Module 6 - Homework 6

Colt Thomas

March 26, 2021

Problem 1

For the joint PMF shown below, determine the correlation coefficient $\gamma_{X,Y}$.

$$p_{X,Y}(0,0) = \frac{1}{8}; p_{X,Y}(0,1) = \frac{1}{8}; p_{X,Y}(1,0) = \frac{1}{4}; p_{X,Y}(1,1) = \frac{1}{2}$$

The correlation coefficient between two random variables is defined as:

$$\gamma_{X,Y} = \frac{cov(x,y)}{\sigma_X \sigma_Y}$$

First find the covariance:

$$cov(x,y) = E[XY] - E[Y]E[X],$$

$$E[XY] = \sum_{i=0}^{1}\sum_{j=0}^{1} x_i y_j p_{X,Y}(x_i, y_j)$$

$$E[XY] = (0)(0)\frac{1}{8} + (0)(1)\frac{1}{8} + (1)(0)\frac{1}{4} + (1)(1)\frac{1}{2}$$

$$E[XY] = \frac{1}{2}$$

$$E[X] = \sum_{i=0}^{1}\sum_{j=0}^{1} x_i p_{X,Y}(x_i, y_j)$$

$$E[X] = (0)\frac{1}{8} + (0)\frac{1}{8} + (1)\frac{1}{4} + (1)\frac{1}{2}$$

$$E[X] = \frac{3}{4}$$

$$E[Y] = \sum_{i=0}^{1}\sum_{j=0}^{1} y_j p_{X,Y}(x_i, y_j)$$

$$E[Y] = (0)\frac{1}{8} + (1)\frac{1}{8} + (0)\frac{1}{4} + (1)\frac{1}{2}$$
$$E[Y] = \frac{5}{8}$$

The covariance is then....

$$cov(x, y) = \frac{1}{2} - \frac{3}{4} * \frac{5}{8}$$
$$cov(x, y) = \frac{1}{32}$$

Now we need to find the variance:

$$Var(X) = \sum_{i=0}^{1}\sum_{j=0}^{1}(x_i - E[X])^2 p_{X,Y}(x_i, y_j)$$
$$Var(X) = (0 - \frac{3}{4})^2(\frac{1}{8}) + (0 - \frac{3}{4})^2(\frac{1}{8}) + (1 - \frac{3}{4})^2(\frac{1}{4}) + (1 - \frac{3}{4})^2(\frac{1}{2})$$
$$Var(X) = \frac{3}{16}$$

$$Var(Y) = \sum_{i=0}^{1}\sum_{j=0}^{1}(y_j - E[Y])^2 p_{X,Y}(x_i, y_j)$$
$$Var(Y) = (0 - \frac{5}{8})^2(\frac{1}{8}) + (1 - \frac{5}{8})^2(\frac{1}{8}) + (0 - \frac{5}{8})^2(\frac{1}{4}) + (1 - \frac{5}{8})^2(\frac{1}{2})$$
$$Var(Y) = \frac{15}{64}$$

And the correlation coefficient is:

$$\gamma_{X,Y} = \frac{1/32}{\sqrt{3/16}\sqrt{15/64}}$$
$$\boxed{\gamma_{X,Y} = \frac{\sqrt{5}}{15} \approx 0.149}$$

Next use a computer simulation to generate realizations of the random vector(X,Y)and estimate the correlation coefficient below.

After completing a simulation, the correlation coefficient approaches $\boxed{0.1492}$ when using 10,000,000 realizations. Below in figure is a snippet of the equations used for completing this calculation. Matlab code is included in the appendix.

$$\hat{r}_{X,Y} = \frac{\overline{E[XY]} - \bar{x}\bar{y}}{\sqrt{(\overline{xsq} - \bar{x}^2)(\overline{ysq} - \bar{y}^2)}}$$

where

$$\overline{E[XY]} = \frac{1}{M}\sum_{m=1}^{M} x_m y_m; \quad \overline{xsq} = \frac{1}{M}\sum_{m=1}^{M} x_m^2; \quad \overline{ysq} = \frac{1}{M}\sum_{m=1}^{M} y_m^2; \quad \bar{x} = \frac{1}{M}\sum_{m=1}^{M} x_m; \quad \bar{y} = \frac{1}{M}\sum_{m=1}^{M} y_m$$

Problem 2

For a real Gaussian random vector $X = [X_1 X_2 ... X_n]^T$ with covariance matrix C_X, there always exist a matrix A such that $C_X = AA^T$. In terms of the Singular Value Decomposition (SVD),$C_X = UDU^T$ and the standard normalvectorZ, the transformation is

$$X = UD^{1/2}Z + \mu_X$$

where $\mu_x = [\mu_1 \mu_2 ... \mu_n]^T$. Use the randn and svd functions, and the facts above to write a function that generates an n-dimensional real Gaussian random vector. Let the function prototype be

$$\text{function x = gaussRandomVector}(\mu_X, C_X)$$

Use this function to generate one 2-D plot of Gaussian random vector samples for appropriate choices of μ_X and C_X. Also, plot the elliptical contours of the joint distribution of X on the same graph. You can utilize the ellipse.m function posted on Blackboard.

In my Matlab code, I created the gaussRandomVector() that will produce a shifted gaussian RV based on the input mean and covariance vectors. Figure shows a plot of a realization generated by this function. Additional details can be found in the comments for gaussRandomVector.m and homework6.m under the Problem 2 section.

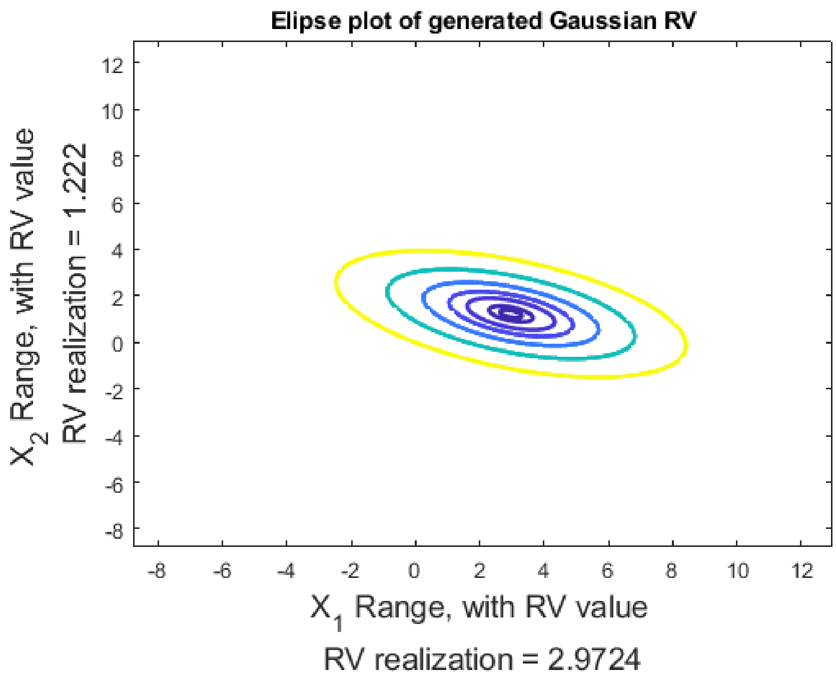

Figure 1: Plot of the realization from running the code for this problem. Note that the centers of the ellipse lines up with the X_1 and X_2 axis.

Matlab Code - Main Function

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Homework6.m
% Colt Thomas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% %%%%%%%%%%%%%%%
% Problem 1
%%%%%%%%%%%%%%%%%

% discrete PMF
p = [1/8 1/8 1/4 1/2]
M = 10e6;

% Create a joint realization of X and Y for computation
[x,y,P,pX,pY] = xyJointRealizations(p,M);

% Calculate components of the correlation coefficient.
E_XY = 1/M * sum(x.*y);

xsq = 1/M * sum(x.^2);

ysq = 1/M * sum(y.^2);

xbar = 1/M * sum(x);

ybar = 1/M * sum(y);

% Note that correlation is covariance divided by the product of the X and Y
% standard deviations.
corr_coeff = (E_XY - xbar*ybar)/sqrt((xsq - xbar^2)*(ysq - ybar^2))

%% %%%%%%%%%%%%%%%
% Problem 2
%%%%%%%%%%%%%%%%%

% This problem defines a desired mean and covariance matrix and produces
% realizations of a vector gaussian matrix.

% A Standard normal gaussian vector example
% mu_X = [1;-1];
```

```
% C_X = [1,0;0,1];

% Non-normal gaussian vector example
mu_X = [3;2];
C_X = [1,1;1,4];

% Generate a gausian vector, which is a vector of gaussian RV
% realizations based on the inputs.
[X] = gaussRandomVector(mu_X, C_X)


% The alpha variable will expand/contract the gradiant contours on the
% plot. Must be between 0 and 1.
alpha = 0.5;
s = -2*log(1-alpha);

% The limits are scaled to keep the distribution in the plot, the tolerance
% is the distance from the center of the elipse that will be shown
window = 10;
limits = [-window+min(X),window+max(X),-window+min(X),window+max(X)];

% The elipse function produces contour plot variables for visualization.
% The two centers are based on the gaussian RV values generated above, and
% the covariance acts as the rotation matrix.
Npts = 1000;
[X_1,X_2,Z,v] = ellipse(X,C_X,s,limits,Npts);

% Plotting code below
figure(1)
csq = [1/16 1/4 1 2 4 8 16].*s(1);
[cmat,h] = contour(X_1,X_2,Z,csq,'LineWidth',2);
% contour(Z,csq);
title("Elipse plot of generated Gaussian RV")
xlabel({'X_1 Range, with RV value ',['RV realization = ',num2str(X(1))]},'FontSize',14);
ylabel({'X_2 Range, with RV value ',['RV realization = ',num2str(X(2))]},'FontSize',14);
```

Matlab Code - Gaussian RV Generating Function

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% gaussRandomVector.m
% Colt Thomas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [X] = gaussRandomVector(mu_X, C_X)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input:
% mu_X = 1xn Gaussian mean vector
% C_X= Covariance matrix  of respective gaussians

% Output:
% x -  n dimensional real Gaussian random vector
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% perform the singular value decomposition of covariance matrix
[U,S,V] = svd(C_X);

% normal gaussian RV realizations
Z = randn(1,length(mu_X));

% Generate the gaussian RV shifted according to input mean and covariance
X = U*S.^(1/2)*Z' + mu_X;
```

Matlab Code - Joint Realization code Function

```matlab
function [x,y,P,pX,pY] = xyJointRealizations(p,M)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input:
% p = [pXY(0,0) pXY(0,1) pXY(1,0) pXY(1,1)], i.e. p = [.125 .125 .25 .5]
% M = Number of realizations

% Output:
% x -  M realizations of RV X
% y -  M realizations of RV Y
% P -  Estimation of of the joint PMF
%      [pXYest(0,0) pXYest(0,1); pXYest(1,0) pXYest(1,1)]
% pX - Estimation of marginal pmf of X = [pX(0) pX(1)]
% pY - Estimation of marginal pmf of Y = [pY(0) pY(1)]
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Inititalization
X = -1*ones(M,2);
P = zeros(2);
pX = zeros(1,2);
pY = zeros(1,2);

%% Realization Estimation
U = rand(M,1);
p2 = cumsum(p);

p00_idx = find(U <= p2(1));
num_p00 = numel(p00_idx);
X(p00_idx,:) = ones(num_p00,1)*[0 0];

p01_idx = find(U > p2(1) & U <= p2(2));
num_p01 = numel(p01_idx);
X(p01_idx,:) = ones(num_p01,1)*[0 1];

p10_idx = find(U > p2(2) & U <= p2(3));
num_p10 = numel(p10_idx);
X(p10_idx,:) = ones(num_p10,1)*[1 0];

p11_idx = find(U > p2(3));
```

```
num_p11 = numel(p11_idx);
X(p11_idx,:) = ones(num_p11,1)*[1 1];

x = X(:,1);
y = X(:,2);

%% Joint PMF estimation
% Counts
 p00_count = numel(find(X(:,1)==0 & X(:,2)==0));
 p01_count = numel(find(X(:,1)==0 & X(:,2)==1));
 p10_count = numel(find(X(:,1)==1 & X(:,2)==0));
 p11_count = numel(find(X(:,1)==1 & X(:,2)==1));
% Joint PMF estimate
P(1,1) = p00_count/M;
P(1,2) = p01_count/M;
P(2,1) = p10_count/M;
P(2,2) = p11_count/M;
% Marginal pmf of X estimate
pX(1) = P(1,1) + P(1,2);
pX(2) = P(2,1) + P(2,2);
% Marginal pmf of Y estimate
pY(1) = P(1,1) + P(2,1);
pY(2) = P(1,2) + P(2,2);
```

Matlab Code - Ellipse Helper Function

```matlab
function [X,Y,Z,v] = ellipse(m,A,s,limits,Npts)
% Returns an eqution of an ellips with center m and rotation matrix A
% USAGE:
%    [X,Y,Z,v] = ellipse(m,A,s,limits,Npts)
% INPUTS:
%    m = [m1 m2]'
%    A = 2x2 rotation matrix
%    s = -2*log(1-alpha) where alpha is between 0 and 1
%    limits = [xmin xmax ymin ymax]
%    Npts = number of points for the x-y plotting grid
% OUTPUTS:
%    X = x-values of grid
%    Y = Y-values of grid
%    Z = equation of the ellipse
%    v = contour level of ellipse
%
xmin = limits(1);
xmax = limits(2);
ymin = limits(3);
ymax = limits(4);

x = linspace(xmin,xmax,Npts);
y = linspace(ymin,ymax,Npts);
[X,Y] = meshgrid(x,y);

Z = A(1,1)*(X-m(1)).^2 + (A(1,2) + A(2,1))*(X-m(1)).*(Y-m(2)) + ...
    A(2,2)*(Y-m(2)).^2;

v = [s s];
```