實驗4

實驗 4.1

實驗代碼重要部分解釋：

```
sampleSize = size( Data.samples, DOWN );
maxNorm    = realmin;
for iObservation = 1:sampleSize
    observationNorm = norm( Data.samples(iObservation,:) );
    if observationNorm > maxNorm
        maxNorm = observationNorm;
    end
end
enclosingBallRadius        = maxNorm;
enclosingBallRadiusSquared = enclosingBallRadius .^ 2;
```

- 這部分是爲了求出離原點最遠的點，用來給出bias的修正值，以防梯度下降法無法收斂

```
maxNumSteps = 1000;

for iStep = 1:maxNumSteps

    isAnyObsMisclassified = false;

    for iObservation = 1:sampleSize;

        inputObservation = Data.samples( iObservation, : );
        desiredLabel     = Data.labels(  iObservation    ); % +1 or -1

        perceptronOutput = sum( Model.weights .* inputObservation, ACROSS ) + Model.bi
        margin           = desiredLabel * perceptronOutput;

        isCorrectLabel   = margin > 0;

        % -------------------------------------------------------------
        % If the model misclassifies the observation, update the
        % weights and the bias.
        %

        if ~isCorrectLabel

            isAnyObsMisclassified = true;

            weightCorrection = desiredLabel  * inputObservation;
            Model.weights    = Model.weights + weightCorrection;

            biasCorrection   = desiredLabel .* enclosingBallRadiusSquared;
            Model.bias       = Model.bias   + biasCorrection;

            displayPerceptronState( Data, Model );

        end % if this observation misclassified.

    end % loop over observations

    if ~isAnyObsMisclassified
        disp( 'Done!' );
        break;
    end

end % outer loop
```

- 以上代碼爲感知器算法的核心
- 其中`perceptronOutput = sum( Model.weights .* inputObservation, ACROSS ) + Model.bias;`計算了`g(x) = w'*x+w0`
- 其中`desiredLabel * inputObservation;`是`g(x)`中`w`矩陣的修正值
- 其中`desiredLabel .* enclosingBallRadiusSquared;`是`g(x)`中`w0`的修正值
- 多次迭代直至最後沒有分錯的點爲止（或者到1000次）

實驗 4.2

LS.m代碼分析：

```
m(:,1)=[0 0 0 0 0]';
m(:,2)=[1 1 1 1 1]';
S=[.9 .3 .2 .05 .02;
     .3 .8 .1 .2 .05;
     .2 .1 .7 .015 .07;
     .05 .2 .015 .8 .01;
     .02 .05 .07 .01 .75];
P=[1/2 1/2];
```

- 此段爲設置題目要求的Gauss Distribution的mean和covariance

```
% Generate X1 and the required class labels
N1=200;
randn('seed',0)
X1=[mvnrnd(m(:,1),S,fix(N1/2)); mvnrnd(m(:,2),S,N1-fix(N1/2))]';
z1=[ones(1,fix(N1/2)) 2*ones(1,N1-fix(N1/2))];

% Generate X2 and the required class labels
N2=200;
randn('seed',100)
X2=[mvnrnd(m(:,1),S,fix(N2/2)); mvnrnd(m(:,2),S,N2-fix(N2/2))]';
z2=[ones(1,fix(N2/2)) 2*ones(1,N2-fix(N2/2))];
```

- 這一段爲生成X1,X2
- 爲了數據的可再現，代碼中還使用了固定的隨機數種子

```
% Compute the Bayesian classification error based on X2
S_true(:,:,1)=S;
S_true(:,:,2)=S;
[z]=bayes_classifier(m,S_true,P,X2);
err_Bayes_true=sum(z~=z2)/sum(N2)
```

- 這一段使用了bayes算法進行分類，並計算了它的正確率

```
% 2. Augment the data vectors of X1
X1=[X1; ones(1,sum(N1))];
y1=2*z1-3;

% Augment the data vectors of X2
X2=[X2; ones(1,sum(N2))];
y2=2*z2-3;
```

- 這一段將 `X1`, `X2` 增廣化

```
% Compute the classification error of the LS classifier based on X2
[w]=SSErr(X1,y1,0);
SSE_out=2*(w'*X2>0)-1;
err_SSE=sum(SSE_out.*y2<0)/sum(N2)
```

- 這一段使用X1對SSE算法進行訓練，使用它對X2進行分類，並計算了正確率

SSErr.m代碼分析：

```
[l,N]=size(X);
w=inv(X*X'+C*eye(l))*(X*y');
```

- SSE本質上就只做了`a = inv(Y'*Y)*Y'*b`這個公式的事情，求得的解，即爲分類

實驗 4.3

代碼分析：

```
% This example deals with 2 classes
c1=[1 2;2 3;3 3;4 5;5 5]  % the first class 5 observations
c2=[1 0;2 1;3 1;3 2;5 3;6 5] % the second class 6 observations
scatter(c1(:,1),c1(:,2),6,'r'),hold on;
scatter(c2(:,1),c2(:,2),6,'b');

% Number of observations of each class
n1=size(c1,1)
n2=size(c2,1)

%Mean of each class
mu1=mean(c1)
mu2=mean(c2)
```

- 生成數據，計算各自的mean

```
% Center the data (data-mean)
d1=c1-repmat(mu1,size(c1,1),1)
d2=c2-repmat(mu2,size(c2,1),1)

% Calculate the within class variance (SW)
s1=d1'*d1
s2=d2'*d2
sw=s1+s2
invsw=inv(sw)

% in case of two classes only use v
 v=invsw*(mu1-mu2)'
```

- d1的計算即，`x(j)-m1`的公式，d2同理
- s1就是原公式中的前半部分，`sum((x(j)-m1)(x(j)-m1)')`，s2也同理
- sw爲類內散度矩陣
- invsw爲sw的逆矩陣
- 由`w`正比於`inv(sw) * (m2-m1)`得到代碼中的`v`

```
% project the data of the first and second class respectively
y2=c2*v
y1=c1*v
```

- 完成對`c1`, `c2`的投影