# 



# Specific Learning Difficulties Cover Note

**Student ID:** 150402149

**Advice for assessors and examiners**

Guidelines for markers assessing coursework and examinations of students diagnosed with Specific Learning Difficulties (SpLDs) –

As far as the learning outcomes for the module allow, examiners are asked to mark exam scripts sympathetically, ignoring the types of errors that students with SpLDs make and to focus on content and the student’s understanding of the subject.

Specific learning difficulties such as Attention Deficit Disorders, dyslexia and or dyspraxia may affect student performance in the following ways:

* The candidate’s spelling, grammar and punctuation may be less accurate than expected
* The candidate’s organisation of ideas may be confused, affecting the overall structure of written work
* The candidate’s proof reading may be weak with some errors undetected, particularly homophones and homonyms which can avoid spell checkers

**Under examination conditions, these difficulties are likely to be exacerbated. Errors are likely to become more marked towards the end of scripts.**

Useful approaches can include:

* Reading the passage quickly for content
* Including positive/constructive comments amongst the feedback so that students can work with specialist study skills tutors on developing new coping strategies
* Using clear English and when correcting; explain what is wrong and give examples
* Using non-red coloured pens for comments/corrections

**Colleagues in schools are asked to ensure that students with specific learning difficulties access the support provided by the** [Disability and Dyslexia Service](http://www.dds.qmul.ac.uk/about/index.html)**.**

For more information regarding marking guidelines see DDS webpage



http://www.dds.qmul.ac.uk/staffinfo/index.html and the [Institutional Marking Practices for Dyslexic Students](http://adshe.org.uk/wp-content/uploads/marking_guidelines_for_good_practice.docx)

**Disability and Dyslexia Service**

Student Services

Room 2.06 Francis Bancroft Building

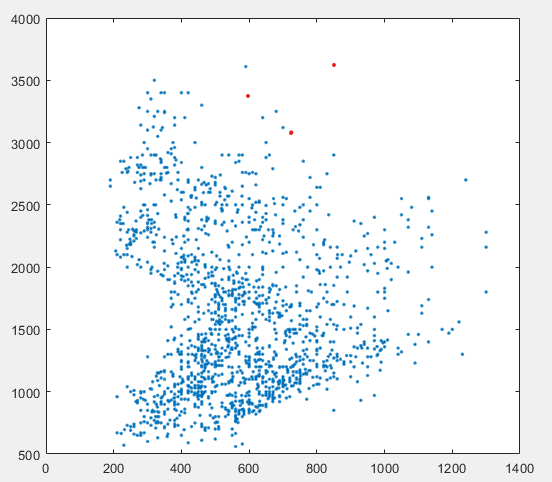
[www.dds.qmul.ac.uk](http://www.dds.qmul.ac.uk/)

Tel: 020 7882 2756

Email: dds@qmul.ac.uk

### Alteration or misuse of this document will result in disciplinary action

**Task 1**



load('PB12.mat');

load('PB\_data.mat');

J = f1;

J(:,2) = f2;

plot(J(:,1),J(:,2), ' .');

**Task 2**

To make x the data set for a selected phoneme.

phoneme = 1;

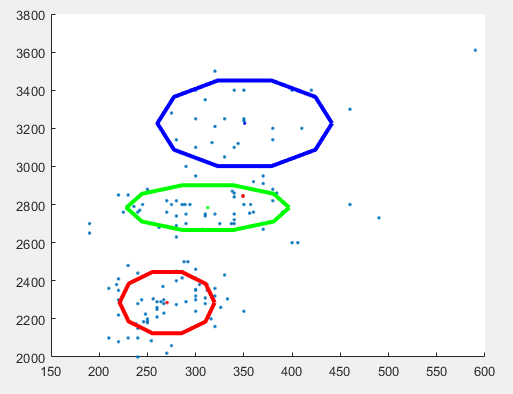
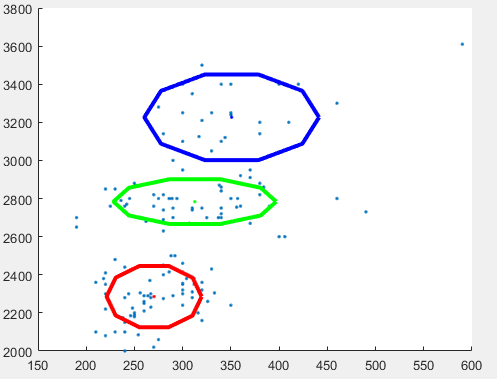
temp = [phno,J];

temp(temp(:,1) ~= phoneme, :) = [];

x = temp(:,2:3)

k = 3

The was very consistent in the location it identified for the clusters at a k = 3. Run results were all extremely similar.

MoG Model Phoneme 1 k = 3

*mu*

1.0e+03 \*

0.3508 0.2704 0.3126

3.2263 2.2855 2.7839

*s2*

(:,:,1) =

1.0e+04 \*

0.4103 0

0 2.7830

(:,:,2) =

1.0e+04 \*

0.1214 0

0 1.4278

(:,:,3) =

1.0e+03 \*

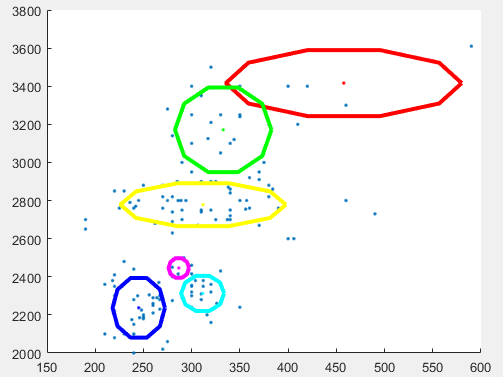
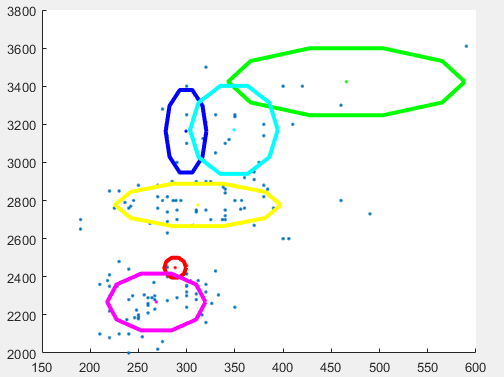
3.5626 0

0 7.6578

*p*

0.1839 0.4351 0.3810

At k = 6 there was far more variance in the final results. In addition to sometimes finding overlapping clusters.

This indicates to me that 6 is probably too many clusters to search for in this dataset.

*mu*

1.0e+03 \*

0.2965 0.4677 0.2449 0.3119 0.2808 0.3459

3.1462 3.4250 2.1543 2.7774 2.3375 3.1816

*s2*

(:,:,1) =

1.0e+04 \*

0.0168 0

0 2.4986

(:,:,2) =

1.0e+04 \*

0.7527 0

0 1.7344

(:,:,3) =

1.0e+03 \*

0.2830 0

0 7.2432

(:,:,4) =

1.0e+03 \*

3.6978 0

0 6.8953

(:,:,5) =

1.0e+03 \*

1.2183 0

0 6.9907

(:,:,6) =

1.0e+04 \*

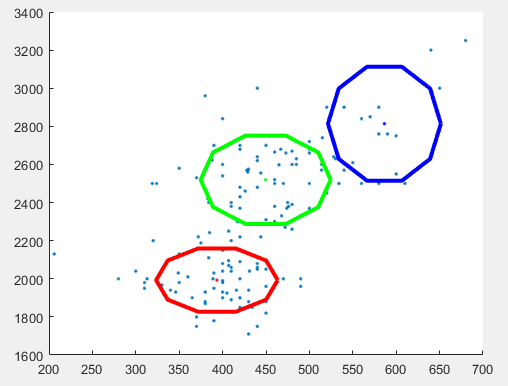
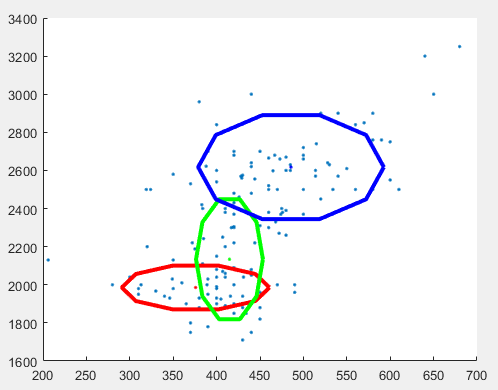
0.1105 0

0 2.8344

*p*

0.0449 0.0228 0.1255 0.3656 0.3086 0.1326

Phoneme 2 didn’t behave as consisted at k = 3. The results were mostly similar but sometimes gave a extremely anomalous result.

Normal result Anomalous result

*MoG model of Normal Result Phoneme 2 k = 3*

*mu*

1.0e+03 \*

0.4494 0.3934 0.5860

2.5190 1.9928 2.8120

*s2*

(:,:,1) =

1.0e+04 \*

0.2790 0

0 2.9782

(:,:,2) =

1.0e+04 \*

0.2456 0

0 1.5213

(:,:,3) =

1.0e+04 \*

0.2129 0

0 4.9421

*p*

0.4692 0.4346 0.0961

*MoG model of Anomalous Results Phoneme 2 k = 3*

*mu*

1.0e+03 \*

0.3758 0.4148 0.4857

1.9856 2.1338 2.6173

*s2*

(:,:,1) =

1.0e+03 \*

3.6202 0

0 7.2884

(:,:,2) =

1.0e+04 \*

0.0742 0

0 5.4796

(:,:,3) =

1.0e+04 \*

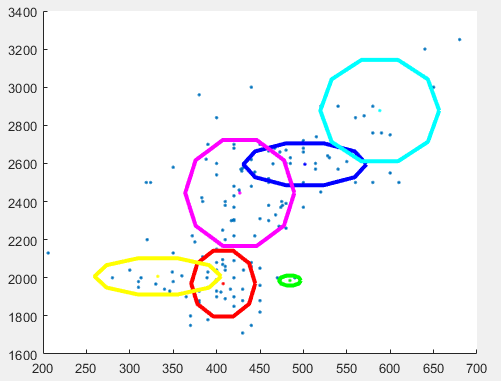
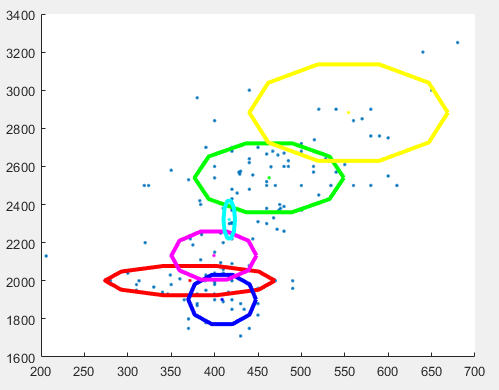
0.5723 0

0 4.1122

*p*

0.2137 0.3388 0.4474

Phoneme 2 at k = 6 was extremely erratic with no run looking similar to previous runs. Evening failing on the occasional run.

*MoG model of Result Phoneme 2 k = 6*

*mu*

1.0e+03 \*

0.3716 0.4629 0.4087 0.5543 0.3992 0.4169

2.0008 2.5406 1.9019 2.8825 2.1325 2.3211

*s2*

(:,:,1) =

1.0e+03 \*

4.7928 0

0 3.2795

(:,:,2) =

1.0e+04 \*

0.3700 0

0 1.8059

(:,:,3) =

1.0e+03 \*

0.7624 0

0 9.2258

(:,:,4) =

1.0e+04 \*

0.6537 0

0 3.5425

(:,:,5) =

1.0e+03 \*

1.2003 0

0 8.7890

(:,:,6) =

1.0e+03 \*

0.0221 0

0 5.3580

*p*

0.1458 0.4064 0.1785 0.0933 0.1204 0.0556

**Task 3**

Code:

%Loading in combind dataset

phonemex = 1;

phonemey = 2;

temp1 = [phno,J];

temp2 = [phno,J];

temp1((temp1(:,1) ~= phonemex), :) = [];

temp2((temp2(:,1) ~= phonemey), :) = [];

pointsToClassify = [temp1;temp2];

%Loading in model 1 values

load('ph1k3mu.mat')

load('ph1k3p.mat')

load('ph1k3s2.mat')

mu1 = mu;

p1 = p;

s21 = s2;

%Loading in model 2 values

load('ph2k3mu.mat')

load('ph2k3p.mat')

load('ph2k3s2.mat')

mu2 = mu;

p2 = p;

s22 = s2;

k = 3;

[Labelsk3, accuracyk3] = Classify\_Set(k, mu1, s21, p1, mu2, s22, p2, pointsToClassify, phonemex, phonemey);

disp(1-accuracyk3);

%Loading in model 1 values

load('ph1k6mu.mat')

load('ph1k6p.mat')

load('ph1k6s2.mat')

mu1 = mu;

p1 = p;

s21 = s2;

%Loading in model 2 values

load('ph2k6mu.mat')

load('ph2k6p.mat')

load('ph2k6s2.mat')

mu2 = mu;

p2 = p;

s22 = s2;

k = 6;

[Labelsk6, accuracyk6] = Classify\_Set(k, mu1, s21, p1, mu2, s22, p2, pointsToClassify, phonemex, phonemey);

disp(1-accuracyk6)

function [predictedLabels, accuracy] = Classify\_Set(k, mu1, s21, p1, mu2, s22, p2, pointsToClassify, label1, label2)

predictedLabels = {};

for x = 1:size(pointsToClassify,1)

predictedLabels{x,1} = Classifier(k, mu1, s21, p1, mu2, s22, p2, pointsToClassify(x,2:3));

end

correctCounter = 0;

for y = 1:size(pointsToClassify,1)

if predictedLabels{y,1} == 1;

if pointsToClassify(x,1) == label1

correctCounter = correctCounter + 1;

end

else

if pointsToClassify(x,1) == label2

correctCounter = correctCounter + 1;

end

end

end

accuracy = correctCounter/size(pointsToClassify,1);

function label = Classifier(kval, mu1 , s21 , p1, mu2, s22, p2, point)

probability1 = zeros(1,3);

probability2 = zeros(1,3);

for i=1:kval

probability1(1,i) = p1(i)\*det(s21(:,:,i))^(-0.5)\*exp(-0.5\*sum((point'-mu1(:,i))'\*inv(s21(:,:,i)).\*(point'-mu1(:,i))',2));

end

probability1 = probability1./repmat(sum(probability1,2),1,kval);

for i=1:kval

probability2(1,i) = p2(i)\*det(s22(:,:,i))^(-0.5)\*exp(-0.5\*sum((point'-mu2(:,i))'\*inv(s22(:,:,i)).\*(point'-mu2(:,i))',2));

end

probability2 = probability2./repmat(sum(probability2,2),1,kval);

%Maximum Likelihood Criterion

maxProb1 = max(probability1);

maxProb2 = max(probability2);

if maxProb1 < maxProb2

label = 1;

else

label = 2;

end

**Results**

My classifier functions by calculations the probability of a point within the two models that I have saved values for. For this is finds the highest probability in the Mixture of gaussian probability list and assign the label of this class.

At k3 my classifier was 42.76% Misclassification. Not great.

My k6 classification was slightly better at 38.49% misclassification. I believe that the increase in k improved the classification as it reduced the overlapping between cluster identified by phoneme 1 data and phoneme 2 data.

I was feeding in the oogonial training points into the classifier which might have caused these high rates of misclassification.

**Task 4**

I chose to the use k = 6 models as this had a lower misclassification rate. With 200 points per line.

%Creating grid of points.

numperrow = 200;

x = linspace(190,1300, numperrow);

y = linspace(560,3610, numperrow);

gridVectors = zeros(numperrow\*numperrow,2);

counter = 1;

for i = 1:numperrow

for j = 1:numperrow

gridVectors(counter,1) = x(i);

gridVectors(counter,2) = y(j);

counter = counter + 1;

end

end

load('ph1k6mu.mat')

load('ph1k6p.mat')

load('ph1k6s2.mat')

mu1 = mu;

p1 = p;

s21 = s2;

%Loading in model 2 values

load('ph2k6mu.mat')

load('ph2k6p.mat')

load('ph2k6s2.mat')

mu2 = mu;

p2 = p;

s22 = s2;

k = 6;

%Labeling the grid of data

Labels = Classify\_SetQ4(k, mu1, s21, p1, mu2, s22, p2, gridVectors);

%loading labels into grid

grid = zeros(numperrow);

counter = 1;

for i = 1:numperrow

for j = 1:numperrow

grid(i,j) = Labels{counter,1};

counter = counter + 1;

end

end

imagesc(grid);

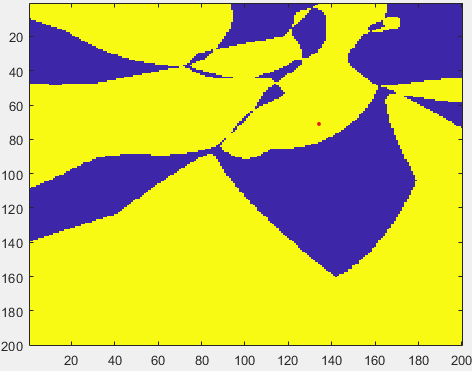
function predictedLabels = Classify\_SetQ4(k, mu1, s21, p1, mu2, s22, p2, pointsToClassify)

predictedLabels = {};

for x = 1:size(pointsToClassify,1)

predictedLabels{x,1} = Classifier(k, mu1, s21, p1, mu2, s22, p2, pointsToClassify(x,:));

end



Example of the classification grid at row size 20

2 2 2 2 2 2 2 2 2 1 1 1 1 2 2 2 1 1 1 1

1 2 2 2 2 2 2 2 2 1 1 1 1 1 2 2 2 1 1 1

1 1 1 2 2 2 2 2 2 1 1 2 1 2 2 1 1 1 1 1

1 1 1 1 1 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1

1 1 1 1 1 2 2 2 1 2 2 1 2 2 2 1 1 1 1 1

2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 1

2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 1 2 2 2 2

2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 2 2

2 2 2 1 1 1 1 2 2 1 1 1 1 1 1 1 1 2 2 2

2 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 2 2 2

1 1 1 1 1 2 2 2 2 2 1 1 1 1 1 1 1 2 2 2

1 1 1 1 2 2 2 2 2 2 2 1 1 1 1 1 1 2 2 2

1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2

2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2