EECE 5644: Machine Learning / Pattern Recognition

Assignment 1

Jiayun Xin

NUID: 001563582

College of Engineering

Northeastern University Boston, Massachusetts

Spring, 2022

# Q1

Code for this question can be found in Appendix A.

## Part A

This problem produced a 10000 sample, 4-dimensional real valued random vector X with the following characteristics: p(x)=p(x|L=0)P(L=0)+p(x|L=1)p(L=1) where L is the true class label that indicates which class-label-conditioned pdf generates the data. The class conditional PDFs are p(x|L=0)=g(x|mo,Co) and p(x|L=1)=g(x|m1,C1) where m is the mean vector and C is the covariance matrix.

The particular parameters and class priors are shown below. A plot of the vector X generated using these parameters is shown in Figure 1.

A picture containing table

Description automatically generated



1.

A picture containing text, document, screenshot, receipt

Description automatically generated

2.

Multiple gamma values were used and ROC is shown below.

The estimated and theoretical minimum error point are marked on the plot.

Chart

Description automatically generated

Figure 1: ROC Curve for ERM Classification

3.

|  |  |  |
| --- | --- | --- |
|  | Gamma | Min.Perror |
| Theoretical | 2.33 | 2.82% |
| Emprical | 1.68 | 2.70% |

Table 1

The table shows that empirically selected gamma value has a lower minimum P(error) than theoretically optimal threshold.

Chart, line chart

Description automatically generated

Figure 2

Figure 3 displays the values of error proportion as gamma grows and the point with the lowest error proportion is marked. When the gamma is set as the minimum value, input can be classified with the best result.

## Part B

1.

The values remain the same with part A.

A picture containing text, document, screenshot, receipt

Description automatically generated

2.

Like part A, multiple gamma values were used and ROC is shown below.

The theoretical minimum error point is marked on the plot.

Chart

Description automatically generated

Figure 3

3.

The table 2 shows the comparison of threshold between the theoretical values and naïve Bayesian. As we can see, they produced similar gamma value, but the minimum Perror of naïve Bayesian is a little higher than the theoretical because of the lack of true statistics of data being classified.

Comparing the figure 2 and 4, they have almost the same shape.

|  |  |  |
| --- | --- | --- |
|  | Gamma | Min.Perror |
| Theoretical | 2.33 | 2.82% |
| Emprical | 1.68 | 2.70% |
| Naïve Bayesian | 1.78 | 4.48% |

Table 2

Chart, line chart

Description automatically generated

Figure 4

## Part C

1.

Fisher LDA classification rule:

Logo

Description automatically generated

WLDA: generalized eigendecom-position of within and between class scatter matrices

2.

The projection figure is shown below and the tau for minimum error is mark in figure.

The ROC curve is shown in figure 6.

The theoretical minimum error point is marked on the plot.

Chart, line chart

Description automatically generated

Figure 5

Chart

Description automatically generated

Figure 6

3.

The table 3 lists the minimum Perror results of all three methods. As we can see, Fisher LDA has a lower probability of error than Naïve Bayesian method and a little higher error probability than ERM method.

|  |  |
| --- | --- |
|  | Min.Perror |
| Theoretical | 2.82% |
| Emprical | 2.70% |
| Naïve Bayesian | 4.48% |
| Fisher LDA | 3.63% |

Table 3

The shape of figure 7 has almost the same shape with figure 2 and 4.

Chart, line chart

Description automatically generated

Figure 7

# Q2

Code for this question can be found in Appendix B.

## Part A

1.

mean vector m: To ensure overlap between class-conditional pdfs, the mean vectors were set by twice average standard deviation of all Gaussians.

covariance matrix c: it was set to be diagonal.

A picture containing scatter chart

Description automatically generated

The figure 8 shows the true data distribution of the three classes.

Chart, scatter chart

Description automatically generated

Figure 8

2.

The following decision rule was used to classify the data samples:

Text, schematic

Description automatically generated with medium confidence

Where C is the number of classes, lambda is the loss and p mean a pdf. 0-1 loss is selected as follow:

A picture containing text, clock

Description automatically generated

The confusion table is as below:

Table

Description automatically generated

Table 4

3.

The visualization of the data is shown as figure 9. As we can see, different marker shape marks different class. The green marker relates to correct classification and red marker is wrong classification.

Chart, scatter chart

Description automatically generated

Figure 9

## Part B

After repeating the ERM classification rule and resetting loss matrices of 10 and 100 times about not making mistakes when L = 3, the table 5, table 6, figure 10 and figure 11 show the results respectively. We can see that there are more misclassified points as loss matrices of ‘L = 3’ cares more and more. The overall probability of error increases from 27% to 43% to 54 for 1, 10 and 100 times sensitivity.

Depends on the table 5 and 6, we can see that more and more data was classified to class 3 and class 3 had a good classified-accuracy. But it causes more class 1 and class 2 data misclassified because the relevant confusion value become lower.

Table

Description automatically generated

Table 5: confusion matrix when lambda is 10

Table

Description automatically generated

Table 6: confusion matrix when lambda is 100

Chart, scatter chart

Description automatically generated

Figure 10

Chart, scatter chart

Description automatically generated

Figure 11

# Q3

Code for this question can be found in Appendix C.

I split the whole question into two 2 parts and talk about wine quality dataset and human activity recognition dataset separately.

A regularization term is used to covariance to ensure that the regularized covariance matrix Cregularized has all eigenvalues larger than the parameter. The regularization term formula is:



Where lambda is set as 0.1

## Part A

The probability of error is 54.49%. The count error is 2669. The confusion matrix is shown as table 7 in which the first column represents the decision class label, and the first row relates to the true class label. From the confusion matrix, there are many 0 values in the table. Class 0, 1, 2 and 10 do not show up. Class 4, 8 and 9 have mall class priors, so they are not considered. Class 7 with 0.1797 class prior has a large overlap with other classes. Class 5 and 6 are the two classes with highest class prior, so they are chosen to decide class label.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | True Class Label | | | | | | | | | | | |
| Decision |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0.2500 | 0.0123 | 0.0034 | 0.0036 | 0.0023 | 0.0229 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0.2000 | 0.1963 | 0.1935 | 0.1128 | 0.0466 | 0.0343 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0.5500 | 0.7914 | 0.8030 | 0.8835 | 0.9511 | 0.9429 | 1 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 7: confusion matrix for white wine quality dataset

The Principal Component Analysis was used, and the result is shown as figure 12. As we can see, the shape of it like a triangle but Gaussian distribution. Different classes are overlapped because the mean (7.65) of between two classes is smaller than the average standard deviation (10.34).

Chart, scatter chart

Description automatically generated

Figure 12

## Part B

The probability of error for this dataset is 1.73%. The count error is 178. The confusion matrices result is shown as table 8. As we can see, every value belongs to class 2 and 6 was classified correctly. All 6 classes have a good class distribution with little overlaps.

Table

Description automatically generated

Table 8: confusion matrix for human activity recognition dataset

The figure 13 shows the PCA results on human activity dataset. Compared with the PCA results of wine quality, human activity dataset has a clearer distribution. The average distance between means is 1.20 which is much higher than the average standard deviation 0.2. It explains the little overlap between each classes.

Chart, scatter chart

Description automatically generated

Figure 13

# Appendix A

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%EECE5644 Spring 2022

%Homework #1

%Problem #1

%Significant parts of this code were derived from the following sources

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; close all;

%Initialize Parameters and Generate Data

N = 10000; %Number of data points

n = 4; %Dimensions of data

p0 = 0.7; %Prior for label 0

p1 = 0.3; %Prior for label 1

u = rand(1,N)>=p0; %Determine posteriors

%Create appropriate number of data points from each distribution

N0 = length(find(u==0));

N1 = length(find(u==1));

N = N0 + N1;

label=[zeros(1,N0) ones(1,N1)];

%Parameters for two classes

%mu: mean vector

%singma: covariance matrix

mu0 = [-1;-1;-1;-1];

Sigma0 = [2,-0.5,0.3,0;

-0.5,1,-0.5,0;

0.3,-0.5,1,0;

0,0,0,2];

mu1 = [1;1;1;1];

Sigma1 = [1,0.3,-0.2,0;

0.3,2,0.3,0;

-0.2,0.3,1,0;

0,0,0,3];

%Generate data as prescribed in assignment description

r0 = mvnrnd(mu0, Sigma0, N0);

r1 = mvnrnd(mu1, Sigma1, N1);

%Combine data from each distribution into a single dataset

x=zeros(N,n);

x(label==0,:)=r0;

x(label==1,:)=r1;

%Part A: ERM Classification with True Knowledge

discScore=log(evalGaussian(x',mu1,Sigma1)./evalGaussian(x',mu0,Sigma0));

sortDS=sort(discScore);

%Generate vector of gammas for parametric sweep

logGamma=[min(discScore)-eps sort(discScore)+eps];

for ind=1:length(logGamma)

decision=discScore>logGamma(ind);

Num\_pos(ind)=sum(decision);

pFP(ind)=sum(decision==1 & label==0)/N0;

pTP(ind)=sum(decision==1 & label==1)/N1;

pFN(ind)=sum(decision==0 & label==1)/N1;

pTN(ind)=sum(decision==0 & label==0)/N0;

%Two ways to make sure I did it right

pFE(ind)=(sum(decision==0 & label==1) + sum(decision==1 & label==0))/N;

pFE2(ind)=(pFP(ind)\*N0 + pFN(ind)\*N1)/N;

end

%Calculate Theoretical Minimum Error

logGamma\_ideal=log(p0/p1);

decision\_ideal=discScore>logGamma\_ideal;

pFP\_ideal=sum(decision\_ideal==1 & label==0)/N0;

pTP\_ideal=sum(decision\_ideal==1 & label==1)/N1;

pFE\_ideal=(pFP\_ideal\*N0+(1-pTP\_ideal)\*N1)/(N0+N1);

%Estimate Minimum Error

%If multiple minimums are found choose the one closest to the theoretical %minimum

[min\_pFE, min\_pFE\_ind]=min(pFE);

if length(min\_pFE\_ind)>1

[~,minDistTheory\_ind]=min(abs(logGamma(min\_pFE\_ind)-logGamma\_ideal));

min\_pFE\_ind=min\_pFE\_ind(minDistTheory\_ind);

end

%Find minimum gamma and corresponding false and true positive rates

minGAMMA=exp(logGamma(min\_pFE\_ind));

min\_FP=pFP(min\_pFE\_ind);

min\_TP=pTP(min\_pFE\_ind);

%Plot

figure;

plot(pFP,pTP,'DisplayName','ROC Curve','LineWidth',2);

hold all;

plot(min\_FP,min\_TP,'o','DisplayName','Estimated Min. Error','LineWidth',2);

plot(pFP\_ideal,pTP\_ideal,'+','DisplayName',...

'Theoretical Min. Error','LineWidth',2);

xlabel('Prob. False Positive');

ylabel('Prob. True Positive');

title('Mininimum Expected Risk ROC Curve');

legend 'show';

grid on; box on;

fprintf('Theoretical: Gamma=%1.2f, Error=%1.2f%%\n',...

exp(logGamma\_ideal),100\*pFE\_ideal);

fprintf('Estimated: Gamma=%1.2f, Error=%1.2f%%\n',minGAMMA,100\*min\_pFE);

figure;

plot(logGamma,pFE,'DisplayName','Errors','LineWidth',2);

hold on;

plot(logGamma(min\_pFE\_ind),pFE(min\_pFE\_ind),...

'ro','DisplayName','Minimum Error','LineWidth',2);

xlabel('Gamma');

ylabel('Proportion of Errors');

title('Probability of Error vs. Gamma');

grid on;

legend 'show';

%Part 2: Naive Bayesian Classifier

Sigma\_NB=eye(4); %Assumed covariance

%Generate data to illustrate assumptions

r0\_NB = mvnrnd(mu0, Sigma\_NB, N0);

r1\_NB = mvnrnd(mu1, Sigma\_NB, N1);

%Evaluate for different gammas

discScore\_NB=...

log(evalGaussian(x' ,mu1,Sigma\_NB)./evalGaussian(x' ,mu0,Sigma\_NB));

logGamma\_NB=[min(discScore\_NB)-0.1 sort(discScore\_NB)+0.1];

for ind=1:length(logGamma\_NB)

decision=discScore\_NB>logGamma\_NB(ind);

Num\_pos\_NB(ind)=sum(decision);

pFP\_NB(ind)=sum(decision==1 & label==0)/N0;

pTP\_NB(ind)=sum(decision==1 & label==1)/N1;

pFN\_NB(ind)=sum(decision==0 & label==1)/N1;

pTN\_NB(ind)=sum(decision==0 & label==0)/N0;

pFE\_NB(ind)=(sum(decision==0 & label==1) + sum(decision==1 & label==0))/(N0+N1);

pFE2\_NB(ind)=pFP(ind)\*p0+pFN(ind)\*p1;

end

%Estimated Minimum Error

[min\_pFE\_NB, min\_pFE\_ind\_NB]=min(pFE\_NB);

minGAMMA\_NB=exp(logGamma(min\_pFE\_ind\_NB));

min\_FP\_NB=pFP\_NB(min\_pFE\_ind\_NB);

min\_TP\_NB=pTP\_NB(min\_pFE\_ind\_NB);

%Plot Results

figure;

plot(pFP\_NB,pTP\_NB,'DisplayName','ROC Curve','LineWidth',2); hold all;

plot(min\_FP\_NB,min\_TP\_NB,'o','DisplayName',...

'Estimated Min. Error','LineWidth',2);

xlabel('Prob. False Positive');

ylabel('Prob. True Positive');

title('Mininimum Expected Risk ROC Curve (Naive Bayesian)');

legend 'show';

grid on; box on;

figure;

plot(logGamma\_NB,pFE\_NB,'DisplayName','Errors','LineWidth',2);

hold on;

plot(logGamma\_NB(min\_pFE\_ind\_NB),pFE\_NB(min\_pFE\_ind\_NB),'ro',...

'DisplayName','Minimum Error','LineWidth',2);

xlabel('Gamma');

ylabel('Proportion of Errors');

title('Probability of Error vs. Gamma (Naive Bayesian Estimate)')

grid on;

legend 'show';

fprintf('Estimated for NB: Gamma=%1.2f, Error=%1.2f%%\n', minGAMMA\_NB,100\*min\_pFE\_NB);

%Part 3: Fisher LDA

%Compute Sample Mean and covariances

mu0\_hat=mean(r0)';

mu1\_hat=mean(r1)';

Sigma0\_hat=cov(r0);

Sigma1\_hat=cov(r1);

%Compute scatter matrices

Sb=(mu0\_hat-mu1\_hat)\*(mu0\_hat-mu1\_hat)';

Sw=Sigma0\_hat+Sigma1\_hat;

%Eigen decompostion to generate WLDA

[V,D]=eig(inv(Sw)\*Sb);

[~,ind]=max(diag(D));

w=V(:,ind);

y=w'\*x';

w=sign(mean(y(find(label==1))-mean(y(find(label==0)))))\*w;

y=sign(mean(y(find(label==1))-mean(y(find(label==0)))))\*y;

%Evaluate for different taus

tau=[min(y)-0.1 sort(y)+0.1];

for ind=1:length(tau)

decision=y>tau(ind);

Num\_pos\_LDA(ind)=sum(decision);

pFP\_LDA(ind)=sum(decision==1 & label==0)/N0;

pTP\_LDA(ind)=sum(decision==1 & label==1)/N1;

pFN\_LDA(ind)=sum(decision==0 & label==1)/N1;

pTN\_LDA(ind)=sum(decision==0 & label==0)/N0;

pFE\_LDA(ind)=(sum(decision==0 & label==1) + sum(decision==1 & label==0))/(N0+N1);

end

%Estimated Minimum Error

[min\_pFE\_LDA, min\_pFE\_ind\_LDA]=min(pFE\_LDA);

minTAU\_LDA=tau(min\_pFE\_ind\_LDA);

min\_FP\_LDA=pFP\_LDA(min\_pFE\_ind\_LDA);

min\_TP\_LDA=pTP\_LDA(min\_pFE\_ind\_LDA);

%Plot results

figure;

plot(y(label==0),zeros(1,N0),'o','DisplayName','Label 0');

hold all;

plot(y(label==1),ones(1,N1),'o','DisplayName','Label 1');

ylim([-1 2]);

plot(repmat(tau(min\_pFE\_ind\_LDA),1,2),ylim,'--',...

'DisplayName','Tau for Min. Error','LineWidth',2); grid on;

xlabel('y');

title('Fisher LDA Projection');

legend 'show';

figure;

plot(pFP\_LDA,pTP\_LDA,'DisplayName','ROC Curve','LineWidth',2);

hold all;

plot(min\_FP\_LDA,min\_TP\_LDA,'o','DisplayName',...

'Estimated Min. Error','LineWidth',2);

xlabel('Prob. False Positive');

ylabel('Prob. True Positive');

title('Mininimum Expected Risk ROC Curve (Fisher LDA)');

legend 'show';

grid on;

box on;

figure;

plot(tau,pFE\_LDA,'DisplayName','Errors','LineWidth',2);

hold on;

plot(tau(min\_pFE\_ind\_LDA),pFE\_LDA(min\_pFE\_ind\_LDA),'ro',...

'DisplayName','Minimum Error','LineWidth',2);

xlabel('Tau');

ylabel('Proportion of Errors');

title('Probability of Error vs. Tau (Fisher LDA)')

grid on;

legend 'show';

fprintf('Estimated for LDA: Tau=%1.2f, Error=%1.2f%%\n', minTAU\_LDA,100\*min\_pFE\_LDA);

function g = evalGaussian(x ,mu,Sigma)

%Evaluates the Gaussian pdf N(mu, Sigma ) at each column of X

[n,N] = size(x);

C = ((2\*pi)^n \* det(Sigma))^(-1/2); %coefficient

E = -0.5\*sum((x-repmat(mu,1,N)).\*(inv(Sigma)\*(x-repmat(mu,1,N))),1);%exponent

g = C\*exp(E); %finalgaussianevaluation

end

# Appendix B

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%EECE5644 Sprint 2022

%Homework #1

%Problem #2

%Significant parts of this code were derived from the following sources

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

close all;

N = 10000; %number of samples

n = 3; %number of dimensions

C = 3; %number of classes

p = [0.3, 0.3, 0.4]; %class priors

%class conditional distributions

sigma(:,:,1) = [rand 0 0

0 rand 0

0 0 rand];

sigma(:,:,2) = [rand 0 0

0 rand 0

0 0 rand];

sigma(:,:,3) = [rand 0 0

0 rand 0

0 0 rand];

sigma(:,:,4) = [rand 0 0

0 rand 0

0 0 rand];

averageStdDev = trace(sum(sqrt(sigma),3))/16; %offset means by 2 std devs

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

mu(:,2) = [2\*averageStdDev; 0; 0];

mu(:,3) = [averageStdDev; averageStdDev\*sqrt(3); 0];

mu(:,4) = [averageStdDev; averageStdDev\*(sqrt(3)/3); averageStdDev\*sqrt(8/3)];

% Data generation and labelling

label = rand(1,N);

for i = 1:length(label)

if label(i) < p(1)

label(i) = 1;

elseif label(i) < (p(2)+p(1))

label(i) = 2;

elseif label(i) < ((p(3)/2)+p(2)+p(1)) %two subclasses for the last class, will be combined later

label(i) = 3;

else

label(i) = 4;

end

end

NumClass = [sum(label==1),sum(label==2),sum(label==3),sum(label==4)];

x = zeros(n,N);

x(:, label==1) = mvnrnd(mu(:,1), sigma(:,:,1), NumClass(1))';

x(:, label==2) = mvnrnd(mu(:,2), sigma(:,:,2), NumClass(2))';

x(:, label==3) = mvnrnd(mu(:,3), sigma(:,:,3), NumClass(3))';

x(:, label==4) = mvnrnd(mu(:,4), sigma(:,:,4), NumClass(4))';

% Combine labels 2 and 3 into one class under label 2

for i = 1:length(label)

if label(i) == 4

label(i) = 3;

end

end

NumClass = [sum(label==1),sum(label==2),sum(label==3)];

% plot generated data

figure

scatter3(x(1, label==1),x(2, label==1),x(3, label==1),'bo')

hold on

scatter3(x(1, label==2),x(2, label==2),x(3, label==2),'r\*')

scatter3(x(1, label==3),x(2, label==3),x(3, label==3),'gx')

title('True Data Distributions')

legend('Class 1','Class 2','Class 3')

xlabel('x1')

ylabel('x2')

zlabel('x3')

hold off

%% Part A - MAP Classifier with True Knowledge

% Evaluate class conditional pdfs

pxgivenl(1,:) = mvnpdf(x', mu(:,1)', sigma(:,:,1))';

pxgivenl(2,:) = mvnpdf(x', mu(:,2)', sigma(:,:,2))';

pxgivenl(3,:) = .5\*mvnpdf(x', mu(:,3)', sigma(:,:,3))' + .5\*mvnpdf(x', mu(:,4)',sigma(:,:,4))'; %two distributions for class 3

% Find class posteriors

px = p\*pxgivenl; %total probability

plgivenx = pxgivenl.\*repmat(p',1,N)./repmat(px,C,1); %class posterior functions

% Loss matrix, 0-1 loss provides minimum probability of error

lossMatrix = ones(3,3)-eye(3);

expectedRisks = lossMatrix\*plgivenx;

[~,decisions] = min(expectedRisks,[],1);

% Make confusion matrix and plot data

figure

shapes = ['o','\*','x'];

for i = 1:C %each decision

for j = 1:C %each class label

confusionMatrix(i,j) = sum(decisions==i & label==j)/NumClass(j);

if i == j

scatter(i,j) = scatter3(x(1,decisions==i & label==j), ...

x(2,decisions==i & label==j), ...

x(3,decisions==i & label==j), ...

'g',shapes(j), ...

'DisplayName', ...

['Class ' num2str(j) ' Correct Classification']);

hold on

else

scatter(i,j) = scatter3(x(1,decisions==i & label==j), ...

x(2,decisions==i & label==j), ...

x(3,decisions==i & label==j), ...

'r',shapes(j), ...

'DisplayName', ...

['Class ' num2str(j) ' Incorrect Classification']);

hold on

end

end

end

title('Correct vs. Incorrect Classification')

legend([scatter(1,1) scatter(2,1) scatter(2,2) scatter(3,2) scatter(3,3) scatter(1,3)])

xlabel('x1')

ylabel('x2')

zlabel('x3')

hold off

%% higher loss

% Loss matrix, Lambda\_10

lossMatrix10 = [0 1 10

1 0 10

1 1 0];

expectedRisks10 = lossMatrix10\*plgivenx;

[~,decisions10] = min(expectedRisks10,[],1);

% Make confusion matrix and plot data

figure

for i = 1:C %each decision

for j = 1:C %each class label

confusionMatrix10(i,j) = sum(decisions10==i & label==j)/NumClass(j);

if i == j

scatter(i,j) = scatter3(x(1,decisions10==i & label==j),...

x(2,decisions10==i & label==j), ...

x(3,decisions10==i & label==j), ...

'g',shapes(j),'DisplayName', ...

['Class ' num2str(j) ' Correct Classification']);

hold on

else

scatter(i,j) = scatter3(x(1,decisions10==i & label==j),...

x(2,decisions10==i & label==j), ...

x(3,decisions10==i & label==j), ...

'r',shapes(j),'DisplayName', ...

['Class ' num2str(j) ' Incorrect Classification']);

hold on

end

end

end

title('Correct vs. Incorrect Classification (Lambda=10)')

legend([scatter(1,1) scatter(2,1) scatter(2,2) scatter(3,2) scatter(3,3) scatter(1,3)])

xlabel('x1')

ylabel('x2')

zlabel('x3')

hold off

% Loss matrix, Lambda\_100

lossMatrix100 = [0 1 100

1 0 100

1 1 0];

expectedRisks100 = lossMatrix100\*plgivenx;

[~,decisions100] = min(expectedRisks100,[],1);

% Make confusion matrix and plot data

figure

for i = 1:C %each decision

for j = 1:C %each class label

confusionMatrix100(i,j) = sum(decisions100==i & label==j)/NumClass(j);

if i == j

scatter(i,j) = scatter3(x(1,decisions100==i & label==j),...

x(2,decisions100==i & label==j),...

x(3,decisions100==i & label==j), ...

'g',shapes(j),'DisplayName', ...

['Class ' num2str(j) ' Correct Classification']);

hold on

else

scatter(i,j) = scatter3(x(1,decisions100==i & label==j),...

x(2,decisions100==i & label==j),...

x(3,decisions100==i & label==j), ...

'r',shapes(j),'DisplayName', ...

['Class ' num2str(j) ' Incorrect Classification']);

hold on

end

end

end

title('Correct vs. Incorrect Classification (Lambda=100)')

legend([scatter(1,1) scatter(2,1) scatter(2,2) scatter(3,2) scatter(3,3) scatter(1,3)])

xlabel('x1')

ylabel('x2')

zlabel('x3')

hold off

# Appendix C

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%EECE5644 Spring 2022

%Homework #1

%Problem #3A

%Significant parts of this code were derived from the following sources

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Wine Dataset Class Conditional/Prior Estimation

clear all;

close all;

%%

% import data from excel

x = readmatrix('winequality-white.csv', 'Range', 'A2:K4899')';

label = readmatrix('winequality-white.csv', 'Range', 'L2:L4899')';

%%

N = size(x,2); %number of samples

n = size(x,1); %number of dimensions

C = 11; %number of classes

alpha = 0.1; %for regularization

sigmaTotal = cov(x'); %for regularization

%%

% Estimate class conditionals (gaussian) and class priors

for i = 1:C

% estimate class prior

p(i) = sum(label==i-1)/N;

% estimate mean

mu(:,i) = mean(x(:,label==i-1),2);

% estimate covariance matrix

sigma(:,:,i) = cov(x(:,label==i-1)');

% regularize covariance matrix

sigma(:,:,i) = sigma(:,:,i) + eye(size(sigma,1))\*alpha\*trace(sigmaTotal)/rank(sigmaTotal);

end

%% Wine Dataset ERM

% Evaluate class conditional pdfs

for i = 1:C

if sum(isnan(sigma(:,:,i)))==0

pxgivenl(i,:) = mvnpdf(x', mu(:,i)', sigma(:,:,i))';

else

pxgivenl(i,:) = zeros(1,4898); %zero for classes without data

end

end

% Find class posteriors

px = p\*pxgivenl; %total probability

plgivenx = pxgivenl.\*repmat(p',1,N)./repmat(px,C,1); %class posterior functions

% Loss matrix, 0-1 loss provides minimum probability of error

lossMatrix = ones(C,C)-eye(C);

expectedRisks = lossMatrix\*plgivenx;

[~,decisions] = min(expectedRisks,[],1);

decisions = decisions-1; %because classes start at 0

% Total error probability estimate

countError = sum(label~=decisions);

pE = countError/N;

% Make confusion matrix

for i = 1:C %each decision

for j = 1:C %each class label

if sum(isnan(sigma(:,:,j)))==0

confusionMatrix(i,j) = sum(decisions==i-1 & label==j-1)/sum(label==j-1);

else

confusionMatrix(i,j) = 0;

end

end

end

%% Wine Dataset PCA and Plotting

% Without classes, sample based estimates of distribution (gaussian)

muHat = mean(x,2);

% sigmaTotal is calculated above

% Make data zero-mean

xzm = x - muHat\*ones(1,N);

% Get and sort eignvalues/eigenvectors

[Q,D] = eig(sigmaTotal);

[d,ind] = sort(diag(D),'descend');

Q = Q(:,ind);

D = diag(d);

% Calculate the first two principal components for visualization

y = Q(:,1:2)'\*xzm;

% Percent of variance maintained

percentVar = trace(D(1:2,1:2))/trace(D);

% Plot data after PCA

figure

hold on

plot(y(1,label==3),y(2,label==3),'g\*','DisplayName','Class 3')

plot(y(1,label==4),y(2,label==4),'m\*','DisplayName','Class 4')

plot(y(1,label==5),y(2,label==5),'c\*','DisplayName','Class 5')

plot(y(1,label==6),y(2,label==6),'r\*','DisplayName','Class 6')

plot(y(1,label==7),y(2,label==7),'y\*','DisplayName','Class 7')

plot(y(1,label==8),y(2,label==8),'b\*','DisplayName','Class 8')

plot(y(1,label==9),y(2,label==9),'k\*','DisplayName','Class 9')

title('PCA on Wine Dataset')

xlabel('y1')

ylabel('y2')

legend

hold off

% Average distance between means and average standard deviation

counterDist = 0;

counterSig = 0;

for i = 1:C

if sum(isnan(sigma(:,:,i)))==0

counterSig = counterSig+1;

standardDev(i) = trace(sqrt(sigma(:,:,i)))/size(sigma,1);

end

for j = 1:C

if sum(isnan(sigma(:,:,j)))==0

if i < j

counterDist = counterDist+1;

distances(i) = sqrt(sum((x(:,i)-x(:,j)).^2));

end

end

end

end

averageDistance = sum(distances)/counterDist;

averageStdDev = sum(standardDev)/counterSig;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%EECE5644 Spring 2022

%Homework #1

%Problem #3B

%Significant parts of this code were derived from the following sources

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Human Activity Dataset Class Conditional/Prior Estimation

clear all;

close all;

%%

% import data from excel

x\_test = readmatrix('X\_test.txt', 'Range', 'B2:UP2948')';

x\_train = readmatrix('X\_train.txt', 'Range', 'B2:UP7353')';

label\_test = readmatrix('y\_test', 'Range', 'A2:A2948')';

label\_train = readmatrix('y\_train', 'Range', 'A2:A7353')';

x = horzcat(x\_test,x\_train);

label = horzcat(label\_test,label\_train);

N = size(x,2); %number of samples

n = size(x,1); %number of dimensions

C = 6; %number of classes

%%

alpha = 0.1; %for regularization

sigmaTotal = cov(x'); %for regularization

% Estimate class conditionals (gaussian) and class priors

for i = 1:C

% estimate class prior

p(i) = sum(label==i)/N;

% estimate mean

mu(:,i) = mean(x(:,label==i),2);

% estimate covariance matrix

sigma(:,:,i) = cov(x(:,label==i)');

% regularize covariance matrix

sigma(:,:,i) = sigma(:,:,i) + eye(size(sigma,1))\*alpha\*trace(sigmaTotal)/rank(sigmaTotal);

end

%% Human Activity Dataset ERM

% Evaluate class conditional pdfs

for i = 1:C

if sum(isnan(sigma(:,:,i)))==0

pxgivenl(i,:) = mvnpdf(x', mu(:,i)', sigma(:,:,i))';

else

pxgivenl(i,:) = zeros(1,4898); %zero for classes without data

end

end

% Find class posteriors

px = p\*pxgivenl; %total probability

plgivenx = pxgivenl.\*repmat(p',1,N)./repmat(px,C,1); %class posterior functions

% Loss matrix, 0-1 loss provides minimum probability of error

lossMatrix = ones(C,C)-eye(C);

expectedRisks = lossMatrix\*plgivenx;

[~,decisions] = min(expectedRisks,[],1);

% Total error probability estimate

countError = sum(label~=decisions);

pE = countError/N;

% Make confusion matrix

for i = 1:C %each decision

for j = 1:C %each class label

if sum(isnan(sigma(:,:,j)))==0

confusionMatrix(i,j) = sum(decisions==i & label==j)/sum(label==j);

else

confusionMatrix(i,j) = 0;

end

end

end

%% Human Activity Dataset PCA and Plotting

% Without classes, sample based estimates of distribution (gaussian)

muHat = mean(x,2);

% sigmaTotal is calculated above

% Make data zero-mean

xzm = x - muHat\*ones(1,N);

% Get and sort eignvalues/eigenvectors

[Q,D] = eig(sigmaTotal);

[d,ind] = sort(diag(D),'descend');

Q = Q(:,ind);

D = diag(d);

% Calculate the first three principal components for visualization

y = Q(:,1:3)'\*xzm;

% Percent of variance maintained

percentVar = trace(D(1:3,1:3))/trace(D);

% Plot data after PCA

figure

scatter3(y(1,label==1),y(2,label==1),y(3,label==1),'b\*','DisplayName','Class 1')

hold on

scatter3(y(1,label==2),y(2,label==2),y(3,label==2),'g\*','DisplayName','Class 2')

scatter3(y(1,label==3),y(2,label==3),y(3,label==3),'m\*','DisplayName','Class 3')

scatter3(y(1,label==4),y(2,label==4),y(3,label==4),'c\*','DisplayName','Class 4')

scatter3(y(1,label==5),y(2,label==5),y(3,label==5),'r\*','DisplayName','Class 5')

scatter3(y(1,label==6),y(2,label==6),y(3,label==6),'k\*','DisplayName','Class 6')

title('PCA on Human Activity Dataset')

xlabel('y1')

ylabel('y2')

zlabel('y3')

legend

hold off

% Average distance between means and average standard deviation

counterDist = 0;

counterSig = 0;

for i = 1:C

if sum(isnan(sigma(:,:,i)))==0

counterSig = counterSig+1;

standardDev(i) = trace(sqrt(sigma(:,:,i)))/size(sigma,1);

end

for j = 1:C

if sum(isnan(sigma(:,:,j)))==0

if i < j

counterDist = counterDist+1;

distances(i) = sqrt(sum((x(:,i)-x(:,j)).^2));

end

end

end

end

averageDistance = sum(distances)/counterDist;

averageStdDev = sum(standardDev)/counterSig;