ECE 2372 - Homework 1

Please upload your solutions to Canvas.

Jiyang Liu 4731134

1. Consider n independent tosses of m biased coins. The probability of heads for each coin is p. We know that the exact probability of k heads in n tosses is calculated by the binomial distribution, and we can find an empirical estimate by finding the ratio of the number of times the coin lands on heads to n.

$$\hat{p}_i = \frac{\text{number of times coin}i\text{lands on heads}}{n} \quad \text{where} \quad i = 1, ..., m$$

a. Fill the table with the exact probability values that at least one coin out of m will have no heads when you flip each coin $n = 10$ times.

1a.

Use the binomial distribution:

$$Bin(m|N,\mu) = \binom{N}{m}\mu^m(1-\mu)^{N-m}$$

For m=1, p=0.04,

$$P = (1-0.04)^{10} = 0.665$$

For m=1, p=0.75,

$$P = (1-0.75)^{10} = 9.537 \times 10^{-7}$$

For m=1000, p=0.04,

$$P = 1 - Bin(0|1000,0.665) = 1 - 1 * 1 * (1-0.665)^{1000} = 1$$

For m=1000, p=0.75,

$$P = 1 - Bin(0|1000,9.537 \times 10^{-7}) = 1 - 1 * 1 * (1 - 9.537 \times 10^{-7})^{1000} = 0.000953$$

For m=1000000, p=0.04,

$$P = 1 - Bin(0|1000000,0.665) = 1 - 1 * 1 * (1-0.665)^{1000000} = 1$$

For m=1000000, p=0.04,

$$P = 1 - Bin(0|1000000,9.537 \times 10^{-7}) = 1 - 1 * 1 * (1 - 9.537 \times 10^{-7})^{1000000} = 0.615$$

	m=1	m=1,000	m=1,000,000
p=0.04	0.665	1	1
p=0.75	9.537×10^{-7}	0.000953	0.615

b. In this part, we will compare the the exact probability with its bound suggested by Hoeffing's inequality. Consider ten tosses of two coins with $p = 0.5$, calculate exact probability of the event stated below.

$$\mathbb{P}[max_i \mid \hat{p}_i - p_i \mid > \epsilon]$$

Sketch this probability as a function of $\epsilon \in [0, 1]$. On the same figure, show the Hoeffding's bound. Comment on the tightness of the bound.

1b.

$$P = [max_i|\widehat{p_i} - p_i| \geq \varepsilon] = 1 - P[|\widehat{p_1} - p_1| < \varepsilon] \;\&\&\; P[|\widehat{p_2} - p_2| < \varepsilon]$$

The two coins are the same, therefore,

$$P[|\widehat{p_1} - p_1| < \varepsilon] = P[|\widehat{p_2} - p_2| < \varepsilon]$$

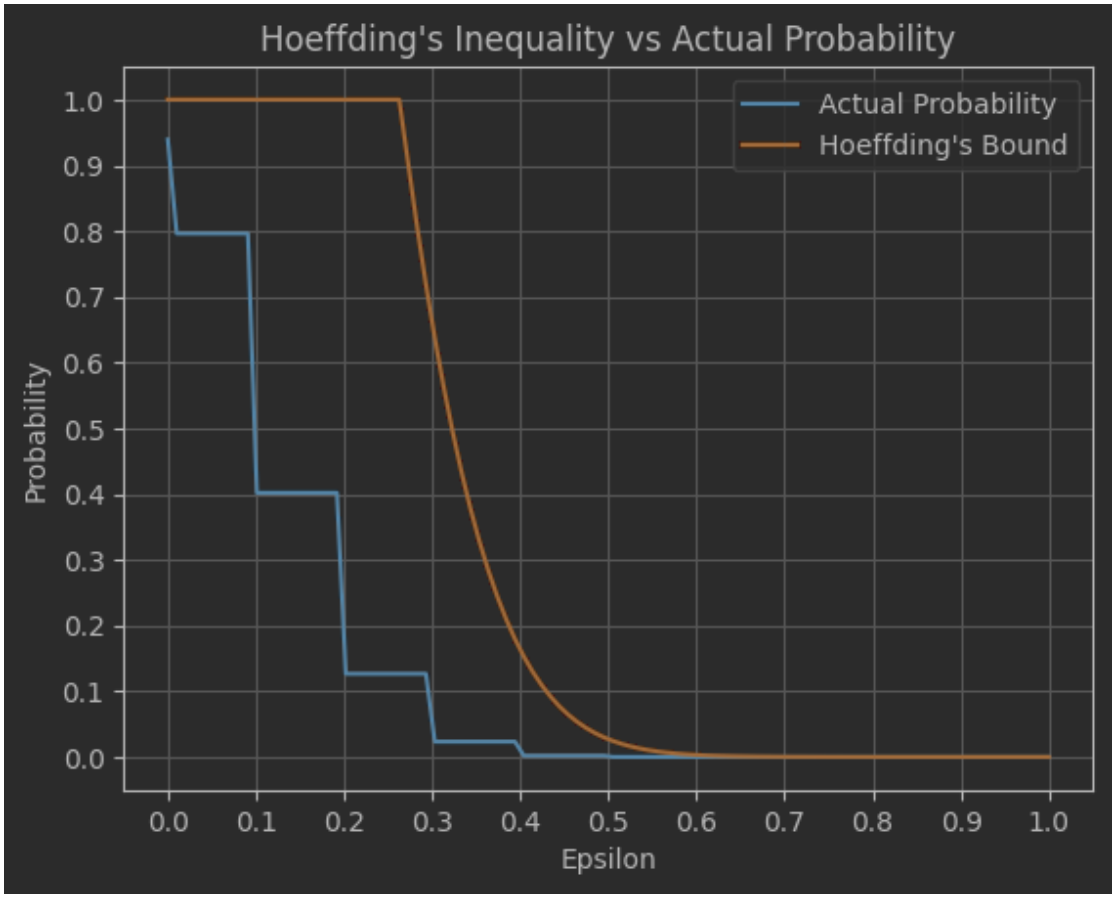
To calculate the probability,

$$P[|\widehat{p_1} - p_1| \leq \varepsilon] = P[|\widehat{np_1} - np_1| \leq n\varepsilon] = P[np_1 - n\varepsilon \leq \widehat{np_1} \leq np_1 + n\varepsilon]$$
$$= P[\widehat{np_1} \leq np_1 + n\varepsilon] - P[\widehat{np_1} \leq np_1 - n\varepsilon]$$
$$= F[np_1 + n\varepsilon] - F[np_1 - n\varepsilon]$$

Where $n = 10,\ p_1 = 0.5$,

$$P = [max_i|\widehat{p_i} - p_i| \geq \varepsilon] = 1 - (F[5 + 10\varepsilon] - F[5 - 10\varepsilon])^2$$

The specific calculation code is in the HW1.ipynb.



2. For X that is a discrete random variable with a probability mass function $g_k(x)$, prove that the expression given below minimizes the probability of error for a classifier with k classes. Note that we have examine this for the continuous case in Lecture 3.

$$f^*(x) = \arg\max_k \mu_k(x)$$
$$= \arg\max_k \pi_k \; g_k(x)$$

Consider an arbitrary classifier f and denote the decision regions

$$\gamma_k(f) = \{x: f(x) = k\}$$

To minimize the probability of error for a classifier with k classes,

$$1 - R(f) = P[f(X) = Y]$$

$$= \sum_{K=0}^{K-1} \pi_k P\left[f(X) = x_i | Y = k\right]$$

$$= \sum_{K=0}^{K-1} \pi_k \sum_{\{i: f(x_i)=k\}} P[f(X) = x_i | Y = k]$$

To maximize this, we should select f such that

$$x \epsilon \gamma_k(f) \leftrightarrow \pi_k g_k(x) \; is \; maximized$$

Therefore, the optimal

$$f^*(x) = arg_k max \mu_k(x)$$

Or equivalently

$$f^*(x) = arg_k max \pi_k g_k(x)$$

3. Suppose we have a scalar input x and we are considering a binary classification problem with the following class conditional probabilities

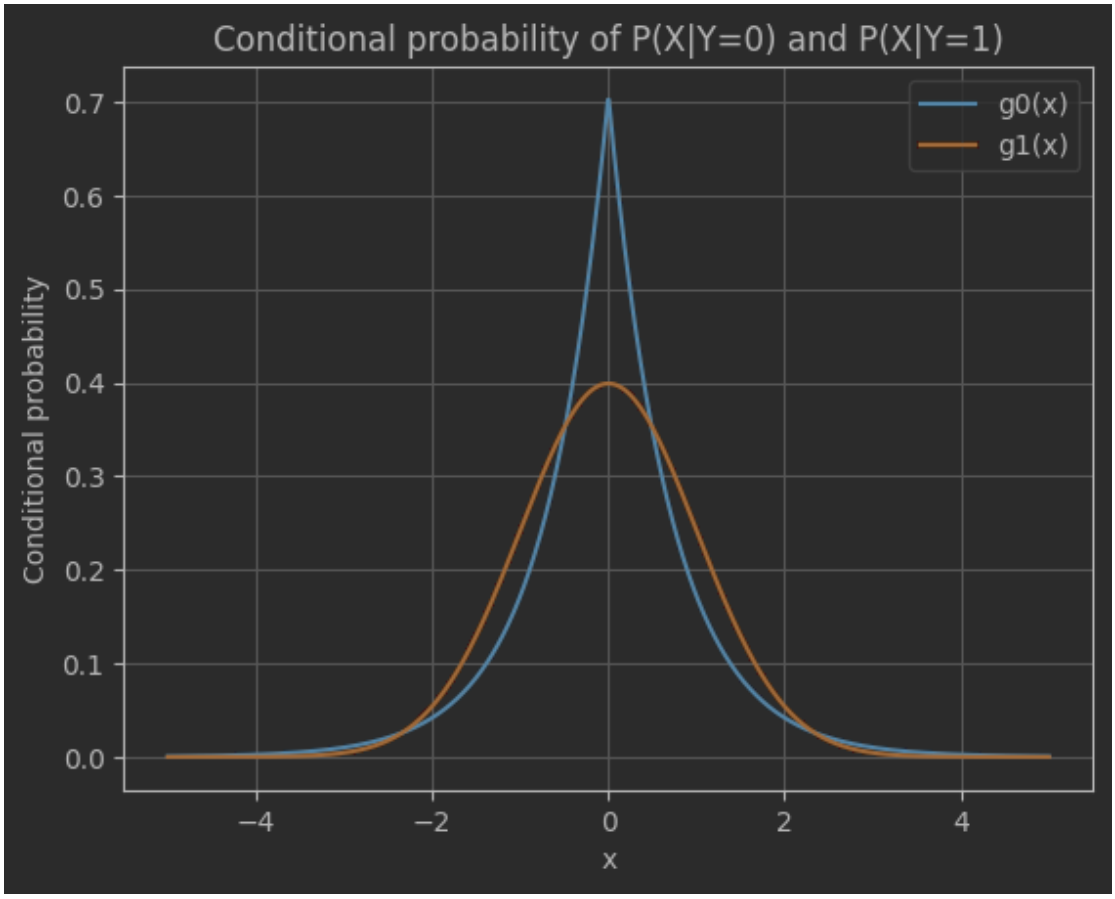
$$X|Y = 0 \sim g_0(x) = \frac{1}{\sqrt{2}} e^{-\sqrt{2}\,|x|} \quad \text{and}$$
$$X|Y = 1 \sim g_1(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

a. Sketch $g_0(x)$ and $g_1(x)$ overlayed in one figure.

b. Let the class prior probabilities be equal, e.g., $\pi_0 = \pi_1$. Express the optimum classification rule in terms of minimizing the risk or the probability of error). How would you relate this rule to your sketch of class conditional probabilities in part (a)?

c. Calculate the Bayes risk for the classification rule you derived in part (b).

3a. The specific calculation code is in the HW1.ipynb



3b.

The prior probabilities are equal, which is,

$$P(Y=0)=P(Y=1)=0.5$$

The optimal classification rule,

$$1-R(f)=\left\{\pi_0 g_0(x)\underset{<_1}{\overset{>}{}}^{0}\pi_1 g_1(x)\right\}$$

$$=\left\{0.5\frac{1}{\sqrt{2}}e^{-\sqrt{2}|x|}\underset{<_1}{\overset{>}{}}^{0}0.5\frac{1}{\sqrt{2\pi}}e^{-x^2/2}\right\}$$

$$=\left\{g_0(x)\underset{<_1}{\overset{>}{}}^{0}g_1(x)\right\}$$

If $g_0(x)$ is greater than $g_1(x)$, then the class would be $k=0$. Otherwise, the class would be $k=1$.

In the sketch when $g_0(x)$ is on the top then the class would be $k=0$. Otherwise, the class would be $k=1$.

3c.

$$g_0(x)=g_1(x)$$

$$0.5\frac{1}{\sqrt{2}}e^{-\sqrt{2}|x|}=0.5\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$$

$$\sqrt{\pi}e^{-\sqrt{2}|x|}=e^{-x^2/2}$$

$$0.5log\pi - \sqrt{2}|x| = -\frac{x^2}{2}$$

We get,

$$x_1 = 0.489$$
$$x_2 = 2.339$$

Therefore, the decision region is,

$$\gamma_0(f^*) = (-\infty, -x_2) \cup (-x_1, x_1) \cup (x_2, \infty)$$
$$\gamma_1(f^*) = (-x_2, -x_1) \cup (x_1, x_2)$$

Then,

$$1 - R(f^*) = \int_{\gamma_1} g_1(x)dx + \int_{\gamma_0} g_0(x)dx$$
$$R(f^*) = 1 - \left(\int_{\gamma_1} g_1(x)dx + \int_{\gamma_0} g_0(x)dx\right)$$
$$= 1 - \left(\int_{\gamma_1} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}dx + \int_{\gamma_0} \frac{1}{\sqrt{2}} e^{-\sqrt{2}|x|}dx\right)$$

4. In our second lecture, we discuss LDA and obtain a linear classifier of the form:

$$\hat{f}(x) = \begin{cases} 1 & if\ a^T x + b \geq 0 \\ 0 & if\ o.w. \end{cases}$$

Please provide a derivation for a and b in this decision rule.

In the linear classifier we have,

$$d_M{}^2(x: \hat{\mu}_0, \hat{\Sigma}) - 2\log \hat{\pi}_0 \overset{0}{\lesseqgtr} d_M{}^2(x: \hat{\mu}_1, \hat{\Sigma}) - 2\log \hat{\pi}_1$$

Where the Mahalanobis distance is,

$$d_M(x: \mu, \Sigma) = \sqrt{(x - \hat{\mu}_k)^T \hat{\Sigma}^{-1} (x - \hat{\mu}_k)}$$

Substitute and move terms, we get,

$$\hat{f}(x) = (x - \hat{\mu}_1)^T \hat{\Sigma}^{-1} (x - \hat{\mu}_1) - 2\log \hat{\pi}_1 - (x - \hat{\mu}_0)^T \hat{\Sigma}^{-1} (x - \hat{\mu}_0) + 2\log \hat{\pi}_0$$

$$= \left(x^T \hat{\Sigma}^{-1} x - \hat{\mu}_1{}^T \hat{\Sigma}^{-1} x - x\hat{\Sigma}^{-1} \hat{\mu}_1 + \hat{\mu}_1{}^T \hat{\Sigma}^{-1} \hat{\mu}_1\right) - \left(x^T \hat{\Sigma}^{-1} x - \hat{\mu}_0{}^T \hat{\Sigma}^{-1} x - x\hat{\Sigma}^{-1} \hat{\mu}_0 + \hat{\mu}_0{}^T \hat{\Sigma}^{-1} \hat{\mu}_0\right)$$
$$+ \left(2\log \hat{\pi}_0 - 2\log \hat{\pi}_1\right)$$

$$= \left(-\hat{\mu}_1{}^T \hat{\Sigma}^{-1} x - x\hat{\Sigma}^{-1} \hat{\mu}_1 + \hat{\mu}_1{}^T \hat{\Sigma}^{-1} \hat{\mu}_1\right) - \left(-\hat{\mu}_0{}^T \hat{\Sigma}^{-1} x - x\hat{\Sigma}^{-1} \hat{\mu}_0 + \hat{\mu}_0{}^T \hat{\Sigma}^{-1} \hat{\mu}_0\right) + 2\log \frac{\hat{\pi}_0}{\hat{\pi}_1}$$

$$= \left(\hat{\mu}_0{}^T \hat{\Sigma}^{-1} x - \hat{\mu}_1{}^T \hat{\Sigma}^{-1} x + x\hat{\Sigma}^{-1} \hat{\mu}_0 - x\hat{\Sigma}^{-1} \hat{\mu}_1\right) - \hat{\mu}_0{}^T \hat{\Sigma}^{-1} \hat{\mu}_0 + \hat{\mu}_1{}^T \hat{\Sigma}^{-1} \hat{\mu}_1 + 2\log \frac{\hat{\pi}_0}{\hat{\pi}_1}$$

Notice that the dimensions of $x\hat{\Sigma}^{-1}\hat{\mu}_0$ is,

$$(1 \times d) * (d \times d) * (d \times 1) = 1 \times 1$$

Which is a single value, therefore,

$$\hat{x}\hat{\Sigma}^{-1}\hat{\mu}_0 = \left(\hat{x}\hat{\Sigma}^{-1}\hat{\mu}_0\right)^T = \hat{\mu}_0^T\hat{\Sigma}^{-1}x$$

$$\hat{x}\hat{\Sigma}^{-1}\hat{\mu}_1 = \left(\hat{x}\hat{\Sigma}^{-1}\hat{\mu}_1\right)^T = \hat{\mu}_1^T\hat{\Sigma}^{-1}x$$

Substitute and normalized, we get,

$$\hat{f}(x) = \left[\hat{\Sigma}^{-1}(\hat{\mu}_0 - \hat{\mu}_1)\right]^T x + \left(-\frac{1}{2}\hat{\mu}_0^T\hat{\Sigma}^{-1}\hat{\mu}_0 + \frac{1}{2}\hat{\mu}_1^T\hat{\Sigma}^{-1}\hat{\mu}_1 + \log\frac{\hat{\pi}_0}{\hat{\pi}_1}\right)$$

$$= a^T x + b$$

$$\begin{cases} a = \hat{\Sigma}^{-1}(\hat{\mu}_0 - \hat{\mu}_1) \\ b = -\frac{1}{2}\hat{\mu}_0^T\hat{\Sigma}^{-1}\hat{\mu}_0 + \frac{1}{2}\hat{\mu}_1^T\hat{\Sigma}^{-1}\hat{\mu}_1 + \log\frac{\hat{\pi}_0}{\hat{\pi}_1} \end{cases}$$

5. For this problem I would like you to implement LDA and a slight variation of it. Then you will test on some data provided.

   a. Implement LDA in MATLAB or Python using the lecture 4 notes. Please come up with your version of LDA as it is simple enough to implement.

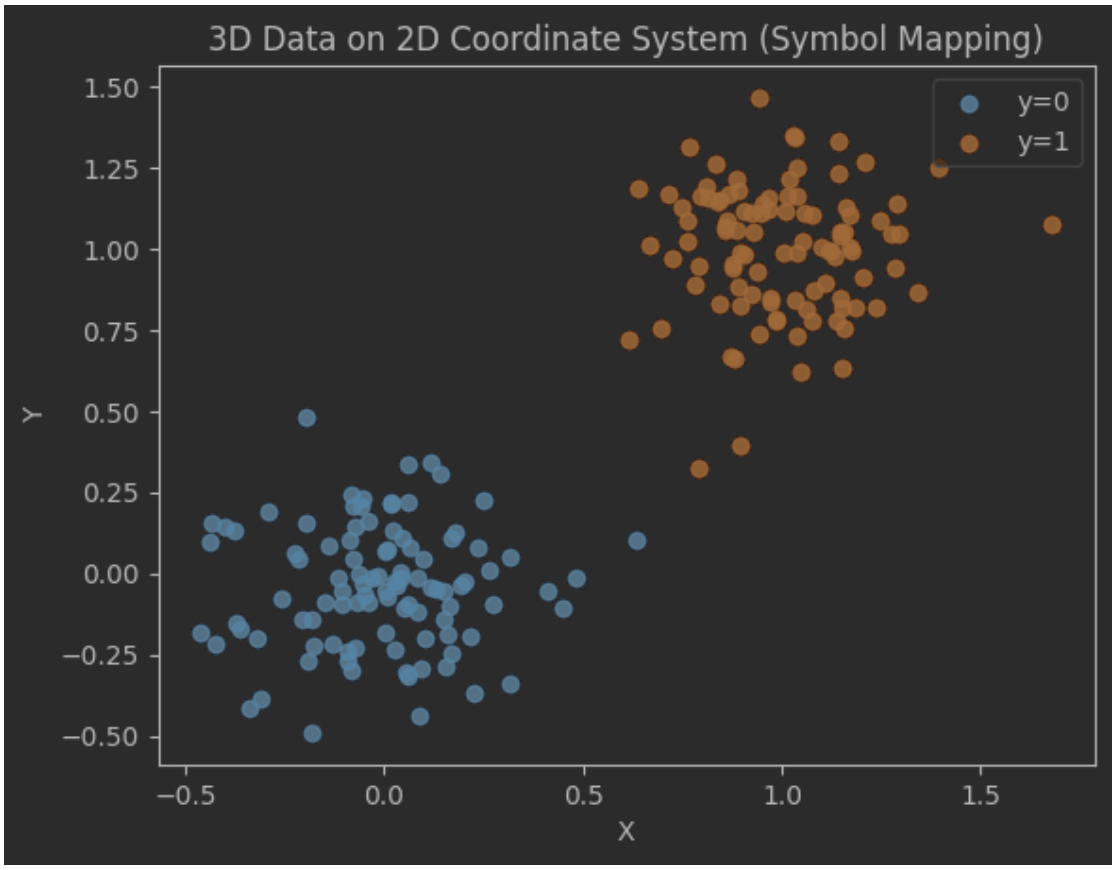
   b. Now implement another version of your algorithm by letting $\Sigma = \sigma^2 I$, here you can use the same covariance estimate you used in part(a) for LDA and replace it with $\frac{1}{d}trace(\hat{\Sigma})I$.

5a, 5b The specific algorithm code is in the HW1.ipynb

   c. Run your algorithms on the synthetic data sets. You will find four different synthetic data sets posted on Canvas, and compare and commend on the performance of both algorithms on all of these sets.

The validation and data visualization code are in HW1.ipynb

Synthetic1:

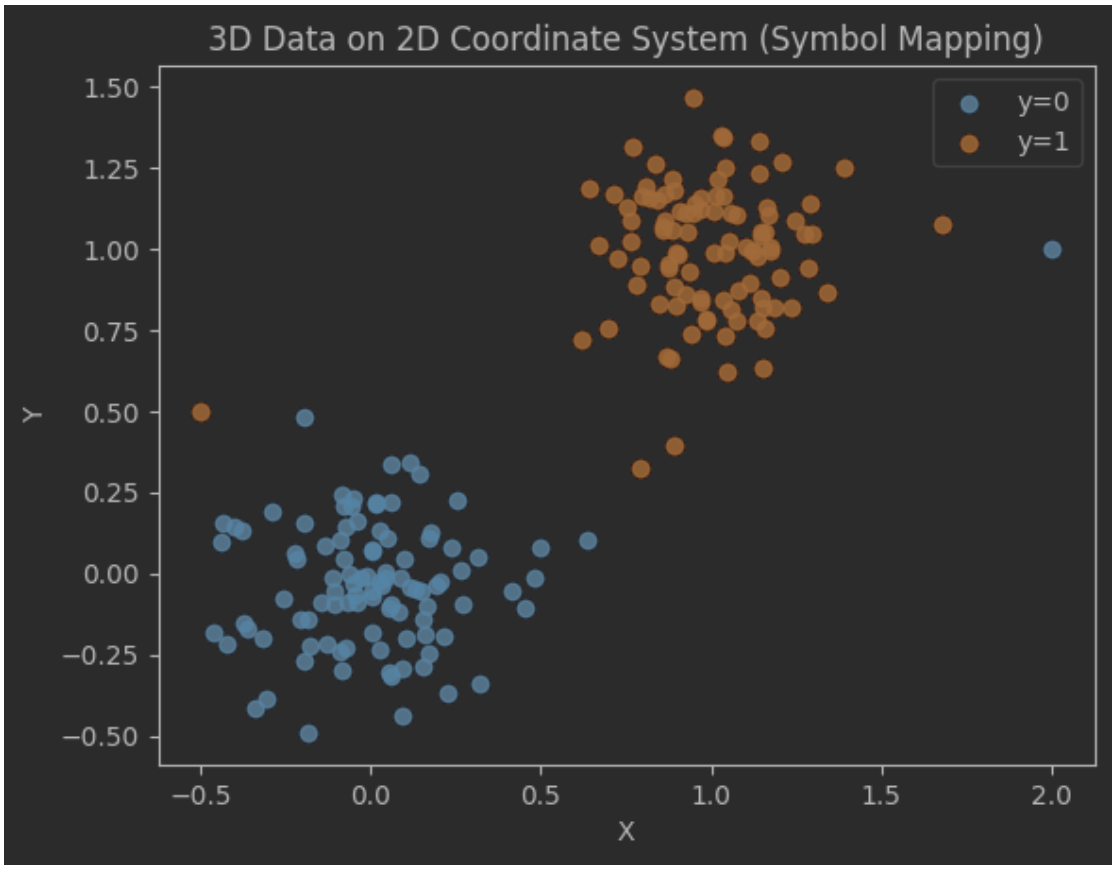



```
synthetic1
a =  [25.79723763 27.17261469]
b =  -25.78410886820642
sigma =  [[0.03780866 0.00115064]
 [0.00115064 0.0371704 ]]
sigma_simple =  [[0.03748953 0.        ]
 [0.         0.03748953]]
risk: 0.0 ,risk_simple: 0.0
```

Discussion:

In Synthetic1, there is basically no outlier in the data, and the similar points of the two classifications are relatively concentrated, and the distance between the central areas is very large. Both classifiers completed the classification perfectly, with a risk of 0.

Synthetic2:

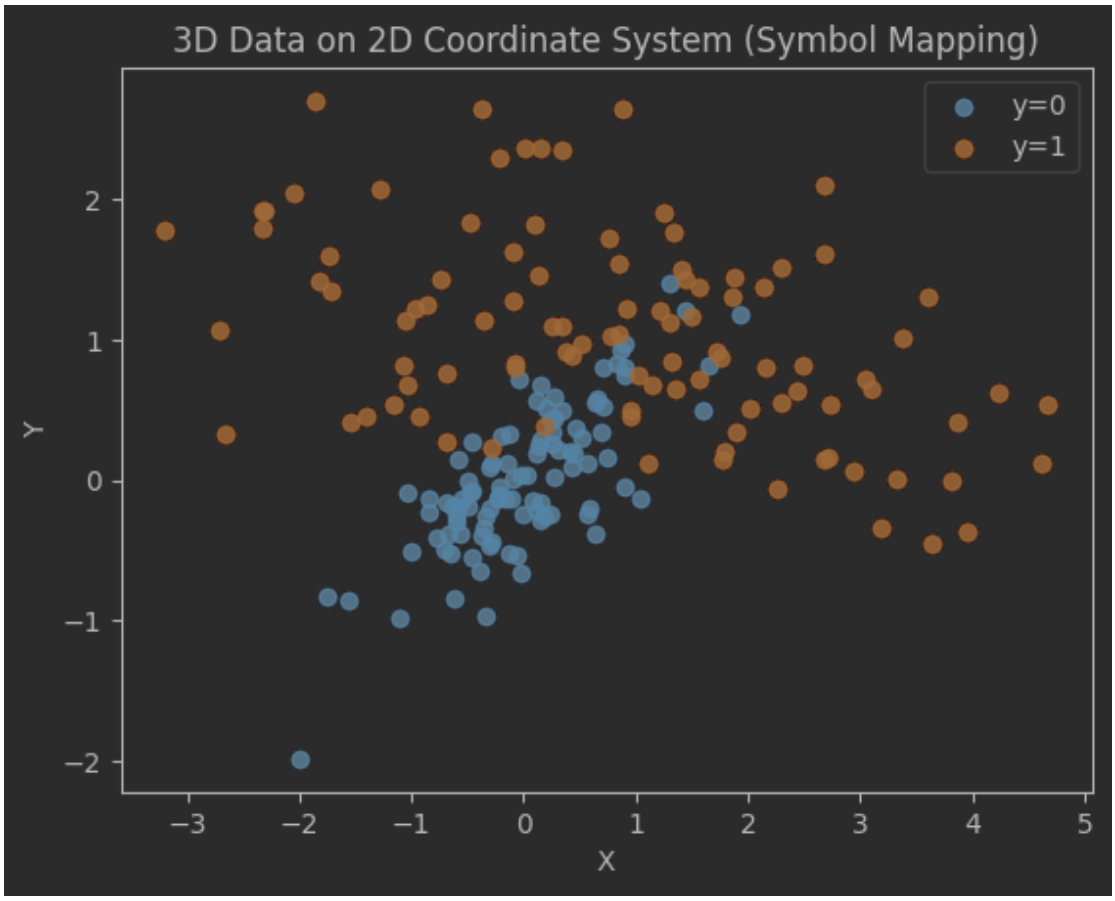



```
synthetic2
a =  [ 9.39193027 20.19938674]
b =  -14.453993274326724
sigma =  [[0.06990821 0.01541904]
 [0.01541904 0.04335511]]
sigma_simple =  [[0.05663166 0.        ]
 [0.         0.05663166]]
risk: 0.015 ,risk_simple: 0.01
```

Discussion:

In Synthetic2, it is similar to Synthetic1, but with a small number of outliers. Both classifiers have a small risk, and the risk of the simple classifier is slightly smaller.

Synthetic3:

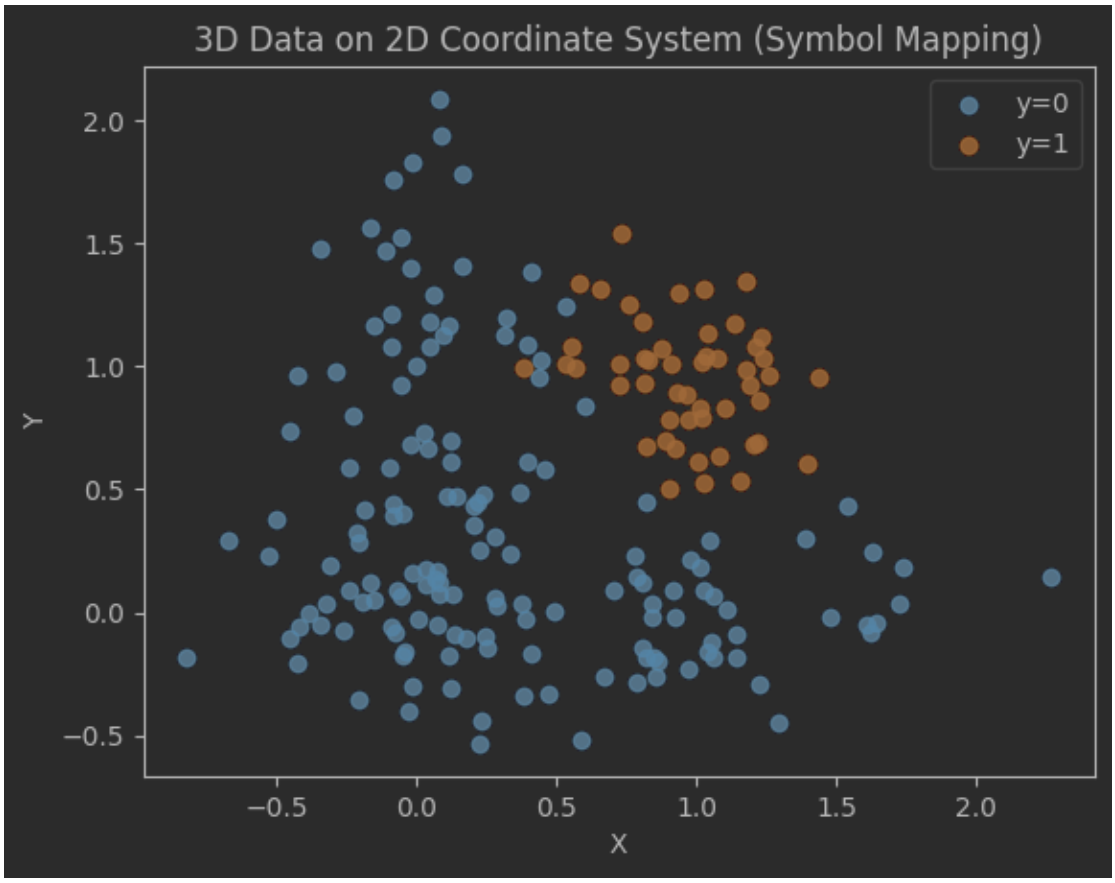



```
synthetic3
a =  [0.67001069 2.89413401]
b =  -1.800296710549185
sigma =  [[ 1.87364691 -0.14718935]
 [-0.14718935  0.3848668 ]]
sigma_simple =  [[1.12925686 0.        ]
 [0.         1.12925686]]
risk: 0.18 ,risk_simple: 0.21
```

Discussion:

In Synthetic3, the two types of data are mixed together and cannot be separated well, so the risk of the classifier is much higher. The risk of simple classifiers is higher than that of ordinary classifiers.

Synthetic4:




```
synthetic4
a =  [3.43030007 3.29444822]
b =  -3.2297169527803327
sigma =  [[ 0.2572706  -0.07174716]
 [-0.07174716  0.26100942]]
sigma_simple =  [[0.25914001 0.        ]
 [0.         0.25914001]]
risk: 0.26 ,risk_simple: 0.315
```

Discussion:

In Synthetic4, the two types of data are mixed to the highest degree, so its risk is the highest. The risk of simple classifiers is higher than that of ordinary classifiers.

In general, simple classifiers perform better when there are fewer outliers in the data; when the degree of data mixing is high, the risk of simple classifiers is higher, but not much worse, and the calculation of simple classifiers The amount should be less. The reason may be that the simple classifier only takes the trace of the deviation matrix, so its classification is more general and its ability to resist noise is stronger.

d. Apply both algorithms on "trainLDA.mat" to train your LDA classifier and test it on "testLDA.mat". Show the testing error you get with both of your classifiers. Do not use the test data in training.

Training:

```
a =  [ 0.00220269  0.09932374  0.15834145  0.00925714  1.17087421  0.02522099
 -0.04957705  0.00733097  0.03326467]
b =  -3.356989690285877
risk: 0.35064935064935066 ,risk_simple: 0.354978354978355
```

Testing:

```
n =  231
testLDA
risk: 0.37662337662337664 ,risk_simple: 0.42857142857142855
```