EEEM007: Advanced Signal Processing

Lab Experiments: Pattern Recognition

Author(s):
Kausthub Krishnamurthy

Page Count:
14

URN:
6562233

Due Date:
May 7, 2019

Contents

1 Introduction **1**

2 Experiment 1 **1**
2.1 Results . . . 1
2.2 Discussion . . . 2

3 Experiment : **2**
3.1 Results . . . 2
3.2 Discussion . . . 2

4 Experiment : **2**
4.1 Results . . . 2
4.2 Discussion . . . 2

5 Experiment : **2**
5.1 Results . . . 2
5.2 Discussion . . . 2

6 Experiment : **2**
6.1 Results . . . 2
6.2 Discussion . . . 3

7 Experiment : **3**
7.1 Results . . . 3
7.2 Discussion . . . 3

8 Conclusions **3**

A Appendix A: Source Code **5**
A.1 Experiment 1 . . . 5
A.2 Plot 2 Classes . . . 6
A.3 Experiment 2 . . . 6
A.4 Experiment 3 . . . 8
A.5 Experiment 4 . . . 10
A.6 Experiment 5 . . . 12
A.7 Experiment 6 . . . 12

1 Introduction

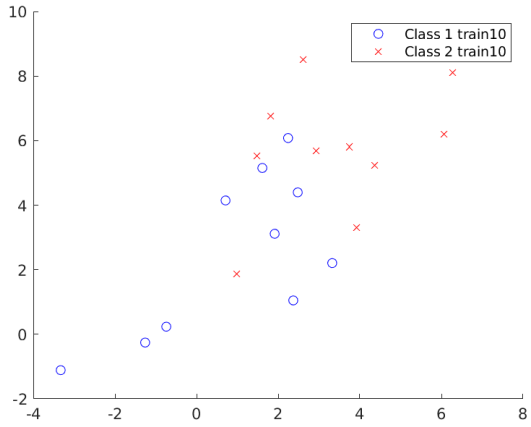
This report includes the documentation of a number of experiments carried out as stipulated by the handout provided. The six experiments attempted will explore the effect of the following variables on a pattern recognition system.

1. The effect of training sample size on classifier performance
2. The effect of test set size on classifier performance
3. The effect of the size of test set on the reliability of the empirical error count estimator
4. To explore the relationship between class separability and error probability
5. The effect, on the classifier error probability, of discrepancies between the true and assumed class probability distribution models.
6. Comparing kNN with a Gaussian classifier.

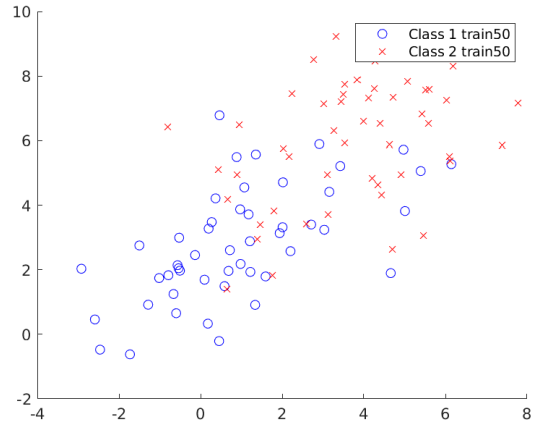
Experiments 1-4 and 6 will each be presented in the following sections, and though some attempt was made at Experiment 5, only a partial solution will be presented in this report.

2 Experiment 1

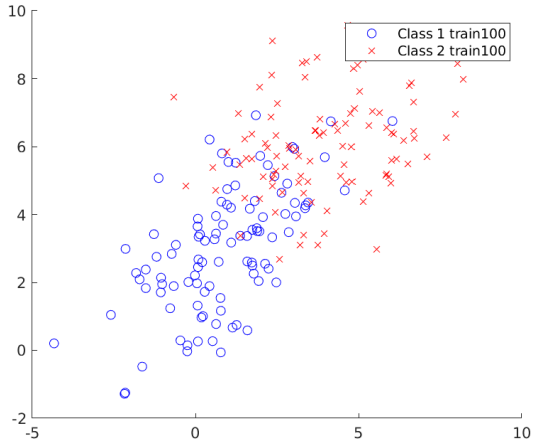
write a description of the experiment
write a prediction about the outcomes based off theory



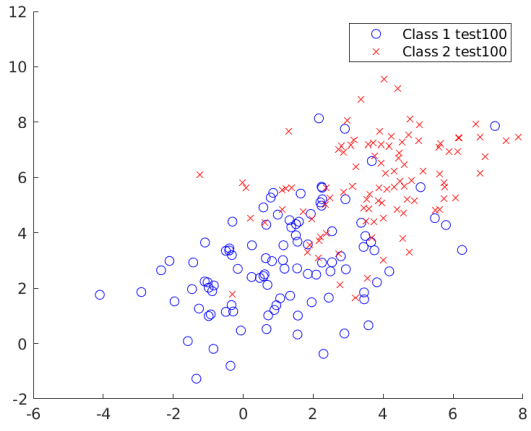
(a) 10 sample training set



(b) 50 sample training set



(c) 100 sample training set



(d) 100 sample test set

Figure 2.1: Visualisation of examples of test and training data generated for Experiment 1.

2.1 Results

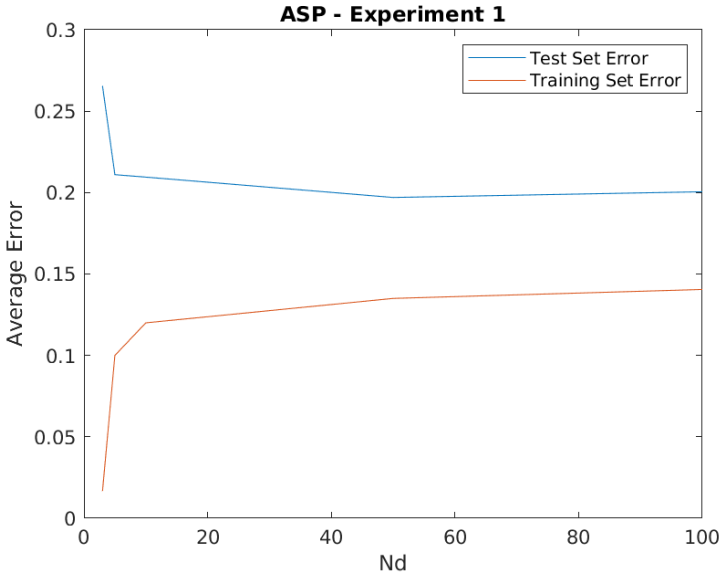


Figure 2.2: Experiment 1 Results

2.2 Discussion

3 Experiment :

write a description of the experiment
write a prediction about the outcomes based off theory

3.1 Results

results
graphs

3.2 Discussion

4 Experiment :

write a description of the experiment
write a prediction about the outcomes based off theory

4.1 Results

results
graphs

4.2 Discussion

5 Experiment :

write a description of the experiment
write a prediction about the outcomes based off theory

5.1 Results

results
graphs

5.2 Discussion

6 Experiment :

write a description of the experiment
write a prediction about the outcomes based off theory

6.1 Results

results
graphs

6.2 Discussion

7 Experiment :

write a description of the experiment
write a prediction about the outcomes based off theory

7.1 Results

results
graphs

7.2 Discussion

8 Conclusions

References

A Appendix A: Source Code

A.1 Experiment 1

Listing 1: Matlab Code developed to simulate experiment 1.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% @author: Kausthub Krishnamurthy
% URN: 6562233
% EEEM007 Advanced Signal Processing - Lab Experiments
% Filename: asp_exp1.m
% Date started: 11-Mar-2019
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Clear everything and setup
clear all
close all
clc

a = 1;
b = 3;
c = 3;
d = 3;
f = 1;

mu1 = [a;b]
mu2 = [a+d; b+d]
cova = [c f;f c] % cova is the covariance matrix. didn't use cov since it's a built
    ↪ in function


%% Effect of training sample size on classifier performance
X1_test = mvnrnd(mu1,cova,100);
X2_test = mvnrnd(mu2,cova,100);
plot2Classes(X1_test, X2_test, 1, "test")

test_labels = [zeros(100,1); ones(100,1)];
sampleSizes = [3 5 10 50 100]

E_train = [];
E_test = [];

for i = 1:length(sampleSizes)
    e_train = [];
    e_test = [];
    for iters = 1:10
        Nd = sampleSizes(i);
```

```
41          X1_design = mvnrnd(mu1,cova,Nd);
42          X2_design = mvnrnd(mu2,cova,Nd);
43          X = [X1_design; X2_design];
44          plot2Classes(X1_design, X2_design, 1, "train")
45          train_labels = [zeros(Nd,1); ones(Nd,1)];
46          mdl = fitcnb(X, train_labels);
47          train_prediction = predict(mdl, X);
48          test_prediction = predict(mdl, [X1_test; X2_test]);
49
50          e_train(iters) = sum(xor(train_prediction, train_labels))/length(
                ↪ train_prediction);
51          e_test(iters) = sum(xor(test_prediction, test_labels))/length(
                ↪ test_prediction);
52      end
53      E_train(i) = sum(e_train)/iters;
54      E_test(i) = sum(e_test)/iters;
55  end
56
57  fig = figure
58  plot(sampleSizes, E_test)
59  hold on
60  plot(sampleSizes, E_train)
61  legend('Test Set Error', 'Training Set Error')
62  title(sprintf("ASP - Experiment 1"))
63  xlabel('Nd')
64  ylabel('Average Error')
65  % saveas(fig,'./Exp1-results/ErrorComparison.png')
```

A.2 Plot 2 Classes

Listing 2: Matlab Code developed to help plot classes in order to visualize the process. Predominantly used in Experiment 1.to simulate experiment 1.

```
1  function plot2Classes(w1,w2, exp, name)
2
3      fig = figure
4      scatter(w1(:,1),w1(:,2), 'bo')
5      hold on
6      scatter(w2(:,1),w2(:,2), 'rx')
7      legend(sprintf('Class 1 ' + name + length(w1)), sprintf('Class 2 '+ name +
           ↪ length(w2)))
8  % saveas(fig,sprintf("./Exp%i-results/",exp) + name + length(w1) +'.png')
9  end
```

A.3 Experiment 2

Listing 3: Matlab Code developed to simulate experiment 2.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% @author: Kausthub Krishnamurthy
% URN: 6562233
% EEEM007 Advanced Signal Processing - Lab Experiments
% Filename: asp_exp2.m
% Date started: 18-Apr-2019
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Clear everything and setup

clear all
close all
clc

a = 1;
b = 3;
c = 3;
d = 3;
f = 1;

%% Effect of test set size on classifier performance

sampleSizes = [3 5 10 20 50 100 200 500]
dimensionSizes = [5, 10, 15]

E_train = [];
E_test = [];

for k = 1:length(dimensionSizes)
    dims = dimensionSizes(k)
    mu1 = randi(5,dims,1)
    if dims == 5
        mu2 = mu1 + 2*rand(dims,1)
    elseif dims == 10
        mu2 = mu1 + 1.6*rand(dims,1)
    else dims == 15
        mu2 = mu1 + 1.35*rand(dims,1)
    end
    cova = eye(dims)
    X1_design = mvnrnd(mu1,cova,100);
    X2_design = mvnrnd(mu2,cova,100);
    for i = 1:length(sampleSizes)
        e_train = [];
        e_test = [];
        for iters = 1:10
            Nd = sampleSizes(i);
```

```
            X1_test = mvnrnd(mu1,cova,Nd);
            X2_test = mvnrnd(mu2,cova,Nd);
            X = [X1_design; X2_design];
            train_labels = [zeros(100,1); ones(100,1)];
            test_labels = [zeros(Nd,1); ones(Nd,1)];
            mdl = fitcnb(X, train_labels);
            train_prediction = predict(mdl, X);
            test_prediction = predict(mdl, [X1_test; X2_test]);

            e_train(iters) = sum(xor(train_prediction, train_labels))/length(
                ↪ train_prediction);
            e_test(iters) = sum(xor(test_prediction, test_labels))/length(
                ↪ test_prediction);
        end
        E_train(i) = sum(e_train)/iters;
        E_test(i) = sum(e_test)/iters;
    end

    fig = figure
    plot(sampleSizes(1:7), E_test(1:7))
    hold on
    plot(sampleSizes(1:7), E_train(1:7))
    hold on
    E_true = [E_test(8), E_test(8), E_test(8), E_test(8), E_test(8), E_test(8),
        ↪ E_test(8)]
    plot(sampleSizes(1:7), E_true)
    legend('Test Set Error', 'Training Set Error', 'True Error')
    title(sprintf("ASP - Experiment 2 -" + " d=" + dims))
    xlabel('Nd')
    ylabel('Average Error')
% saveas(fig,"./Exp2-results/ErrorComparison_" + dims + ".png")
% w = waitforbuttonpress
end
```

A.4 Experiment 3

Listing 4: Matlab Code developed to simulate experiment 3.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% @author: Kausthub Krishnamurthy
% URN: 6562233
% EEEM007 Advanced Signal Processing - Lab Experiments
% Filename: asp_exp3.m
% Date started: 03-May-2019
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Clear everything and setup
```

```
10
11 clear all
12 close all
13 clc
14
15 a = 1;
16 b = 3;
17 c = 3;
18 d = 3;
19 f = 1;
20
21 %% effect of the size of test set on the reliability of the empirical error count
       ↪ estimator
22
23 sampleSizes = [3 5 10 20 50 100 200 500];
24
25 dims = 5;
26 mu1 = randi(5,dims,1);
27 mu2 = mu1 + 2*rand(dims,1);
28 cova = eye(dims);
29
30 X1_design = mvnrnd(mu1,cova,100);
31 X2_design = mvnrnd(mu2,cova,100);
32 Xd = [X1_design; X1_design];
33 train_labels = [zeros(100,1); ones(100,1)];
34
35 for i = 1:10
36     X1_test = mvnrnd(mu1,cova,500);
37     X2_test = mvnrnd(mu2,cova,500);
38     Xt{i} = [X1_test; X2_test];
39 end
40 test_labels = [zeros(500,1); ones(500,1)];
41
42 testSizes = [5, 10, 20, 50:50:500];
43 E_train_mean = [];
44 E_test_mean = [];
45 E_train_std = [];
46 E_test_std = [];
47
48 for i = 1:length(sampleSizes)
49     e_train = [];
50     e_test = [];
51     for iters = 1:10
52         mdl = fitcnb(Xd, train_labels);
53         train_prediction = predict(mdl, Xd);
54         test_prediction = predict(mdl, Xt{i});
55         e_train(iters) = sum(xor(train_prediction, train_labels))/length(
               ↪ train_prediction);
```

```
56          e_test(iters) = sum(xor(test_prediction, test_labels))/length(
                ↪ test_prediction);
57      end
58      E_train_mean(i) = sum(e_train)/iters;
59      E_test_mean(i) = sum(e_test)/iters;
60  % E_train_std(i) = (1/9)*sum((e_train.*E_train_mean(i)).^2);
61  % E_test_std(i) = (1/9)*sum((e_test.*E_test_mean(i)).^2);
62      E_train_std(i) = std(e_train);
63      E_test_std(i) = std(e_test);
64  end
65
66  fig = figure;
67  plot(sampleSizes, E_test_mean)
68  hold on
69  plot(sampleSizes, E_train_mean)
70  hold on
71  E_true = [E_test_mean(end), E_test_mean(end), E_test_mean(end), E_test_mean(end),
        ↪ E_test_mean(end), E_test_mean(end), E_test_mean(end), E_test_mean(end)];
72  plot(sampleSizes, E_true)
73  legend('Test Set Error', 'Training Set Error', 'True Error')
74  title(sprintf("ASP - Experiment 3 -" + " d=" + dims))
75  xlabel('Nd')
76  ylabel('Average Error')
77  saveas(fig,"./Exp3-results/ErrorComparison_" + dims + ".png")
78
79  fig = figure;
80  plot(sampleSizes, E_test_std)
81  hold on
82  plot(sampleSizes, E_train_std)
83  legend('Test Set Error', 'Training Set Error')
84  title(sprintf("ASP - Experiment 3 -" + " d=" + dims))
85  xlabel('Nd')
86  ylabel('Error % Standard Deviation')
87  saveas(fig,"./Exp3-results/ErrorStandardDeviation_" + dims + ".png")
```

A.5 Experiment 4

Listing 5: Matlab Code developed to simulate experiment 4.

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % @author: Kausthub Krishnamurthy
3  % URN: 6562233
4  % EEEM007 Advanced Signal Processing - Lab Experiments
5  % Filename: asp_exp4.m
6  % Date started: 7-May-2019
7  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8
```

```
%% Clear everything and setup
clear all
close all
clc

%% create multiple distribution pairs monotonically increasing.
dims = 5;
numSequences = 10;
classSizes = 100;

mu1 = [];
mu2 = [];
cova = [];
dm2 = [];

for i = 1:numSequences
    mu1{i} = randi(5,dims,1);
    mu2{i} = randi(5,dims,1);
    cova{i} = randi(5,dims,1) .* eye(5);
    dm2(1,i) = (mu1{i}-mu2{i})'*inv(cova{i})*(mu1{i}-mu2{i});
    dm2(2,i) = i;
end
ordered_dm2 = sortrows(dm2', 1)';

for i = 1:numSequences
    X1 = mvnrnd(mu1{ordered_dm2(2,i)},cova{ordered_dm2(2,i)},classSizes);
    X2 = mvnrnd(mu2{ordered_dm2(2,i)},cova{ordered_dm2(2,i)},classSizes);
    Xt{i} = [X1; X2];
    ordered_m1{i} = mu1{ordered_dm2(2,i)};
    ordered_m2{i} = mu2{ordered_dm2(2,i)};
    ordered_cova{i} = cova{ordered_dm2(2,i)};
end
labels = [zeros(classSizes,1); ones(classSizes,1)];

for i = 1:numSequences
    mdl = fitcnb(Xt{i}, labels);
    prediction = predict(mdl, Xt{i});
    e(i) = sum(xor(prediction, labels))/length(prediction);
end

fig = figure;
plot(ordered_dm2(1,:), e)
xlabel('Mahalanobis Distance')
ylabel('Error')
title(sprintf("ASP - Experiment 4 -" + " d=" + dims))
saveas(fig,"./Exp4-results/MahalError_" + dims + ".png")
```

A.6 Experiment 5

Listing 6: Matlab Code developed to simulate experiment 5.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% @author: Kausthub Krishnamurthy
% URN: 6562233
% EEEM007 Advanced Signal Processing - Lab Experiments
% Filename: asp_exp5.m
% Date started: 7-May-2019
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a = 1;
b = 3;
c = 3;
d = 3;
f = 1;

mu1 = [a;b]
mu2 = [a+d; b+d]
cova = eye(2)
```

A.7 Experiment 6

Listing 7: Matlab Code developed to simulate experiment 6.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% @author: Kausthub Krishnamurthy
% URN: 6562233
% EEEM007 Advanced Signal Processing - Lab Experiments
% Filename: asp_exp6.m
% Date started: 6-May-2019
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Clear everything and setup
clear all
close all
clc

a = 1;
b = 3;
c = 3;
d = 3;
f = 1;

mu1 = [a;b]
```

```
mu2 = [a+d; b+d]
cova = [c f;f c] % cova is the covariance matrix. didn't use cov since it's a built in function


%% Comparing kNN with a gaussian classifier
X1_test = mvnrnd(mu1,cova,100);
X2_test = mvnrnd(mu2,cova,100);

test_labels = [zeros(100,1); ones(100,1)];
sampleSizes = [3 5 10 50 100];
% sampleSizes = [3, 5:5:100];

kND_best = [];
E_train_best = [];
E_test_best = [];

E_train_mean_k = [];
E_test_mean_k = [];
for i = 1:length(sampleSizes)
    Nd = sampleSizes(i);
    E_train_mean = [];
    E_test_mean = [];
    if Nd*2 < 51
        kVals = 1:2:(2*Nd);
    else
        kVals = 1:2:51;
    end
    for kIter = 1:length(kVals)
        k = kVals(kIter);
        e_train = [];
        e_test = [];
        for iters = 1:10
            X1_design = mvnrnd(mu1,cova,Nd);
            X2_design = mvnrnd(mu2,cova,Nd);
            X = [X1_design; X2_design];
            train_labels = [zeros(Nd,1); ones(Nd,1)];
            mdl = fitcknn(X, train_labels);
            mdl.NumNeighbors = k;
            train_prediction = predict(mdl, X);
            test_prediction = predict(mdl, [X1_test; X2_test]);

            e_train(iters) = sum(xor(train_prediction, train_labels))/length(train_prediction);
            e_test(iters) = sum(xor(test_prediction, test_labels))/length(test_prediction);
        end
        E_train_mean(kIter) = sum(e_train)/iters;
```

```
66          E_test_mean(kIter) = sum(e_test)/iters;
67      end
68      E_train_mean_k{i} = E_train_mean;
69      E_test_mean_k{i} = E_test_mean;
70  end
71  for i = 1:length(E_test_mean_k)
72      [val, idx] = min(E_test_mean_k{i});
73      kND_best(i) = kVals(idx);
74      E_train_best(i) = E_train_mean_k{i}(idx);
75      E_test_best(i) = val;
76  end
77
78  fig = figure;
79  plot(sampleSizes, E_test_best)
80  hold on
81  plot(sampleSizes, E_train_best)
82  legend('Test Set Error', 'Training Set Error')
83  title(sprintf("ASP - Experiment 6"))
84  xlabel('Nd')
85  ylabel('Average Error')
86  saveas(fig,'./Exp6-results/ErrorComparison.png')
87
88  fig = figure;
89  plot(sampleSizes, kND_best)
90  title(sprintf("ASP - Experiment 6"))
91  xlabel('Nd')
92  ylabel('k')
93  saveas(fig,'./Exp6-results/BestkVals.png')
```