**IE 6318 Data Mining and Analytics**

**Homework 4**

Question 1:

|  |
| --- |
| function Lpred = myBayesPredict(Dtrain, Ltrain, Dtest, opt)  %% 1. Use Naive Bayes Function to Make classification  % Assume the features are independent, then we can use Naive Bayes for prediction  if opt==1  NB = fitcnb(Dtrain,Ltrain); % construct a Naive Bayes model NB  Lpred = predict(NB, Dtest); % apply the trained model NB to predict class of test samples in Dtest  end    %% 2. Use the discriminant function G(x) = likelihood\*prior for classification  % In a general case with correlated features, we can assume the features  % follows multivariate normal distribution, then we can use function "mvnpdf"  % to calculate the likelihood P(X|Wj) directly  % Decision Rule: select the class that maximizes P(X|Wj)P(Wj) - likelihood\*prior    if opt==2  C = unique(Ltrain);  Lpred = [];    for iC = 1:length(C) % For each class i, calculate P(X|Wj)P(Wj) for all testing samples  cl = C(iC);  idx = find(Ltrain==cl);  data = Dtrain(idx,:);  mu = mean(data); % feature mean vector  sigma = cov(data); % feature covariance matrix  P = length(idx)/length(Ltrain);    % For each testing sample, calculate P(X|Wj)P(Wj) = likelihood of class i \* prior of class i  for j = 1:size(Dtest,1)  x = Dtest(j, :);  likelihood = mvnpdf(x,mu,sigma); % likelihood of the current class i  prior = P; % prior of the current class i    % Record values of the discriminat function G(X)  % In the following matrix G, each row represent a class, and  % each column represent a testing sample  G(iC, j) = likelihood\*prior;  end  end    % For each testing sample, find the index of the class that have maximum  % value of likelihood\*prior  [~, pred] = max(G);  Lpred = C(pred);  end  %% 3. Use the derived discriminant function G(x) for classification  % based on the the assumption of Multivariate Normal Distribution for features  if opt==3  C = unique(Ltrain);  Lpred = [];    for iC = 1:length(C)  cl = C(iC);  idx = find(Ltrain==cl);  data = Dtrain(idx,:);    mu = mean(data)';  sigma = cov(data);  P = length(idx)/length(Ltrain);  W = -0.5\*inv(sigma);  w = inv(sigma)\*mu;  w0 = (-0.5\*mu'\*inv(sigma)\*mu)-(0.5\*log(det(sigma))+log(P));    for j = 1:size(Dtest,1)  x = Dtest(j, :)';  % The closed form of the derived discriminant function G(X)  G(iC, j) = x'\*W\*x + w'\*x + w0;  end  end    [~, pred] = max(G);  Lpred = C(pred);    end |

Question 2:

|  |
| --- |
|  |