Computer Assignment (CA) NO. 7: Visualization

Tyler Berezowsky

March 11, 2015

1 Problem Statement

The goal of this assignment is to help you visualize what covariance means in terms of the shape of a pdf. We will rely heavily on ~~MATLAB's~~ matplotlib's 3D visualization tools. We will work exclusively with two random variables. The tasks to be accomplished are:

1. Generate a large number of random vectors of dimension 2 consisting of two independent uniformly distributed random numbers over the range [0,1]. Estimate the pdf and plot in 3D using MATLAB's functions like surf, mesh or the equivalent. Describe the shape of this plot.

2. Generate a set of Gaussian random vectors that have a 2x2 covariance matrix, estimate the pdf, and plot in 3D. To keep things simple, use a mean of [6,6] for each example. Use the following covariance matrices:

$$cov_1(X,Y) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{1.1}$$

$$cov_2(X,Y) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{1.2}$$

$$cov_3(X,Y) = \begin{pmatrix} 5 & 0 \\ 0 & 2 \end{pmatrix} \tag{1.3}$$

$$cov_4(X,Y) = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \tag{1.4}$$

$$cov_5(X,Y) = \begin{pmatrix} 5 & 0.5 \\ 0.5 & 2 \end{pmatrix} \tag{1.5}$$

2 Approach and Results

2.1 2 Dimensional Uniform Distribution

To develop a multivariate uniform distribution in $\Re 2$, a two dimensional vector x with length N consisting of uniformly distributed values was generated.

$$x = \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{1,2} \\ \cdots & \cdots \\ x_{N,1} & x_{N,2} \end{pmatrix} \tag{2.1}$$

This was fed into a two dimensional histogram function with a bin size B for each axis. The bins were then normalized by dividing by the total number of possibilities N^2. The resulting PDF was plotted in $\Re 3$. An illustration with $N = 1 \times 10^6$ and $B = 20$ can be seen in figure 2.1 below.

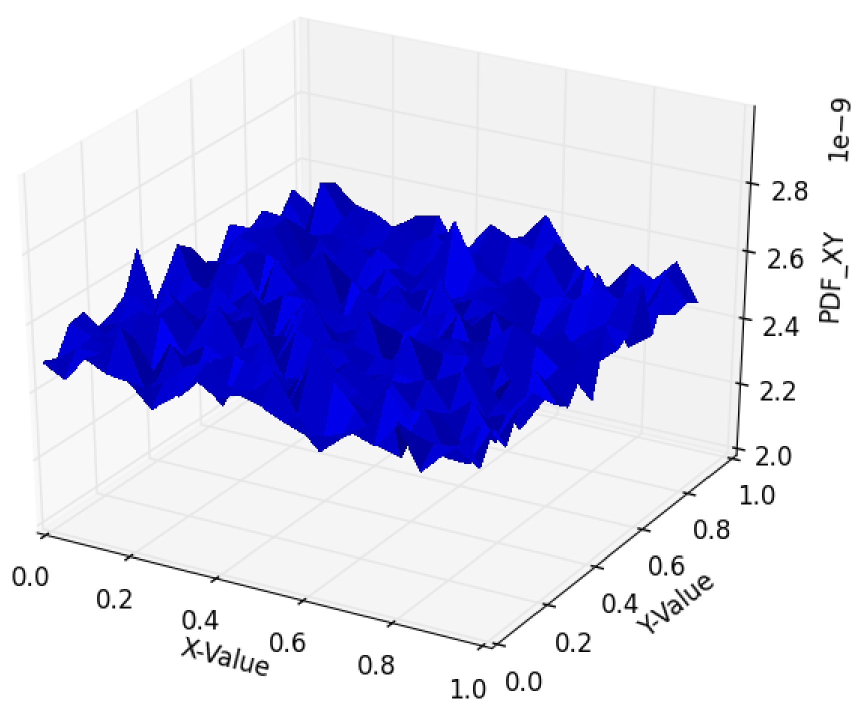

Figure 2.1: 2 dimensional uniform distribution.

As expected, the distribution is flat. The illustration does exhibit slight distortion from the expected shape due the resolution provided by the small range of the plot's axes. The variance of the PDF confirms the "flatness" of the plot. $\sigma = 2.62 \times 10^{-21}$.

2.2 2 Dimensional Normal Distribution

The $\Re 2$ multivariate normal distribution was generated utilizing a built in function `multivariate_normal()` which is fed the length (N), dimension (2), and covariance matrix (cov_k). The output of the function is identical to the shape of x described in the previous section, and therefore is treated also identically to illustrate in $\Re 3$.

To aid visualization, contour lines were added in addition to the surface plots. Two sets of plots were generated for covariance matrices, one consisting of both the surface and contour plots, and another consisting of only the contour plots. The result with $N = 1 \times 10^6$, $B = 25$, $\mu = (6,6)$ and the covariance matrices 1 to 5 can be seen below in figures 2.2 and 2.3.

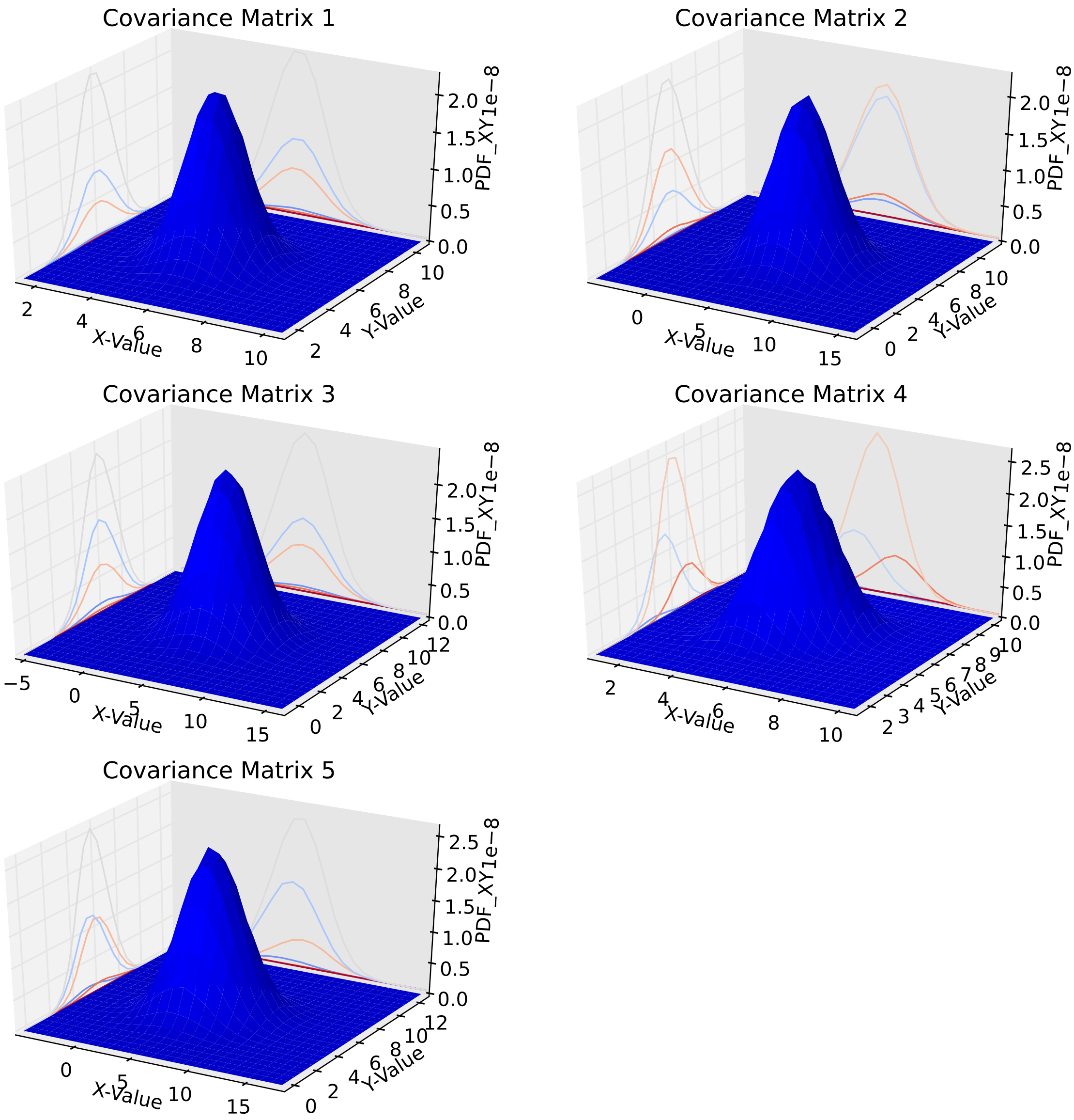

Figure 2.2: 2 dimensional normal distribution illustrated through surface and contour lines.

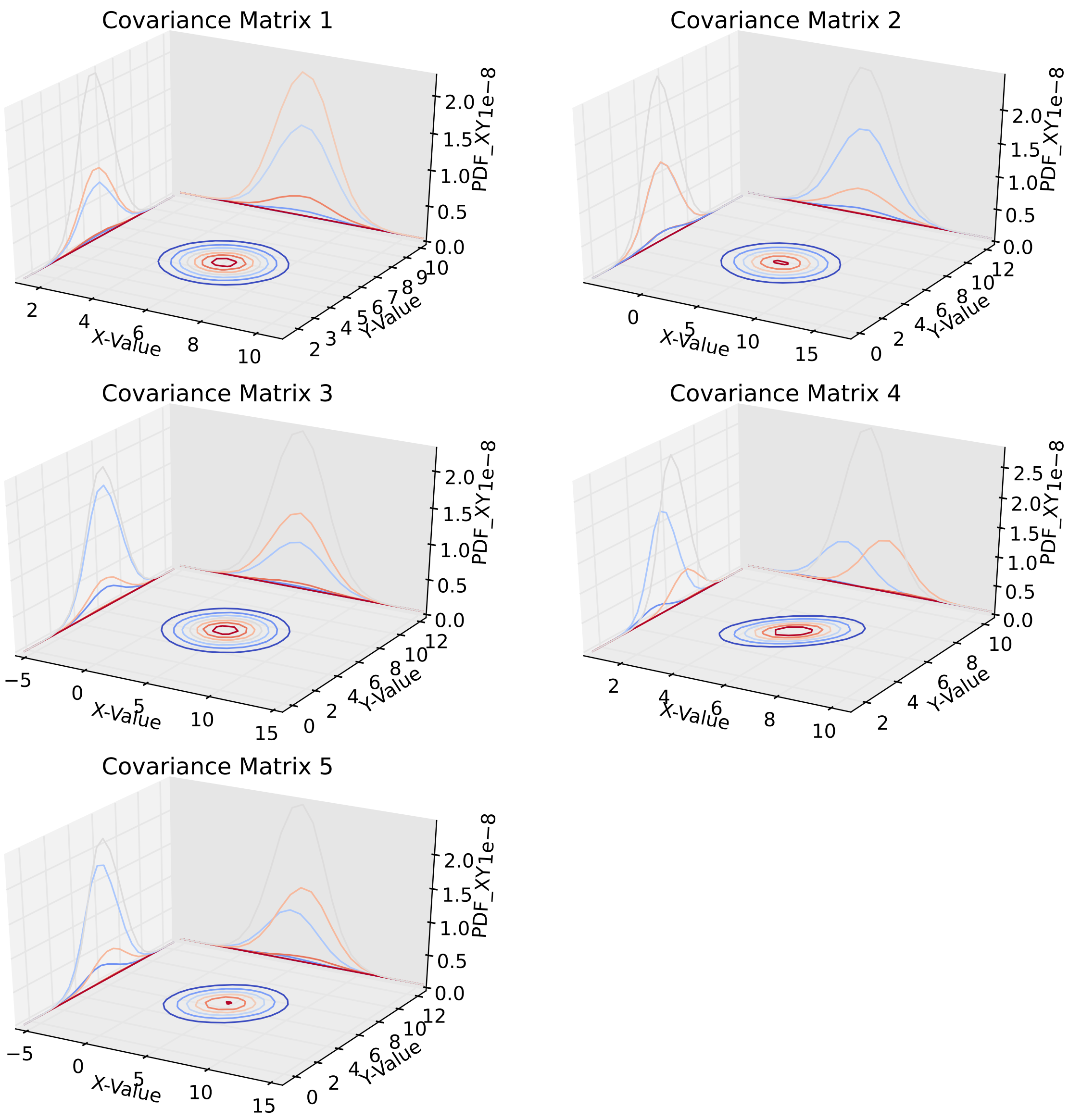

Figure 2.3: 2 dimensional normal distribution illustrated throughcontour lines.

cov_1 and cov_2 are identical matrices, but their multivariate distributions are clearly different in there range of possible values. There shape is nearly identical as displayed by the contour lines with a circular base and symmetrical Gaussian projections on the XZ and YZ plane. For matrix cor_3 the distribution's projections on the XY plane are still circular, but the projections on the XZ are YZ are no longer symmetrical to each other. There is a higher tendency and a wider range for X compared to Y.

For the matrices cor_4 and $corr_5$ the distribution's projections are not symmetrical in the XZ and YZ planes, and the projections in XY plane are now elliptical. The matrices cov_1, cov_2, and cov_3 are all "fundamentally identity matrices" (Is there a word?) where the PDF of X and Y maintain independence in the distribution. This is not the case for cov_4 and cov_5.

3 Python Code

Listing 1: Script for task (1).

```python
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm

'''
PART I
'''


def cahist(N, BINS):
    N = int(N)
    x, y = np.random.rand(2, N)
    H, xbins, ybins = np.histogram2d(x, y, bins=BINS)
    return H, xbins, ybins

N = 1e6
pdf_xy, xbins, ybins = cahist(N, 20)
print pdf_xy
pdf_xy = pdf_xy/(N**2)

fig = plt.figure()
X, Y = np.meshgrid(xbins[:-1], ybins[:-1])  # create meshgrid from bin edges
ax = fig.gca(projection='3d')
ax.set_zlim(2e-9, 3e-9)
surf = ax.plot_surface(X, Y, pdf_xy, rstride=1, cstride=1, linewidth=0,
                       antialiased=False)  # plot ampltude of pdf_xy
ax.set_xlabel('X-Value')
ax.set_ylabel('Y-Value')
ax.set_zlabel('PDF_XY')
fig.patch.set_facecolor('white')
```

Listing 2: Script for tast (2).

```python
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm


'''
PART II
np.radom.rand() generate a normal distribution with a \sigma^2 = 1 and
    \mu = 0
'''


def cahistn(N, mu, corr_mat, BINS):
    N = int(N)
    normdist = np.random.multivariate_normal(mu, corr_mat, N)
    H, xbins, ybins = np.histogram2d(normdist[:, 0], normdist[:, 1],
                                     bins=BINS)
```

```
    return H, xbins, ybins

# LOAD CONSTANTS
N = 1e6
mu = [6, 6]
BINS = 25

# LOAD CORRELATION MATRICES
corr1 = [[1, 0], [0, 1]]
corr2 = [[1, 0], [0, 1]]
corr3 = [[5, 0], [0, 2]]
corr4 = [[1, 0.5], [0.5, 1]]
corr5 = [[5, 0.5], [0.5, 2]]
corr_mats = [corr1, corr3, corr3, corr4, corr5]



fig = plt.figure(figsize=(10, 10))
for index, corr_mat in enumerate(corr_mats):
    pdf_xy, xbins, ybins = cahistn(N, mu, corr_mat, BINS)
    pdf_xy = pdf_xy/(N**2)

    X, Y = np.meshgrid(xbins[:-1], ybins[:-1])  # create meshgrid from binedges
    ax =  fig.add_subplot(3, 2, index+1, projection='3d')
    #surf = ax.plot_surface(X, Y, pdf_xy, rstride=1, cstride=1, linewidth=0,
    #                        antialiased=False)  # plot ampltude of pdf_
    cset = ax.contour(X, Y, pdf_xy, zdir='z', offset=0, cmap=cm.coolwarm)
    cset = ax.contour(X, Y, pdf_xy, zdir='x', offset=min(xbins), cmap=cm.coolwarm)
    cset = ax.contour(X, Y, pdf_xy, zdir='y', offset=max(ybins), cmap=cm.coolwarm)

    ax.set_xlabel('X-Value')
    ax.set_ylabel('Y-Value')
    ax.set_zlabel('PDF_XY')

    str_title = 'Covariance Matrix ' \
                + str(index+1)
    plt.title(str_title)
    fig.patch.set_facecolor('white')
    plt.tight_layout()
```

4 Conclusions

Visualizing data in $\Re3$ is entertaining. A two dimensional uniform uncorrelated PDF is flat. A two dimensional normal distribution which is uncorrelated has circular projection on the XY plane. A two dimensional normal distribution which is correlated has an elliptical projection on the XY plane.