COURSE 02457

Non-Linear Signal Processing: Answers to Exercise 2

Checkpoint 2.1

One way of reducing the dimensions of a large data set following a gaussian distribution, or the amount of information, 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.

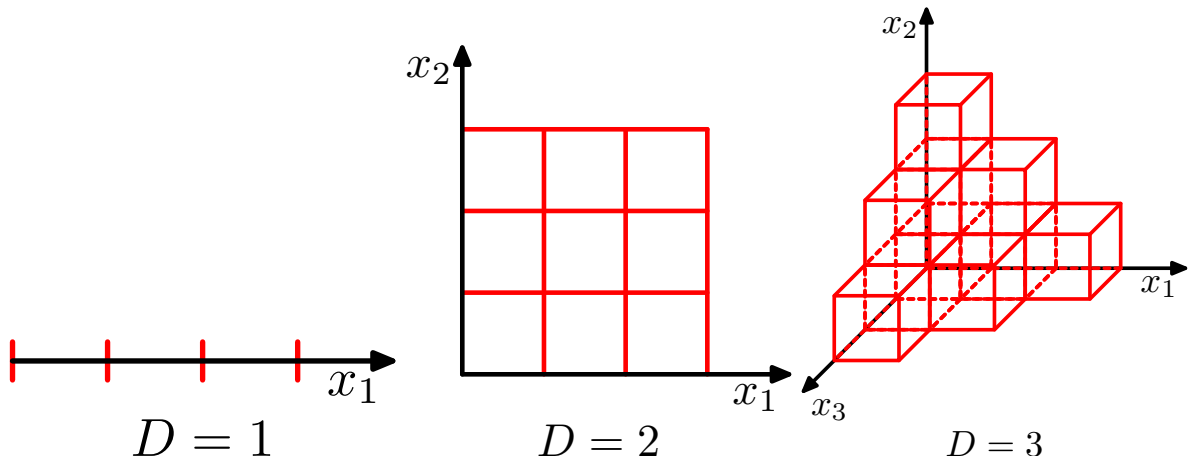
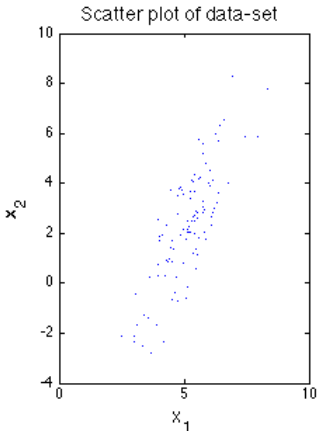


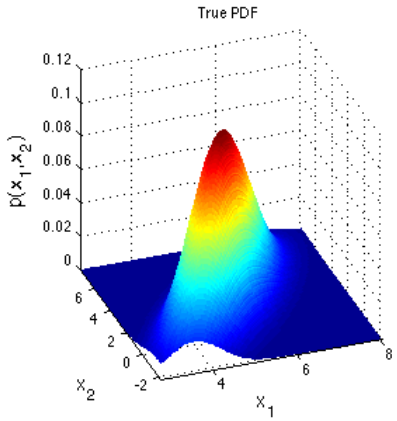
Figure 1: The size of the interior of a bin increases as the dimensionality increases, and so one needs more data points to get good averages in each bin.

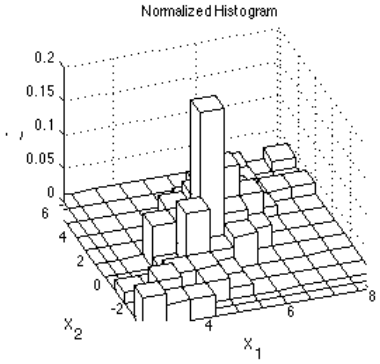
It turns out that a good measure for how many sample points N you need if you want to use $nbins$ number of bins in D dimensions with the average number of points per bin being ρ:

$$N = nbins^D \cdot \rho \tag{1}$$

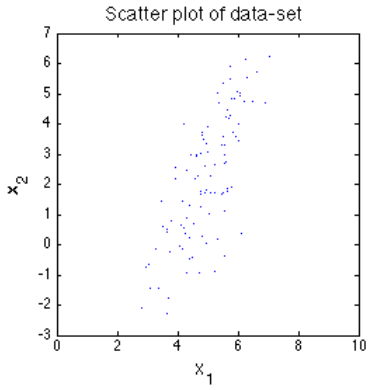
N = 100, nbins = 10

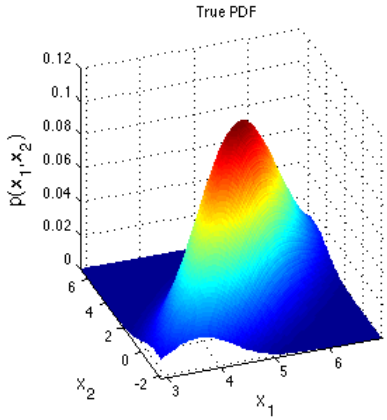


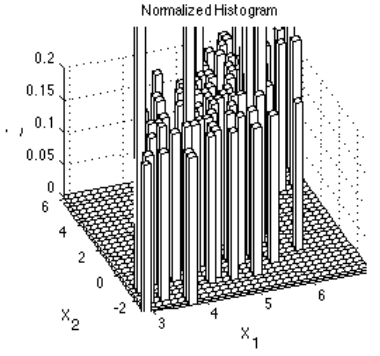




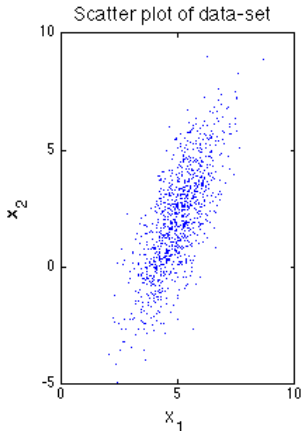
N = 100, nbins = 30

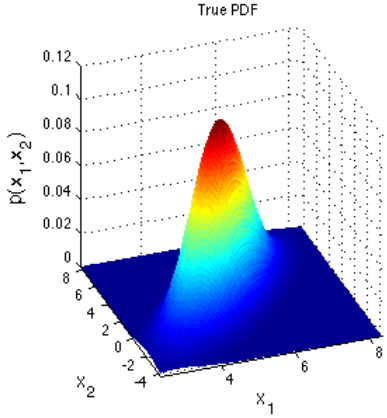


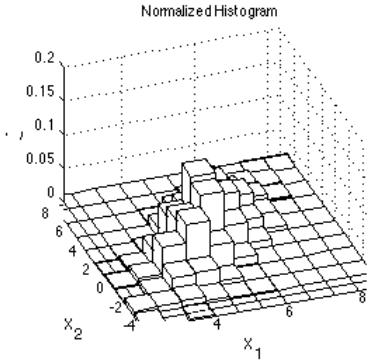




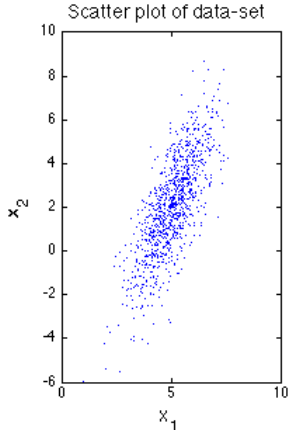
N = 1000, nbins = 10

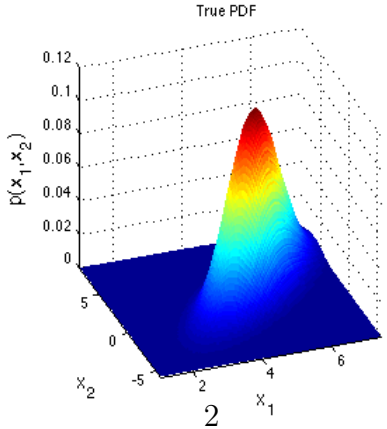


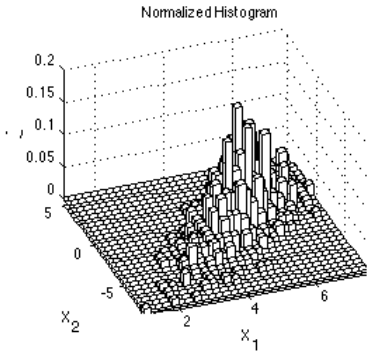


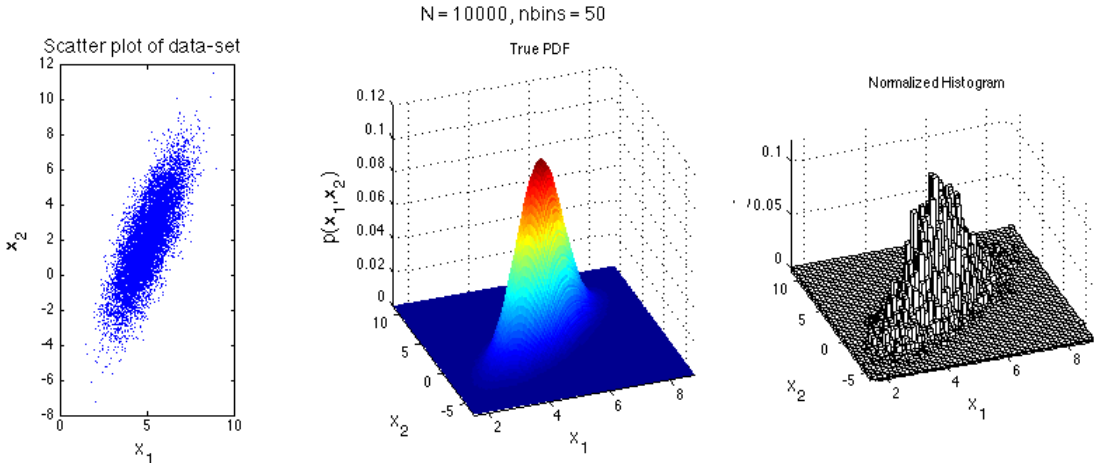


N = 1000, nbins = 30








N = 10000, nbins = 50
Scatter plot of data-set
True PDF
Normalized Histogram
x2
x1
p(x1,x2)
X2
X1

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.

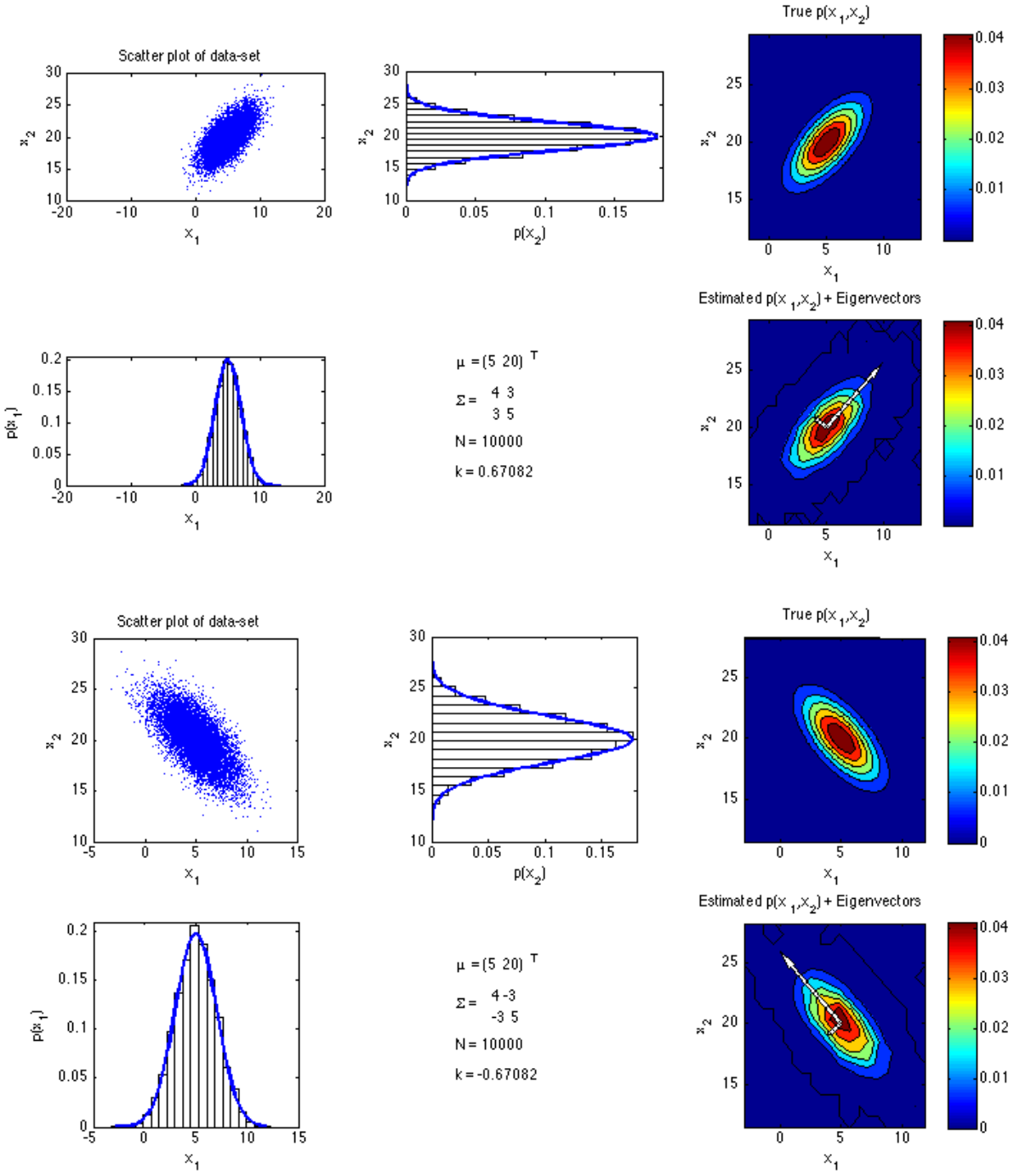


Figure 3: Checkpoint 2.2. True and estimated pdf's together with data generated with covariance matrices with (top) positive and (bottom) negative covariance.

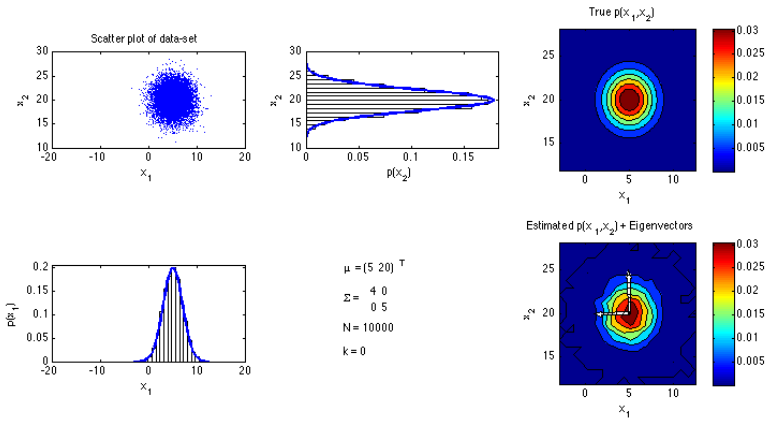


Figure 4: Data set where the variables have zero covariance.

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. (14) (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.

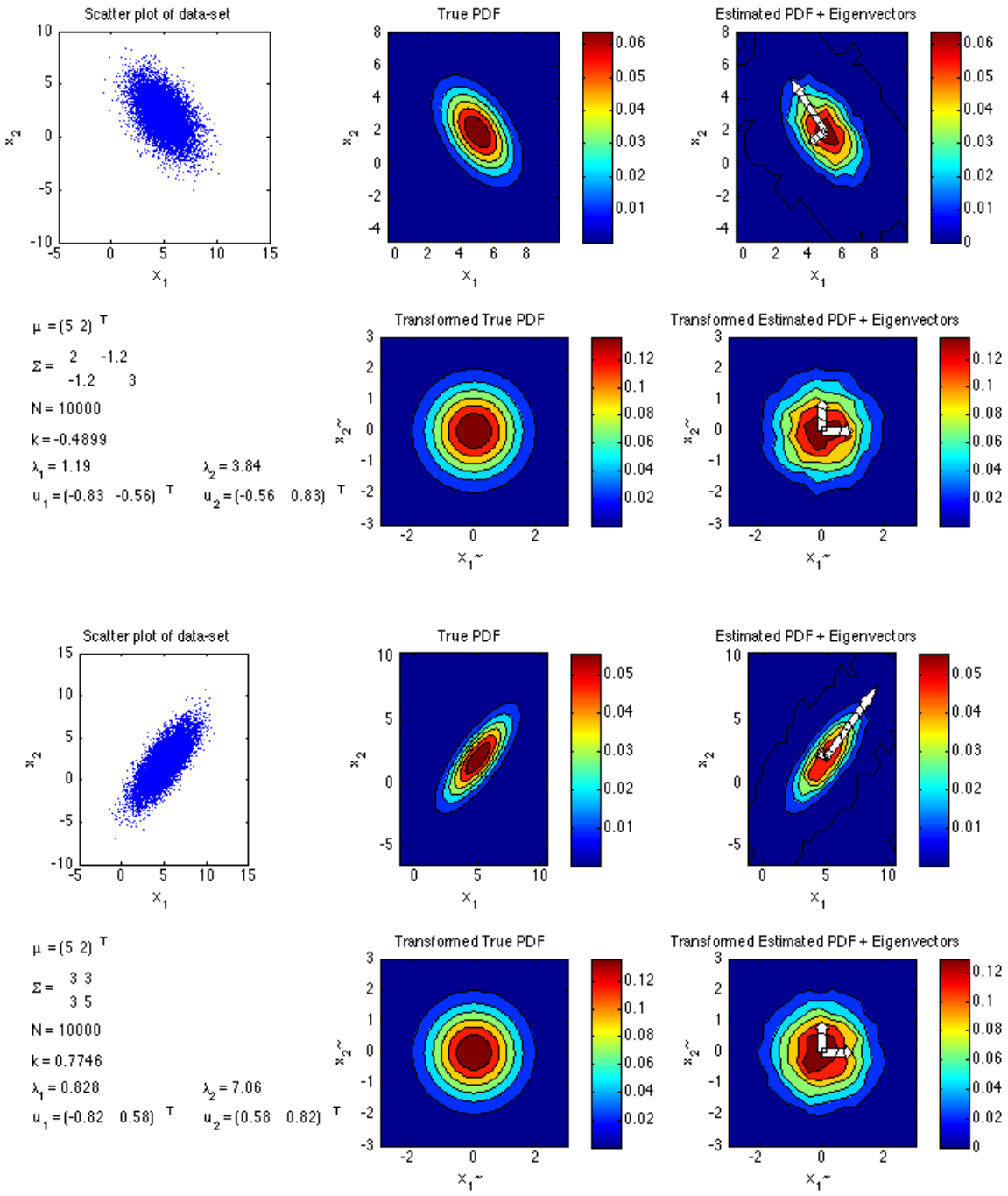


Figure 5: Two different data sets that are transformed (rotated and scaled).

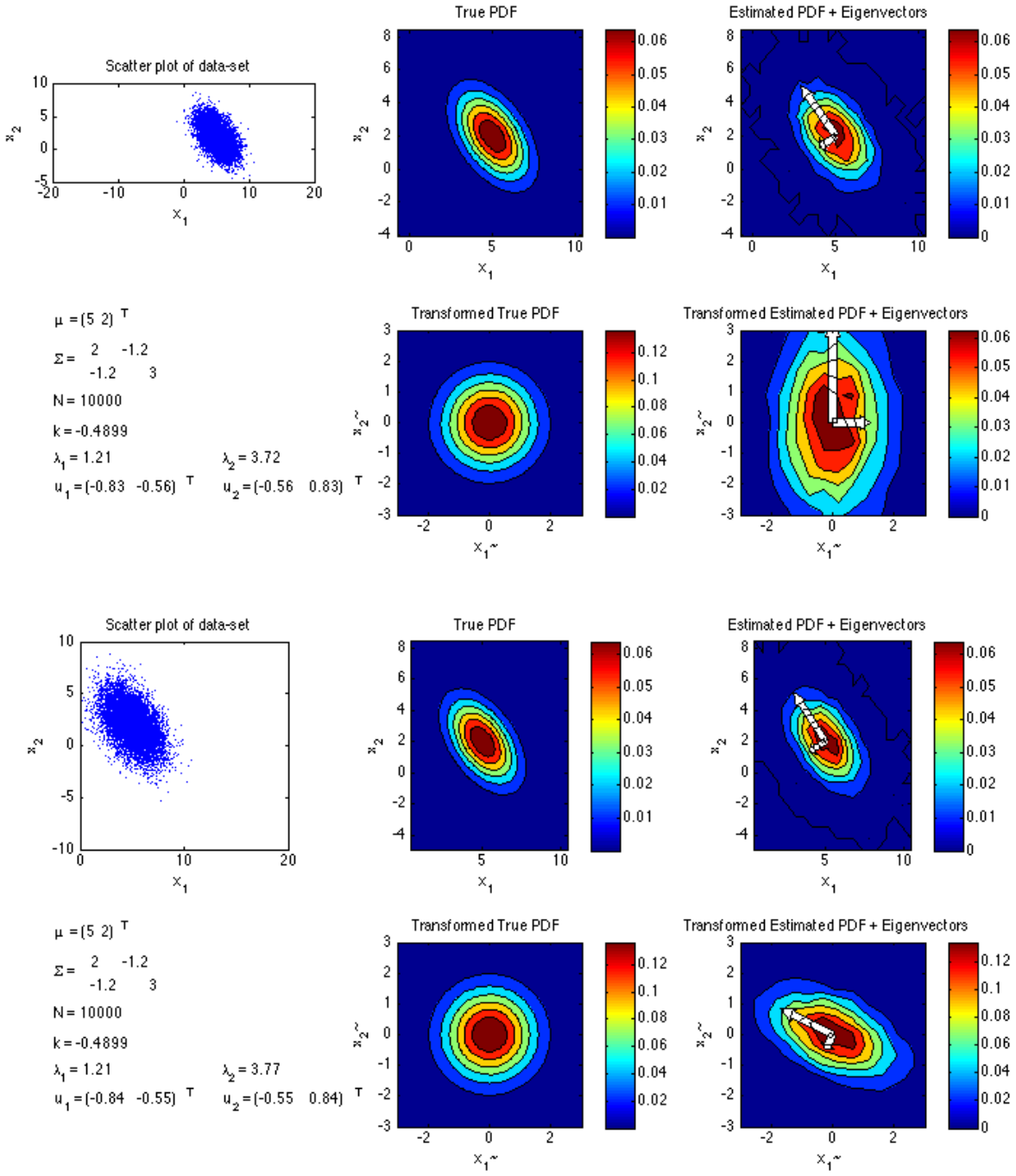


Figure 6: **Top:** Data that has been rotated (only) using the eigenvectors of the estimated covariance matrix. **Bottom:** Data that has been scaled (only) using the eigenvectors of the estimated covariance matrix. Neither of these two procedures results in zero mean, unit variance and zero covariance.

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. 7. 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.

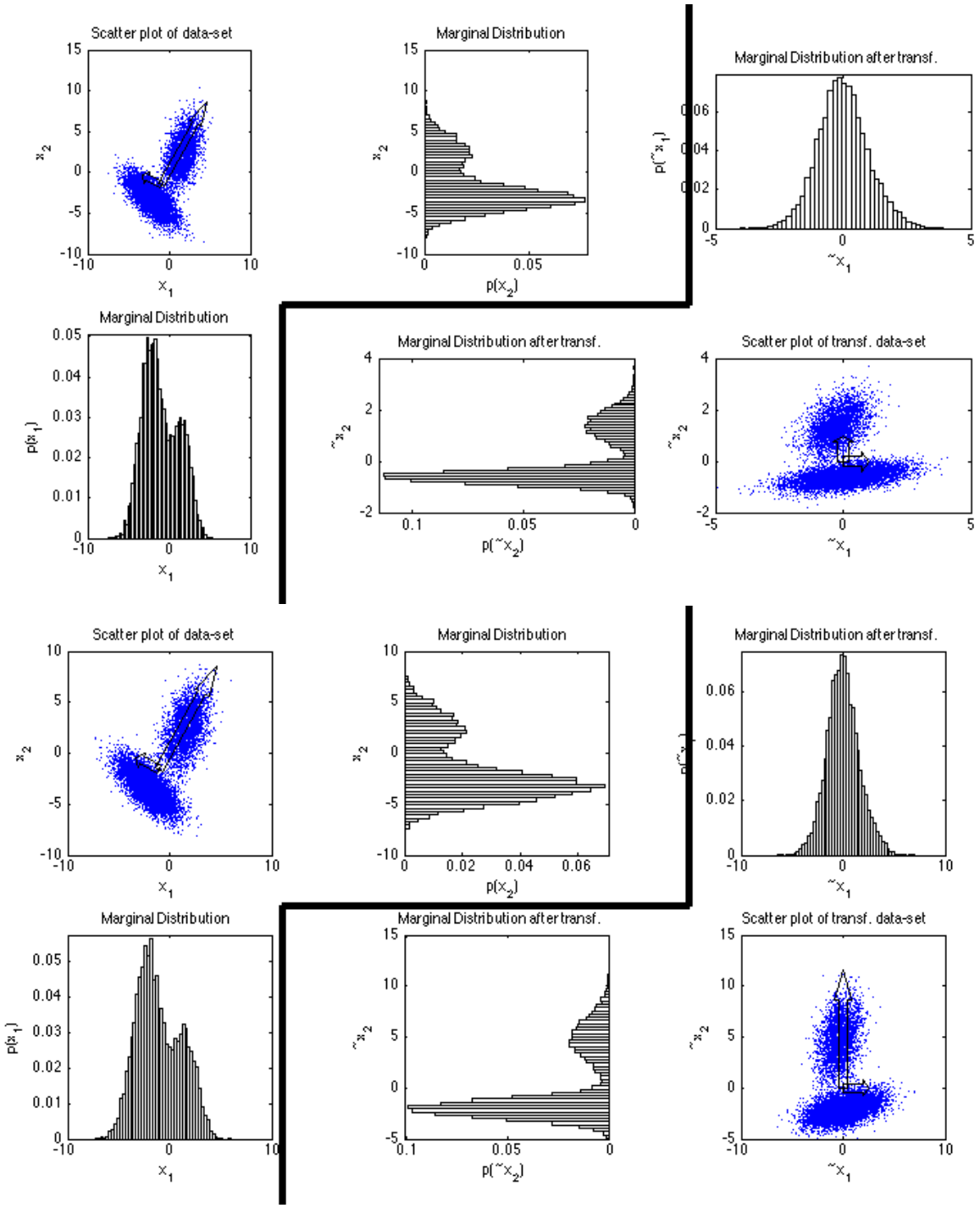


Figure 7: **Top:** Raw data set and fully transformed (rotated and scaled) data. **Bottom:** Raw data and rotated data set.

Script **main2e** is almost the same as **main2d**, only this time the gaussians in the mixture are more separate, see Fig. 8. Here, projecting the 2D information down to only one dimension is an obvious choose.

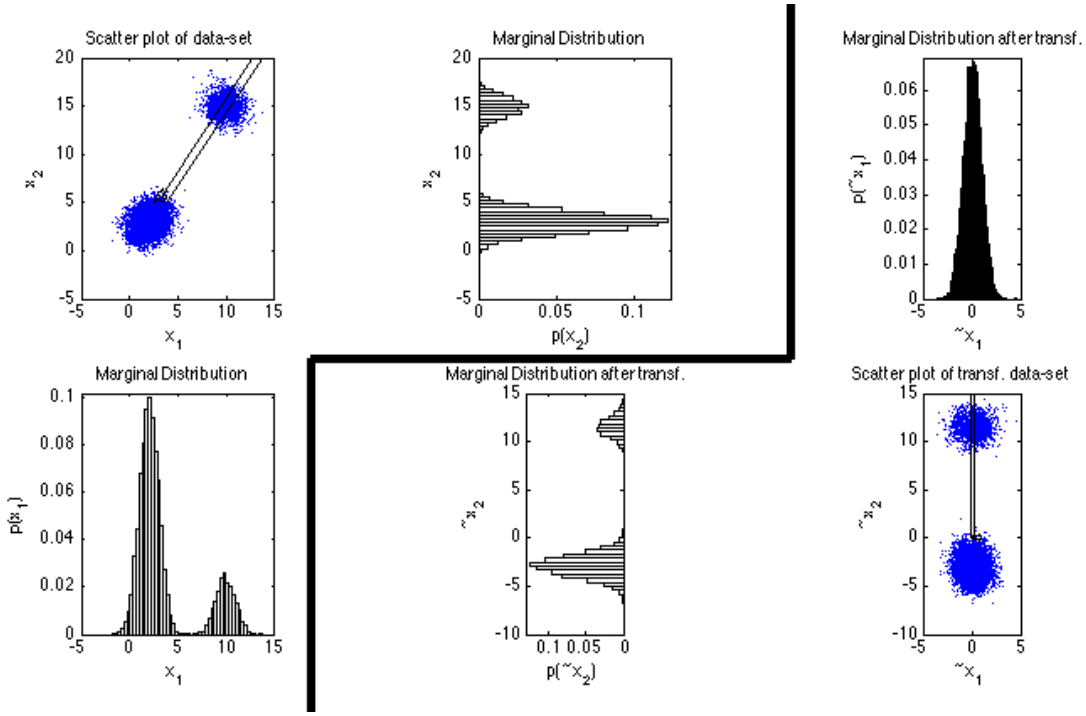


Figure 8: In this case, there is no point in representing the data in two dimensions, so it is projected down to the largest eigenvector.