Contents

Vincent Purcell - HW 1 - ECE487

```
clear; clc; close all;
```

Problem 2.7

```
% Initializing the variables used for functions
m = [[1;1],[4;4],[8;1]];
S(:,:,1) = 2.*[1,0;0,1];
S(:,:,2) = 2.*[1,0;0,1];
S(:,:,3) = 2.*[1,0;0,1];
P1 = [0.333;0.333;0.334];
P2 = [0.8;0.1;0.1];
N = 1000;

% Generated Gaussian classes
[X,y] = generate_gauss_classes(m,S,P1,N);
[X_prime,y_prime] = generate_gauss_classes(m,S,P2,N);

% Classify the Equiprobable and Unevenly Distributed Datasets using
% Euclidean and Bayesian Classification functions
z_bc = bayes_classifier(m,S,P1,X');
z_ec = euclidean_classifier(m,X');
z_prime_bc = bayes_classifier(m,S,P2,X_prime');
z_prime_ec = euclidean_classifier(m,X_prime');
```

Compute Error for Bayesian and Euclidean Classifications of X

```
z_bc_error = compute_error(z_bc,y);
z_ec_error = compute_error(z_ec,y);
```

Compute Error for Bayesian and Euclidean Classifications of X Prime

```
z_prime_bc_error = compute_error(z_prime_bc,y_prime);
z_prime_ec_error = compute_error(z_prime_ec,y_prime);
```

Conclusion for Problem 2.7

Based on the Errors calculated above and printed out below I could make the following conclusions. For the equiprobable dataset the euclidean and bayesian classifications produce extremely similar results and thus the same error. For the dataset with 0.8,0.1,0.1 class distribution the bayesian classification produces much better results than the euclidean classification.

```
fprintf('Bayesian Classification error for Equiprobable Dataset: %.3f\n', z_bc_error)
fprintf('Euclidean Classification error for Equiprobable Dataset: %.3f\n', z_ec_error)
fprintf('Bayesian Classification error for Unevenly Distributed Dataset: %.3f\n', z_prime_bc_error)
fprintf('Euclidean Classification error for Unevenly Distributed Dataset: %.3f\n', z_prime_ec_error)
```

```
Bayesian Classification error for Equiprobable Dataset: 0.075
Euclidean Classification error for Equiprobable Dataset: 0.075
Bayesian Classification error for Unevenly Distributed Dataset: 0.036
Euclidean Classification error for Unevenly Distributed Dataset: 0.070
```

Problem 2.8

```
% Initializing the variables used for functions
m = [[1;1],[8;6],[13;1]];
S(:,:,1) = 6.*[1,0;0,1];
S(:,:,2) = 6.*[1,0;0,1];
S(:,:,3) = 6.*[1,0;0,1];
P1 = [0.333;0.333;0.334];
N = 1000;

% Generated Gaussian classes
[X,y] = generate_gauss_classes(m,S,P1,N);
[Z,y_z] = generate_gauss_classes(m,S,P1,N);

% Classify class using K Nearest Neighbors with k=1 and calcualte
% classification error, X was used as the testing set and Z was used as the
% training set
z_1 = k_nn_classifier(Z',y_z,1,X');
z_1_error = compute_error(z_1,y);

% Classify class using K Nearest Neighbors with k=11 and calculate
% classification error
z_11 = k_nn_classifier(Z',y_z,11,X');
z_11_error = compute_error(z_11,y);
```

Conclusion for Problem 2.8

Based on the error values found above. One could conclude that when using the K Nearest Neighbors Algorithm if you increase k the classification error would go down. Below the is the classification error of X for k equal to 1 and k equal to 11.

```
fprintf('K Nearest Neighbor Classification error of X with k=1: %.3f\n', z_1_error)
fprintf('K Nearest Neighbor Classification error of X with k=11: %.3f\n', z_11_error)
```

```
K Nearest Neighbor Classification error of X with k=1: 0.107
K Nearest Neighbor Classification error of X with k=11: 0.089
```

Functions Received From Textbook

```
The following functions were received from the Textbook
Pattern Recognition - Theodoridis, Koutroumbas
```

Generate Gaussian Classes Function

Received from page 80 of the Text

```
function [X,y]=generate_gauss_classes(m,S,P,N)
    [l,c]=size(m);
    X=[];
    y=[];
    for j=1:c
    % Generating the [p(j)*N)] vectors from each distribution
        t=mvnrnd(m(:,j),S(:,:,j),fix(P(j)*N));
        % The total number of points may be slightly less than N
        % due to the fix operator
        X=[X;t];
        y=[y ones(1,fix(P(j)*N))*j];
    end
end
```

Bayes Classifier Function

Received from page 81 of the Text

```
function z=bayes_classifier(m,S,P,X)
    [l,c]=size(m); % l=dimensionality, c=no. of classes
    [l,N]=size(X); % N=no. of vectors
    for i=1:N
        for j=1:c
            t(j)=P(j)*comp_gauss_dens_val(m(:,j),...
                S(:,:,j),X(:,i));
        end
        % Determining the maximum quantity Pi*p(x|wi)
        [num,z(i)]=max(t);
    end
end
```

Gaussian Function Evaluation Function

Received from page 79 of the Text

```
function z=comp_gauss_dens_val(m,S,x)
    [l,q]=size(m); % l=dimensionality
    z=(1/((2*pi)^ (l/2)*det(S)^ 0.5) )...
        *exp(-0.5*(x-m)'*inv(S)*(x-m));
end
```

Euclidean Classifier Function

Received from page 82 of the Text

```
function z=euclidean_classifier(m,X)
    [l,c]=size(m); % l=dimensionality, c=no. of classes
    [l,N]=size(X); % N=no. of vectors
    for i=1:N
        for j=1:c
            t(j)=sqrt((X(:,i)-m(:,j))'*(X(:,i)-m(:,j)));
        end
        % Determining the maximum quantity Pi*p(x|wi)
        [num,z(i)]=min(t);
    end
end
```

Compute Classification Error Function

Received from page 83 of the Text

```
function clas_error=compute_error(y,y_est)
    [q,N]=size(y); % N= no. of vectors
    c=max(y); % Determining the number of classes
    clas_error=0; % Counting the misclassified vectors
    for i=1:N
        if(y(i)~=y_est(i))
            clas_error=clas_error+1;
        end
    end
    % Computing the classification error
    clas_error=clas_error/N;
end
```

K Nearest Neighbor Classification Function

Received from page 83 of the Text

```
function z=k_nn_classifier(Z,v,k,X)
    [l,N1]=size(Z);
    [l,N]=size(X);
    c=max(v); % The number of classes
    % Computation of the (squared) Euclidean distance
    % of a point from each reference vector
    for i=1:N
        dist=sum((X(:,i)*ones(1,N1)-Z).^ 2);
        %Sorting the above distances in ascending order
        [sorted,nearest]=sort(dist);
        % Counting the class occurrences among the k-closest
        % reference vectors Z(:,i)
        refe=zeros(1,c); %Counting the reference vectors per class
        for q=1:k
            class=v(nearest(q));
            refe(class)=refe(class)+1;
        end
        [val,z(i)]=max(refe);
    end
end
```