

ECE 8527 Homework Number 1: Gaussian Distributions and Maximum Likelihood Decoding

Andrew Powell

February 3, 2014

1 Introduction

Homework 1 of ECE 8527 deals with data generated as gaussian (normal) distributions and decoding error rates as maximum likelihood estimates. Gaussian distributions, whose probability density function (PDF) is shown in Equation 1, is applied in many practical applications due to the central limit theorem, which states a large set of random variates with unknown distributions will tend to a normal distribution so long as the random variates' means and variances are finite [3]. Random variates are particular outcomes to a random variable. The particular outcomes of this homework assignment are the data sets generated in MATLAB.

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$

The goal of maximum likelihood estimation is to determine the value(s) of one or more parameters (such as mean or error rate) that makes a particular set of data the most likely to occur. Finding the values can be found through a likelihood function which maps the possible values of one or more parameters for a particular set of data to the *likelihood* of the particular set of data occurring [2]. The value(s) that maximize the likelihood function are the maximum likelihood estimates of the parameter(s).

The tasks of the homework are broken down into 4 problems. Problem 1 involves the generation of two equally sized, gaussian distributed data sets with same variance but differing means and then determining an error rate based on the data sets, using an optimal maximum likelihood decoder. Problem 2 involves the creation of two sets of gaussian distributed data and then mapping how the error rate changes with respect to the data sets' variance. Problem 3 involves the generation of two two-dimensional multivariate gaussian distributed data sets whose mean vectors covariance matrices are identity matrices and then again determining an error rate, using an optimal maximum likelihood decoder. Finally, Problem 4 involves demonstrating how the support region of the first multivariate data set from Problem 3 changes when manipulating its covariance matrix.

2 Discussion

2.1 Problem 1

The goal of Problem 1 is to determine the theoretical error rate of a statistical model and then determine the error rate, using maximum likelihood estimation. The statistical model consists of two one-dimensional data sets of 10,000 data points. Both data sets are gaussian distributed with variances equal to 1. The first data set has a mean of 1 and the second data set has a mean of -1. The thresholds necessary for decoding which data set to which each data point

belongs are 0, 0.5, and -0.5. Data sets one and two are respectively referred to as ω_1 and ω_2 . Each data point (i.e. the feature) is a continuous random variable symbolically referred to as \mathbf{x} . The possible error rates are represented by the random variable θ . The maximum likelihood estimate of the error rate is $\hat{\theta}$. It is assumed the thresholds given for Problem 1 are really the priors $P(\omega_1)$ and $P(\omega_2)$. A threshold of zero would imply both priors are 0.5. A threshold of .5 would imply $P(\omega_1) = 0.6225$ and $P(\omega_2) = 0.3775$. A threshold of -.5 would simply flip the priors.

First, the error rate is obtained through a theoretical prediction and then finally obtained by a maximum likelihood estimate.

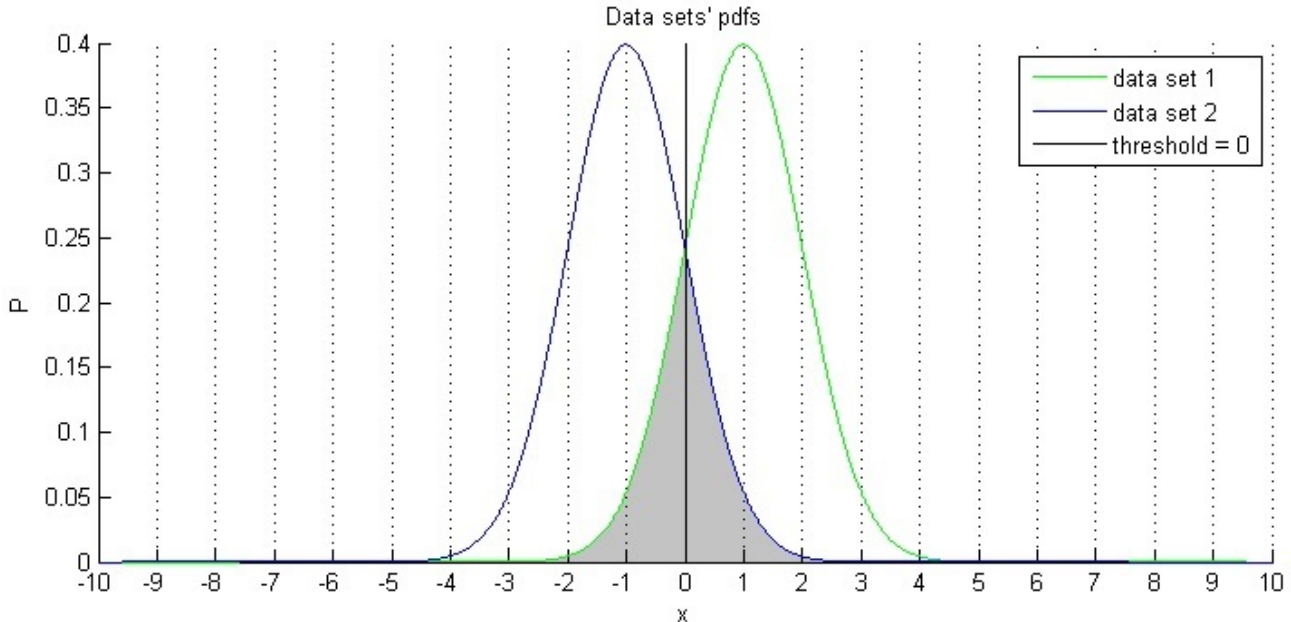
2.1.1 Theoretical Prediction

The probability of error given each data point is always Equation 2, which, for the two class case, simply states the probability of error is the probability that the data set decoded is from the opposite data set.

$$P(\text{error}|\mathbf{x}) = \begin{cases} P(\omega_2|\mathbf{x}) & \mathbf{x} \in \omega_1 \\ P(\omega_1|\mathbf{x}) & \mathbf{x} \in \omega_2 \end{cases} \quad (2)$$

In the context of this problem, the probability of error is simply the areas beneath each data set's gaussian PDF that fall into the wrong side of the thresholds. Figure 1 graphically shows where the two erroneous areas are beneath each data set's PDF.

Figure 1: Gray area to the left of the threshold corresponds to $P(\omega_2|\mathbf{x}) \mathbf{x} \in \omega_1$, whereas to the right of the threshold corresponds to $P(\omega_1|\mathbf{x}) \mathbf{x} \in \omega_2$



Averaging together the two erroneous areas shown in Figure 1 results in the average probability of error, which is the theoretically error rate. Instead of performing calculations by hand, a MATLAB script is written to perform the necessary calculations, as shown in Listing 1.

```
1 % prediction (the parameters are known)
  change = .002;
3 domain = -10:change:10;
```

```

errorRate = @(threshold) (sum(normpdf([-10:change:threshold], 1, 1))+ ...
5     sum(normpdf([threshold:change:10], -1, 1)))*change/2;
printErrorRate = @(threshold) disp(char(['threshold : ' num2str(threshold)], ...
7     ['Error rate: ' num2str(errorRate(threshold))]));
disp('Predicted error rates');
9 printErrorRate(threshold1);
printErrorRate(threshold2);
11 printErrorRate(threshold3);

```

Listing 1: MATLAB script for determining theoretical results

The theoretical error rates produced from each threshold are shown in Listing 2.

```

1 Problem 1
3 Predicted Error Rates
threshold : 0
5 Error rate: 0.1589
threshold : 0.5
7 Error rate: 0.18791
threshold : -0.5
9 Error rate: 0.18791

```

Listing 2: Theoretical results

2.1.2 Maximum Likelihood Estimation

In order to perform maximum likelihood estimation, there are number of assumptions made. The reasons for the assumptions are made clear throughout this section.

The first assumption is that the total number of correctly decoded data points, symbolically referred to as c , and the total number of decoded data points, symbolically referred to as t , are allowed to be used for generating the estimates. Threshold decoding is used to determine what these values are (Listing 3).

```

1 % create two 1 dimensional gaussian random variables
createGrv = @(mean, variance, length) randn(length,1)*sqrt(variance)+mean;
3 n = 10e3;
5 % the data (omegal and omega2, respectively)
% together, grv1 and grv2 have 2*n = t random data points
7 grv1 = createGrv(1, 1, n);
grv2 = createGrv(-1, 1, n);
9
% thresholds
11 threshold1 = 0;
threshold2 = .5;
13 threshold3 = -.5;
15 ...
17 % anonymous function used for generating the number of correctly
% decoded data points
19 correct = @(threshold) sum(grv1 > threshold)+sum(grv2 < threshold);

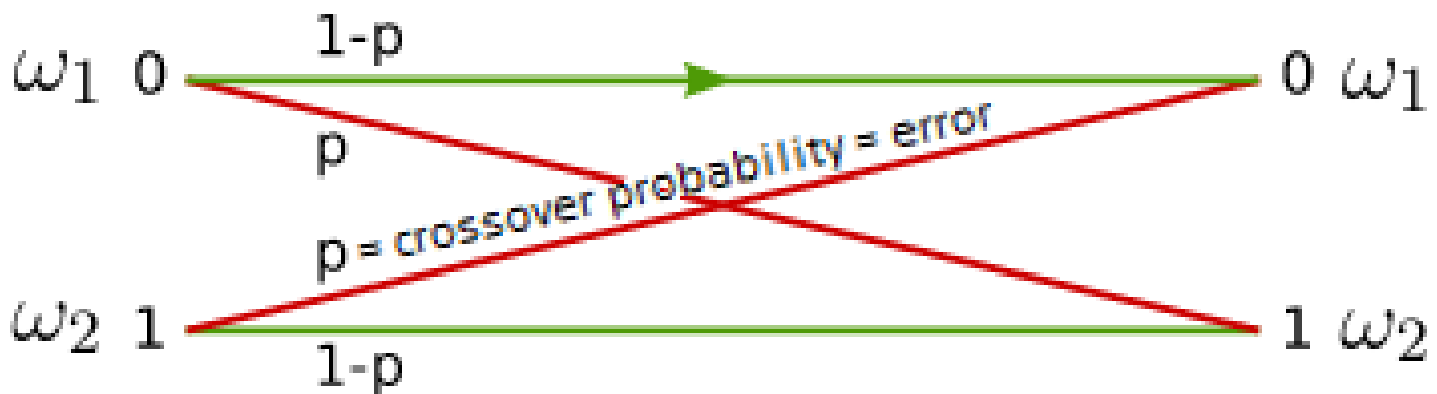
```

Listing 3: MATLAB script for determining c and t

The second assumption is the statistical model can be interpreted as a binary symmetric channel (BSC), a common communications model in which a binary message, typically either a 0_2 or a

1_2 , is transmitted at one end of the channel and then received at the other. There is a probability the received message can be decoded incorrectly as the opposite bit. The probability the message is incorrectly decoded is the crossover probability p [1]. In the context of Problem 1, ω_1 and ω_2 are 0_2 and 1_2 , respectively. The crossover probability p is the error rate.

Figure 2



http://en.wikipedia.org/wiki/Binary_symmetric_channel

The distribution of the combined data sets is conveniently assumed binomial to follow the BSC interpretation of the statistical model. The probability mass function (PMF) of a binomial distribution is shown in Equation 3. Equation 3 is modified to incorporate the parameters mentioned for solving for the maximum likelihood estimation of the error rate.

$$\binom{t}{c} (1-p)^c p^{(n-c)} \quad (3)$$

$$L(\theta|D) = \binom{t}{c} (1-\theta)^c \theta^{(n-c)} \quad (4)$$

Since the error rate is estimated and the data is given, Equation 3 is rewritten as the likelihood function needed for getting the estimated error rate, as shown in Equation 4. Listing 4 shows how the likelihood function is generated in MATLAB. The MATLAB implementation of the likelihood function is actually a function of $1 - \theta$.

```

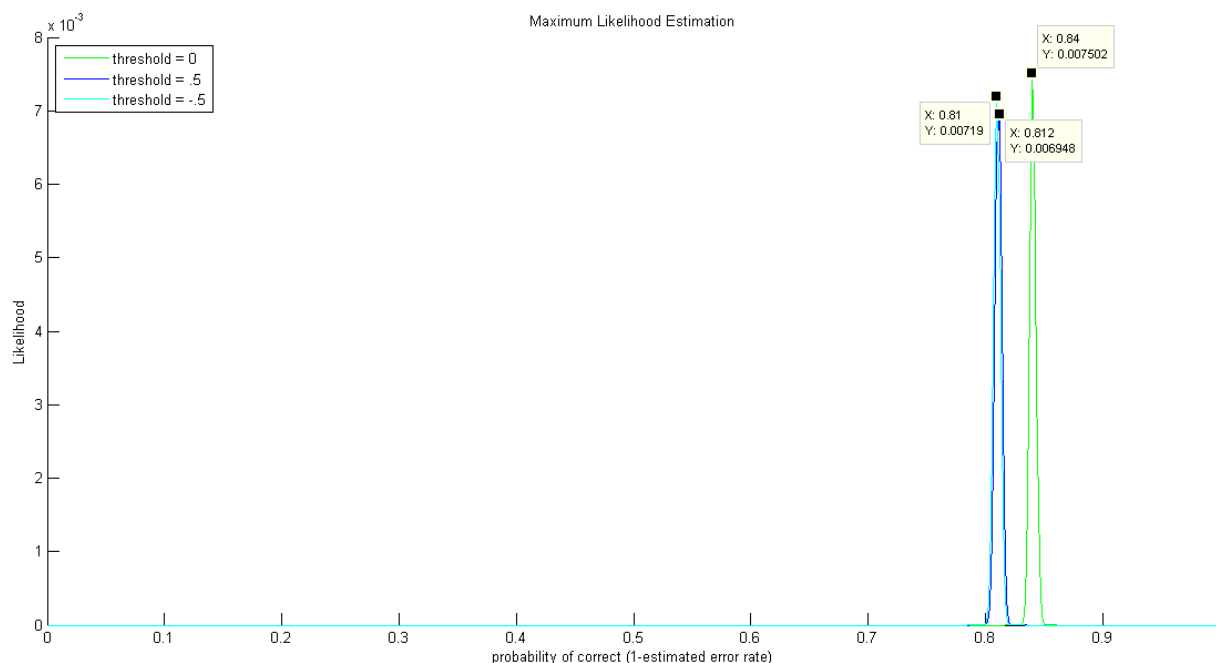
1 % generate the possible probabilities
probabilities = 0:.002:1;
3
4 % recall, grv1 and grv2 are the data sets
5 correct = @(threshold) sum(grv1 > threshold) + sum(grv2 < threshold);
6
7 % this anonymous function is used to generate the number of correctly decoded
8 % data points for each threshold
9 generateLikelihood = @(threshold) binopdf(correct(threshold), 2*n, probabilities);

```

Listing 4: MATLAB script for generating the likelihood function

Plotting the likelihood functions for the three thresholds generates the plots shown in Figure 3.

Figure 3: Likelihood Functions. The MATLAB Datatips indicate what the maximums are.



As the final step, the $1 - \hat{\theta}$, the $1 - \theta$ that maximizes Equation 4, for each threshold are converted to $\hat{\theta}$, the estimated error rates shown in Listing 5. These error rates are not precise as the theoretical error rates, but are fairly close.

```

1 Estimated Error rates
threshold : 0
3 Error rate: 0.16
threshold : 0.5
5 Error rate: 0.188
threshold : -0.5
7 Error rate: 0.19

```

Listing 5: Estimated Results

2.2 Problem 2

The goal of Problem 2 is to plot how the error rate of a statistical model changes as the variance of the data sets changes within the range 0.1 and 2. The two data sets always have 1,000 data points each and have a gaussian distribution. One data set always has a mean of 1, whereas the other has a mean of -1. The priors for both data sets are assumed to be 0.5, thus the discriminant function is simply Equation 5.

$$g_1(\mathbf{x}) - g_2(\mathbf{x}) = 2\mathbf{x} \quad (5)$$

As shown in Listing 6, discriminant function is encoded as a threshold set to zero.

```

1 % an anonymous function for easily generating gaussian distributions
createGrv = @(mean, variance, length) randn(length,1)*sqrt(variance)+mean;
3
% the size of each distribution

```

```

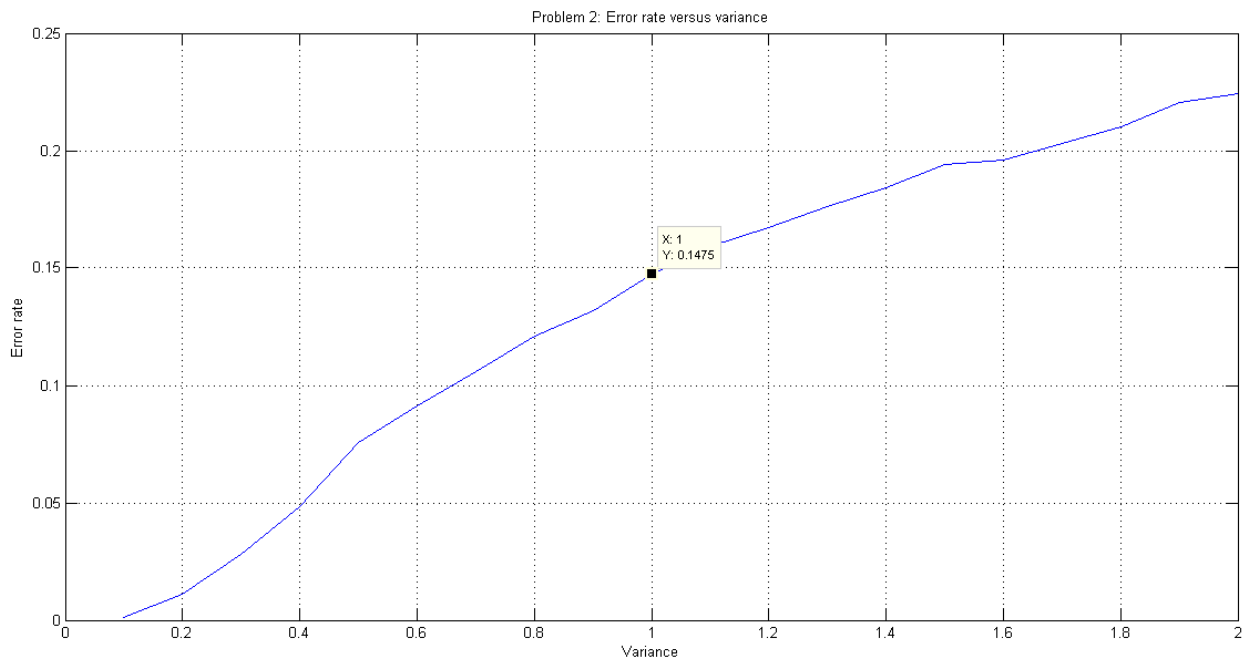
5 n = 1e3;
7 % the anonymous that generates the error rates are generated
errorRate = @(variance) (sum(createGrv(1, variance, n) < 0)+ ...
9     sum(createGrv(-1, variance, n) > 0))/ ...
    (2*n);
11 % the range of variances
13 variances = .1:.1:2;
15 % finally the error rates are generated
errorRates = errorRate(variances);

```

Listing 6: MATLAB script for generating the error rates

Figure 4 is the plot of how the error rate changes with respect to the variance.

Figure 4: Error rate versus variance



2.3 Problem 3

For Problem 3, the theoretical and estimated error rate is computed for another two-class problem but with two-dimensional multivariate data sets, ω_1 and ω_2 . Both data sets have a covariance matrix Σ equal to the identity matrix. ω_1 has a mean vector of $\mu_1 = [-1, 1]^T$, whereas ω_2 has a mean vector of $\mu_2 = [1, 1]^T$. \mathbf{x} in the context of Problem 3 refers to data (*column*) vectors since the two-dimensional data sets are aggregations are vectors, instead of one-dimensional points. The number of data vectors per data set is chosen as 10,000 data vectors. Listing 7 shows how the data sets are created in MATLAB.

```

% n is the number of data vectors per multivariate normally
2 % distributed data sets
n = 10e3;
4 % this anonymous function generates the two dimensional data

```

```

6 createGrv = @(mean, variance)mvnrnd(mean, variance, n);
8 % the parameters are established (mean vector / covariance matrix)
meanVector1 = [-1 1]';    covarianceMatrix1 = [1 0; 0 1];
10 meanVector2 = [1 1]';    covarianceMatrix2 = eye(2);

12 % finally the data is created (grv1 and grv2 are omega1 and
% omega2, respectively)
14 grv1 = createGrv(meanVector1, covarianceMatrix1)';
grv2 = createGrv(meanVector2, covarianceMatrix2)';
16
% priors
18 prior1 = .5; prior2 = .5;

```

Listing 7: MATLAB script for generating the data sets ω_1 and ω_2

The discriminant function for decoding each data vector \mathbf{x} is shown in Equation 6. The 0 in the discriminant function's vector $[2, 0]^T$ causes the second element of the discriminant function to always result in 0. Therefore, it is same to assume the second element in *any* data vector has no impact on whether ω_1 or ω_2 is decided. A closer examination also reveals the discriminant function in Equation 6 effectively becomes the discriminant function shown in Equation 5; the multivariate two-class problem can be considered a one dimensional two-class problem with an extra, trivial dimension.

$$g_1(\mathbf{x}) - g_2(\mathbf{x}) = [2, 0]\mathbf{x} \quad (6)$$

2.3.1 Theoretical Prediction

Considering the observation last stated in Section 2.3, the theoretical error rate calculated in Problem for a threshold of 0 is adopted as the threshold for Problem 3.

```

Problem 3
2
Predicted Error Rate
4 threshold : 0
Error rate: 0.1589

```

Listing 8: Theoretical results

2.3.2 Maximum Likelihood Estimation

The estimated error rate $\hat{\theta}$ is solved the same way as the estimated errors for Problem 1. First the discriminant function is implemented as an anonymous function from the data sets' the priors, covariance matrices, and mean vectors. The general case gaussian classifiers is implemented for the purpose of experimenting with the data sets' parameters. Listing 9 shows how the discriminate function is implemented and utilized as MATLAB script.

```

1 % The discriminant function is a quadratic equation whose constants
% A, b, and c and defined as follows for gaussian classifiers
3 A = 1/2*(covarianceMatrix2^(-1)-covarianceMatrix1^(-1));
b = covarianceMatrix1^(-1)*meanVector1-covarianceMatrix2^(-1)*meanVector2;
5 c = 1/2*(meanVector2'*covarianceMatrix2^(-1)*meanVector2 - ...
meanVector1'*covarianceMatrix1^(-1)*meanVector1) + log(prior1/prior2) - ...
7 1/2*logm(abs(covarianceMatrix1)/abs(covarianceMatrix2));

9 % this anonymous function is the discriminant function itself
g = @(x)x'*A*x+b'*x+c; % discriminant function

```

```

11 % these vectors together bin the results from the discriminant function
13 % for each data vector, however g1 are the results generated from the
15 % data vectors of grv1 and g2 are the results generated from the data
17 % vectors generated from the grv2
19 g1 = zeros(1, n);      % positive values indicate correct decoding!
21 g2 = zeros(1, n);      % negative values indicate correct decoding!
23 for i = 1:n
    temp = g(grv1(:,i));
    g1(i) = temp(1);
    temp = g(grv2(:,i));
    g2(i) = temp(1);
end

```

Listing 9: MATLAB script for implementing discriminant function

The binomial PMF representing the total data's distribution is generated as the statistical model's likelihood function. Listing 10 shows the MATLAB code that generates the likelihood functions, and Figure 5 shows the plot of the likelihood function changing with respect to $1 - \theta$.

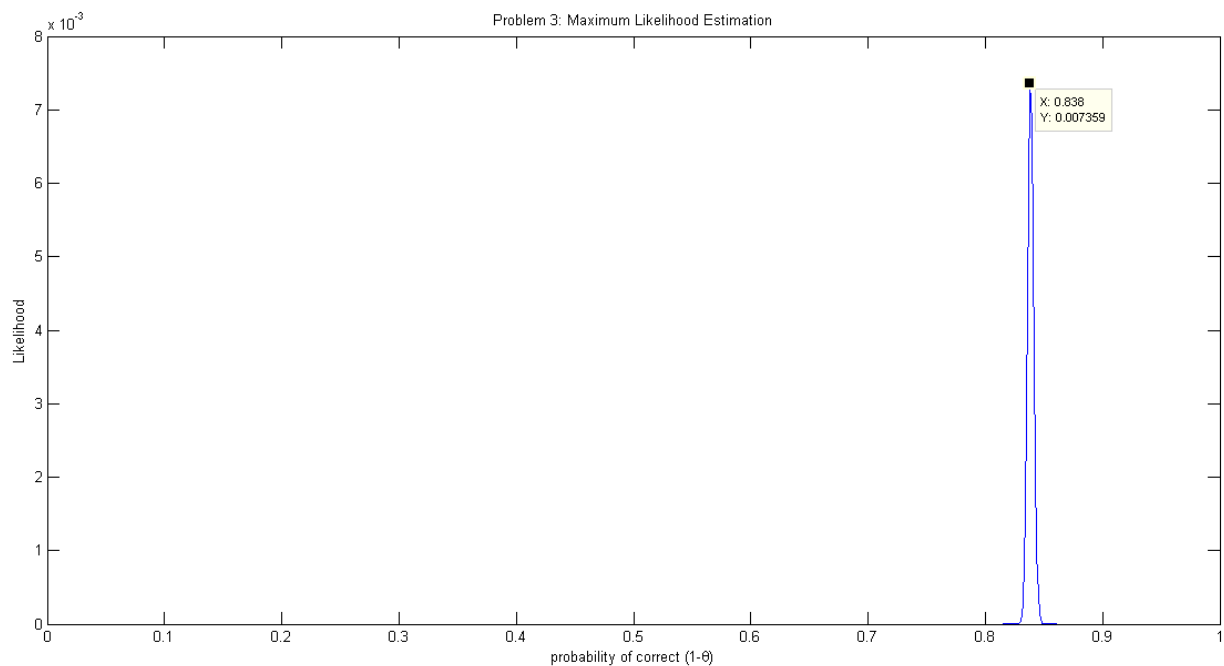
```

1 % the domain
probabilities = 0:.002:1;
3
5 % the number of correctly decoded data vectors are summed
correct = sum(g2 < 0) + sum(g1 > 0);
7
% the likelihood is finally generated
likelihood = binopdf(correct, 2*n, probabilities);

```

Listing 10: MATLAB script for generating the likelihood function

Figure 5: Likelihood Function. The MATLAB Datatip indicates what the maximum is.



The estimated error rate is finally determined as the following.


```

Estimated Error rate
Error rate: 0.162

```

Listing 11: Estimated Results

2.4 Problem 4

Problem 4 simply shows how the support region of the first multivariate data set from Problem 3 changes when its covariance matrices is modified. For each figure showing a support region, the corresponding covariance matrix precedes.

Figure 6

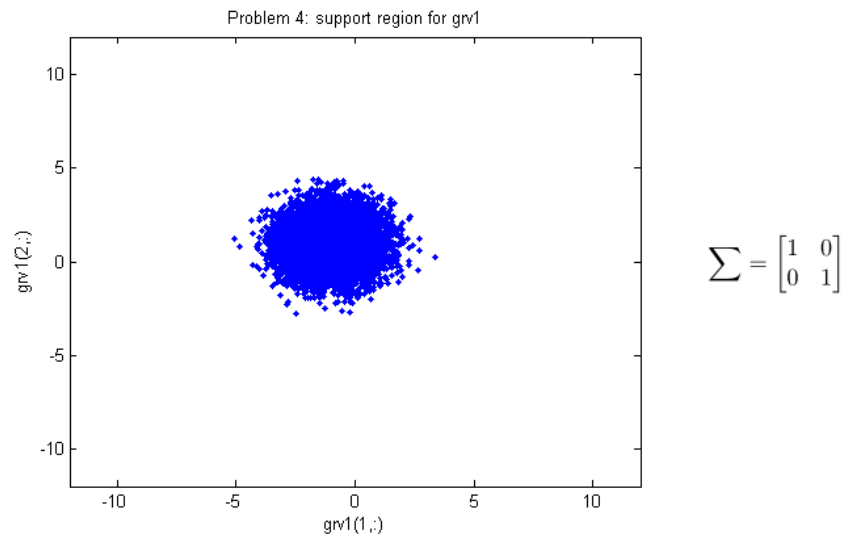


Figure 7

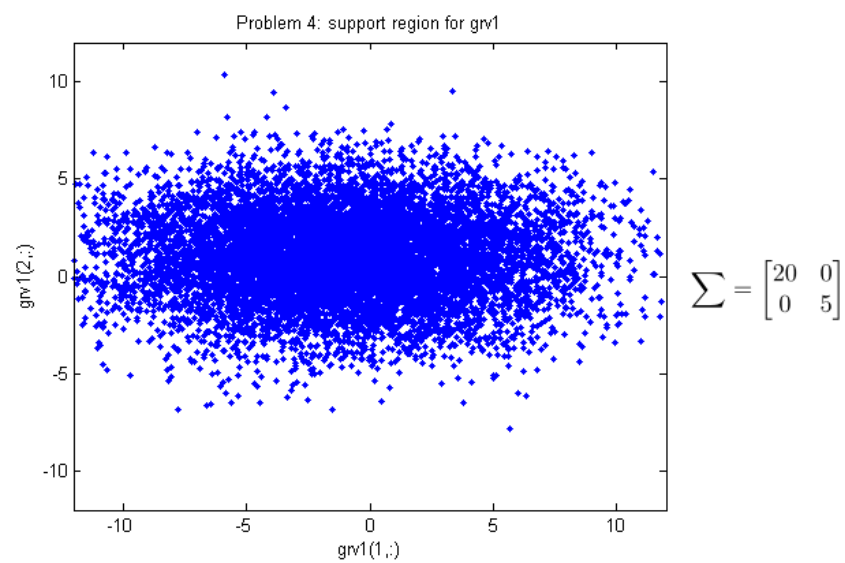


Figure 8

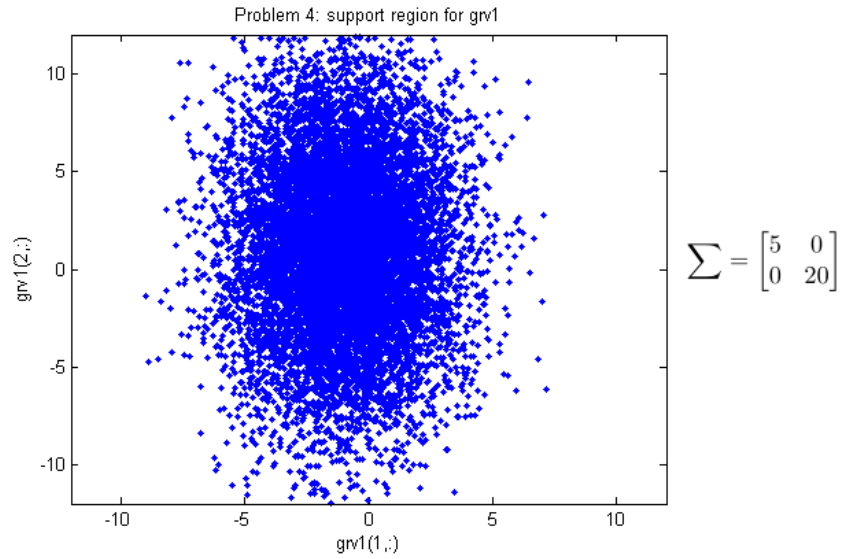


Figure 9

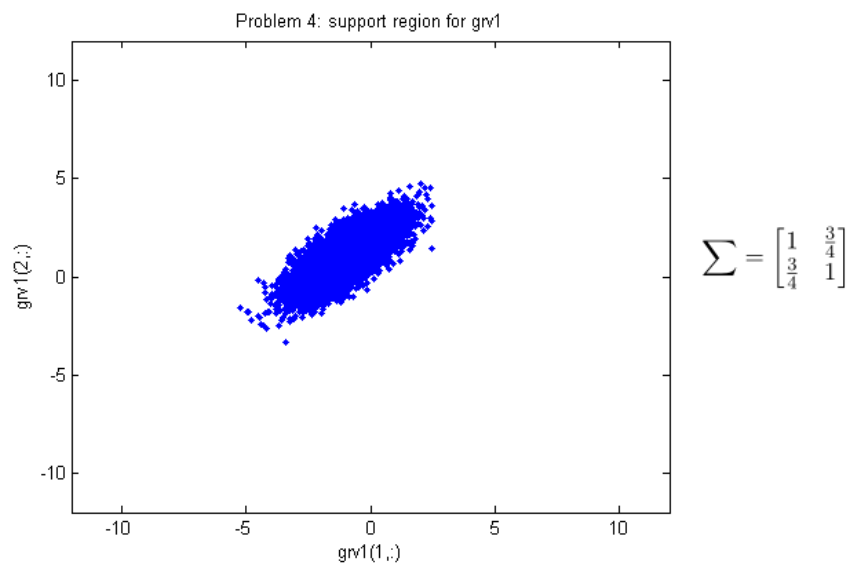
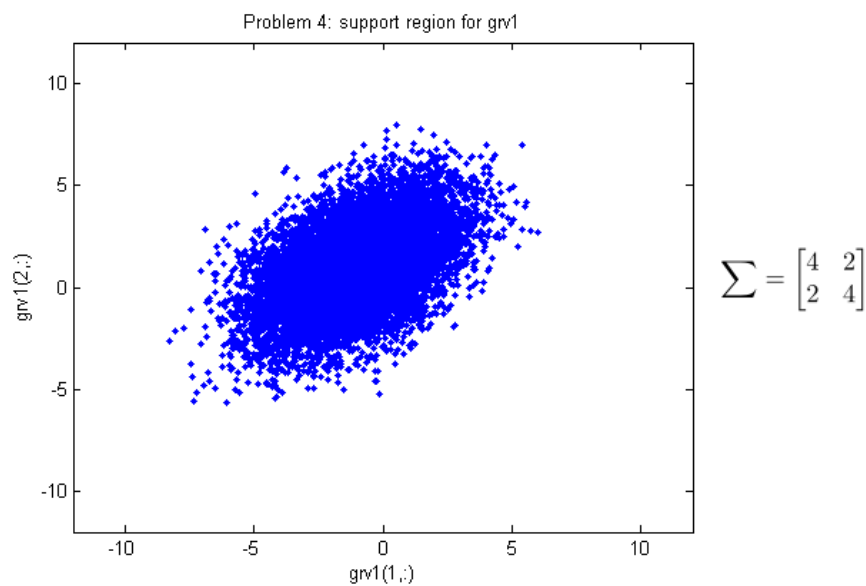


Figure 10



3 Appendix

3.1 MATLAB script

```

1 % clear everything
2 close all;

4 %% Problem 1
5 % clear workspace
6 clear all;

8 % display problem 1
9 disp('Problem 1');
10 disp(' ');

12 % create two 1 dimensional gaussian random variables
13 createGrv = @(mean, variance, length) randn(length,1)*sqrt(variance)+mean;
14 n = 10e3;

16 % the data
17 grv1 = createGrv(1, 1, n);
18 grv2 = createGrv(-1, 1, n);

20 % thresholds
21 threshold1 = 0;
22 threshold2 = .5;
23 threshold3 = -.5;

24

26 % prediction (the parameters are known)
27 change = .002;
28 domain = -10:change:10;
29 errorRate = @(threshold) (sum(normpdf([-10:change:threshold], 1, 1))+ ...
30     sum(normpdf([threshold:change:10], -1, 1)))*change/2;
31 printErrorRate = @(threshold) disp(char(['threshold : ' num2str(threshold)], ...
32     ['Error rate: ' num2str(errorRate(threshold))]));
33 disp('Predicted error rates');
34 printErrorRate(threshold1);
35 printErrorRate(threshold2);

```

```
printErrorRate(threshold3);
36 disp(' ');

38 % plot figures
figure();
40 hold on
plot(domain, normpdf(domain, 1, 1), 'g');
42 plot(domain, normpdf(domain, -1, 1), 'b');
plot(0*ones(1, 20), linspace(.4, 0, 20), 'k');
44 hold off
title('Data sets' pdfs');
46

48 % maximum likelihood estimation
% source1: http://statgen.iop.kcl.ac.uk/bgim/mle/sslike\_5.html
50 % source2: http://www.gaussianwaves.com/2010/01/maximum-likelihood-estimation-2/
probabilities = 0:change:1;
52 correct = @(threshold)sum(grv1 > threshold)+sum(grv2 < threshold);
generateLikelihood = @(threshold)binopdf(correct(threshold), 2*n, probabilities);
54

% plot
56 figure();
hold on
58 plot(probabilities, generateLikelihood(threshold1), 'g');
plot(probabilities, generateLikelihood(threshold2), 'b');
60 plot(probabilities, generateLikelihood(threshold3), 'c');
hold off
62 title('Maximum Likelihood Estimation');
xlabel('probability of correct');
64 ylabel('Likelihood');

66 % get error rates
[~,errorIndex] = max(generateLikelihood(threshold1));
68 errorRate1 = 1-probabilities(errorIndex);
[~,errorIndex] = max(generateLikelihood(threshold2));
70 errorRate2 = 1-probabilities(errorIndex);
[~,errorIndex] = max(generateLikelihood(threshold3));
72 errorRate3 = 1-probabilities(errorIndex);

74 disp('Estimated error rates');
printErrorRate = @(threshold, errorRate)disp(char(['threshold : ' ...
76     num2str(threshold)], ['Error rate: ' num2str(errorRate)]));
printErrorRate(threshold1, errorRate1);
78 printErrorRate(threshold2, errorRate2);
printErrorRate(threshold3, errorRate3);
80 disp(' ');

82 %% Problem 2
% clear workspace
84 clear all;

86 % create data
createGrv = @(mean, variance, length)randn(length,1)*sqrt(variance)+mean;
88 n = 1e3;
errorRate = @(variance)(sum(createGrv(1, variance, n) < 0)+ ...
90     sum(createGrv(-1, variance, n) > 0))/ ...
    (2*n);
92 variances = .1:.1:2;
errorRates = errorRate(variances);
94

% create figure
96 figure();
plot(variances, errorRates);
```

```

98 title('Problem 2: Error rate versus variance');
99 ylabel('Error rate');
100 xlabel('Variance');

102 %% Problem 3

104 % clear workspace
105 clear all;

106 % display problem 3
107 disp('Problem 3');
108 disp(' ');

110 % create the data
111 n = 10e3;
112 createGrv = @(mean, variance)mvnrnd(mean, variance, n);
113 meanVector1 = [-1 1]'; covarianceMatrix1 = [4 2; 2 4];
114 meanVector2 = [1 1]'; covarianceMatrix2 = eye(2);
115 grv1 = createGrv(meanVector1, covarianceMatrix1)';
116 grv2 = createGrv(meanVector2, covarianceMatrix2)';

118 % priors
119 prior1 = .5; prior2 = .5;

122 % discriminant function for the general case
123 A = 1/2*(covarianceMatrix2^(-1)-covarianceMatrix1^(-1));
124 b = covarianceMatrix1^(-1)*meanVector1-covarianceMatrix2^(-1)*meanVector2;
125 c = 1/2*(meanVector2'*covarianceMatrix2^(-1)*meanVector2 - ...
126     meanVector1'*covarianceMatrix1^(-1)*meanVector1) + log(prior1/prior2) - ...
127     1/2*logm(abs(covarianceMatrix1)/abs(covarianceMatrix2));

128 g = @(x)x'*A*x+b'*x+c; % discriminant function
129 g1 = zeros(1, n); % positive values indicate correct decoding!
130 g2 = zeros(1, n); % negative values indicate correct decoding!
131 for i = 1:n
132     temp = g(grv1(:,i));
133     g1(i) = temp(1);
134     temp = g(grv2(:,i));
135     g2(i) = temp(1);
136 end

138 % maximum likelihood estimation
139 probabilities = 0:.002:1;
140 correct = sum(g2 < 0) + sum(g1 > 0);
141 likelihood = binopdf(correct, 2*n, probabilities);
142 figure();
143 plot(probabilities, likelihood);
144 title('Problem 3: Maximum Likelihood Estimation');
145 xlabel('probability of correct');
146 ylabel('Likelihood');
147 [~, maxIndex] = max(likelihood);
148 disp(['Error rate: ' num2str(1-probabilities(maxIndex))]);
149 disp(' ');

152 %% Problem 4
153 figure();
154 plot(grv1(1,:), grv1(2,:), '.');
155 title('Problem 4: support region for grv1');
156 xlabel('grv1(1,:)');
157 ylabel('grv1(2,:)');
158 xlim([-12 12]);
159 ylim([-12 12]);

```

Listing 12: MATLAB script full

References

- [1] Binary symmetric channel. http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Binary_symmetric_channel.html. Accessed: 2014-2-1.
- [2] S.Purcell. Maximum likelihood estimation (mle). http://statgen.iop.kcl.ac.uk/bgim/mle/sslike_3.html, May 2007. Accessed: 2014-2-1.
- [3] Eric W Weisstein. Central limit theorem. <http://mathworld.wolfram.com/CentralLimitTheorem.html>. Accessed: 2014-1-31.