**Course No.: ELEN-857**

**Course Title: Advanced Pattern Recognition Method**

**Department: Electrical and Computer Engineering**

**Project 3: Feature Selection**

**Submitted To: Prepared By:**

**Dr. Robert Y.Li, Professor Name: Mrinmoy Sarkar**

**Department of Electrical Engineering Banner Id: 950-363-260 Telephone: (336) 285-3716; E-mail: eeli@ncat.edu E-mail: msarkar@aggies.ncat.edu**

**Contents:**

1. **Abstract**
2. **Technical Description**
3. **Mathematical Formulation**
4. **Results**
5. **Summary**
6. **Appendix**
7. **Abstract:**

The main purpose of the project is to apply Feature Selection technique to Fisher’s Iris data. Fisher’s Iris data contains a set of measurements related to 3 species of the Iris plant. The three species are Iris Setosa, Iris Versicolor, and Iris Virginica and 4 features names Sepal Length, Sepal Width, Petal Length, Petal Width.

1. **Technical Description:**

Divergence is a measure of separability between two classes and it is defined as the difference between expected values of the likelihood ratio for the two classes under consideration. For better separability between two classes, divergence must be high.

For the computation of divergence, transformed divergence, or the Bhattacharyya distance and the probability of error for two or three feature(s) out of four taken at a time the 150 observations for the three IRIS data classes are used.

1. **Mathematical Formulation:**

For normal distribution, the divergence is estimated as

Where, and are covariance matrices for the two classes and and are the mean vectors.

The average divergence for three classes is:

Transformed divergence is expressed as

Where, D is the divergence.

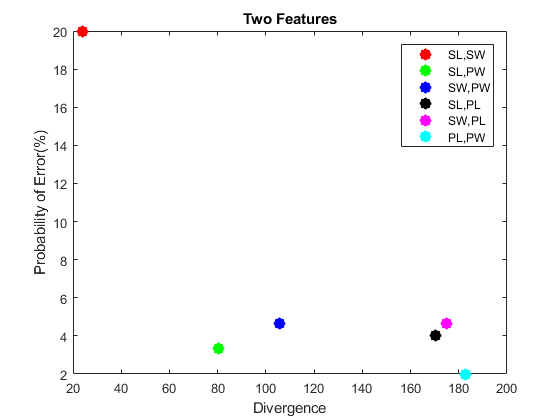
For minimum-distance classifier the decision function is given as below:

For quadratic Bayesian classifier the decision function is given as below:

1. **Results:**

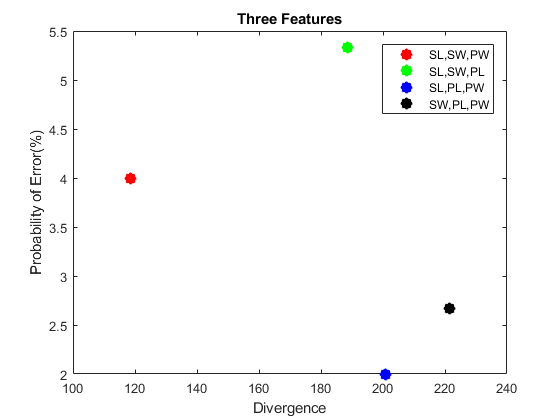
**Two features using divergence measure:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature  Combination | SL,SW | SL,PW | SW,PW | SL,PL | SW,PL | PL,PW |
| Divergence | 23.8078 | 80.3513 | 105.6774 | 170.2532 | 174.9820 | 182.7527 |
| Probability of Error(%) | 20.0000 | 3.3333 | 4.6667 | 4.0000 | 4.6667 | 2.0000 |



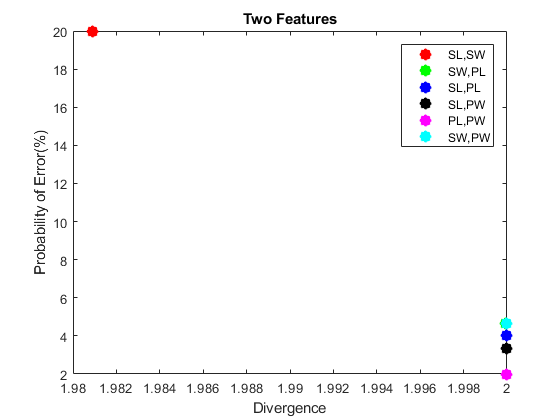
**Three features using divergence measure:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature  Combination | SL,SW,PW | SL,SW,PL | SL,PL,PW | SW,PL,PW |
| Divergence | 118.5123 | 188.6182 | 200.7230 | 221.3827 |
| Probability of Error(%) | 4.0000 | 5.3333 | 2.0000 | 2.6667 |



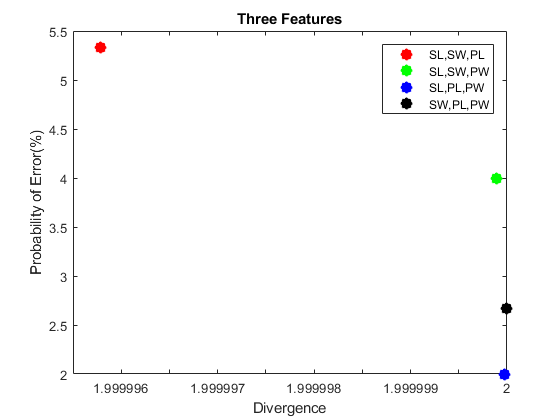
**Two features using transformed divergence measure:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature  Combination | SL,SW | SW,PL | SL,PL | SL,PW | PL,PW | SW,PW |
| Transformed Divergence | 1.9809 | 1.9999 | 2.0000 | 2.0000 | 2.0000 | 2.0000 |
| Probability of Error(%) | 20.0000 | 4.6667 | 4.0000 | 3.3333 | 2.0000 | 4.6667 |



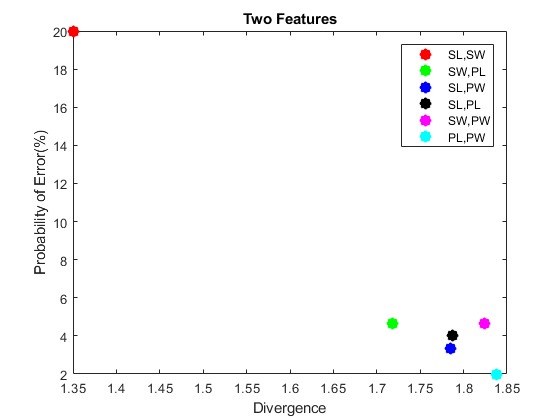
**Three features using transformed divergence measure:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature  Combination | SL,SW,PL | SL,SW,PW | SL,PL,PW | SW,PL,PW |
| Transformed  Divergence | 2.0000 | 2.0000 | 2.0000 | 2.0000 |
| Probability of Error(%) | 5.3333 | 4.0000 | 2.0000 | 2.6667 |



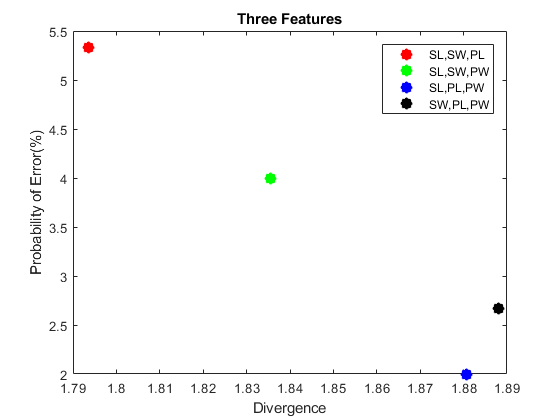
**Two features using Bhattacharyya distance measure:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature  Combination | SL,SW | SW,PL | SL,PW | SL,PL | SW,PW | PL,PW |
| Bhattacharyya  distance | 1.3509 | 1.7183 | 1.7850 | 1.7876 | 1.8243 | 1.8384 |
| Probability of Error(%) | 20.0000 | 4.6667 | 3.3333 | 4.0000 | 4.6667 | 2.0000 |



**Three features using Bhattacharyya distance measure:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature  Combination | SL,SW,PL | SL,SW,PW | SL,PL,PW | SW,PL,PW |
| Bhattacharyya  distance | 1.7937 | 1.8356 | 1.8806 | 1.8881 |
| Probability of Error(%) | 5.3333 | 4.0000 | 2.0000 | 2.6667 |



1. **Summary:**

* Quadratic classifier performs better when using three features than two features.
* The transformed divergence performs so close to the divergence, it slightly has better analysis advantage.
* Bhattacharyya distance gives an analysis close to that of transformed divergence.
* Feature selection using divergence analysis is a useful technique for making better and well-informed feature selection.

1. **Appendix:**

**MATLAB Code:**

|  |
| --- |
| %% file name project4.m  % author: Mrinmoy Sarkar  % email: msarkar@aggies.ncat.edu  % date: 12/1/2017    %%  clear all;  close all;  %% load data to a veriable  data = importdata('iris.txt');    % no. of class is 3 named Iris-setosa, Iris-versicolor and Iris-verginica  % there are 4 attributes named sepal-length, sepal-width, petal-length,  % petal-width  % there are 50 plants for each species    irisSetosad = zeros(50,4);  irisVersicolord = zeros(50,4);  irisVerginicad = zeros(50,4);    n = size(data,1);    indxSeto = 1;  indxVers = 1;  indxVerg = 1;    for i=2:n  x = strsplit(cell2mat(data(i)));  if strcmp(x(5), 'Iris-setosa')  for j=1:4  irisSetosad(indxSeto,j) = str2double(cell2mat(x(j)));  end  indxSeto = indxSeto + 1;  elseif strcmp(x(5), 'Iris-versicolor')  for j=1:4  irisVersicolord(indxVers,j) = str2double(cell2mat(x(j)));  end  indxVers = indxVers + 1;  elseif strcmp(x(5), 'Iris-virginica')  for j=1:4  irisVerginicad(indxVerg,j) = str2double(cell2mat(x(j)));  end  indxVerg = indxVerg + 1;  end  end    featureName = {'SL','SW','PL','PW'};  color = 'rgbkmc';    %% divergence calculation and quadratic Bayesian classifier implementation  for opt = 1:3  option = opt;  for sf=2:3  noOfclass = 3;  classes={};  classes{1} = irisSetosad;  classes{2} = irisVersicolord;  classes{3} = irisVerginicad;    noOfselectedFeature = sf;  featurevector = 1:4;  combOffeature = nchoosek(featurevector,noOfselectedFeature);  combOfclass = nchoosek(1:3,2);  Di = zeros(size(combOffeature,1),1);  Pe = Di;  for ii=1:size(combOffeature,1)  mu = {};  co = {};  for j=1:noOfclass  cl = classes{j};  mu{j} = (mean(cl(:,combOffeature(ii,:))))';  co{j} = (cov(cl(:,combOffeature(ii,:))));  end  D = zeros(noOfclass,1);  a = D;  for j=1:size(combOfclass,1)  D(j) = 0.5\*trace((co{combOfclass(j,1)}-co{combOfclass(j,2)})\*(pinv(co{combOfclass(j,2)})-pinv(co{combOfclass(j,1)})))...  + 0.5\*trace((pinv(co{combOfclass(j,1)})+pinv(co{combOfclass(j,2)}))\*(mu{combOfclass(j,1)}-mu{combOfclass(j,2)})...  \*(mu{combOfclass(j,1)} - mu{combOfclass(j,2)})');    A = 0.5\*(co{combOfclass(j,1)}+co{combOfclass(j,2)});  a(j) = 0.125\*(mu{combOfclass(j,1)} - mu{combOfclass(j,2)})'...  \*pinv(A)\*(mu{combOfclass(j,1)} - mu{combOfclass(j,2)})...  +0.5\*log(det(A)/sqrt(det(co{combOfclass(j,1)})\*det(co{combOfclass(j,2)})));  end  if option == 1 %divergence  Di(ii) = mean(D);  elseif option == 2 %Transformed divergence  Di(ii) = mean(2\*(1-exp(-D)/8));  elseif option == 3 %Bhattacharyya distance  Di(ii) = mean(2\*(1-exp(-a)));  end          % quadratic Bayesian classifier  irisSetosa = irisSetosad(:,combOffeature(ii,:));  irisVersicolor = irisVersicolord(:,combOffeature(ii,:));  irisVerginica = irisVerginicad(:,combOffeature(ii,:));    no\_of\_train\_sample = 50;  no\_of\_test\_sample = 50;  fprintf('quadratic Bayesian classifier for no. of training sample : %d and no. of test sample : %d\n',no\_of\_train\_sample, no\_of\_test\_sample);  % dataset is partisioned as train:test = no\_of\_train\_sample:no\_of\_test\_sample  trainSet = [irisSetosa(1:no\_of\_train\_sample,:); irisVersicolor(1:no\_of\_train\_sample,:); irisVerginica(1:no\_of\_train\_sample,:)];  testSet = [irisSetosa(1:no\_of\_test\_sample,:); irisVersicolor(1:no\_of\_test\_sample,:); irisVerginica(1:no\_of\_test\_sample,:)];    % mean vector calculation  m\_i = [mean(trainSet(1:no\_of\_train\_sample,:));mean(trainSet(no\_of\_train\_sample+1:2\*no\_of\_train\_sample,:));mean(trainSet(2\*no\_of\_train\_sample+1:3\*no\_of\_train\_sample,:))];    % co-varience matrix calculation  cv = [cov(trainSet(1:no\_of\_train\_sample,:));cov(trainSet(no\_of\_train\_sample+1:2\*no\_of\_train\_sample,:));cov(trainSet(2\*no\_of\_train\_sample+1:3\*no\_of\_train\_sample,:))];    confussionMat = zeros(3,3);  % test each sample  for i = 1:no\_of\_test\_sample\*3  testX = testSet(i,:);  ln\_ci = [log(det(cv(1:noOfselectedFeature,:))) log(det(cv(noOfselectedFeature+1:2\*noOfselectedFeature,:))) log(det(cv(2\*noOfselectedFeature+1:3\*noOfselectedFeature,:)))];  ln\_ci = -0.5 \* ln\_ci;  xm\_i = [testX;testX;testX] - m\_i;  xm\_i\_das\_Cinv\_xm\_i = [xm\_i(1,:)\*pinv(cv(1:noOfselectedFeature,:))\*(xm\_i(1,:))' xm\_i(2,:)\*pinv(cv(noOfselectedFeature+1:2\*noOfselectedFeature,:))\*(xm\_i(2,:))' xm\_i(3,:)\*pinv(cv(2\*noOfselectedFeature+1:3\*noOfselectedFeature,:))\*(xm\_i(3,:))'];  xm\_i\_das\_Cinv\_xm\_i = -0.5 \* xm\_i\_das\_Cinv\_xm\_i;  di = ln\_ci + xm\_i\_das\_Cinv\_xm\_i;  [m, m\_index] = max(di);  if i<=no\_of\_test\_sample  confussionMat(m\_index, 1) = confussionMat(m\_index, 1) + 1;  elseif i>no\_of\_test\_sample && i<=2\*no\_of\_test\_sample  confussionMat(m\_index, 2) = confussionMat(m\_index, 2) + 1;  else  confussionMat(m\_index, 3) = confussionMat(m\_index, 3) + 1;  end  end    fprintf('\nConfusion Matrix: |irisSetosa(True) | irisVersicolor(True) | irisVerginica(True)\n');  fprintf('----------------------------------------------------------------------------------------\n');  fprintf('irisSetosa(Predicted) | %2d | %2d | %2d\n', confussionMat(1,1), confussionMat(1,2), confussionMat(1,3));  fprintf('----------------------------------------------------------------------------------------\n');  fprintf('irisVersicolor(Predicted) | %2d | %2d | %2d\n', confussionMat(2,1), confussionMat(2,2), confussionMat(2,3));  fprintf('----------------------------------------------------------------------------------------\n');  fprintf('irisVerginica(Predicted) | %2d | %2d | %2d\n', confussionMat(3,1), confussionMat(3,2), confussionMat(3,3));  fprintf('----------------------------------------------------------------------------------------\n');  fprintf('Correct = %.2f%%\n\n\n',100\*sum(diag(confussionMat))/(3\*no\_of\_test\_sample));  Pe(ii) = 100 - 100\*sum(diag(confussionMat))/(3\*no\_of\_test\_sample);  end    % sort divergence  [Di,indx] = sort(Di);  tpe = Pe;  tcombOffeature = combOffeature;  for jj=1:length(Di)  Pe(jj) = tpe(indx(jj));  combOffeature(jj,:) = tcombOffeature(indx(jj),:);  end  figure  for jj=1:length(Di)  if length(Di)==6  lgnd = strcat(featureName{combOffeature(jj,1)},',',featureName{combOffeature(jj,2)});  else  lgnd = strcat(featureName{combOffeature(jj,1)},',',featureName{combOffeature(jj,2)},',',featureName{combOffeature(jj,3)});  end  disp(lgnd)  plot(Di(jj),Pe(jj),strcat('\*',color(jj)),'LineWidth',7,'DisplayName', lgnd);  legend('-DynamicLegend');  hold on;  end  if length(Di)==6  title('Two Features');  else  title('Three Features');  end  xlabel('Divergence');  ylabel('Probability of Error(%)');  disp('Divergence:')  disp(Di')  disp('Probability of Error(%):')  disp(Pe')  end  end |