})\right), \tag{3}$$

where $\boldsymbol{\mu}$ is a d-dimensional vector, and $\boldsymbol{\Sigma}$ is a $d \times d$ covariance matrix.

2D normal Distribution

Let $\mathbf{x}$ be a 2-dimensional variable, so that $d = 2$ in the above equations. Let $\mathcal{D}$ be a set of N samples from $\mathbf{x}$, so that $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_i = (x_{i,1}, x_{i,2})^T$, $i = 1, \ldots, N$.

It is then possible to construct a 2D histogram of the data-set, $\mathcal{D}$, by defining a Cartesian grid of small areas $\mathcal{A}_{j,k}$, where $j = 1, \ldots, M_1$ and $k = 1, \ldots, M_2$. The histogram is then given by

$$n_{j,k} = \sum_{\mathbf{x}_i \in \mathcal{A}_{j,k}} 1, \quad j = 1, \ldots, M_1, \, k = 1, \ldots, M_2, \tag{4}$$

with $n_{j,k}$ denoting the number of observations of $\mathbf{x}_i$ falling in the region $\mathcal{A}_{j,k}$. The normalized histogram is given by

$$\tilde{n}_{j,k} = \frac{n_{j,k}}{\sum_{j',k'} n_{j',k'}}. \tag{5}$$

If the union of all the areas $\mathcal{A}_{j,k}$ includes all the samples in $\mathcal{D}$, equation (5) simplifies to

$$\tilde{n}_{j,k} = \frac{n_{j,k}}{N}. \tag{6}$$

The normalized histogram can be compared with the histogram approximation to the probability density function

$$P_{j,k} = \int_{\mathcal{A}_{j,k}} p(\mathbf{x}) d\mathbf{x}\,, \quad j = 1, \ldots, M_1,\ k = 1, \ldots, M_2. \tag{7}$$

Alternatively the histogram can be converted to a normalized probability density, simply by dividing the normalized histogram bins with their corresponding areas $\mathcal{A}_{j,k}$

$$p_{j,k} = \frac{\tilde{n}_{j,k}}{\mathcal{A}_{j,k}}. \tag{8}$$

Hereby, we obtain a model for the density that is constant over the area $\mathcal{A}_{j,k}$ of each bin.

Checkpoint 2.1:

Use the program main2a.m to illustrate a 2-dimensional normal probability density function given by a mean, $\boldsymbol{\mu}$, and covariance matrix, $\boldsymbol{\Sigma}$.

Discuss the quality of the histogram as you vary the number of samples, N, from small to large values. Compare you findings with the results from exercise 1 for the 1D normal distribution and relate this to the curse of dimensionality.

Answer to Checkpoint 2.1:

One way of reducing the dimensions, or the amount of information, of a data set that follows a gaussian distribution, is to approximate the gaussian density, and then storing only the mean vector and the covariance matrix, and throw away the data points.

In this check point we will see that in order to approximate a probability density by sampling data points from it and subsequently making a histogram over the data points, one needs the right amount of data points and the right amount of bins. 2 dimensional data was created using the **randmvn()** function, and the histograms were created using **hist2d()**.

For high dimension problems (making high dimension histograms), one need a lot of data points to approximate the probability density. This can be illustrated by imagining a fixed number of data points, and then boxing them into bins in increasingly higher dimensions, see Fig. 1. As the number of dimensions grow, the density of points in each bin/box decreases. This is not good, since we want to calculate the average number of points in each bin, and as the density of points falls, you risk getting some really bad estimates of the density. The estimate of the pdf will also have a really high variance, since if you use the same "histogram" on a new data set of the same size, you would probably get a completely different height of each histogram bin.

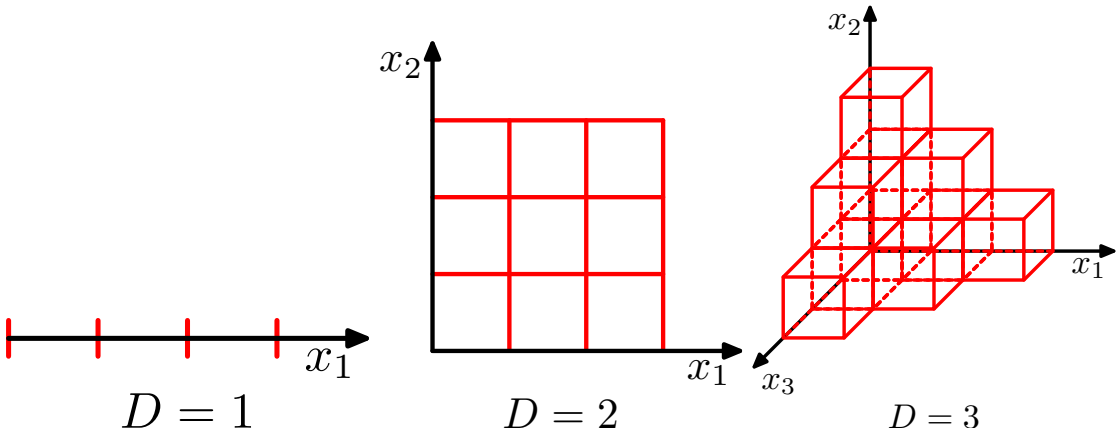


Figure 1: The size of the interior of a bin increases as the dimensionality increases, and so one needs more data points to get good averages in each bin.

It turns out that a good measure for how many sample points N you need if you want to use $nbins$ number of bins in D dimensions with the average number of points per bin being ρ:

$$N = nbins^D \cdot \rho \tag{9}$$

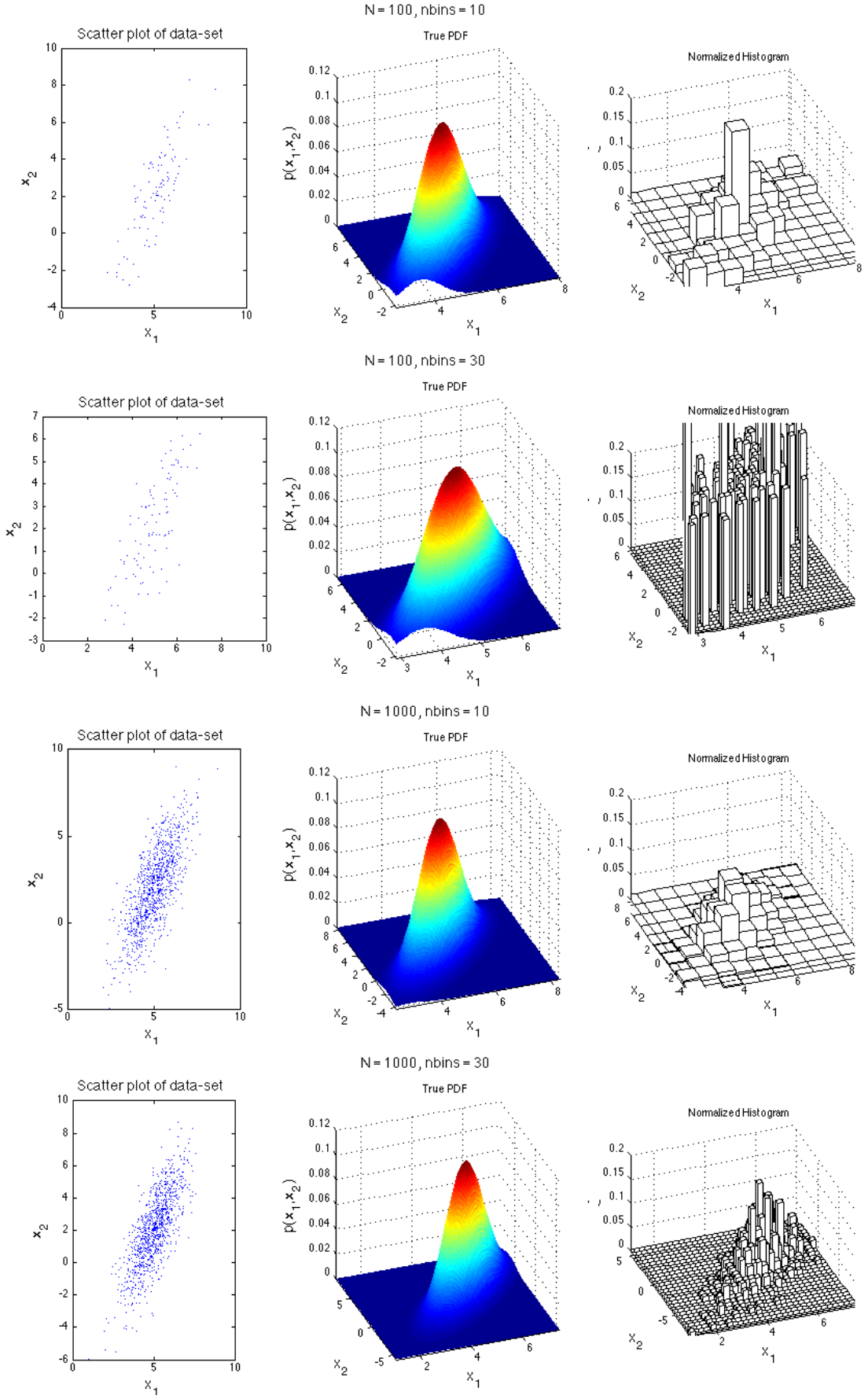
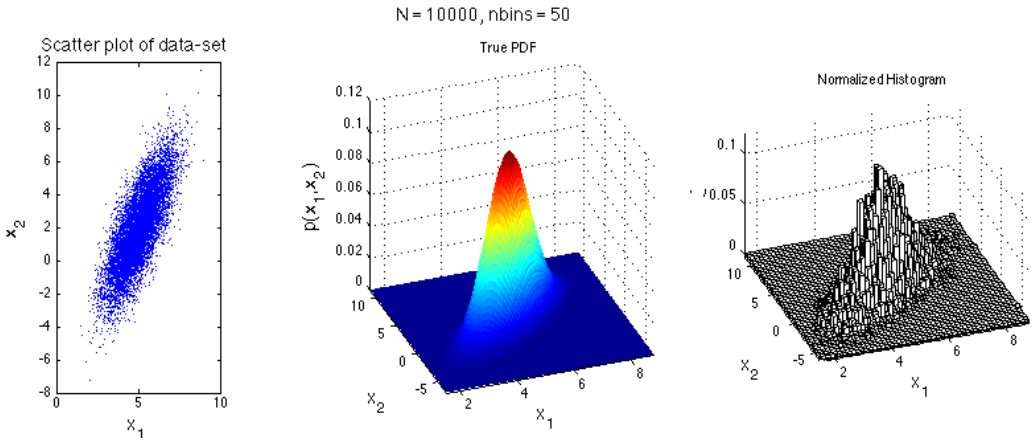


Figure 2: Approximations of a multivariate gaussian distribution with different number of data points and number of bins.



Interpretation of Covariance

A one-dimensional normal distribution is given by its mean, μ, and its variance, σ^2. The variance describes the variation of the variable around its mean.

In two dimensions, each sample consists of two components. Each dimension has a mean and a variance just as in the one-dimensional case. Consider a sample, $\mathbf{x} = (x_1, x_2)^T$ from a 2D normal distribution with mean $\boldsymbol{\mu} = (\mu_1, \mu_2)$. If the variance of x_1, namely σ_1^2 is large, an individual sample of x_1 may well be quite different from μ_1, and similarly for x_2. However, there may be a trend that whenever x_1 is larger than μ_1, x_2 is also larger than μ_2, and that whenever x_1 is smaller than μ_1, x_2 is also smaller than μ_2. In such a case, x_1 and x_2 are not independent, and they are said to be correlated.

Another term is therefore needed to fully describe the variance of the variable, $\mathbf{x}$, namely the covariance between its components, $\mathrm{cov}\,[x_1, x_2] \equiv \mathbb{E}[(x_1 - \mu_1)(x_2 - \mu_2)]$. The covariance matrix of $\mathbf{x}$ is then given by

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \mathrm{cov}\,[x_1, x_2] \\ \mathrm{cov}\,[x_2, x_1] & \sigma_2^2 \end{pmatrix} \equiv \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix}. \tag{10}$$

The terms σ_{12} and σ_{21} are equal since they describe the covariance between the same components. The covariance matrix is therefore always symmetric. The magnitude of the covariance term for a given correlation between the two components also depends on the diagonal variance terms. A useful quantity describing the correlation between the components is the correlation coefficient, ρ. It is the normalized covariance and is given by

$$\rho = \frac{\mathrm{cov}\,[x_1, x_2]}{\sqrt{\sigma_1^2 \sigma_2^2}} = \frac{\sigma_{12}}{\sqrt{\sigma_{11}\sigma_{22}}}. \tag{11}$$

where $\rho \in [-1, 1]$. However, the limiting case $\rho = \pm 1$ corresponds to a perfect linear relationship between x_1 and x_2. In this case the variable, $\mathbf{x}$, is not really 2-dimensional since one component completely defines the other.

Checkpoint 2.2:

Use the program main2b.m to visualize the probability density functions of 2D normal distributions with different covariance matrices. For example, try to fix the variances, σ_1^2 and σ_2^2, while only changing the covariance. Think of an example where there is no correlation between the components and implement this distribution. Comment on the dependence of the orientation and shape of the ellipsoids in the contour plots of quadratic form induced by the covariance matrix.

Answer to Checkpoint 2.2:

In this checkpoint we illustrate the shape of the 2D pdf for different covariance matrices. 2 dimensional data is created using the **randmvn()** function. It is shown that for positive covariances, the ellipsoid shape of the data and the (true and estimated) pdf is tilted to the right. For negative covariance, they are tilted to the left. When there is no correlation, the data and pdf are circular.

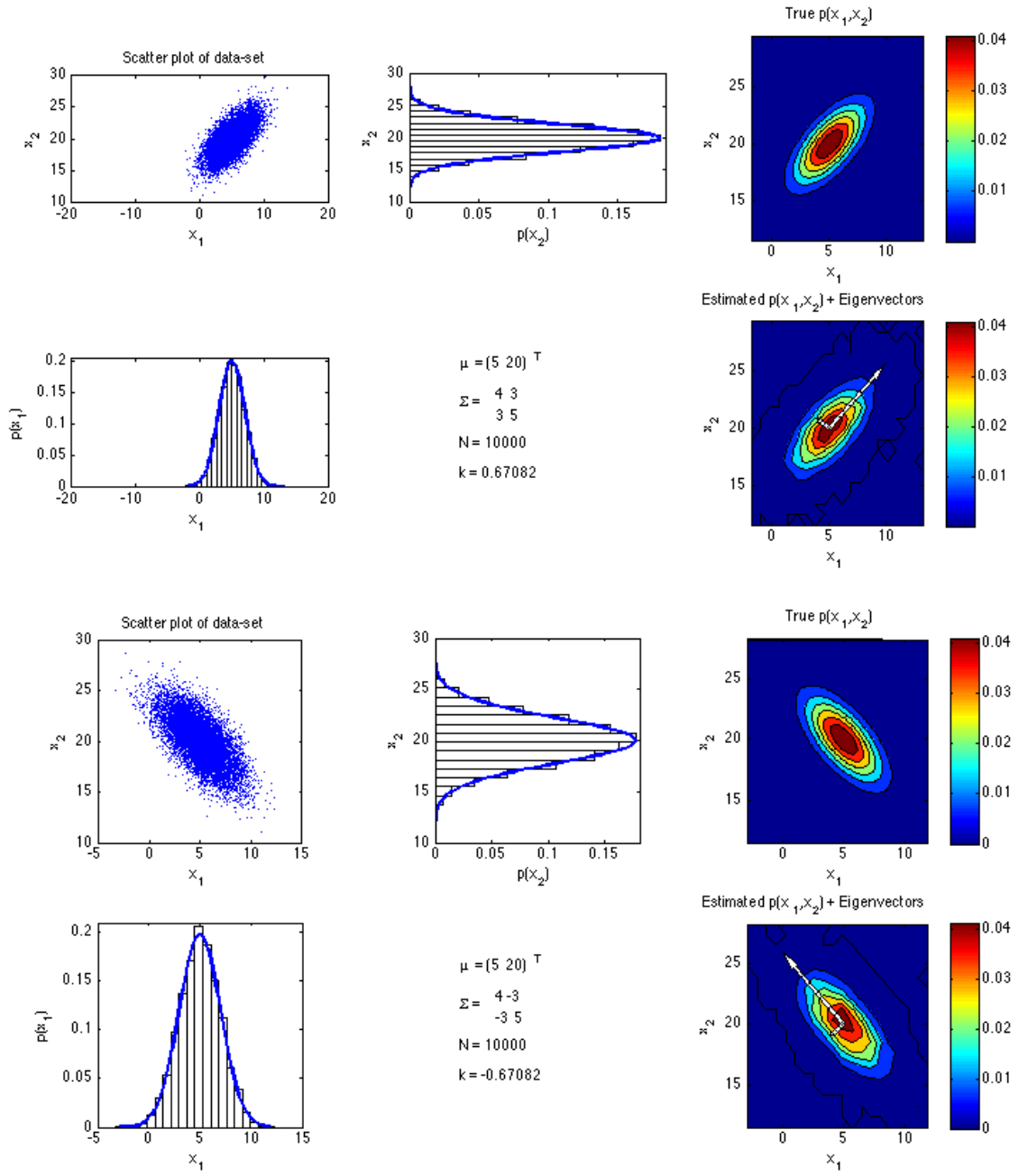


Figure 3: Checkpoint 2.2. True and estimated pdf's together with data generated with covariance matrices with (top) positive and (bottom) negative covariance.

Coordinate Transformation

For some non-linear signal detection algorithms it is desired that the input should have zero mean, unit variance and zero covariance. The advantage of this is that it is possible to use the same algorithm (and not changing the control parameters of it) for variables of very different origins and covariation.

Geometrically, such a normalization corresponds to a coordinate transformation to the

system defined by the eigenvectors of the covariance matrix. Typically, the mean and covariance matrix are not known, and must therefore be estimated from the data-set, $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$:

$$\widehat{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{12}$$

$$\widehat{\mathbf{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \widehat{\mathbf{x}})(\mathbf{x}_i - \widehat{\mathbf{x}})^T . \tag{13}$$

The eigenvalue equation for the covariance matrix is

$$\widehat{\mathbf{\Sigma}}\mathbf{u}_j = \lambda_j \mathbf{u}_j , \quad j = 1, \ldots, d , \tag{14}$$

where λ_j is the j'th eigenvalue and $\mathbf{u}_j$ is the corresponding eigenvector of $\widehat{\mathbf{\Sigma}}$. The transformed input variables are then given by

$$\tilde{\mathbf{x}}_i = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T(\mathbf{x}_i - \widehat{\mathbf{x}}), \tag{15}$$

where

$$\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_d) \tag{16}$$

$$\mathbf{\Lambda} = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_d) . \tag{17}$$

It can be shown that the transformed data-set, $\tilde{\mathcal{D}} = \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_N\}$ has zero mean and a covariance matrix given by the unit matrix.

Checkpoint 2.3:

Use the program main2c.m to calculate the eigenvalues and eigenvectors of the covariance matrix for different distributions. Comment on the geometrical significance of the eigenvalues and eigenvectors. Compare the transformed data-sets from different distributions. What happens if the term $\mathbf{\Lambda}^{-1/2}$ is removed from equation (15)?

Answer to Checkpoint 2.3

2 dimensional data is created using the **randmvn()** function, and an estimate of the mean and covariance is calculated, as well as the eigenvectors and eigenvalues of the estimated covariance matrix. The eigenvectors points in the direction where the data set has the most and the least variance, and thus they can be interpreted as the most significant direction and the least.

When the data is transformed using the eigenvectors and the eigenvalues as in Eq. (15) (after the mean has been subtracted), the coordinate system used to represent the data is rotated to be aligned with the eigenvectors of the covariance matrix. In the new coordinate system, the data has no covariance and the variance in each direction is the same as the eigenvalues. By dividing the rotated data set by the same eigenvalues, the data set is scaled so it has the same variance in all directions.

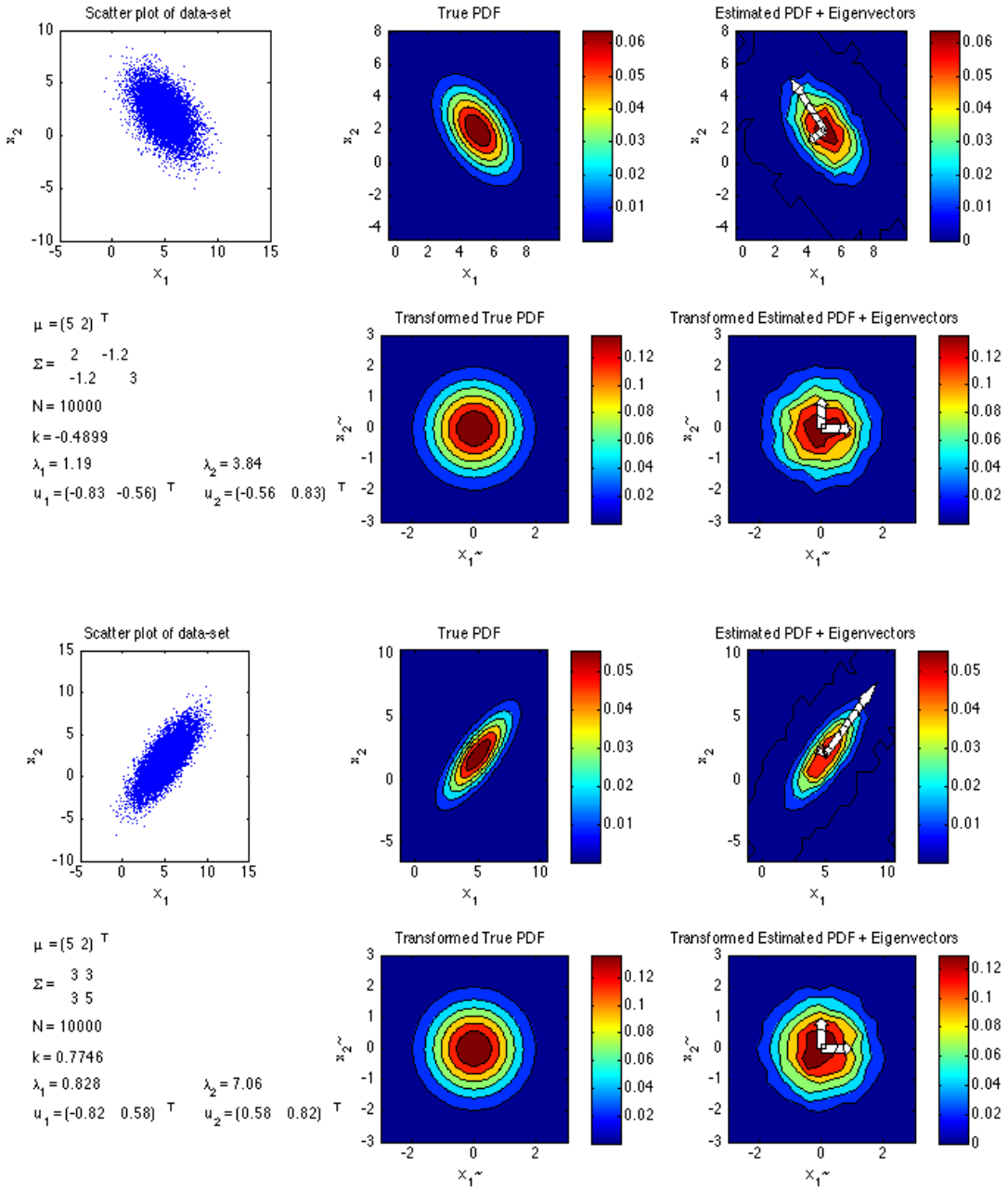


Figure 4: Two different data sets that are transformed (rotated and scaled).

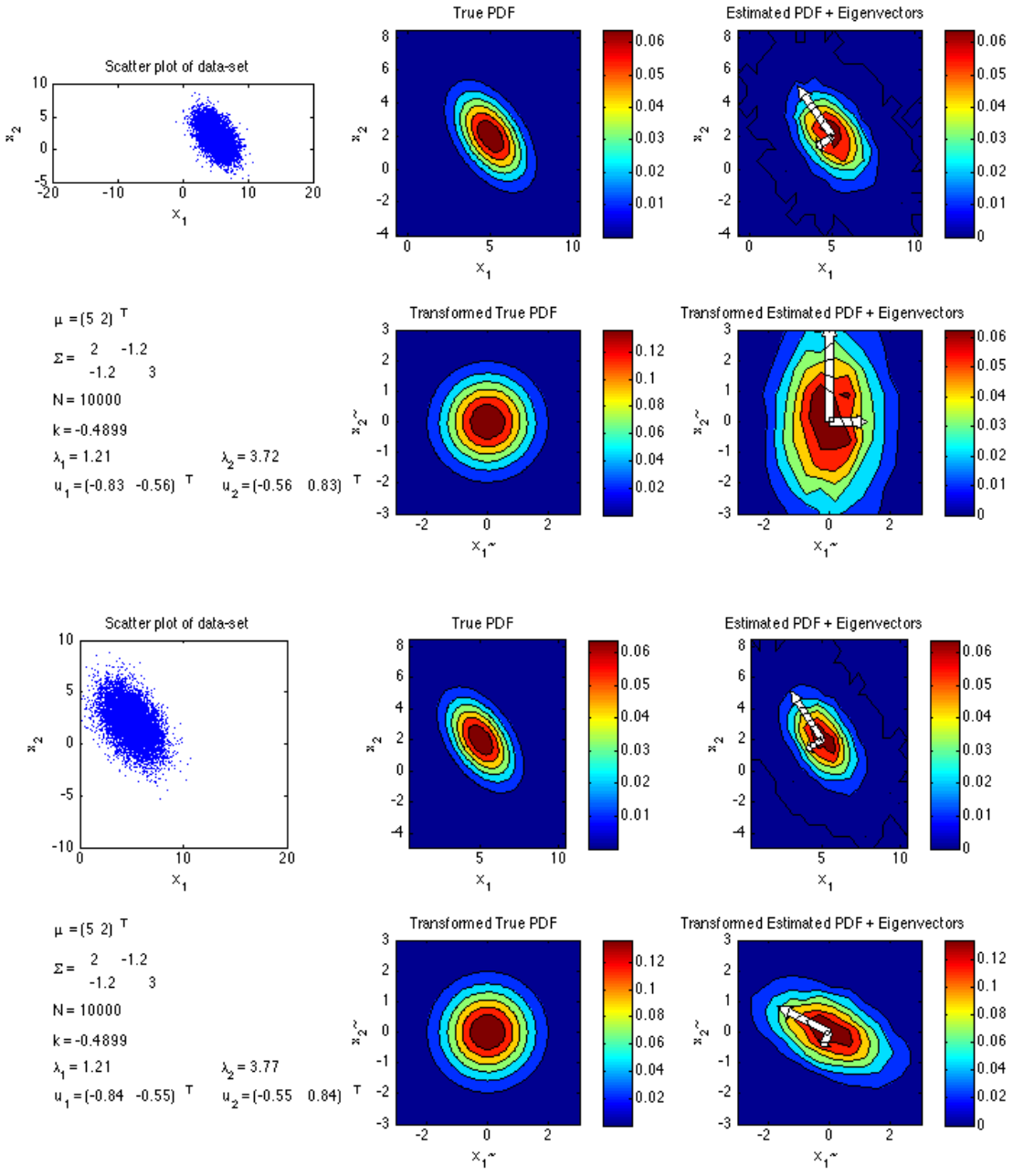


Figure 5: **Top:** Data that has been rotated (only) using the eigenvectors of the estimated covariance matrix. **Bottom:** Data that has been scaled (only) using the eigenvectors of the estimated covariance matrix. Neither of these two procedures results in zero mean, unit variance and zero covariance.

Projection on Eigenvectors

In some cases, the measured data is of a lower "true" dimension than the apparent dimension of the data vector. For example, imagine a data-set of a 3-dimensional variable. If all the data are on a straight line, the true dimension of the data is only 1D. If the

data-set is transformed to a coordinate system, where the variation of the data is along one of the axes, the two other components can be ignored.

Let $\lambda_1, \ldots, \lambda_d$ be the ordered set of eigenvalues of the covariance matrix, such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$. If there exists a number m, such that $\lambda_i \gg \lambda_j$, $i = 1, \ldots, m$, and $j = m+1, \ldots, d$, then the data-set can be transformed to a coordinate system, where most of the signal variance is in an m-dimensional linear subspace spanned by the m'th first eigenvectors in the ordered list. This transformation is again given by the eigenvectors of the covariance matrix ($\mathbf{U}$),

$$\tilde{\mathbf{x}}_i = \mathbf{U}^T(\mathbf{x}_i - \widehat{\mathbf{x}}). \tag{18}$$

If we extract only the first m components of the transformed datavector $\tilde{\mathbf{x}}$ we obtain a signal that carries most of the variation of the original signal. Such reduction of the effective dimensionality of the problem is also known as extraction of features.

Checkpoint 2.4:

Use the programs main2d.m and main2e.m to transform 2D datasets into the eigenvector-space and comment on the "true" dimensionality of the classification problems.

Answer to checkpoint 2.4:

In this last check point we are ready to apply what we have learned about coordinate transformation to project as much information onto one dimension as possible, so we can trow away the other dimension so we don't have to store it.

In script **main2d**, data from two 2-dimensional normal distributions are simulated, and randomly added together according to the prior probability of the gaussians to form a single data set. The data is then transformed (rotated) using the eigenvectors of the covariance matrix to better distinguish between the two gaussians, see Fig. 6. If the data set is only rotated, the resulting two peaks in the marginal distribution are wider but set further apart. If the data set is normalized using the eigenvalues of the covariance of the total data set, one dimension is squeezed together and the other is stretched out so that the total variation in each direction becomes similar. This means that the two peaks in the resulting marginal distribution gets closer, but are sharper.

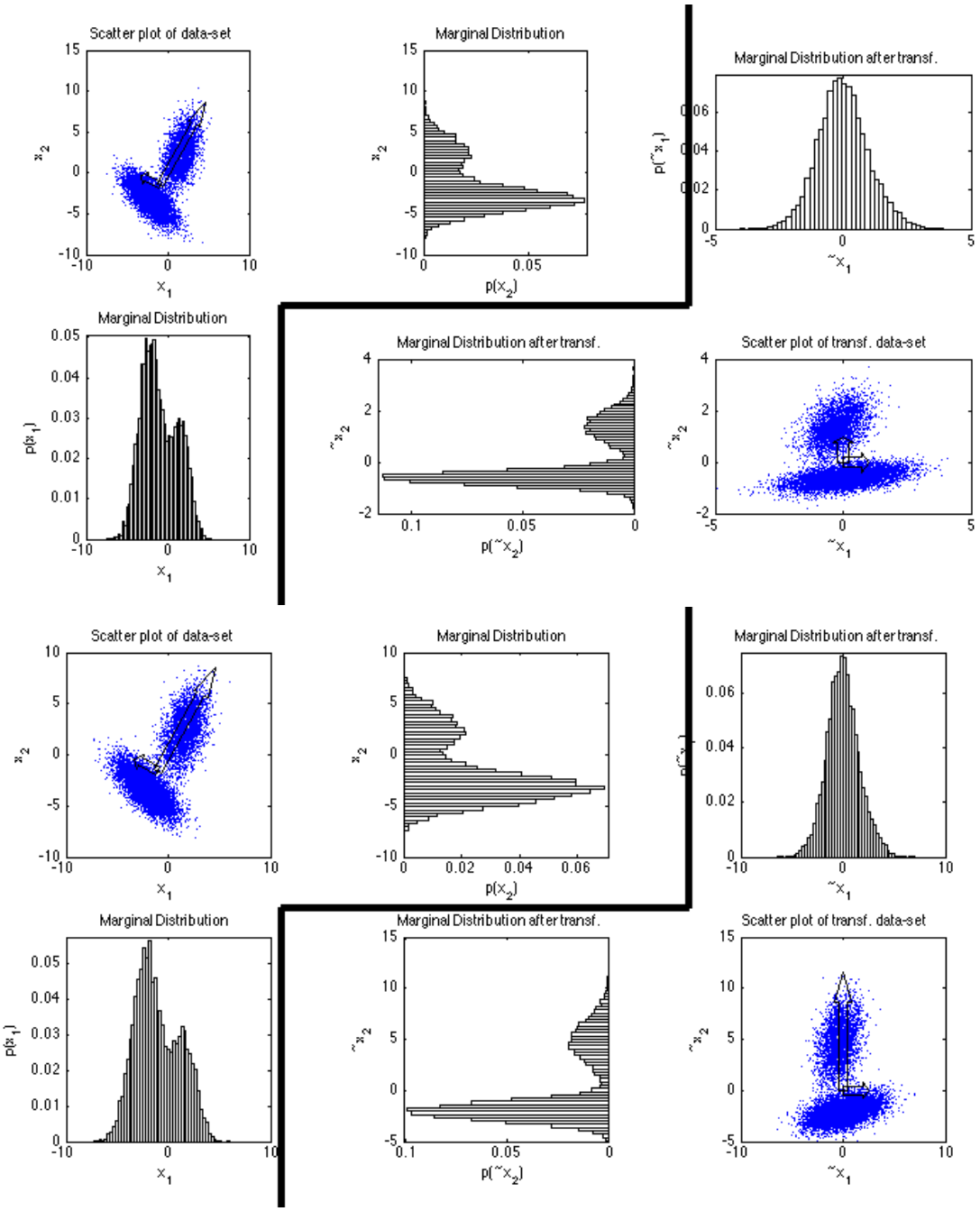


Figure 6: **Top:** Raw data set and fully transformed (rotated and scaled) data. **Bottom:** Raw data and rotated data set.

Script **main2e** is almost the same as **main2d**, only this time the gaussians in the mixture are more separate, see Fig. 7. Here, projecting the 2D information down to only one dimension is an obvious choose.

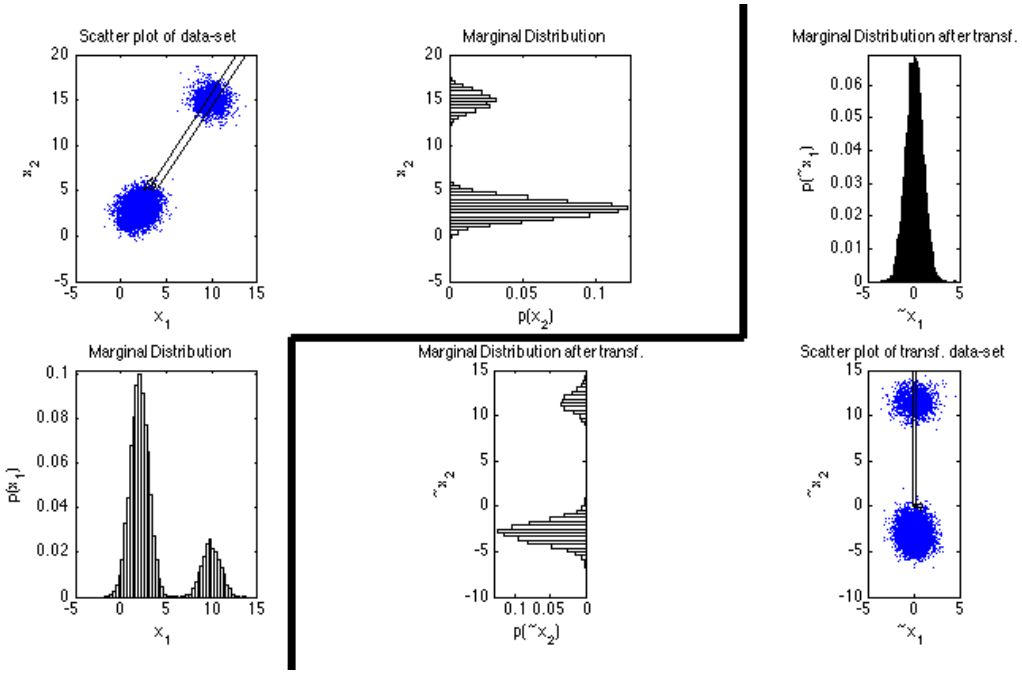


Figure 7: In this case, there is no point in representing the data in two dimensions, so it is projected down to the largest eigenvector.

Challenges (not part of the curriculum)

1) Prove that the transformed data in Eq. (15) is zero mean and has a unit covariance matrix.

2) How would you make the decision on the number of principal components to retain? Hint: Check K.W. Jorgensen, L.K. Hansen: *Model selection for Gaussian kernel PCA denoising.* IEEE Transactions on Neural Networks and Learning Systems **23**(1):163-168 (2012), the references in this paper.

DTU, September 2009,

Karam Sidaros, Lars Kai Hansen