

EECE5644 - Assignment 2

Dhruv Agarwal

October 2024

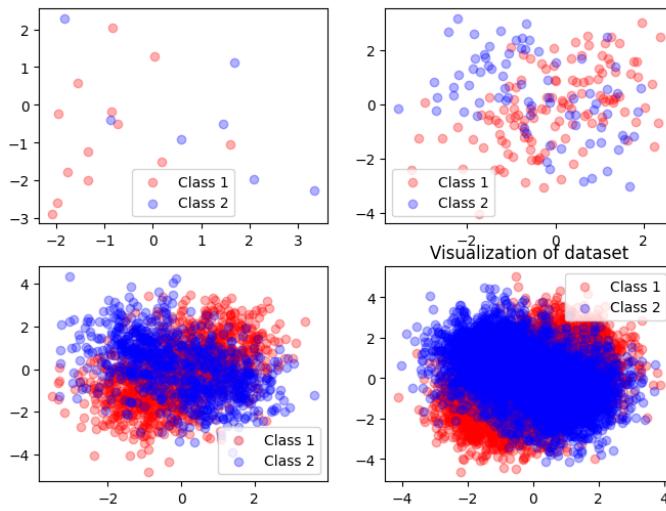
1 Question 1

X is a 2-dimensional real-valued random vector with probability distribution function of the form: $p(x) = p(x|L = 0) * p(L = 0) + p(x|L = 1) * p(L = 1)$ where $p(L = 0) = 0.6$ and $p(L = 1) = 0.4$. $p(x|L = 0) = w_{01} * g(x|m_{01}, C_{01}) + w_{02} * g(x|m_{02}, C_{02})$ and $p(x|L = 1) = w_{11} * g(x|m_{11}, C_{11}) + w_{12} * g(x|m_{12}, C_{12})$ where $g(x|m, C)$ are Gaussian distributions with mean and variance as follows.

$$m_{01} = \begin{bmatrix} -0.9 \\ -1.1 \end{bmatrix} m_{02} = \begin{bmatrix} 0.8 \\ 0.75 \end{bmatrix} m_{11} = \begin{bmatrix} -1.1 \\ 0.9 \end{bmatrix} m_{12} = \begin{bmatrix} 0.9 \\ -0.75 \end{bmatrix} c_{ij} = \begin{bmatrix} 0.75 & 0 \\ 0 & 1.25 \end{bmatrix}$$

I generate:

- $\mathcal{D}_{20}^{\text{train}}$ consists of 20 samples and their labels for training;
- $\mathcal{D}_{200}^{\text{train}}$ consists of 200 samples and their labels for training;
- $\mathcal{D}_{2000}^{\text{train}}$ consists of 2000 samples and their labels for training;
- $\mathcal{D}_{10K}^{\text{validate}}$ consists of 10,000 samples and their labels for validation;



1.1 Part A: MPE Classifier

1.1.1 Analysis

Theoretically optimal classifier that achieves minimum probability of error is:

$$\hat{C} = \arg \max_{C_i} P(C_i | \mathbf{x}) = \arg \max_{C_i} (p(\mathbf{x}|C_i)P(C_i))$$

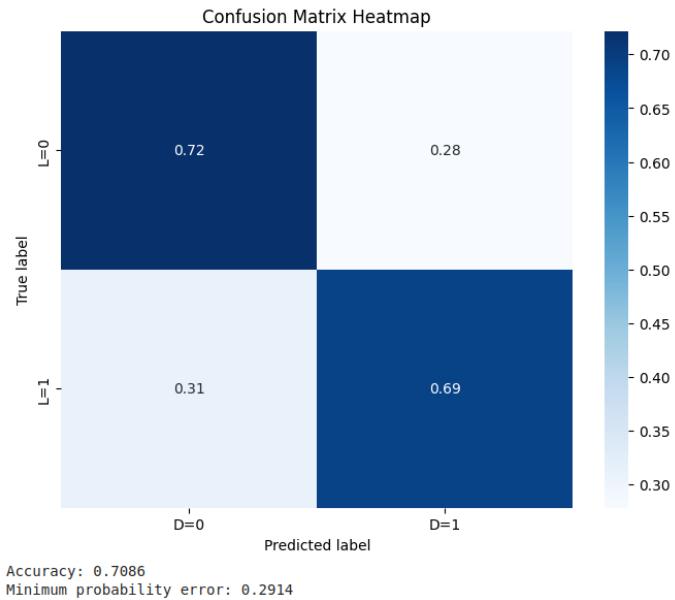
where $P(C_i)$ is the prior as given in the question and

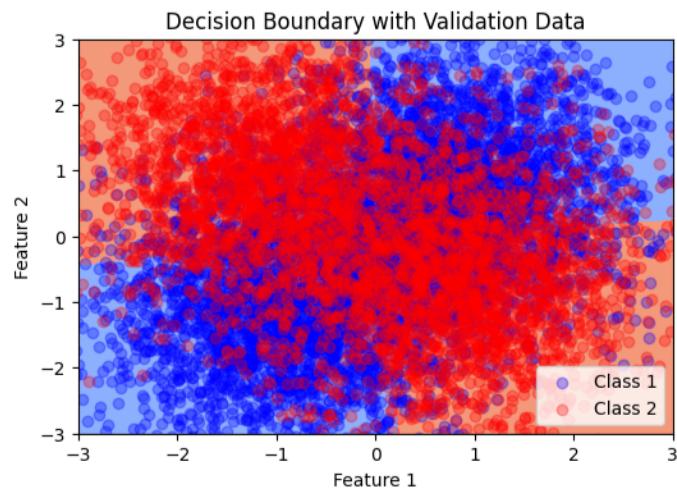
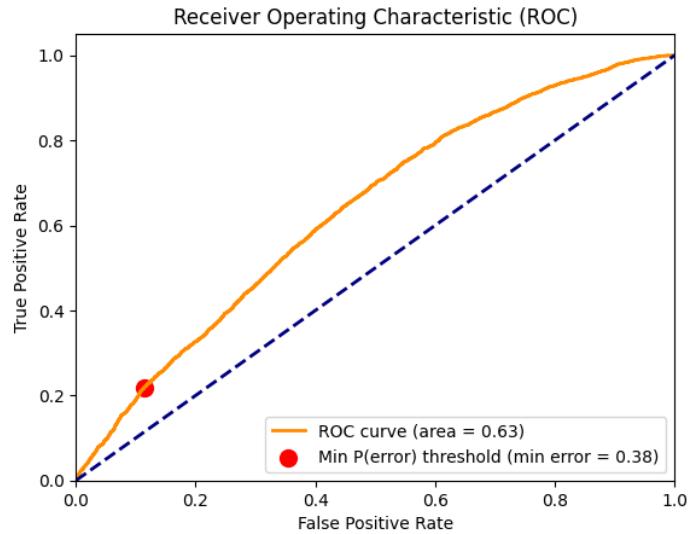
$$p(\mathbf{x}|C_i) = 0.5 \cdot f(\mathbf{x}; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}) + 0.5 \cdot f(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma})$$

$$f(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)\right)$$

- d is the dimensionality of \mathbf{x} ,
- $\boldsymbol{\Sigma}$ is the covariance matrix (assumed to be the same for both distributions),
- $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ are the mean vectors for the two Gaussian distributions,
- $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$

1.1.2 Results





1.2 MLE Classification

1.2.1 Logistic function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$NLL(w, b) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i))]$$

$$dw = \frac{1}{N} \sum_{i=1}^N (\sigma(z_i) - y_i) X_i$$

$$db = \frac{1}{N} \sum_{i=1}^N (\sigma(z_i) - y_i)$$

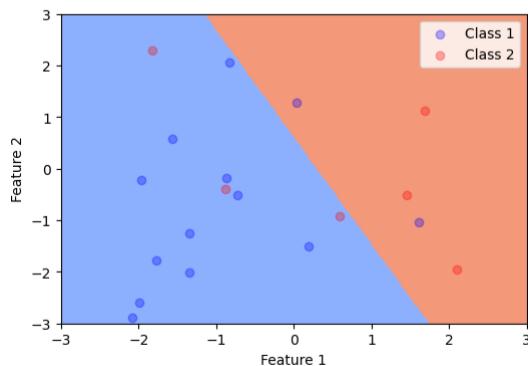
$$w \leftarrow w - \eta \cdot dw$$

$$b \leftarrow b - \eta \cdot db$$

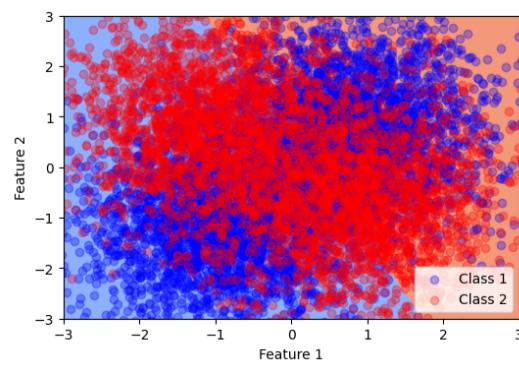
$$\text{predictions} = \begin{cases} 1, & \text{if probabilities} > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

$$\text{accuracy} = \frac{1}{N} \sum_{i=1}^N (y_i == \hat{y}_i)$$

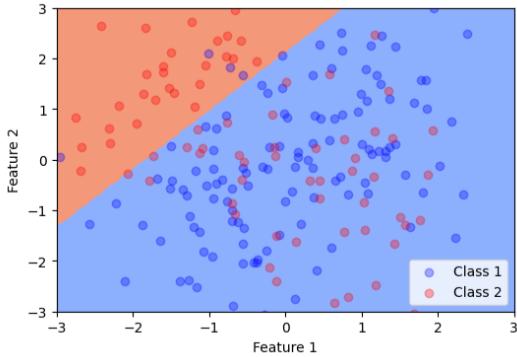
Evaluating Model 1 on the same training dataset:
Accuracy on Dataset 1: 75.00%



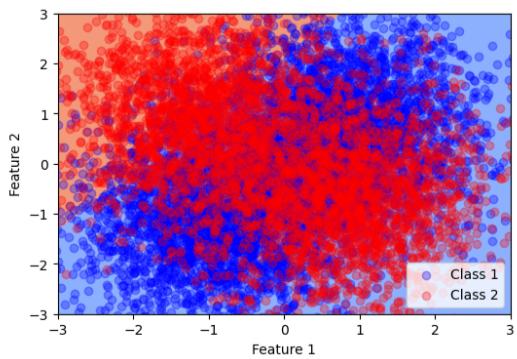
Evaluating Model 1 on validation dataset:
Accuracy on Dataset 1: 49.40%



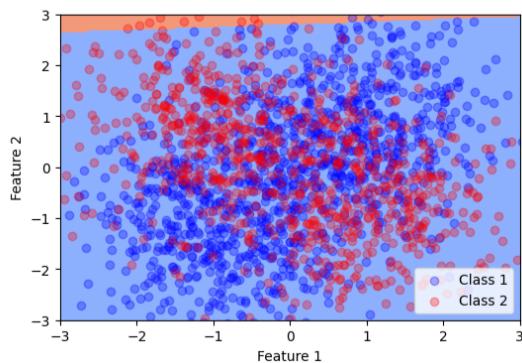
Evaluating Model 2 on the same training dataset:
Accuracy on Dataset 2: 74.00%

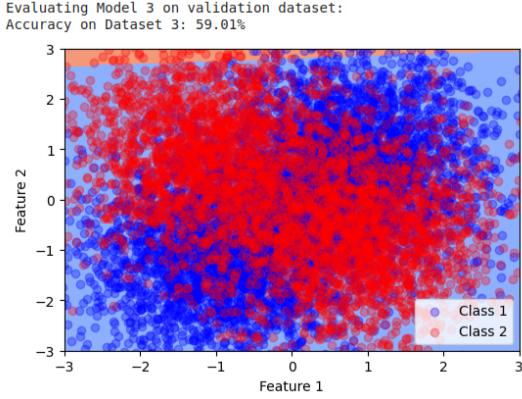


Evaluating Model 2 on validation dataset:
Accuracy on Dataset 2: 65.77%



Evaluating Model 3 on the same training dataset:
Accuracy on Dataset 3: 59.30%





1.2.2 Logistic quadratic function

$$\mathbf{X}_{\text{transformed}} = [X, \ X^2, \ X_i X_j]$$

$$z = \mathbf{X}_{\text{transformed}} \cdot \mathbf{w} + b$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$NLL(w, b) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i))]$$

$$dw = \frac{1}{N} \sum_{i=1}^N (\sigma(z_i) - y_i) X_i$$

$$db = \frac{1}{N} \sum_{i=1}^N (\sigma(z_i) - y_i)$$

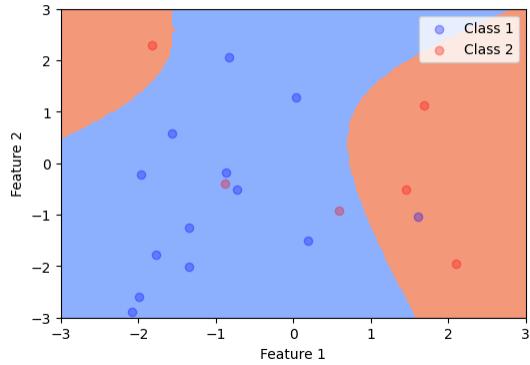
$$w \leftarrow w - \eta \cdot dw$$

$$b \leftarrow b - \eta \cdot db$$

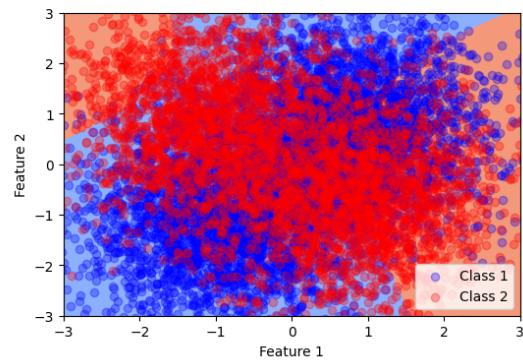
$$\text{predictions} = \begin{cases} 1, & \text{if probabilities} > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

$$\text{accuracy} = \frac{1}{N} \sum_{i=1}^N (y_i == \hat{y}_i)$$

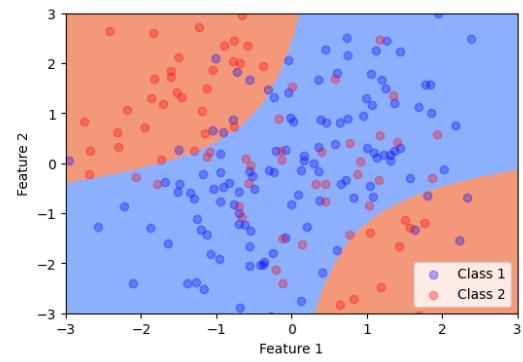
Evaluating Model 1 on the same training dataset:
Accuracy on Dataset 1: 65.00%



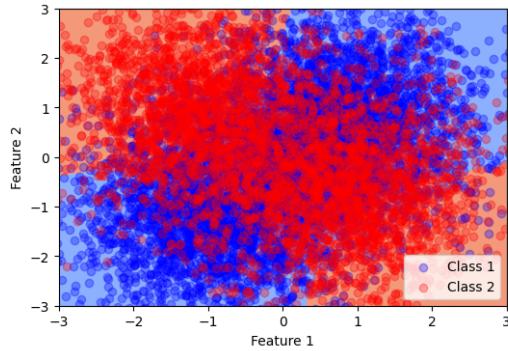
Evaluating Model 1 on validation dataset:
Accuracy on Dataset 1: 58.24%



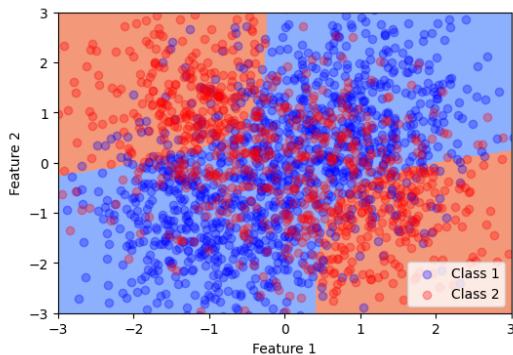
Evaluating Model 2 on the same training dataset:
Accuracy on Dataset 2: 76.50%



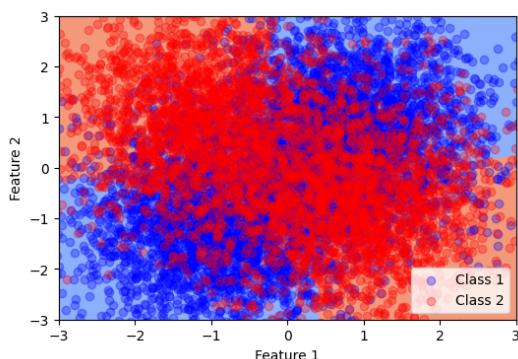
Evaluating Model 2 on validation dataset:
Accuracy on Dataset 2: 70.79%



Evaluating Model 3 on the same training dataset:
Accuracy on Dataset 3: 72.90%



Evaluating Model 3 on validation dataset:
Accuracy on Dataset 3: 71.77%



1.2.3 Conclusion

We see that the MPE performs as good as the best of MLE classifier (optimized by SGD).

The MLE classifier, predictably, shows the best accuracy (on the training data) when the training data is least and the accuracy progressively goes down as it is flooded with training data.

For the validation data, in the linear case, it performs best when the model has neither too less training data (leading to overfitting on it) nor too much data (where it starts to underfit). For the quadratic case, the model accuracy increases with the increase in training data as the model complexity is enough to prevent the model from underfitting.

2 Question 2

y is a scalar real number and x is a two-dimensional real vector.

$$y = c(x, w) + v$$

where $c(x, w)$ is a cubic polynomial in x with weights w .

$$v \sim \mathcal{N}(0, \sigma^2)$$

Given a dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ with N samples of (x, y) pairs, we assume that these samples are independent and identically distributed according to the model.

2.1 Theory

2.1.1 MLE Classifier

$$P(y_i | x) \sim N(c(x_i, w), \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - c(x_i, w))^2}{2\sigma^2}\right)$$

$$\arg \min_w (y_i - c(x_i, w))^2 \text{ for all } i$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix}, \quad w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

$$c(x_i, w) = X_i \cdot w$$

$$\arg \min_w (Y - Xw)^2$$

$$= \arg \min_w (Y - Xw)^T (Y - Xw)$$

$$= \arg \min_w (Y^T Y - w^T X^T Y - Y^T Xw + (Xw)^T (Xw))$$

differentiating:

$$-2X^T Y + 2X^T Xw = 0$$

$$X^T Xw = X^T Y$$

$$\omega = (X^T X)^{-1} X^T Y$$

2.1.2 MAP Classifier

Model:

$$Y = X^\top w + v$$

Likelihood:

$$P(Y|X, w) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(Y - X^\top w)^2}{2\sigma^2}\right)$$

Prior Distribution:

$$P(w) = \frac{1}{(2\pi)^{d/2} |\gamma^2 I|^{1/2}} \exp\left(-\frac{1}{2\gamma^2} w^\top w\right)$$

Log-posterior:

$$L(w) = -\frac{1}{2} \log(2\pi\sigma^2) - \frac{(Y - X^\top w)^2}{2\sigma^2} - \frac{d}{2} \log(2\pi\gamma^2) - \frac{1}{2\gamma^2} w^\top w$$

Finding the Maximum:

$$\begin{aligned} \frac{dL(w)}{dw} &= 0 \Rightarrow -\frac{1}{\sigma^2}(Y - X^\top w)x - \frac{1}{\gamma^2}w = 0 \\ \gamma^2(Y - X^\top w)x &= \sigma^2 w \\ \gamma^2 YX &= \sigma^2 w + \gamma^2(XX^\top)w \\ \hat{w} &= \left(\frac{\sigma^2}{\gamma^2}I + (XX^\top)\right)^{-1} YX \end{aligned}$$

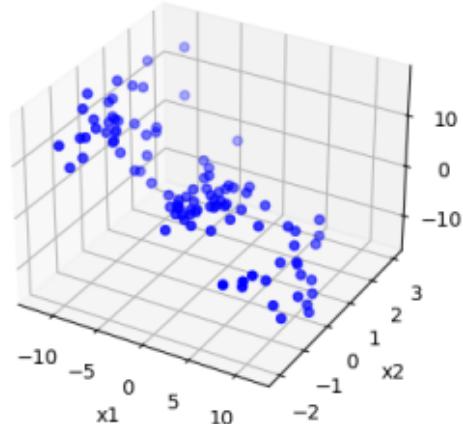
2.1.3 MSE

$$\text{MSE}(X, y, w) = \frac{1}{n} \sum_{i=1}^n (y_i - (w_0 + w_1 X_i + w_2 X_i^2 + w_3 X_i^3))^2$$

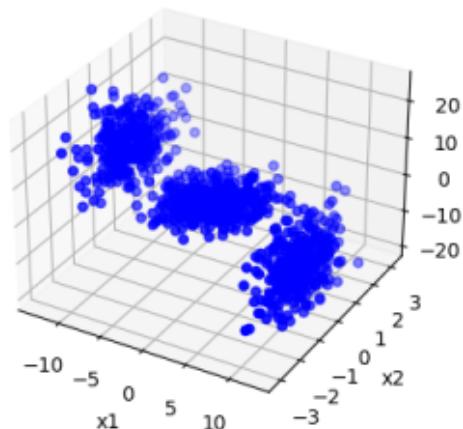
2.2 Results

Plotting the training and the validation dataset

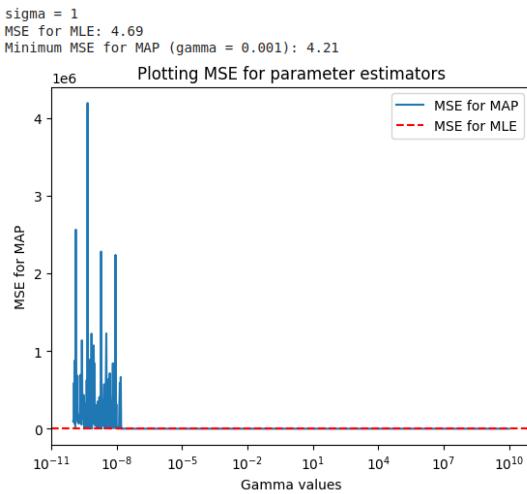
Training Dataset



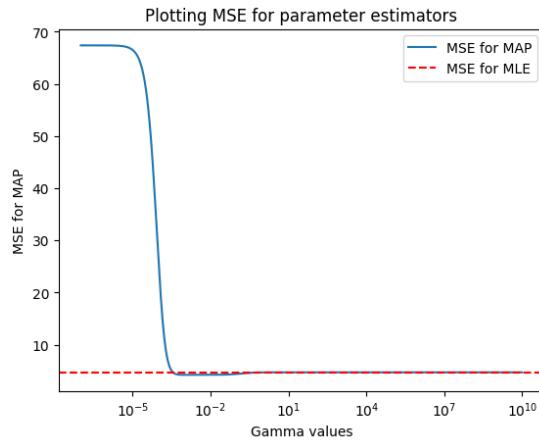
Validation Dataset

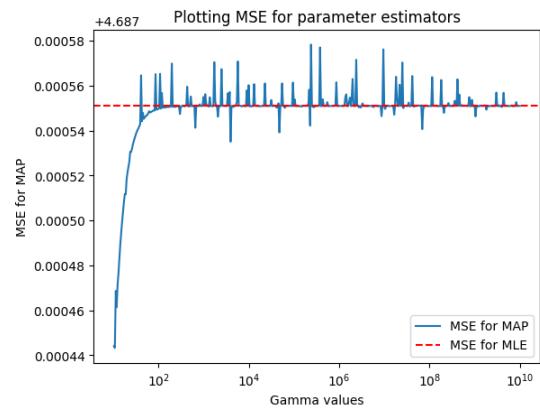
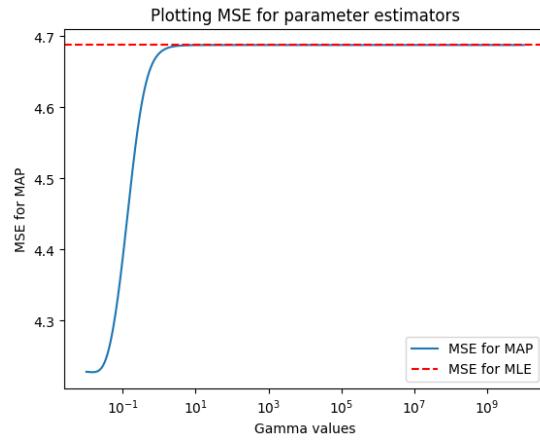


MPE we get when we vary γ



Following are the graphs we get when we zoom into the above graph to get a better idea of the graph





We can see that the minimum MSE for MAP estimator and that we get from MLE are pretty close, but MAP performs slightly better.

3 Question 3

3.1 Theory

1. **True Position:** The true position of the vehicle is represented as $[x_T, y_T]^T$ in 2-dimensional space.
2. **Reference Landmarks:** There are K reference landmarks, each with coordinates $[x_i, y_i]^T$ for $i = 1, \dots, K$.
3. **Range Measurements:** The range measurements r_i from the vehicle to each landmark are given by:

$$r_i = d_T^i + n_i$$

where:

- $d_T^i = \| [x_T, y_T]^T - [x_i, y_i]^T \|$ is the true distance from the vehicle to the i -th landmark.
- n_i is the measurement noise, assumed to be zero mean Gaussian with known variance σ_i^2 , i.e., $n_i \sim \mathcal{N}(0, \sigma_i^2)$.

4. **Prior:** Prior for position of vehicle is given by:

$$p(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2} \begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \end{pmatrix}}$$

5. **Objective:** The objective is to estimate the vehicle's position $[x_T, y_T]^T$ using the measurements r_i .
6. **MAP Estimator:**
Bayesian Inference Equation:

$$p(x, y | r) \propto p(r|x, y)p(x, y)$$

Likelihood Function:

$$\begin{aligned} p(r|x, y) &= \prod_{i=1}^K p(r_i|x, y) \\ p(r_i|x, y) &= \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(r_i - d_T^i)^2}{2\sigma_i^2}} \\ d_T^i &= \sqrt{(x - x_i)^2 + (y - y_i)^2} \\ p(r|x, y) &= \prod_{i=1}^K \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(r_i - d_T^i)^2}{2\sigma_i^2}} \end{aligned}$$

Cost Function (Negative Log-Likelihood):

$$J(x, y) = - \sum_{i=1}^K \log p(r_i|x, y) + C$$

$$J(x, y) = \sum_{i=1}^K \frac{(r_i - d_T^i)^2}{2\sigma_i^2} + C'$$

Prior:

$$p(x, y) = (2\pi\sigma_x\sigma_y)^{-1} \exp\left(-\frac{1}{2}\begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \end{pmatrix}\right)$$

$$\log p(x, y) = -\log(2\pi\sigma_x\sigma_y) - \frac{1}{2}\begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \end{pmatrix}$$

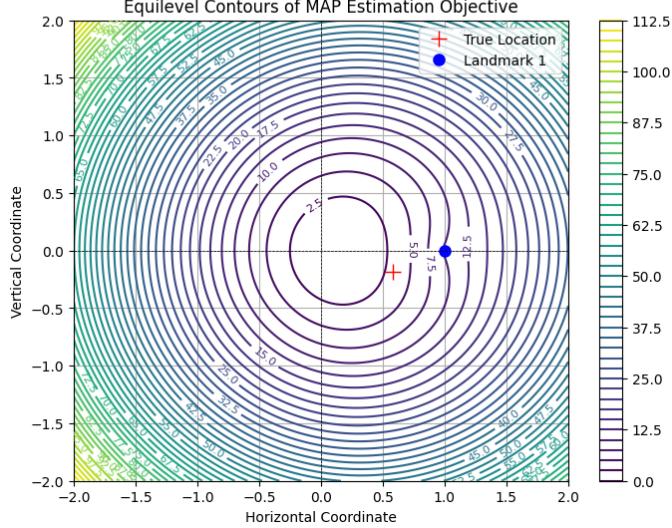
Cost Function for MAP Estimation:

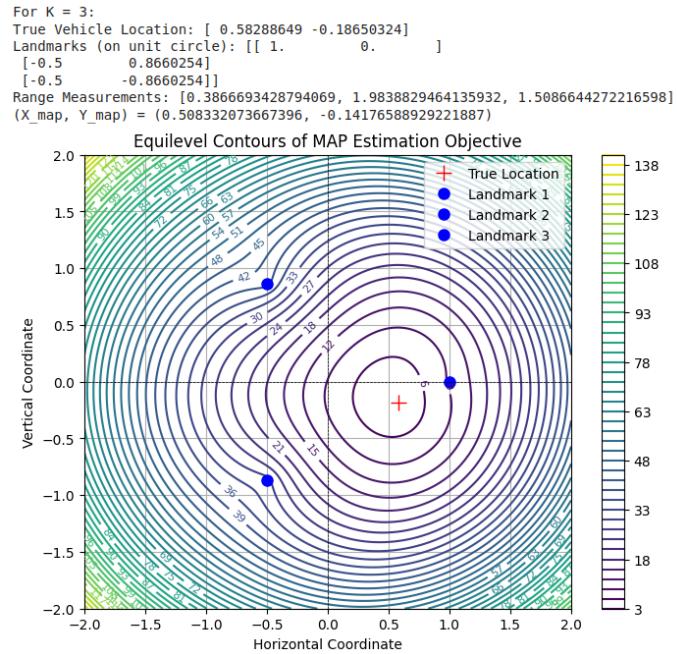
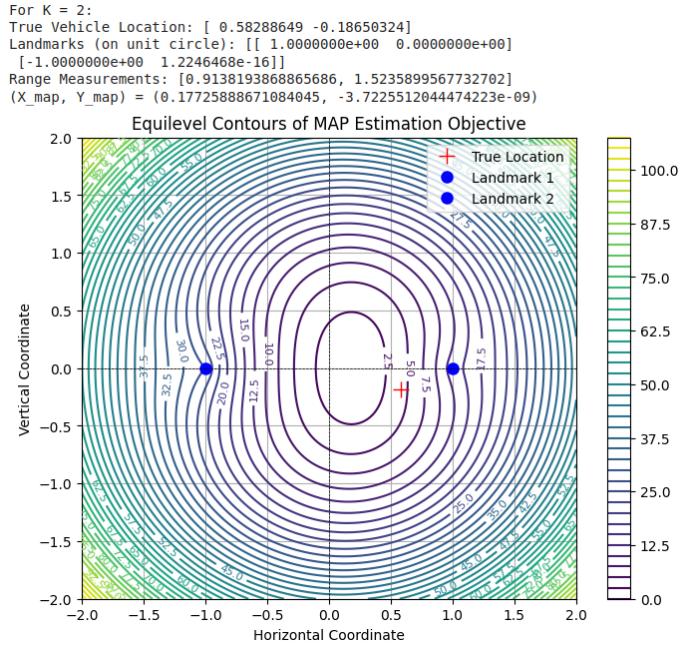
$$J(x, y) = \sum_{i=1}^K \frac{(r_i - d_T^i)^2}{2\sigma_i^2} + \frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right)$$

$$J(x, y) = \sum_{i=1}^K \frac{(r_i - \sqrt{(x - x_i)^2 + (y - y_i)^2})^2}{2\sigma_i^2} + \frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right)$$

3.2 Results

```
For K = 1:
True Vehicle Location: [ 0.58288649 -0.18650324]
Landmarks (on unit circle): [[1. 0.]]
Range Measurements: [0.6512169913943688]
(X_map, Y_map) = (0.1429438511590868, -6.054458016215484e-09)
```

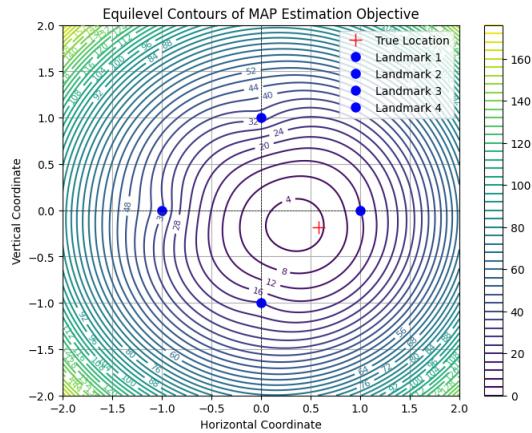




```

For K = 4;
True Vehicle Location: [ 0.58288649 -0.18650324]
Landmarks (on unit circle): [[ 1.0000000e+00  0.0000000e+00]
[ 6.1232340e-17  1.0000000e+00]
[ -1.0000000e+00  1.2246468e-16]
[ -1.8369702e-16 -1.0000000e+00]]
Range Measurements: [0.3166681141836754, 1.4847166476450725, 1.4548106613465324, 0.8610475959474878]
(X_map, Y_map) = (0.3482197898875743, -0.16081472938969188)

```



We can see that as we increase the value of K, the MAP estimate does indeed become more accurate and nearer to the true location (this can be seen as a progressively lower value of counter encircles the true location)

4 Question 4

4.1 Question

$$\lambda(\alpha_i|w_j) = \begin{cases} 0, & \text{if } i = j \forall i, j \in 1, \dots, C \\ \lambda_r, & \text{if } i = C + 1 \\ \lambda_s, & \text{otherwise} \end{cases}$$

To Prove: minimum risk is achieved if w_i is returned if $P(w_i|X) \geq P(w_j|X) \forall j$, $P(w_i|x) \geq 1 - \frac{\lambda_r}{\lambda_s}$ and reject otherwise

$$\text{Risk}(i) = \sum_j \lambda(\alpha_i|w_j) P(w_j|x)$$

$$\text{Risk}(i) = \sum_{j \neq i} \lambda(\alpha_i|w_j) P(w_j|x)$$

$$\text{Risk}(i) = \begin{cases} \lambda_r \sum_{j \neq i} P(w_j|x), & \text{if } i = C + 1 \\ \lambda_s \sum_{j \neq i} P(w_j|x), & \text{if } i \in 1, \dots, C \end{cases}$$

$$\text{Risk}(i) = \begin{cases} \lambda_r, & \text{if } i = C + 1 \\ \lambda_s(1 - P(w_i|x)), & \text{if } i \in 1, \dots, C \end{cases}$$

So, to return w_i ,

$$\lambda_s(1 - P(w_i|x)) \leq \lambda_r$$

$$P(w_i|x) \geq 1 - \frac{\lambda_r}{\lambda_s}$$

If $\lambda_r = 0$, then unless we know i such that $P(w_i|x) = 0$ we should choose reject
If $\frac{\lambda_r}{\lambda_s} > 1$ then the cost of rejection is too great, and we always choose the
 $i = \arg \max_j P(w_j|x)$

5 Question 5

$$\begin{aligned}\mathbf{z} &= [z_1, z_2, \dots, z_K]^T \\ \Theta &= [\theta_1, \theta_2, \dots, \theta_K]^T \\ \sum_{k=1}^K \theta_k &= 1 \\ P(z_k = 1) &= \theta_k, \quad k \in \{1, 2, \dots, K\} \\ P(\mathbf{z}|\Theta) &= \prod_{k=1}^K \theta_k^{z_k}\end{aligned}$$

5.1 ML Estimator

$$\begin{aligned}P(\mathbf{D}|\Theta) &= \prod_{i=1}^N P(\mathbf{z}_i; \Theta) = \prod_{i=1}^N \prod_{k=1}^K \theta_k^{z_{ik}} = \prod_{k=1}^K \theta_k^{\sum_{i=1}^N z_{ik}} \\ \log(P(\mathbf{D}|\Theta)) &= \sum_{k=1}^K \log(\theta_k^{\sum_{i=1}^N z_{ik}}) = \sum_{k=1}^K (\sum_{i=1}^N z_{ik}) \log(\theta_k) \\ N_k &= \sum_{i=1}^N z_{ik}\end{aligned}$$

So, we have to maximize $\sum_{k=1}^K N_k \log(\theta_k)$ constrained to $\sum_{k=1}^K \theta_k = 1$
 Applying Lagrange Multiplier method:

$$\begin{aligned}\mathbf{L}(K|\Theta) &= \sum_{k=1}^K N_k \log(\theta_k) + \lambda(1 - \sum_{k=1}^K \theta_k) \\ \frac{\partial \mathbf{L}}{\partial \theta_k} &= \frac{N_k}{\theta_k} + \lambda(-1) = 0 \\ \theta_k &= \frac{N_k}{\lambda}\end{aligned}$$

Using the constraint $\sum_{k=1}^K \theta_k = 1$

$$\begin{aligned}\sum_{k=1}^K \frac{N_k}{\lambda} &= 1 \rightarrow \lambda = \sum_{k=1}^K N_k \\ \theta_k &= \frac{N_k}{\sum_{k=1}^K N_k} = \frac{\sum_{i=1}^N z_{ik}}{\sum_{k=1}^K \sum_{i=1}^N z_{ik}} = \frac{\sum_{i=1}^N z_{ik}}{\sum_{i=1}^N \sum_{k=1}^K z_{ik}} = \frac{\sum_{i=1}^N z_{ik}}{N}\end{aligned}$$

5.2 MAP Estimator

$$\begin{aligned}
P(\Theta|\mathbf{D}) &= \frac{P(\mathbf{D}|\Theta)P(\Theta)}{P(\mathbf{D})} \\
P(\Theta) &= \frac{1}{B(\alpha)} \prod_{k=1}^K \theta_k^{\alpha_k-1}; B(\alpha) = \frac{\prod_{k=1}^K \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^K \alpha_k)} \\
P(\mathbf{D}|\Theta) &= \prod_{i=1}^N \prod_{k=1}^K \theta_k^{z_{ik}} \\
P(\Theta|\mathbf{D}) &= \frac{(\prod_{i=1}^N \prod_{k=1}^K \theta_k^{z_{ik}}) \prod_{k=1}^K \theta_k^{\alpha_k-1}}{P(\mathbf{D})B(\alpha)}
\end{aligned}$$

Disregarding terms not related to Θ

$$\log(P(\Theta|\mathbf{D})) = \sum_{k=1}^K \left(\sum_{i=1}^N z_{ik} \right) \log(\theta_k) + \sum_{k=1}^K (\alpha_k - 1) \log \theta_k = \sum_{k=1}^K \left(\sum_{i=1}^N z_{ik} + \alpha_k - 1 \right) \log(\theta_k)$$

So, we have to maximize $\sum_{k=1}^K (\sum_{i=1}^N z_{ik} + \alpha_k - 1) \log(\theta_k)$ constrained to $\sum_{k=1}^K \theta_k = 1$

Applying Lagrange Multiplier method:

$$\begin{aligned}
\mathbf{L}(K|\Theta) &= \sum_{k=1}^K \left(\sum_{i=1}^N z_{ik} + \alpha_k - 1 \right) \log(\theta_k) + \lambda \left(1 - \sum_{k=1}^K \theta_k \right) \\
\frac{\partial \mathbf{L}}{\partial \theta_k} &= \frac{\left(\sum_{i=1}^N z_{ik} + \alpha_k - 1 \right)}{\theta_k} + \lambda(-1) = 0 \\
\theta_k &= \frac{\left(\sum_{i=1}^N z_{ik} + \alpha_k - 1 \right)}{\lambda}
\end{aligned}$$

Using the constraint $\sum_{k=1}^K \theta_k = 1$

$$\begin{aligned}
\sum_{k=1}^K \frac{\left(\sum_{i=1}^N z_{ik} + \alpha_k - 1 \right)}{\lambda} &= 1 \rightarrow \lambda = \sum_{k=1}^K \left(\sum_{i=1}^N z_{ik} + \alpha_k - 1 \right) = N + \sum_{k=1}^K \alpha_k - K \\
\theta_k &= \frac{\sum_{i=1}^N z_{ik} + \alpha_k - 1}{N + \sum_{k=1}^K \alpha_k - K}
\end{aligned}$$

Link to the [Github Folder](#) or copy this into your web browser:
<https://github.com/Dhruv-2020EE30592/EECE-5644/tree/main/Assignment-2>