Group Members

Jianshen Liu (jliu120)
Sumukh Atreya (satreya)

CMPS 242 Homework 2

All the code for this homework is available at https://github.com/ljishen/ml-expt/blob/master/hw2/figure.ipynb

Question 1

1) From

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

, we get

$$\mu_{ML} = \frac{1}{5}(-1 + 1 + 10 - 0.5 + 0) = 1.9$$

.

From

$$\sigma^2_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{ML})^2$$

, we get

$$\sigma^2_{ML} = \frac{1}{5}\left[(-1 - 1.9)^2 + (1 - 1.9)^2 + (10 - 1.9)^2 + (-0.5 - 1.9)^2 + (0 - 1.9)^2\right]$$

$$= \frac{1}{5} \cdot 84.2 = 16.84$$

Because

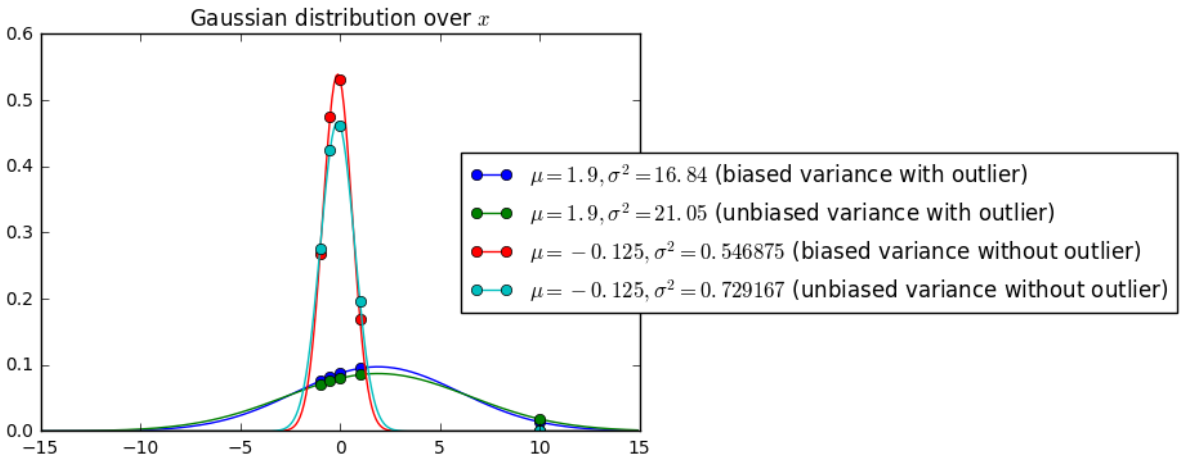
$\mathbb{E}\left[\sigma^2_{ML}\right] = (\frac{N-1}{N})\sigma^2$ and the unbiased variance should be $\mathbb{E}\left[\widetilde{\sigma}^2\right] = \sigma^2$. Therefore,

$$\widetilde{\sigma}^2 = \frac{N}{N-1}\sigma^2_{ML} = \frac{5}{4} \cdot 16.84 = 21.05$$

.

In this given dataset, if we remove the outlier data point 10, we get
$\mu_{ML} = -0.125$, $\sigma^2_{ML} = 0.546875$, $\widetilde{\sigma}^2 = 0.729167$.

Here is the Gaussian distribution curves for these four cases,



We see that the outlier data point affects both the mean and variance, but the impact on variance is greater than on the mean. Specifically, the presence of an outlier makes the variance become much larger. On the other hand, the unbiased variance is slightly larger than the biased one.

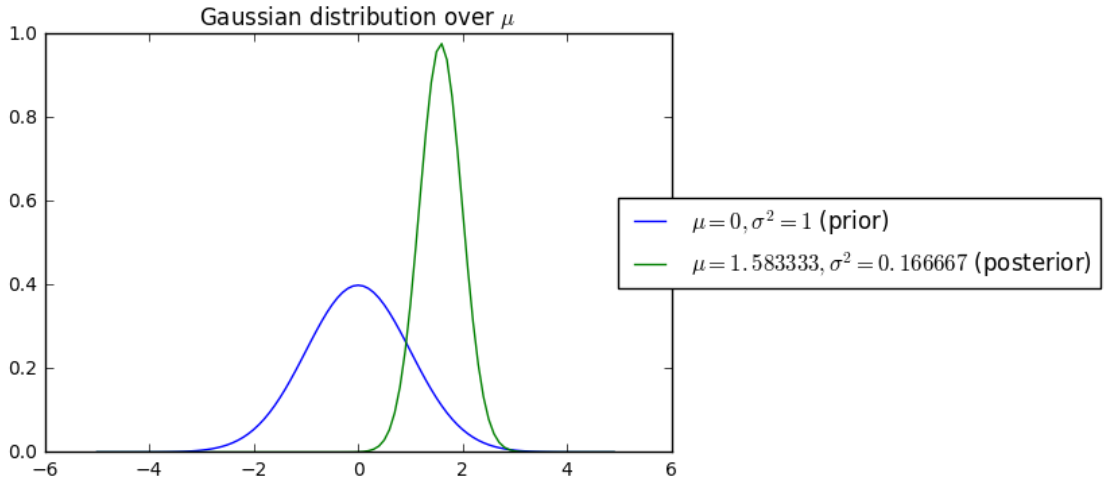
2) From

$\sigma_N^2 = \frac{\sigma_0^2 \sigma_2}{N\sigma_0^2 + \sigma^2}$, $\mu_N = (\frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2})\overline{\mu}_N + (\frac{\sigma^2}{N\sigma_0^2 + \sigma^2})\mu_0$, $\overline{\mu}_N = \frac{1}{N}\sum_{n=1}^{N} x_n$ and

$\sigma^2 = 1$, $\sigma_0^2 = 1$, we get

$\sigma_N^2 = 0.1666667$ and $\mu_N = 1.583333$.

Here is the curves for both the prior and the posterior distributions,



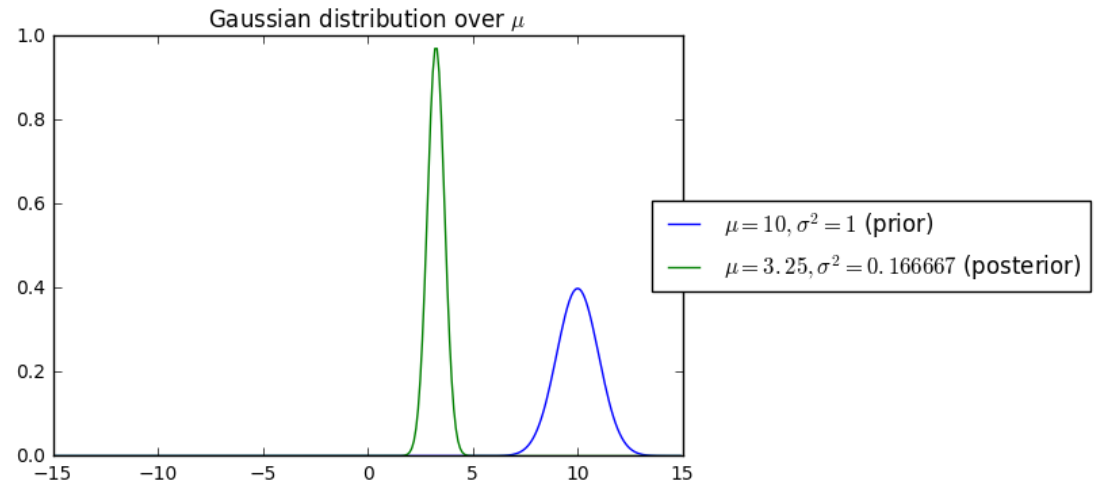
The prior distribution over the mean is a standard normal distribution, after some the data comes in, the variance of the distribution over the mean shrinks, and the mean of the distribution over the mean moves a little to the right, which is closer to the sample mean ($\overline{\mu}_N = 1.9$).

3) This time we have

$\mu_0 = 10$ and $\sigma_0^2 = 1$. By following the calculation formulas above, we get

$\sigma_N^2 = 0.1666667$ and $\mu_N = 3.25$.

Here is the curves for both the prior and the posterior distributions,



This time we see the mean of the distribution over the mean moves a lot to the left after data comes in, still making the mean of the distribution over the mean closer to the sample mean($\overline{\mu}_N = 1.9$).

4) Now the dataset becomes

$$X = \{-1, 1, 10, -0.5, 0, 2, 0.5\}$$

We can compute

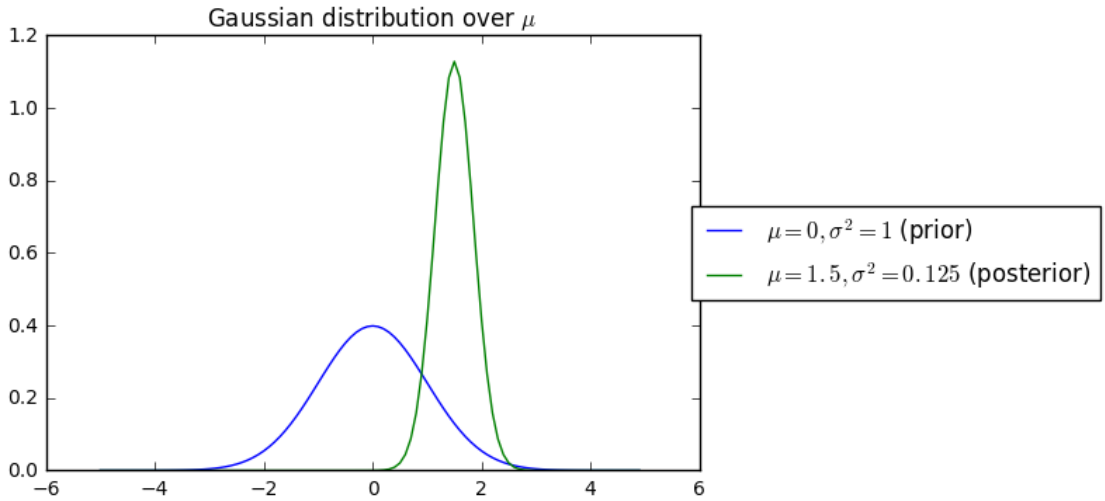
$$\overline{\mu}_N = \frac{1}{7}(-1 + 1 + 10 - 0.5 + 0 + 2 + 0.5) = 1.714286$$

.

By following the calculation formulas above, we get

$\sigma_N^2 = 0.125$ and $\mu_N = 1.5$.

Here is the curves for both the prior and the posterior distributions,



Comparing to the question 2), because of more data comes in, the variance of the distribution over the mean reduced. Also, the the mean of the distribution over the mean reduced the impact from the one outlier, and now it is closer to the sample mean after the adjustment with the help of data.

Question 2

1) MLE estimation of $\mathbf{w}$ is given by **minimizing** this *sum-of-squares-error* function,

$$\arg\min_{\mathbf{w}} E(\mathbf{w}) = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

,

and β_{ML} is given by

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}_{ML} - t_n)\}^2$$

.

Here is our simulation program,

```
from scipy.optimize import minimize
import random
import numpy as np

NUM_POINTS = 100

# generate x
xs = []
for _ in range(NUM_POINTS):
    xs.append(random.randint(-100, 100))

# generate t
def fx(x):
    return 0.1 + (2 * x) + (x**2) + (3 * (x**3))

ts = []
for i in range(NUM_POINTS):
```

```
    t = np.random.normal(fx(xs[i]), 1, 1)
    ts.append(t[0])

# The first polynomial model
def yxw_1(w, x):
    return w[0] + w[1] * x + w[2] * x**2 + w[3] * x**3

# The second polynomial model
def yxw_2(w, x):
    return w[0] + w[1] * x + w[2] * x**2 + w[3] * x**3 + w[4] * x**4 + w[5] * x**5

# sum-of-squares-error function
def sose(w, xs, ts):
    summ = 0
    for i in range(NUM_POINTS):

        # choose yxw_1 or yxw_2 here as a different polynomial model
        summ += (yxw_1(w, xs[i]) - ts[i])**2
    return 1.0 / 2 * summ

# Initial guess of w.
# Choose w0_1 when you use yxw_1 as the polynomial model
# in the sum-of-squares-error function.
# Correspondingly, choose wo_2 when you use yxw_2.
w0_1 = [1, 1, 1, 1]
w0_2 = [1, 1, 1, 1, 1, 1]

res = minimize(sose, w0_1, args=(xs,ts), method='Powell')

if (res.success):
    w_ml = res.x
    print('w_ml = ' + str(w_ml))

    beta_ml = 1.0 / (sose(w_ml, xs, ts) * 2 / NUM_POINTS)
    print('beta_ml = ' + str(beta_ml))
```

Here is the example outputs from this program with the order of the polynomial model is 3

- when the number of generated data points is **100**,
  $\mathbf{w}_{ML} = \begin{bmatrix} 0.21956938 & 1.99305728 & 0.9999889 & 3.00000119 \end{bmatrix}$
  $\beta_{ML} = 1.19304962418$
- when the number of generated data points is **10,000**,
  $\mathbf{w}_{ML} = \begin{bmatrix} 0.10956622 & 2.00028188 & 0.99999992 & 2.99999998 \end{bmatrix}$
  $\beta_{ML} = 1.0186496674$

Comparison to the true parameters:
The true parameters are:
$\mathbf{w}_T = \begin{bmatrix} 0.1 & 2 & 1 & 3 \end{bmatrix}$
$\beta_T = 1$

We ran this program several times and found that most of the time w_1, w_2 and w_3 are very close to the true parameter in $f(x)$, while w_0 and β still have a little distance. As we increase the number of generated data points, we see that the both $\mathbf{w}_{ML}$ and β_{ML} become closer to the true parameters used to generate the data.

2) Here is one example output from our program after we increased the order of the polynomial model to 5,
$\mathbf{w}_{ML} = \begin{bmatrix} 1.003e-01 & 2.000e+00 & 9.999e-01 & 2.999e+00 & 7.886e-10 & -6.484e-12 \end{bmatrix}$
$\beta_{ML} = 1.00303475021$

We notice that even though we set the number of generated data point to 10,000, the result is still sometimes worse than when we use the polynomial of order of 3 and only use 100 sample data points, which implies that the data has been over-fitted.

3) The posterior predictive distribution $p(t|x, \mathbf{x}, \mathbf{t})$ is given by a Gaussian of the form,

$$p(t|x, \mathbf{x}, \mathbf{t}) = \mathcal{N}(t|m(x), s^2(x))$$

where the mean and variance are given by

$$m(x) = \beta \boldsymbol{\phi}(x)^T \mathbf{S} \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) t_n$$

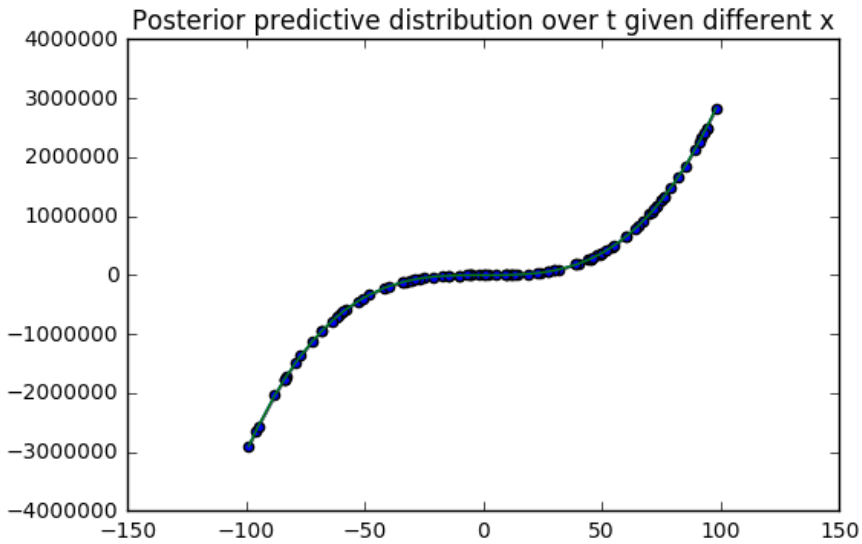
$$s^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^T \mathbf{S} \boldsymbol{\phi}(x).$$

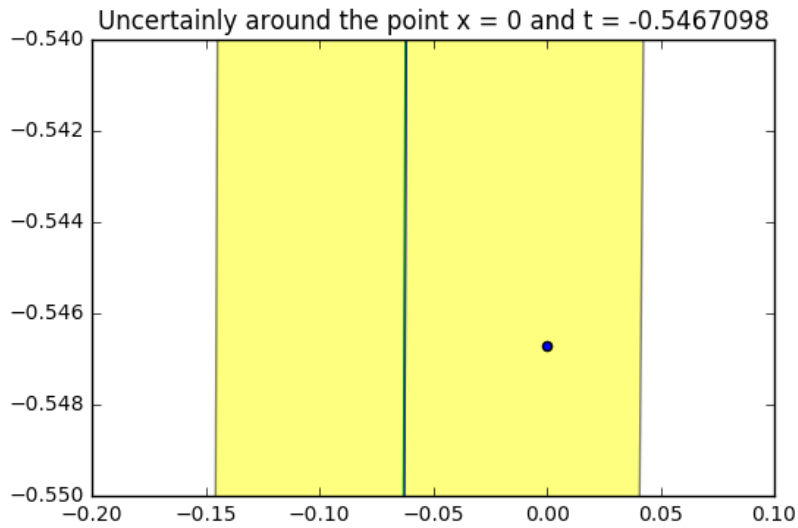
Here the matrix $\mathbf{S}$ is given by

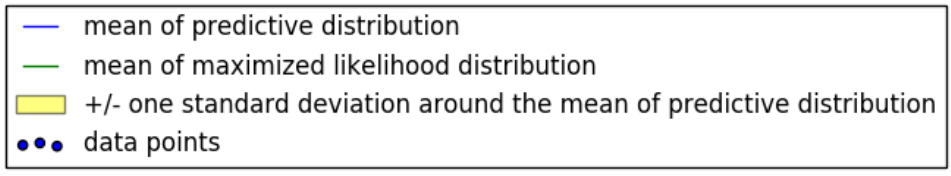
$$\mathbf{S}^{-1} = \alpha \mathbf{I} + \beta \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) \boldsymbol{\phi}(x_n)^T$$

where $\mathbf{I}$ is the unit matrix, and we define the vector $\boldsymbol{\phi}(x)$ with elements $\phi_i(x) = x^i$ for $i = 0, ..., M$.

Here are two generated figures for posterior predictive distribution over t given different x, when we use 100 data points and set $\alpha = 1$, the order of polynomial model $M = 3$







Since the range of t is too large to view the detail uncertainty around the estimate, in the second figure, we zoom in to the point where $x = 0$ and $t = -0.5467098$. We see that the mean of predictive distribution is very close to the mean of maximum likelihood distribution with respect to this point.

We also made statistic results for

- ($\mu_{\mathbf{m}}$) The average **Euclidean distance** between the mean of the distribution used to generate data and the mean of the predictive distribution.
  - $\mu_{\mathbf{m}} = 0.235173$, when the number of data points is **100**
  - $\mu_{\mathbf{m}} = 0.039126$, when the number of data points is **1000**
  - $\mu_{\mathbf{m}} = 0.011207$, when the number of data points is **10,000**
- ($\sigma_{\mathbf{m}}$)The average **Euclidean distance** between the variance of the distribution used to generate data and the variance of the predictive distribution.
  - $\sigma_{\mathbf{m}} = 0.042530$, when the number of data points is **100**
  - $\sigma_{\mathbf{m}} = 0.070124$, when the number of data points is **1000**
  - $\sigma_{\mathbf{m}} = 0.007151$, when the number of data points is **10,000**

We see that as the number of data points increases, the mean and variance of the predictive distribution are both closer to the ones used to generated the data, respectively.

4) When we set $\alpha = 100$, as the point density is too high to see the difference, the plotted figure looks pretty much the same. We also made statistic results for both $\mu_{\mathbf{m}}$ and $\sigma_{\mathbf{m}}$ as we did for the previous question,

- $\mu_{\mathbf{m}} = 0.084659$, when the number of data points is **100**
- $\mu_{\mathbf{m}} = 0.048880$, when the number of data points is **1000**
- $\mu_{\mathbf{m}} = 0.003427$, when the number of data points is **10,000**

- $\sigma_{\mathbf{m}} = 0.115917$, when the number of data points is **100**
- $\sigma_{\mathbf{m}} = 0.004181$, when the number of data points is **1000**
- $\sigma_{\mathbf{m}} = 0.007010$, when the number of data points is **10,000**

We see that both mean the variance are becoming more and more accurate as the number of data points increases, just as we saw in the previous question. Furthermore, the change of α helps to reduce $\mu_{\mathbf{m}}$. In other words, the increase of α increased the accuracy of the mean of the predictive distribution in this case.

Question 3

1) Since the log likelihood function can be written in the form

$$\ln p(\mathbf{x}|\mathcal{C}_k) = -\frac{1}{2\sigma_k^2}\sum_{i=1}^{D}(x_i - \mu_{ki})^2 - \frac{D}{2}\ln\sigma_k^2 - \frac{D}{2}\ln 2\pi$$

Maximizing this equation with respect to μ_{ki}, we obtain the maximum likelihood solution given by $\frac{1}{\sigma_k^2}\sum_{i=1}^{D}(x_i - \mu_{ki}) = 0$. Upon rearranging, we finally get $\mu_{ikML} = \frac{1}{D}\sum_{i=1}^{D} x_i$.

2) With condition of the number of class is 2 and Naive Bayes assumption, the decision rule for choosing the class $\mathcal{C}_1$ is

$$\ln p(\mathbf{x}|\mathcal{C}_1) + \ln p(\mathcal{C}_1) \geq \ln p(\mathbf{x}|\mathcal{C}_2) + \ln p(\mathcal{C}_2)$$

after replacement, we have

$$D\ln\frac{1}{(2\pi\sigma_1^2)^{1/2}} - \sum_{i=1}^{D}\frac{(x_i - \mu_{1i})^2}{2\sigma_1^2} + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \geq D\ln\frac{1}{(2\pi\sigma_2^2)^{1/2}} - \sum_{i=1}^{D}\frac{(x_i - \mu_{2i})^2}{2\sigma_2^2}$$

Then we finally get

$$\frac{1}{2}\left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2}\right)\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i\left(\frac{\mu_{1i}}{\sigma_1^2} - \frac{\mu_{2i}}{\sigma_2^2}\right) + b \geq 0$$

where $$b = -\frac{1}{2}\sum_{i=1}^{D}\left(\frac{\mu_{1i}^2}{\sigma_1^2} - \frac{\mu_{2i}^2}{\sigma_2^2}\right) + D\ln\frac{\sigma_2}{\sigma_1} + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

3) If the cost of misclassifying class $\mathcal{C}_1$ is 10 times of the cost of misclassifying class $\mathcal{C}_2$, then the only reason to choose $\mathcal{C}_1$ is the potential cost of misclassifying of $\mathcal{C}_1$ is not more than the potential cost of misclassifying of $\mathcal{C}_2$. Let's say the cost of misclassifying of $\mathcal{C}_1$ is λ_{12}, and the cost of misclassifying of $\mathcal{C}_2$ is λ_{21}, and $\lambda_{12} = 10\lambda_{21}$, then we have

$$(1 - p(\mathcal{C}_1|\mathbf{x}))\,\lambda_{12} \leq (1 - p(\mathcal{C}_2|\mathbf{x}))\,\lambda_{21}$$

Since there are only two classes, we know

$$1 - p(\mathcal{C}_1|\mathbf{x}) = p(\mathcal{C}_2|\mathbf{x}) \text{ and } 1 - p(\mathcal{C}_2|\mathbf{x}) = p(\mathcal{C}_1|\mathbf{x})$$

Therefore the above decision rule becomes

$$p(\mathcal{C}_2|\mathbf{x})\,\lambda_{12} \leq p(\mathcal{C}_1|\mathbf{x})\,\lambda_{21}$$

We can also write it as

$$10p(\mathcal{C}_2|\mathbf{x}) \leq p(\mathcal{C}_1|\mathbf{x})$$

It means the only reason to classify $\mathbf{x}$ as $\mathcal{C}_1$ is the probability of it is at least 10 times of the probability for the other classification option, because we may incur 10 times of penalty in case of misclassification.

By using Bayes' theorem, we get

$$p(\mathbf{x}|\mathcal{C}_1)\,p(\mathcal{C}_1) \geq 10p(\mathbf{x}|\mathcal{C}_2)\,p(\mathcal{C}_2)$$

We get the following inequation by applying logarithm on both side,

$$\ln p(\mathbf{x}|\mathcal{C}_1) + \ln p(\mathcal{C}_1) \geq \ln p(\mathbf{x}|\mathcal{C}_2) + \ln p(\mathcal{C}_2) + \ln 10$$

Then we finally get decision rule for classify $\mathbf{x}$ as $\mathcal{C}_1$ is

$$\frac{1}{2}\left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2}\right)\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i\left(\frac{\mu_{1i}}{\sigma_1^2} - \frac{\mu_{2i}}{\sigma_2^2}\right) + b \geq 0$$

where $$b = -\frac{1}{2}\sum_{i=1}^{D}\left(\frac{\mu_{1i}^2}{\sigma_1^2} - \frac{\mu_{2i}^2}{\sigma_2^2}\right) + D\ln\frac{\sigma_2}{\sigma_1} + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} - \ln 10$$

4) If $k = 3$, the decision rule for classifying $\mathbf{x}$ as $\mathcal{C}_i$ would be,
for all $i,j \in \{1,2,3\}$, $j \neq i$,

$$p(\mathcal{C}_i|\mathbf{x}) \geq p(\mathcal{C}_j\mathbf{x})$$

Therefore, for example the decision rule for classifying $\mathbf{x}$ as $\mathcal{C}_1$ is

$$p(\mathcal{C}_1|\mathbf{x}) \geq p(\mathcal{C}_2\mathbf{x}) \text{ and } p(\mathcal{C}_1|\mathbf{x}) \geq p(\mathcal{C}_3\mathbf{x})$$

By using the result we have from the previous question, we get

a)
$$\frac{1}{2}\left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2}\right)\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i\left(\frac{\mu_{1i}}{\sigma_1^2} - \frac{\mu_{2i}}{\sigma_2^2}\right) + b_1 \geq 0$$

where $$b_1 = -\frac{1}{2}\sum_{i=1}^{D}\left(\frac{\mu_{1i}^2}{\sigma_1^2} - \frac{\mu_{2i}^2}{\sigma_2^2}\right) + D\ln\frac{\sigma_2}{\sigma_1} + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}$$

b)
$$\frac{1}{2}\left(\frac{1}{\sigma_3^2} - \frac{1}{\sigma_1^2}\right)\sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i\left(\frac{\mu_{1i}}{\sigma_1^2} - \frac{\mu_{3i}}{\sigma_3^2}\right) + b_2 \geq 0$$

where $$b_2 = -\frac{1}{2}\sum_{i=1}^{D}\left(\frac{\mu_{1i}^2}{\sigma_1^2} - \frac{\mu_{3i}^2}{\sigma_3^2}\right) + D\ln\frac{\sigma_3}{\sigma_1} + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_3)}$$

Thus, **a)** and **b)** together is the decision rule for classifying $\mathbf{x}$ as $\mathcal{C}_1$.