Computer Assignment (CA) No. 7:   
Visualization

Truc Le

ECE 3512: Signals – Continuous and Discrete

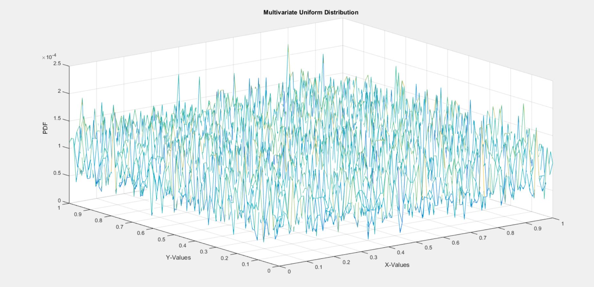
Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 1912

# Problem Statement

The overall purpose of this assignment were to create visual plots of multivariate random distributions. The first part of the assignment was to generate these random numbers over a uniform distribution, estimate the pdf, and then we would have to plot it. As for the second part of the assignment, we required to perform the same operation on several multivariate normal distributions, each with a mean vector of [6 6] and different covariance matrices. The covariance matrices are:

# Approach and Results

In part 1 of the assignment, I would have to create a matrix of size 100000 x 2, with each entry containing a random number generated based on a uniform distribution. I then made two linearly spaced vector from 0 to 1, with a spacing of 0.01 to create 100 bins that will be used in finding the pdf. From here I made a histogram, took the values and normalize them to estimate the pdf, and plotted using the mesh function shown below in Figure 1.

   
Figure 1: Multivariate Uniform PDF

For the second part, the method was basically similar. However, I did not make an initial matrix containing the normally distributed values. But instead, I made two normal random vectors that have each have a mean of 6 then used those to create a multivariate normal pdf using the mvnpdf() function. I generated one data set, and used different covariance matrices to get different plots as shown in Figure 2 to Figure 6 below.

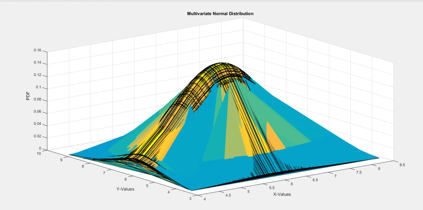
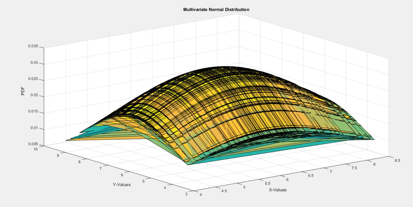
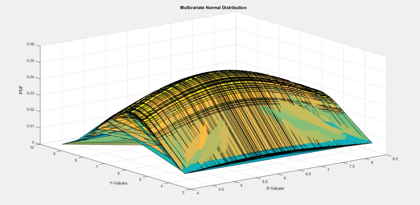
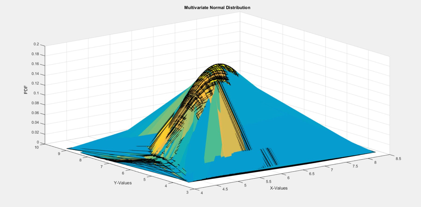
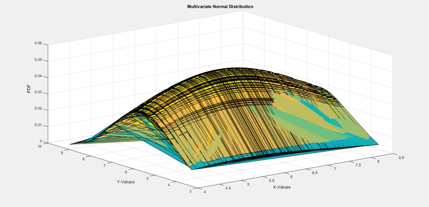


Figure 2: MVN PDF using cov1

  
Figure 3: MVN PDF using cov2

  
Figure 4: MVN PDF using cov3

  
Figure 5: MVN PDF using cov4

  
Figure 6: MVN PDF using cov5

# MATLAB Code

clear all; close all; clc;

%Initialize vector to hold uniform random numbers

xn = zeros(100000,2);

for i = 1:length(xn)

xn(i,:) = rand(1,2);

end

%create axis vectors and make mesh grid

%to make a grid for plotting in 3D

x = 0:0.01:1;

y = 0:0.01:1;

[X,Y] = meshgrid(x,y);

%make a 3D histogram and normalize it to estimate the pdf

%plot using mesh

h1 = hist3(xn,[101 101]);

pdf = h1./(length(xn));

mesh(X,Y,pdf);

title('Multivariate Uniform Distribution');

xlabel('X-Values');

ylabel('Y-Values');

zlabel('PDF');

The above code would find the multivariate pdf of the multivariate uniform distribution. Here, I initialized a vector, created a grid matrix to plot over, and made a histogram and normalized the values to create the pdf.

%%%initialize mean vector and covariance matrices

mu = [6 6];

cov2(:,:,1) = [1 0;0 1];

cov2(:,:,2) = [5 0;0 5];

cov2(:,:,3) = [5 0;0 2];

cov2(:,:,4) = [1 0.5;0.5 1];

cov2(:,:,5) = [5 0.5;0.5 2];

%create normally distributed random vectors use create a grid for 3D

%plotting

x2 = 6 + randn(1,100);

y2 = 6 + randn(1,100);

[X2,Y2] = meshgrid(x2,y2);

%for each covariance matrix, find the multivariate normal pdf of the data

%set and plot

for i = 1:length(cov2)

F = mvnpdf([X2(:),Y2(:)],mu,cov2(:,:,i));

F = reshape(F,length(y2),length(x2));

figure(i+1);

surf(x2,y2,F);

caxis([min(F(:))-.5\*range(F(:)),max(F(:))]);

title('Multivariate Normal Distribution');

xlabel('X-Values');

ylabel('Y-Values');

zlabel('PDF');

end

The second part of the code was somewhat similar to the previous part 1 code except that no initial matrix is created to hold the random numbers. The numbers are contained within their own respective vectors, and the pdf is found using a function rather than a histogram.

# Conclusions

As a result, this assignment basically enforced the visualization of multivariate random distributions and their pdf’s. From this assignment we can see that the pdf’s are 3-dimensional surfaces with the Z-axis representing the pdf values. In the first part of the assignment, the shape that is generated is similar to a square. This can be expected as the univariate graph of a uniform pdf is a square. The graph shows that the numbers will all have similar probabilities. This again expected because in a univariate case, all the numbers will have very similar probabilities. Since the z-axis represents the probabilities, the height for each number will be close, which resulted in the prismatic rectangular shape.

For the second part of the assignment, once more, we get the results we’d expect from analyzing the univariate normal distribution. The covariance matrix defines the support region for each Gaussian curve. The support region is the ellipse found by plotting the probabilities of numbers with the same probability, given that covariance matrix and this would give the altered shape to each curve.