Machine Learning Lab 1

Junming Zhang 29299527 jz1g17@ecs.soton.ac.uk

Problem 1

Answer:

```
x = rand(1000,1); % generate 1000 random numbers within 0 and 1
hist(x,40); % generate the frequency histogram for 40 ranges.
% the frequency histogram has equally 10 ranges and returns
% value of eachbin to nn and the center value of each bin to xx
[nn, xx] = hist(x);
bar(nn); % separate each bin.
```

```
x = randn(1000,1);% generate 1000 random number drawn from a Gaussian
                  % distribution of mean=0 and variance=1.
```

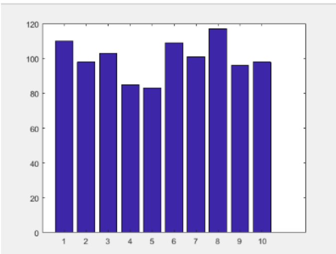

Figure 1.1: result of labone.m

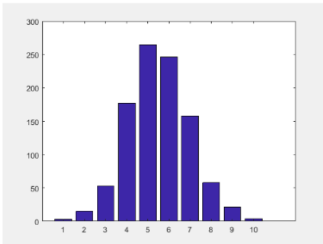

Figure 1.2: result of labone_2.m

It could find 'rand' command is uniform distribution and 'randn' is Gaussian distribution.

```
N = 1000;
x1 = zeros(N,1);% create a 1000*1 matrix with 0
for n=1:N
    % fill the matrix with two uniform random distribute arrays
 x1(n,1) = sum(rand(12,1))-sum(rand(12,1));
end
hist(x1,100); % generate the frequency histogram for 100 ranges.
```

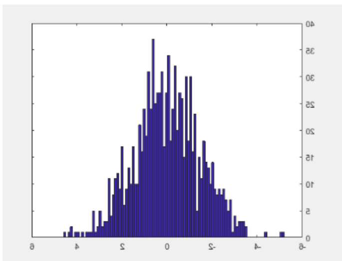

Figure 1.3: result of labone_3.m

It could find the result is similar to a Gaussian distribution. Due to the central limit theorem, in most situations, when independent random variables are added, their properly normalized sum tends toward a normal distribution.

Problem 2

```
C=[2 1;1 2]; % covariance matrix
X=randn(1000,2);% generate 1000*2 matrix with Gassian distribution
A=chol(C); % A^t&A = C
Y=X*A;
% mark Gaussian distribution with cyan potins
% mark another Y with magenta x
plot(X(:,1),X(:,2),'c.', Y(:,1),Y(:,2),'mx');
```

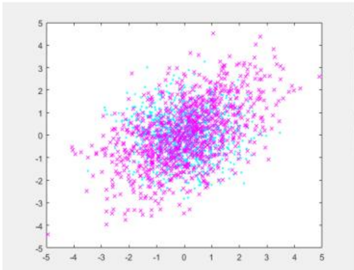

Figure 1.4: result of labtwo.m

This is using a linear transformation to generate the multivariate normal distribution. The n-dimensional random variable is normal distribution N(u,B), suppose m dimensional random variable Y is the linear transformation of X, it means Y=XC, C is a n*m matrix, therefore Y is m dimensional normal distribution N(Uc,C'BC).

In this case, using Cholesky decompose to a covariance matrix and multiply a normal distributed matrix can obtain a linear correlated or negative correlated distribution which like figure 1.4.

```
C=[2 1;1 2];X=randn(1000,2);A=chol(C);Y=X*A;% correlated gaussian distribution
N = 10;% test 10,100,1000,10000
plotArray = zeros(N,1);thRange = linspace(0,2*pi,N);% setting coordinate system
for n=1:N
  theta = thRange(n);
  u = [sin(theta); cos(theta)]
  yp = Y*u;
  var_empirical = var(yp)% matlab command
  var_theoretical = u'*C*u %calculated
  plotArray(n) =  var_empirical - var_theoretical;%difference
end
plot(plotArray)
```

The curve will smooth with the increasing N from 10,100,1000,10000.

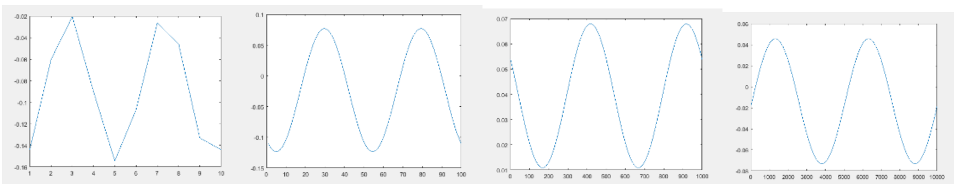

Figure 1.5: result of labtwo2_2.m

$$A - \lambda E = \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = (2-\lambda)^2 - 1 = 3 - 4\lambda + \lambda^2 = (\lambda-1)(\lambda-3)$$

Therefore, the eigenvalue of A is $\lambda_1 = 1$ and $\lambda_1 = 3$

When $\lambda_1 = 1$, $\begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix} * \begin{matrix} x_{11} \\ x_{12} \end{matrix} = \begin{matrix} 0 \\ 0 \end{matrix}$, so $x_{11} = -x_{12}$ and the eigenvector is $\begin{matrix} 1 \\ -1 \end{matrix}$

When $\lambda_1 = 3$, $\begin{vmatrix} -1 & 1 \\ 1 & -1 \end{vmatrix} * \begin{matrix} x_{11} \\ x_{12} \end{matrix} = \begin{matrix} 0 \\ 0 \end{matrix}$, so $x_{11} = x_{12}$ and the eigenvector is $\begin{matrix} 1 \\ 1 \end{matrix}$