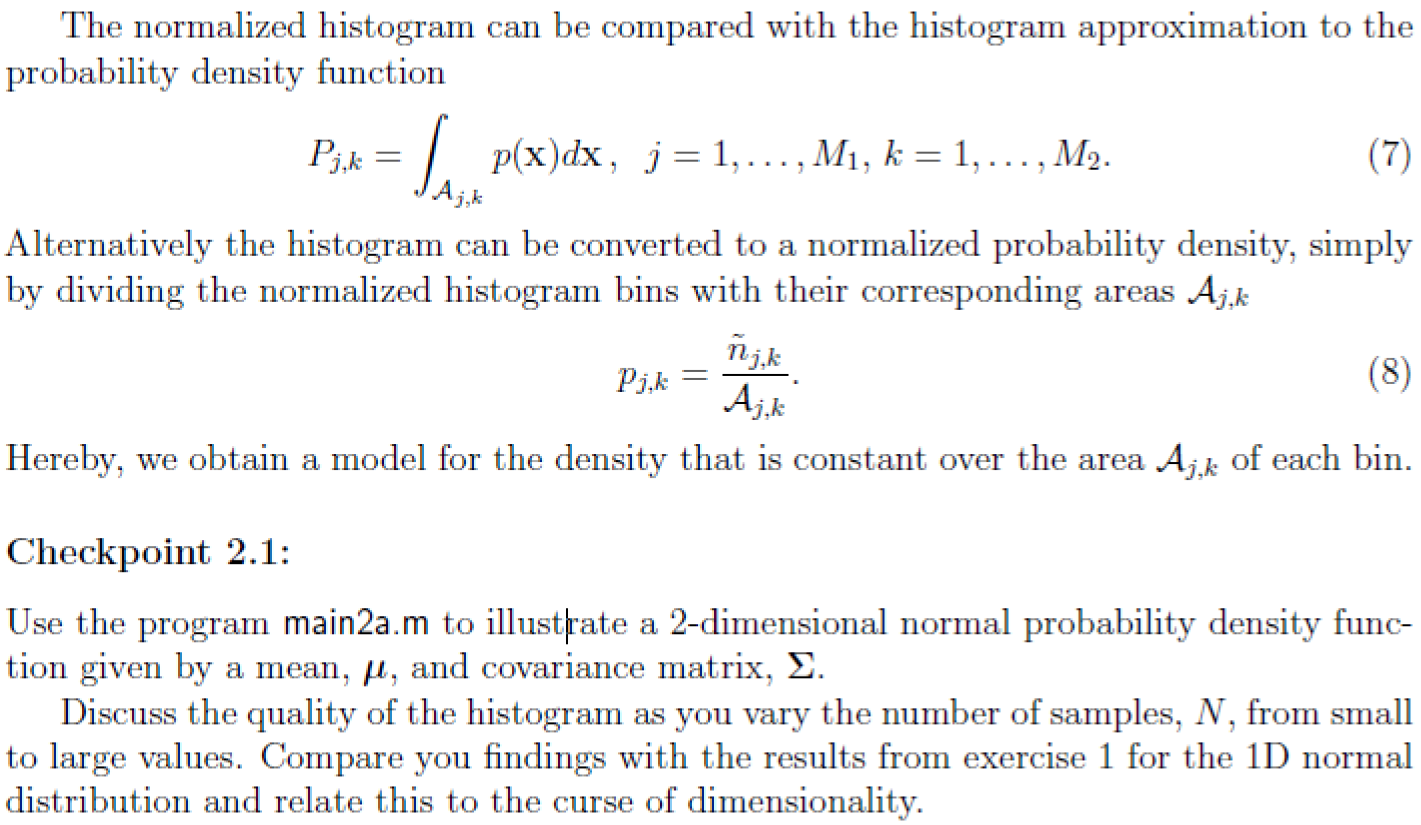
This exercise introduces multivariate normal distribution and focuses on covaraince interpretation.



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.

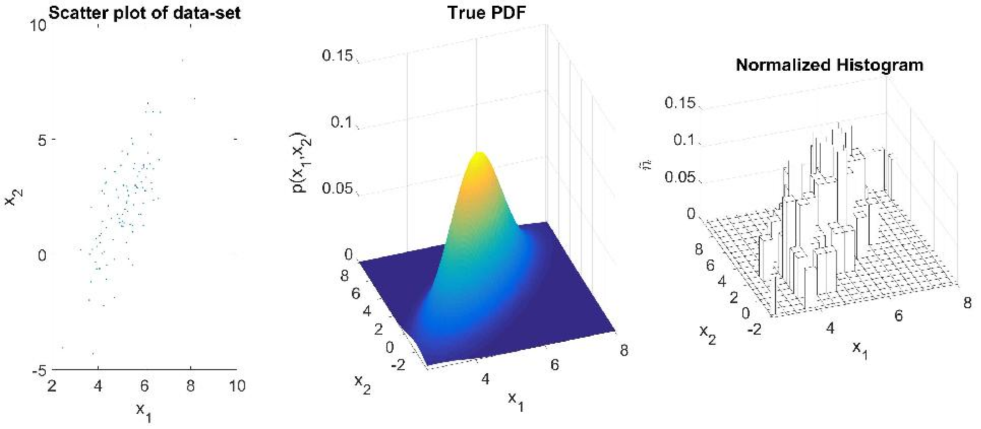
Checkpoint 2.1:

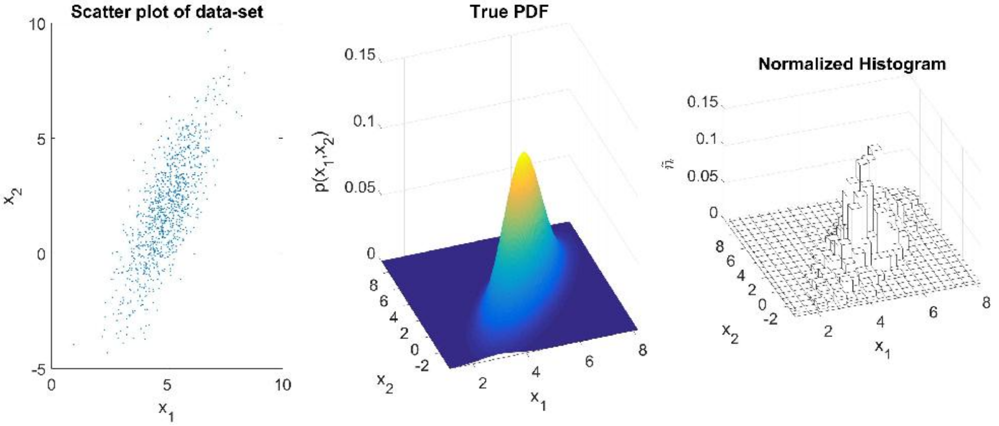
Use the program `main2a.m` to illustrate a 2-dimensional normal probability density function given by a mean, $\boldsymbol{\mu}$, and covariance matrix, $\boldsymbol{\Sigma}$.

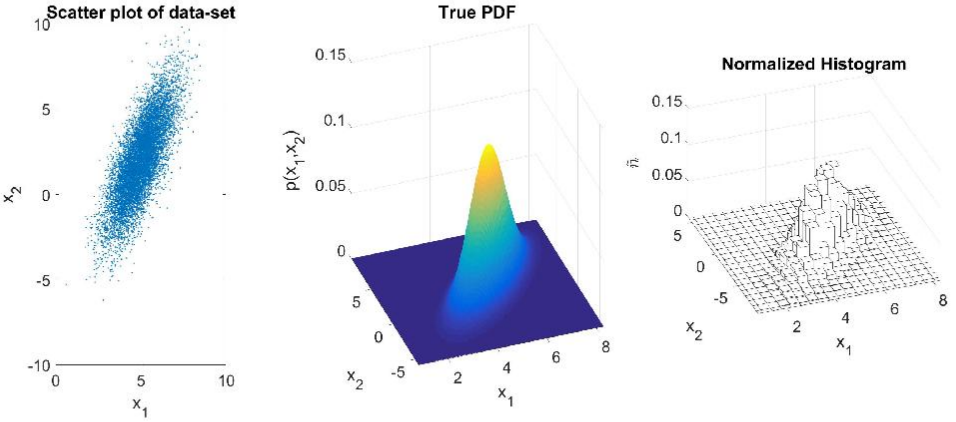
Discuss the quality of the histogram as you vary the number of samples, N, from small to large values. Compare you findings with the results from exercise 1 for the 1D normal distribution and relate this to the curse of dimensionality.

In this exercise a set of data is created which is probability density is 2-d normal distribution. The mean value and covariance matrix for the sample generator are preset. Later on the probed mean value and covariance matrix are compared to the true ones. One can see that as the number of samples is increased, these values will get closer to the true values.

In order to probe the quality of the histogram we have shown the results with three different number of samples: $N = 100$ $N = 1000$ and $N = 1000$0







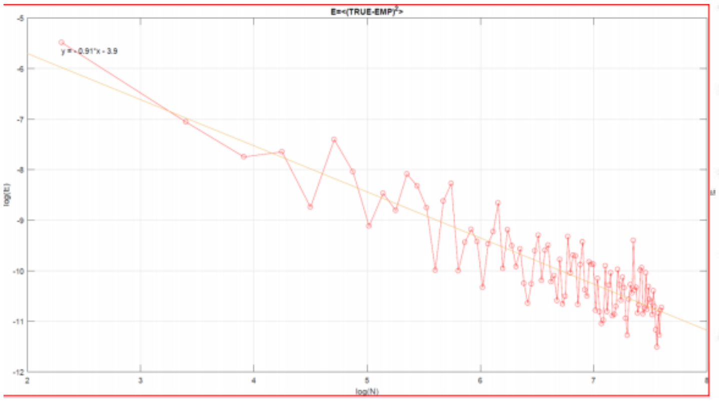
In the case of N=100 the results does not look like a normal distribution at all, but we see how does it look more and more like one with the increments of samples. However, the number of samples needed to get an accuracy similar to the error that we get in the one dimensional case is way bigger, mainly due to the course of dimensionality.

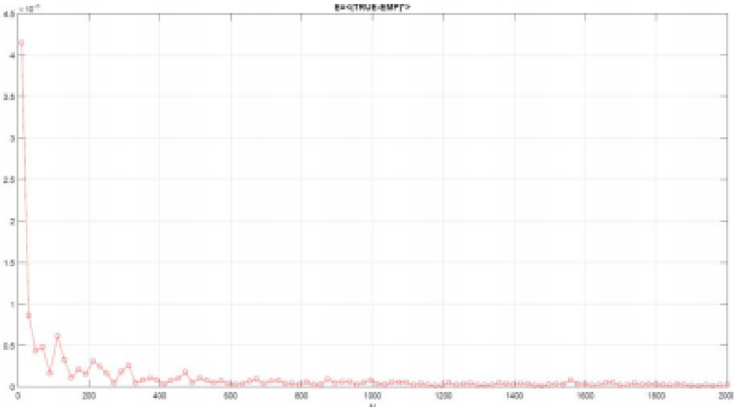
This term is a phenomena that arises when working with high dimensions of data. It mainly states that the density of the distribution decreases exponentionally when introducing new dimensions of data, and with that, in our case the quality of the histogram. In order to mantain the same sampling density we should keep $N^{\frac{1}{D}}$ proportional, being N the number of samples and D the number of dimensions. In our case, if we want to maintain the same density as in the one dimensional case with N=20 samples we should obtain:

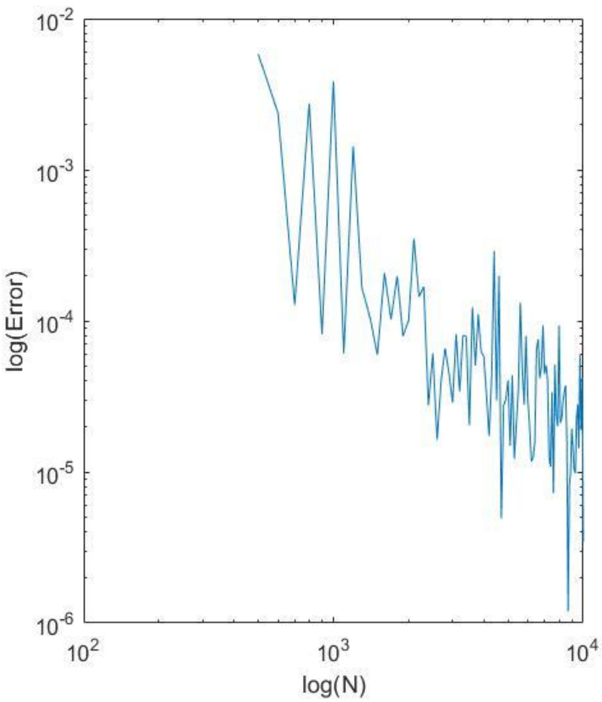
$$20^{\frac{1}{1}} = N_{2D}^{\frac{1}{2}}$$

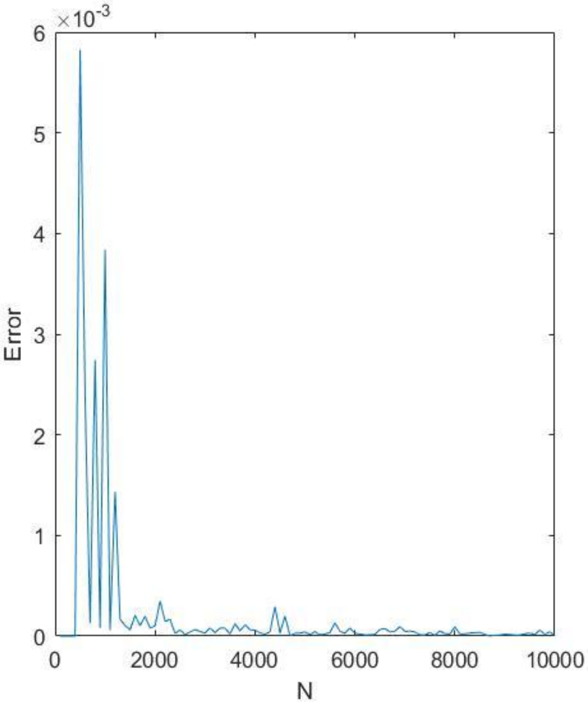
$$N_{2D} = 400$$

We can test this course of dimensionality by comparing the error between the estimated histogram and the true histogram for the one dimensional and the two dimensional case, respectively.









As we can see, the shape is similar, but when we take a look at the scale, the differences are obvious due to the curse of dimensionality.

Checkpoint 2.2:

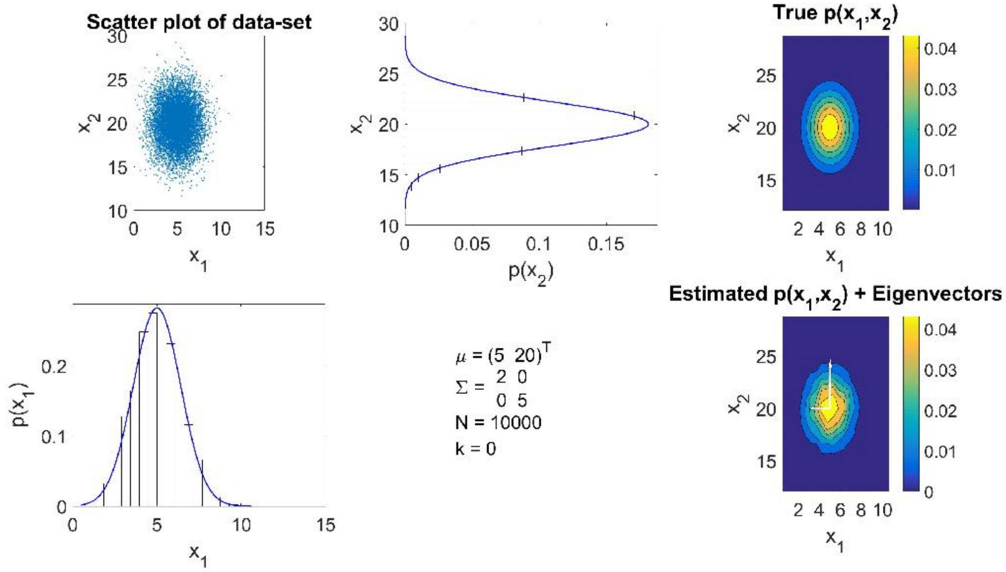
Use the program main2b.m to visualize the probability density functions of 2D normal distributions with different covariance matrices. For example, try to fix the variances, σ_1^2 and σ_2^2, while only changing the covariance. Think of an example where there is no correlation between the components and 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.

We used the script to visualize the interpretation of variance and covariance. Having 2-d normal distribution, each of the variables has it's variance which describies how much x1 and x2 is spread around it's mean values. Additionally there are covariance elements which explain how dependent the variables are on eachother. There may be a trend that whenever x1 is larger than µ1, x2 is also larger than µ2, and that whenever x1 is smaller than µ1, x2 is also smaller than µ2. In such a case, x1 and x2 are not independent, and they are said to be correlated.

After performing this experiment we have came up with 3 different singular cases:

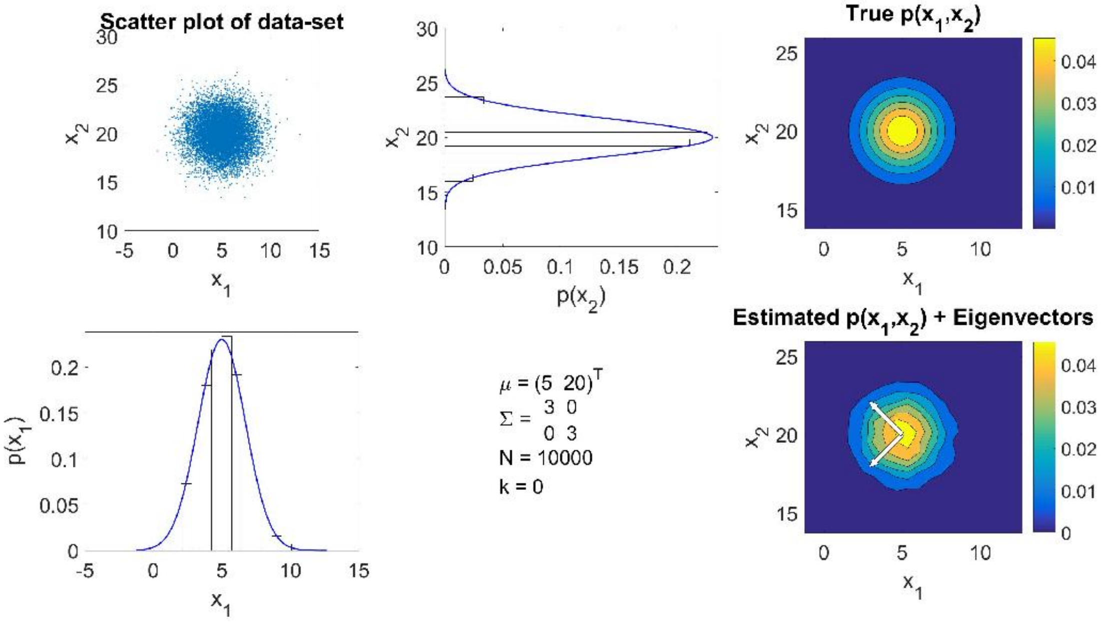
- Uncorrelated distributions: $\sigma_{12} = 0$

In this case we have two completely independent components, so that the values of one component does not affect the other one. One example of this could be a dataset with information about shoe size and yearly income, two completely unrelated components. In this case, as we can see the shape is a vertical elipsoid because σ_{11} is bigger than σ_{12}. In the other case, we would see a distribution that is more spread in the horizontal axis.



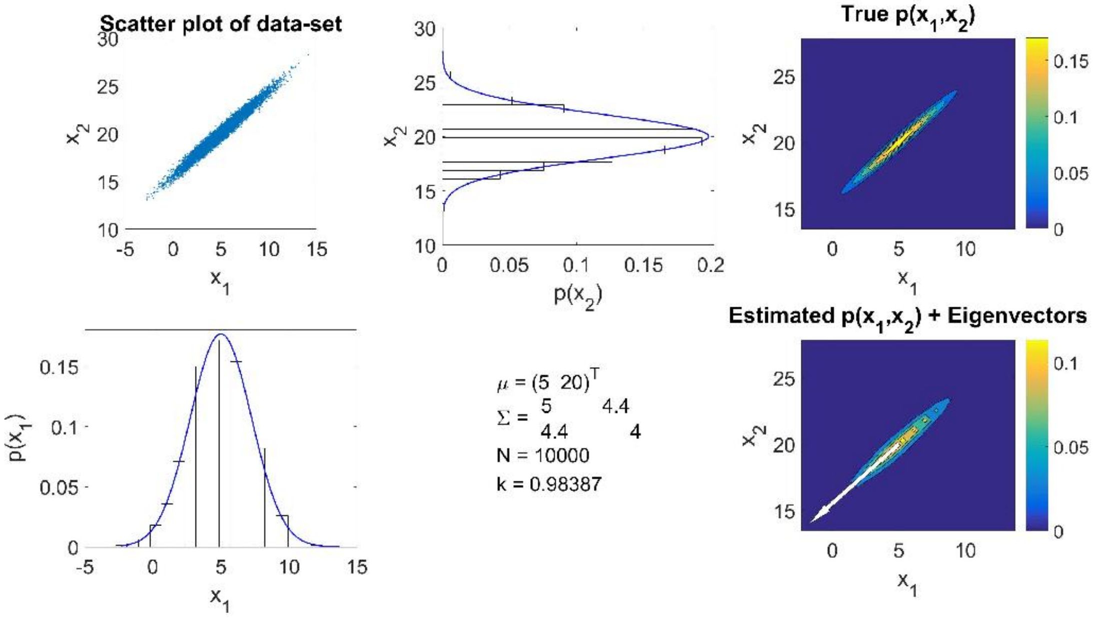
- Isotropic covariance matrix $\sigma_{12} = 0$ | $\sigma_{11} = \sigma_{22}$

Two distributions with the same variance and completely uncorrelated. The contours will adopt a perfect concentric circles shape, as they will be representing the same gaussian distribution (with different means).

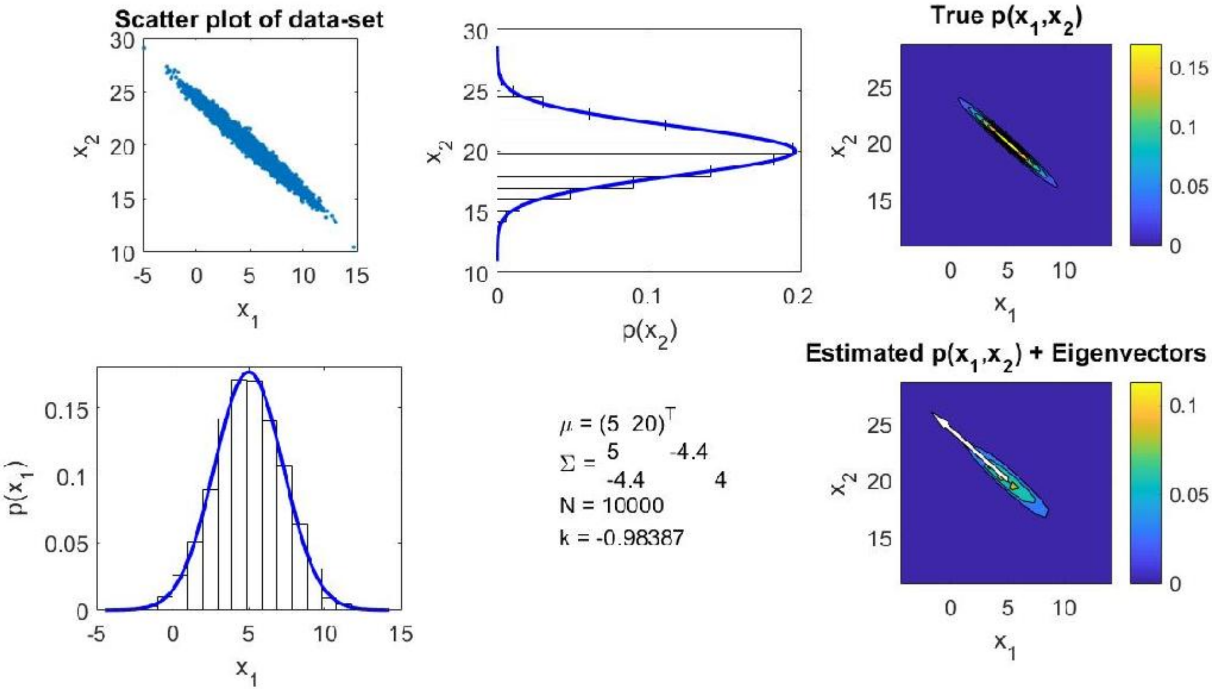


- Linear distribution: $\rho = \frac{\sigma_{12}}{\sqrt{\sigma_{11}\sigma_{22}}} = |1|$

When the correlation coefficient approaches 1 or -1 it is easy to spot how the distribution adopts the shape of a typical linear distribution. In the limit case, when the absolute value of ρ is equal to one the distribution is not two dimensional anymore, as one distribution becomes dependent on the other.



This case can be also used to explain that we can also alter the sign of the covariances to change the inclination of the distributions.


Scatter plot of data-set
x_2
x_1
30
25
20
15
10
-5
0
5
10
15
p(x_2)
0
0.1
0.2
True p(x_1,x_2)
0.15
0.1
0.05
0
p(x_1)
Estimated p(x_1,x_2) + Eigenvectors
μ = (5 20)^T
Σ = 5 -4.4
-4.4 4
N = 10000
k = -0.98387

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, $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$:

$$\widehat{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{11}$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \widehat{\mathbf{x}})(\mathbf{x}_i - \widehat{\mathbf{x}})^T . \tag{12}$$

The eigenvalue equation for the covariance matrix is

$$\widehat{\boldsymbol{\Sigma}}\mathbf{u}_j = \lambda_j \mathbf{u}_j\,, \quad j = 1, \ldots, d\,, \tag{13}$$

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

$$\tilde{\mathbf{x}}_i = \boldsymbol{\Lambda}^{-1/2}\mathbf{U}^T(\mathbf{x}_i - \widehat{\mathbf{x}}), \tag{14}$$

where

$$\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_d) \tag{15}$$

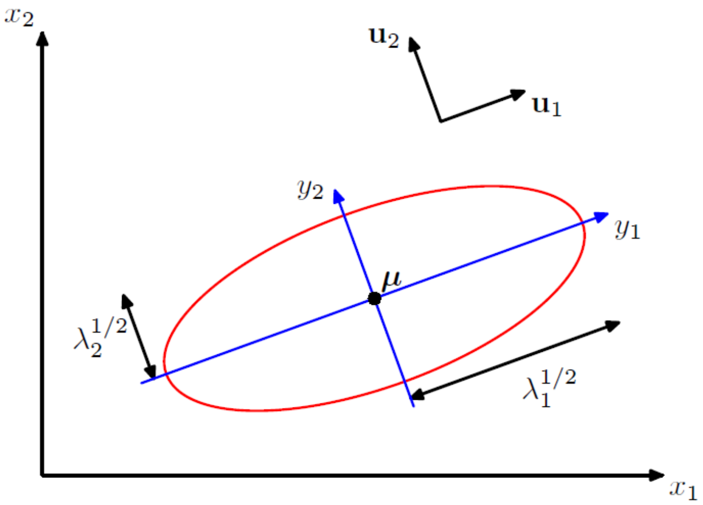
$$\boldsymbol{\Lambda} = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_d)\,. \tag{16}$$

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 $\boldsymbol{\Lambda}^{-1/2}$ is removed from equation (14)?

Let's first take a look at the geometrical significance of the eigenvectors and eigenvalues of a covariance matrix in a gaussian distribution. This is represented in the following image

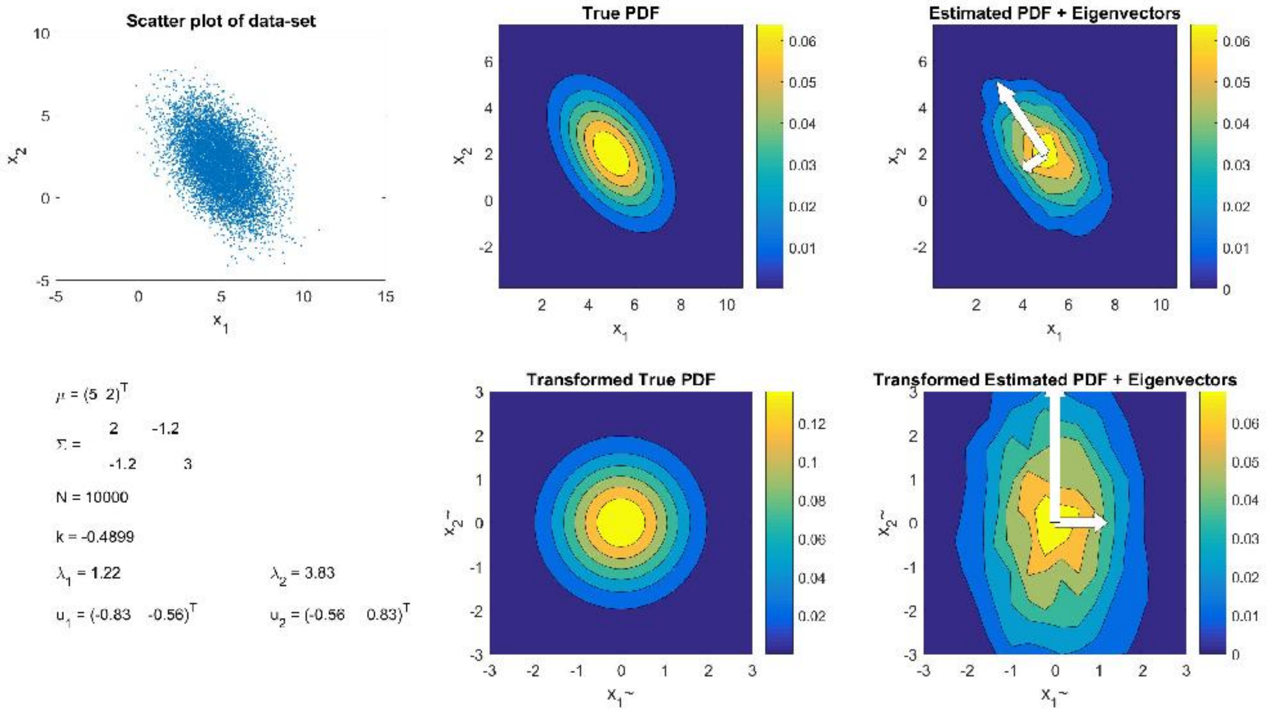


In a two dimensional casse the gaussian distributions adopt an elipsoid form, and the eigenparameters define the distinctive parameters of it. Given that the mean sets the center of the elipse, the eigenvectors define the direction of the axis of the elipse, or the direction on which the data spreads. On the other hand, the eigenvvalues are the factor that determine the size of the axis of the elipse, or how spread is the distribution in each direction given by an eigenvalue.

The eigenvector of the largest eigenvalue tells us the direction in which the singal variance is the largest. It can be used when we want to reduce the dimensionality of the system. In that case we want to cast samples on the axis of largest variance to keep as much information as possible. For example if we cast on axis perpendicular, a lot of samples will be casted to the same point as the variance is very small.

If the covariance is zero then eigenvalues give variances of the variables.If the covariance matrix of our data is a diagonal matrix, such that the covariances are zero, then this means that the variances must be equal to the eigenvalues. The largest eigenvector of the covariance matrix always points into the direction of the largest variance of the data, and the magnitude of this vector equals the corresponding eigenvalue. The second largest eigenvector is always orthogonal to the largest eigenvector, and points into the direction of the second largest spread of the data.

We can also see this relation when executing the script:



From this results we can also see how the coordinate transformation works. We can see that the distribution changed in order to have the eigenvectors as the origin of the coordinate system, rotating around the distribution.

$$\tilde{\mathbf{x}}_i = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T(\mathbf{x}_i - \widehat{\mathbf{x}}), \tag{14}$$

where

$$\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_d) \tag{15}$$

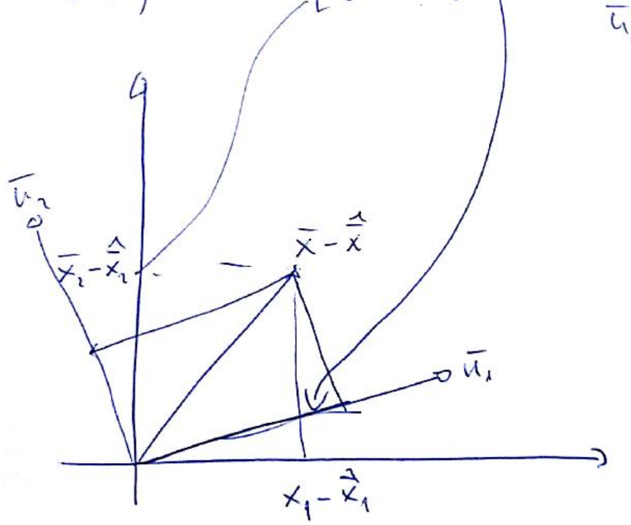
$$\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)\,. \tag{16}$$

At first the system is translated into zero mean system by subtracting the mean value. Then, the first multiplication (U') rotates the coordinate system into the eigenvector coordinate system (what mean that it also removed the covariance elements). In that case the left variances will be equal to the eigenvalues. The other multiplication is just a simple scaling to go to unit variances. Removing that term will leave us with signal of 0 mean and 0 covariance but with not unit variance.

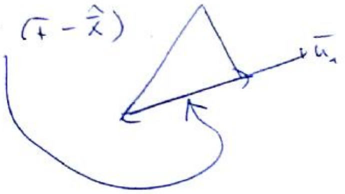
The transformation can be seen graphically in the sketch:

$\bar{\bar{U}}^T \cdot (\bar{x} - \hat{\bar{x}})$ $\qquad$ $\bar{\bar{U}} = [\bar{u}_1 \ \bar{u}_2]$ $\quad$ $\bar{\bar{U}}^T = \begin{bmatrix} \bar{u}_1^T \\ \bar{u}_2^T \end{bmatrix}$

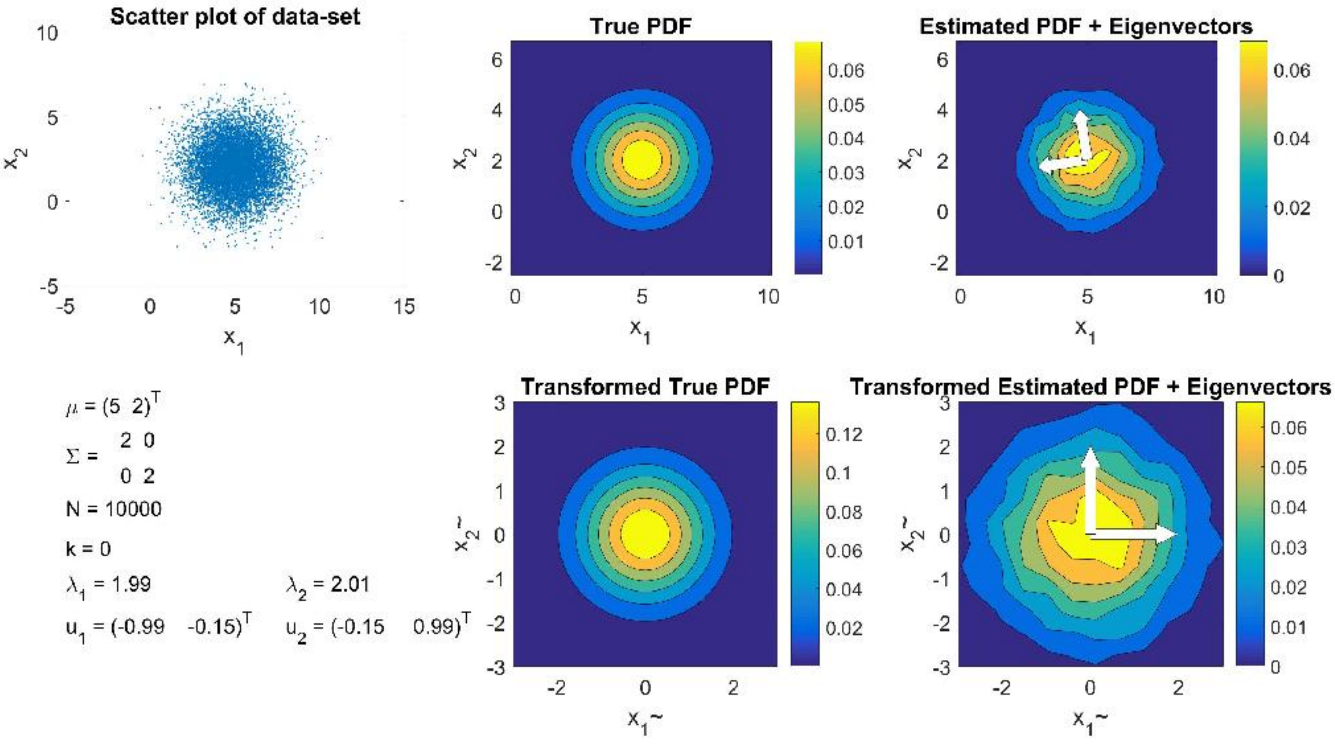
$$\begin{bmatrix} \bar{u}_1^T \\ \bar{u}_2^T \end{bmatrix} (\bar{x} - \hat{\bar{x}}) = \begin{bmatrix} \bar{u}_1^T \cdot (\bar{x} - \hat{\bar{x}}) \\ \bar{u}_2^T \cdot (\bar{x} - \hat{\bar{x}}) \end{bmatrix}$$



$\bar{u}_1^T \cdot (\bar{x} - \hat{\bar{x}})$



Some special cases worth mentioning are when we are working with independent distributions and the variance of the distributions is the same. In this case we can see that the eigenvector's directions move depending on the introduced noise when producing the data.



$\mu = (5 \ \ 2)^T$

$\Sigma = \begin{matrix} 2 & 0 \\ 0 & 2 \end{matrix}$

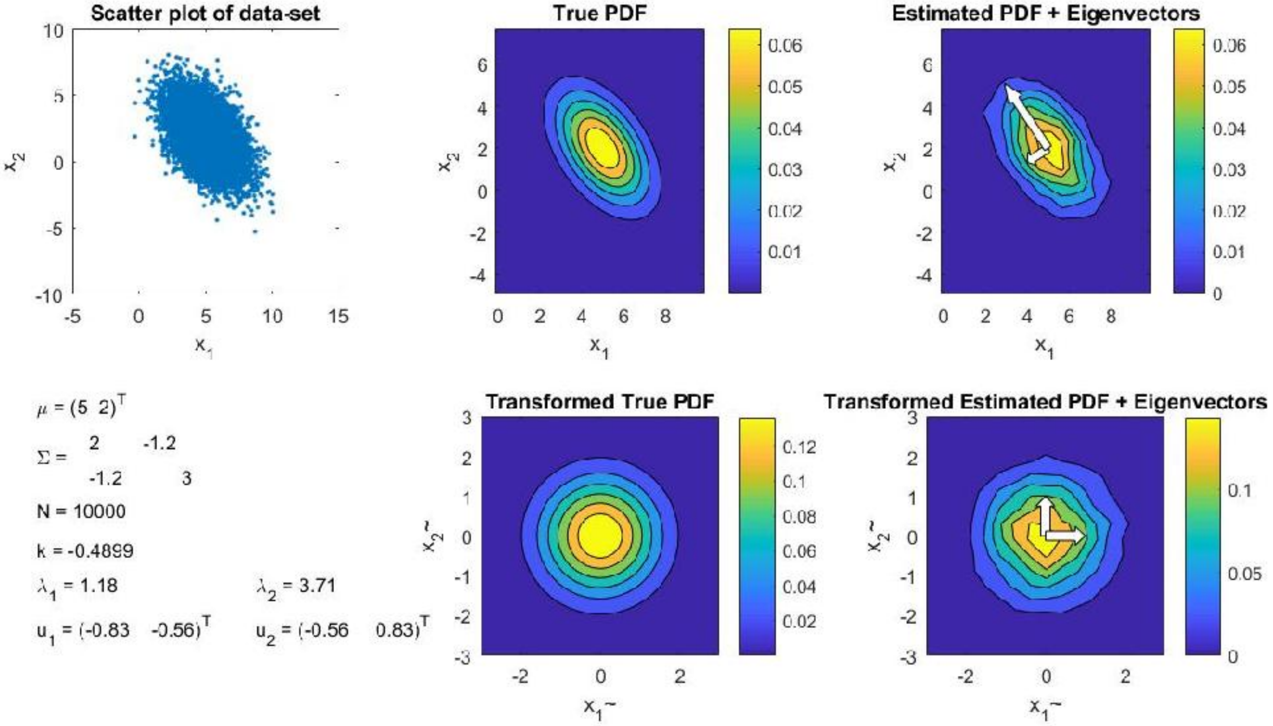
N = 10000

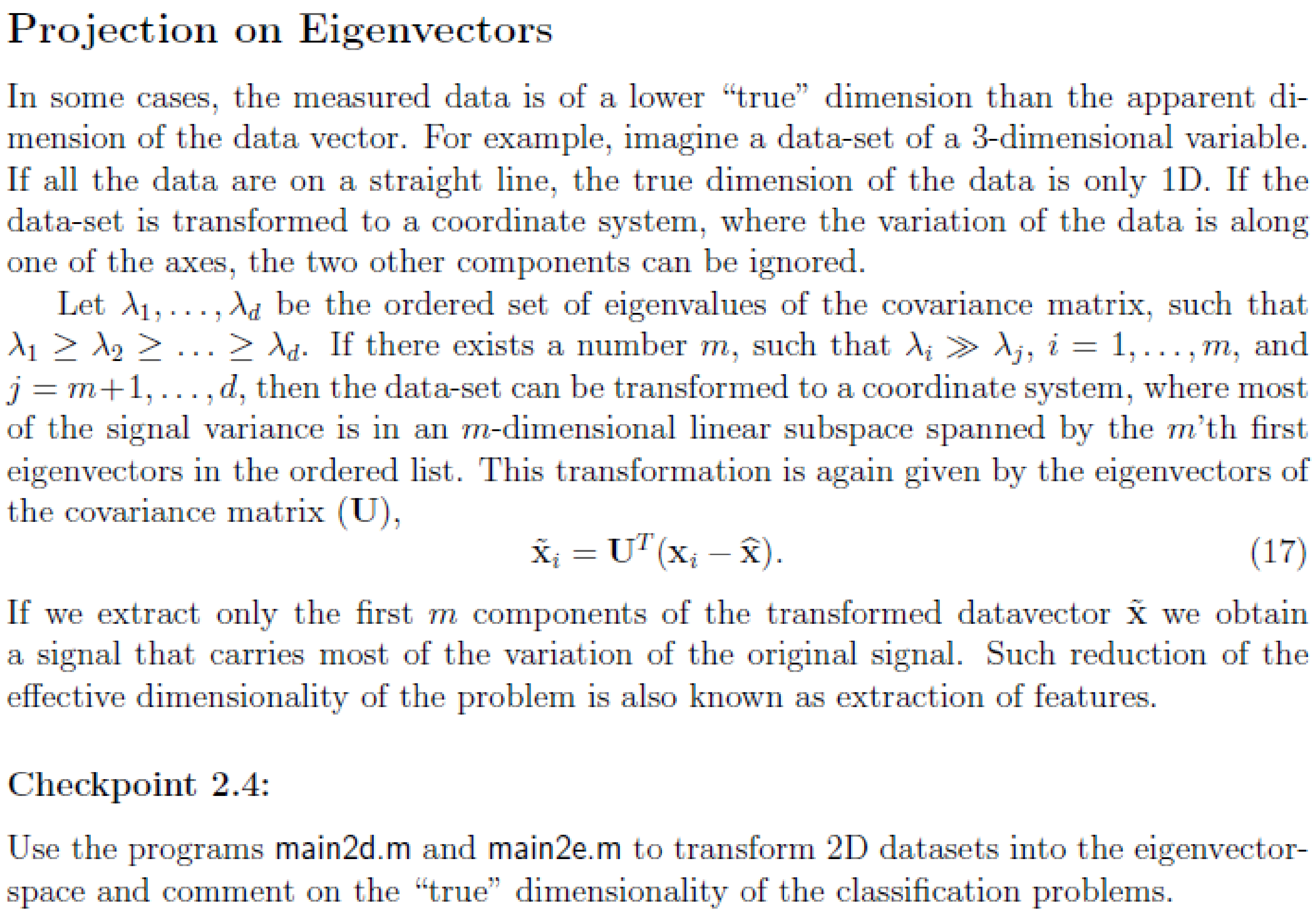
k = 0

$\lambda_1 = 1.99$ $\qquad$ $\lambda_2 = 2.01$

$u_1 = (-0.99 \quad -0.15)^T$ $\qquad$ $u_2 = (-0.15 \quad 0.99)^T$

Regarding the term, it is important to note that it has not been introduced in the previous cases. We can realise because if we take a look at the transformed true PDF of the first example, the distribution are elipses and not a circle as the variance is not scaled. They can be compared with the results when the eigenvalues term is added in the same initial distribution





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 ($\mathbf{U}$),

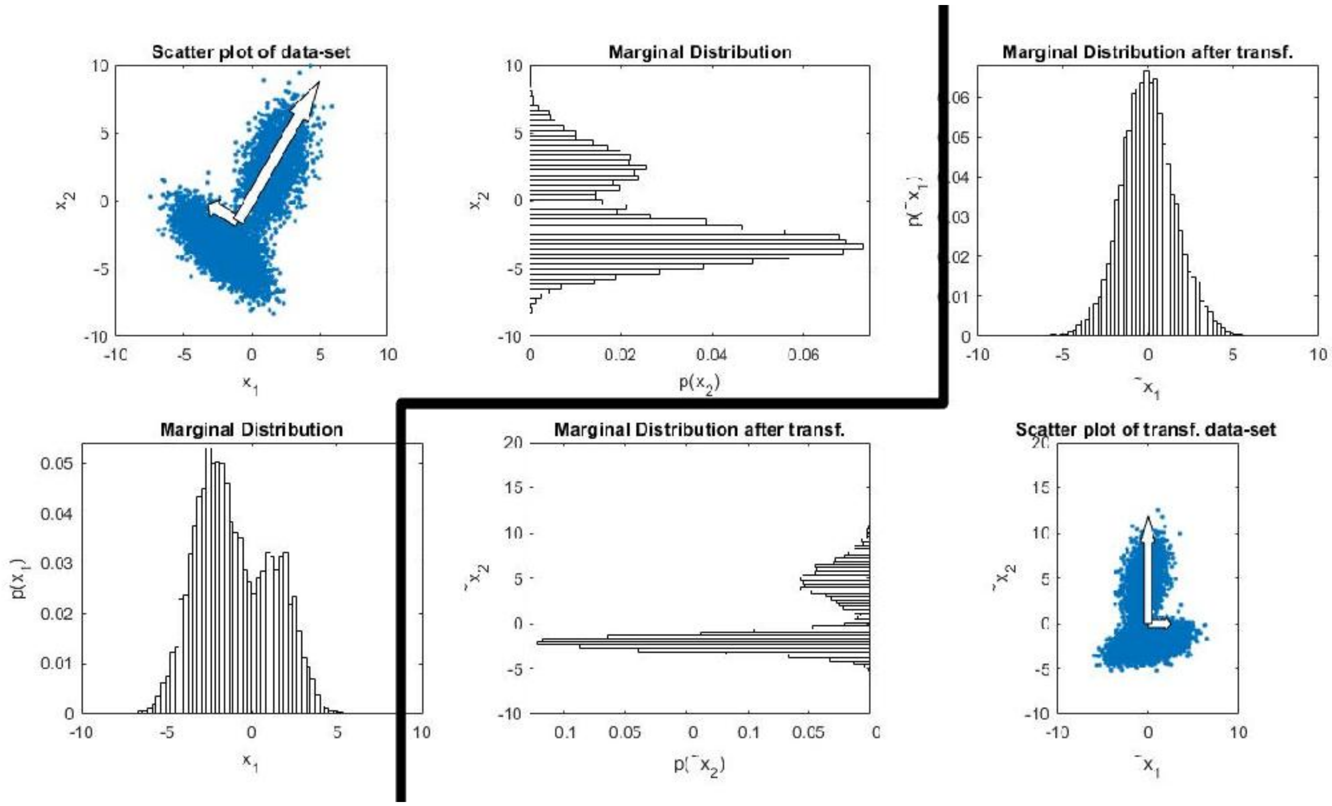
$$\tilde{\mathbf{x}}_i = \mathbf{U}^T(\mathbf{x}_i - \widehat{\mathbf{x}}). \tag{17}$$

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.

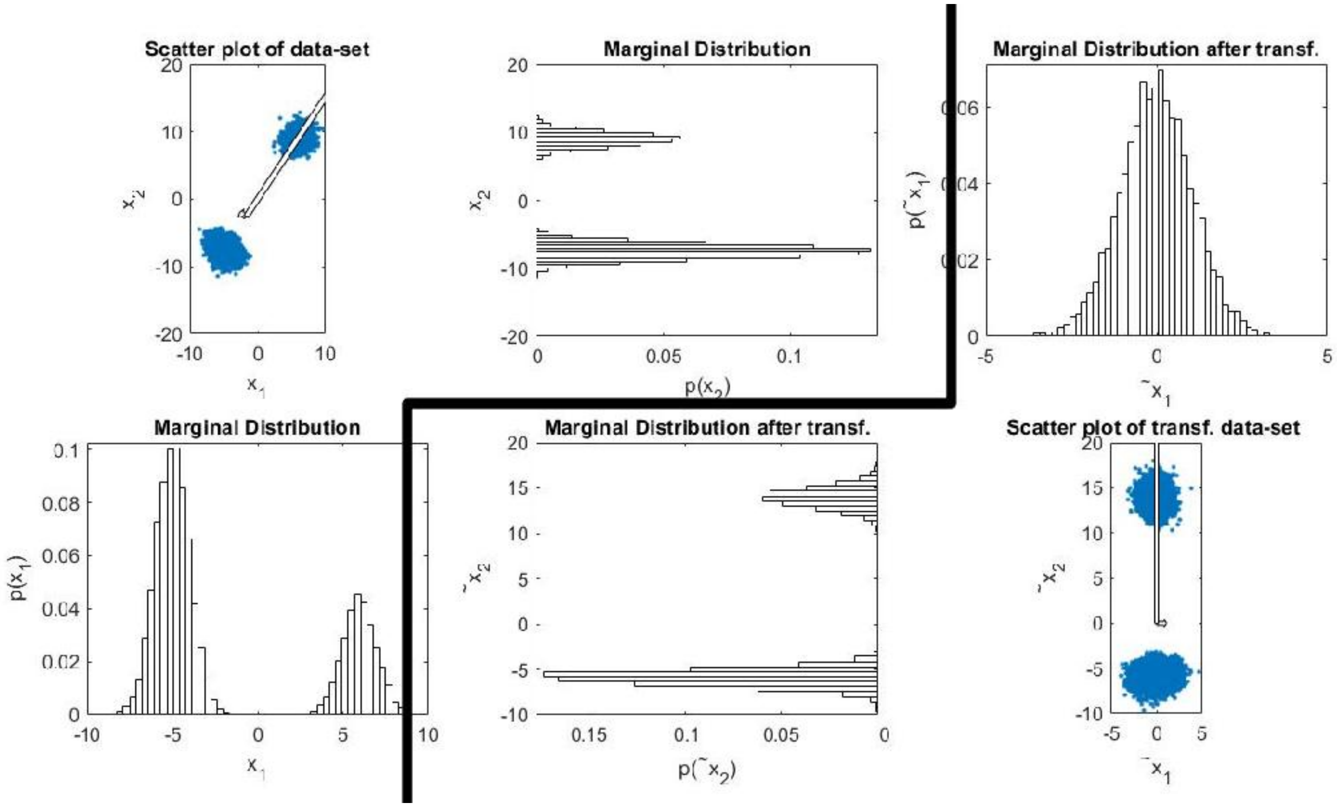
Checkpoint 2.4:

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.

In the programs we are presented with two different datasets based on two gaussian distributions each.
The first one is presented below:

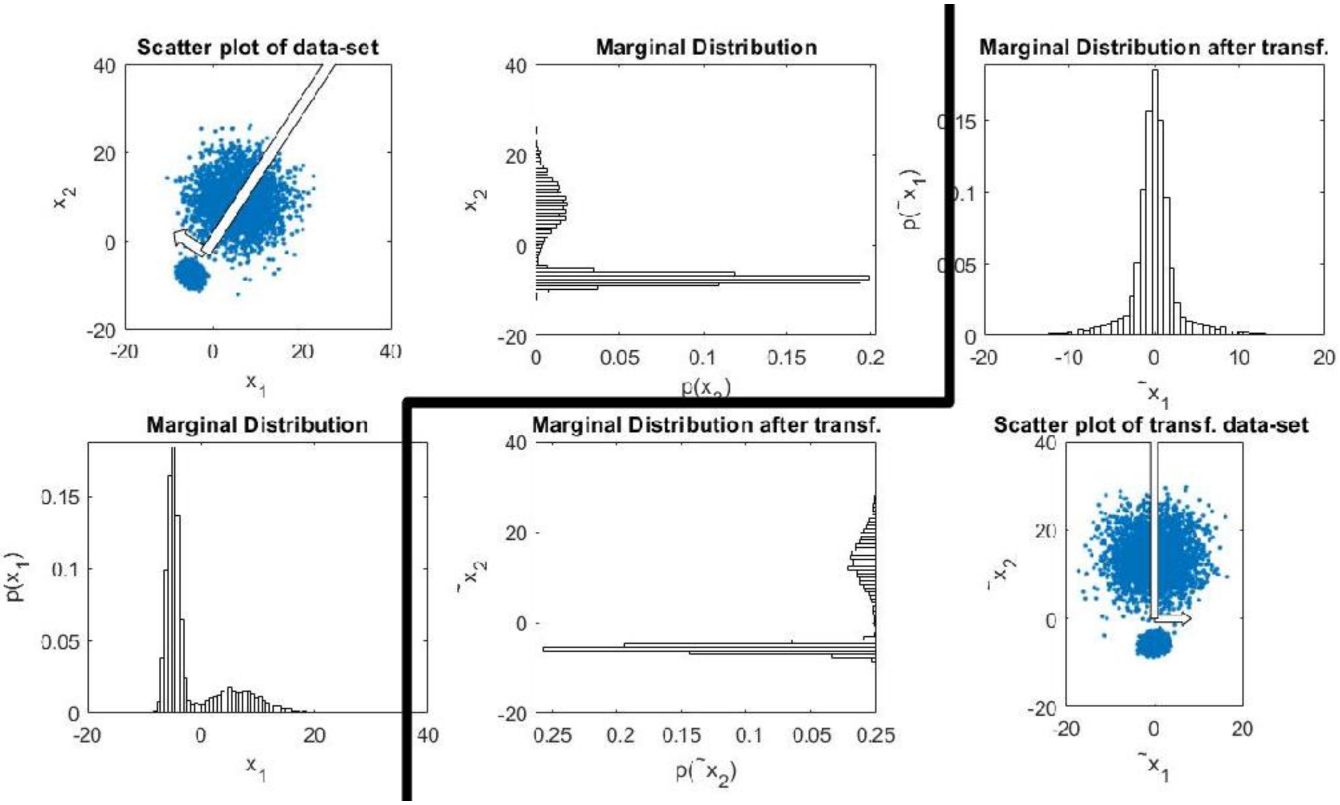


In this figure we can see the same dataset before (left) and after (right) the coordinate transformation. Before the coordinate transformation both axis seem to be holding some features about the data. However, after applying the transformation we can see that the x1 axis is just a normal distribution, which may lead us to think that this dimension is not holding any valuable feature for our model, and all the features are in the axis with the biggest eigenvalue, x2. However, where to place the threshold between the difference of eigenvectors can depend on our dataset, model and application. For example, even though we are seen a normal distribution in the x axis we can note the difference between distributions when looking at the data.



In the case of the next program the number of features to be extracted from the x1 axis is even smaller, as we can note from the relative size of the eigenvectors.

This dimensionality reduction can be useful from a machine learning perspective due to different reasons. First of all, it will reduce the size of your data in memory, with the consecuence improvement in speed. Secondly, this removal of multi-colinearity improves the performance of the ML models. This can be visualized by taking the previous example and adding noise to one of the gaussian distributions.



It can be noted that the x1 axis is not that similar to a gaussian distribution anymore, and it may seem like you can extract some features from that axis as well. However, if you do, you are risking to overfit your model to your training data, as you are just modelling noise in that dimension.

Finally, when reducing the dimensions enough it becomes possible to visualize them. making it easier to work with them.