Northeastern University

Introduction to Machine Learning and Pattern Recognition

Subject Code: EECE 5644

ASSIGNMENT-2

Submitted to:

Prof. Deniz Erdogmus

Submitted by:

Shreyas Kapanaiah Mahesh

NU ID: 002332297

Date of Submission:

October 24th, 2025

QUESTION: 1

The probability density function (pdf) for a 2-dimensional real-valued random vector X is as follows: $p(x) = P(L = 0)p(x|L = 0)+P(L = 1)p(x|L = 1)$. Here L is the true class label that indicates which class-label-conditioned pdf generates the data.

The class priors are $P(L = 0) = 0.6$ and $P(L = 1) = 0.4$. The class class-conditional pdfs are $p(x|L = 0) = w01g(x|m01,C01) + w02g(x|m02,C02)$ and $p(x|L = 1) = w11g(x|m11,C11) + w12g(x|m12,C12)$, where $g(x|m,C)$ is a multivariate Gaussian probability density function with mean vector m and covariance matrix C.

For numerical results requested below, generate the following independent datasets each consisting of iid samples from the specified data distribution, and in each dataset make sure to include the true class label for each sample.

PART-1:

(Q1) PART-1

Given a 2D random vector x with a PDF:

$$p(x) = P(L=0)\,p(x|L=0) + P(L=1)\,p(x|L=1)$$

Class Priors $P(L=0) = 0.6$
$P(L=1) = 0.4$

Class-Conditional PDFs → Multivariate Gaussian Probability

$$p(x|L=0) = 0.5\,g(x|m_{01}, C_{01}) + 0.5\,g(x|m_{02}, C_{02})$$
$$p(x|L=1) = 0.5\,g(x|m_{11}, C_{11}) + 0.5\,g(x|m_{12}, C_{12})$$

Given that $C_{ij} = \begin{bmatrix} 0.75 & 0 \\ 0 & 1.25 \end{bmatrix}$ for all pairs.

Bayes classifier is the optimal classifier that achieves

Decide $L=1$ if $P(L=1|x) > P(L=0|x)$

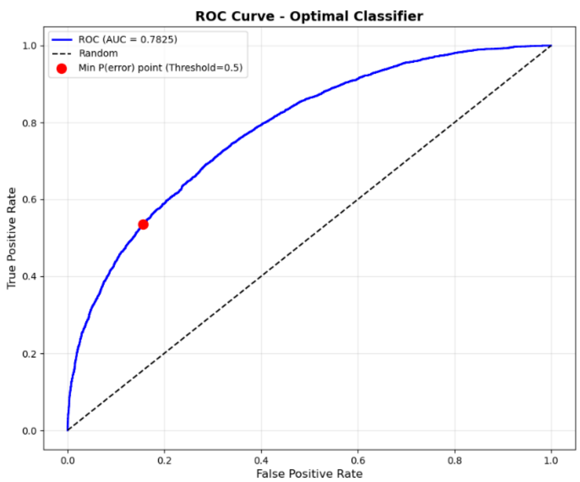
which leads to ....

$$\frac{p(x|L=1)}{p(x|L=0)} \gtrless \frac{(\lambda_{10}-\lambda_{00})}{(\lambda_{01}-\lambda_{11})} * \frac{P(L=0)}{P(L=1)}$$

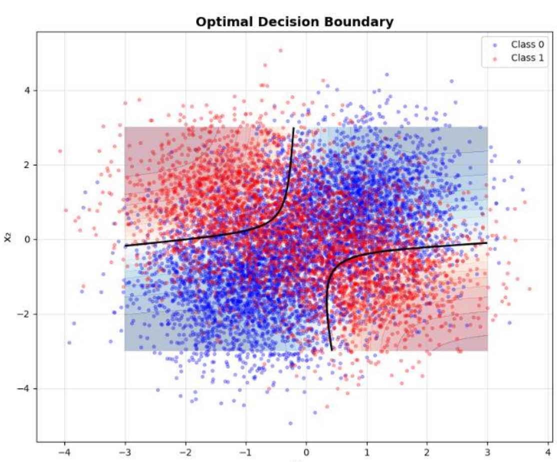
$$\Rightarrow \frac{(1-0)}{(1-0)} * \frac{0.6}{0.4} = 1.5$$

The theoretical optimal threshold γ^* will be 1.5 leading to minimum Probability of error P(error)

	Value
Minimum P(error) Estimate	0.2807
Area Under Curve (AUC)	0.7825
Minimum P(error) operating point	0.5 (Threshold)



Here's the ROC curve for Optimal Classifier. The minimum P(error) point is indicated at the threshold 0.5. The AUC value of 0.7825 indicates an imperfect yet good separability.



Here's the optimal decision boundary plot. The boundary is non-linear and hyperbolic, reflecting the structure necessary to separate the 4 gaussian components, obtaining the theoretical minimum error.

PART-2:

Q1 PART-2

Logistic Regression Models

Mathematical Framework :-

The logistic regression approximates the posterior $P(L=1|x)$:

$$h(x, \omega) = \frac{1}{1+e^{-\omega^T z(x)}}$$

Logistic - Linear Model

$$z(x) = [1, x_1, x_2]^T \Rightarrow 3 \text{ parameters}$$

linear decision boundary.

Logistic - Quadratic Model

$$z(x) = [1, x_1, x_2, x_1^2, x_1 x_2, x_2^2]^T \Rightarrow 6 \text{ parameters}$$

Quadratic Decision Boundary.

ERRORS are estimated on the $D_{10000}^{validate}$ set.

Datasets Generated

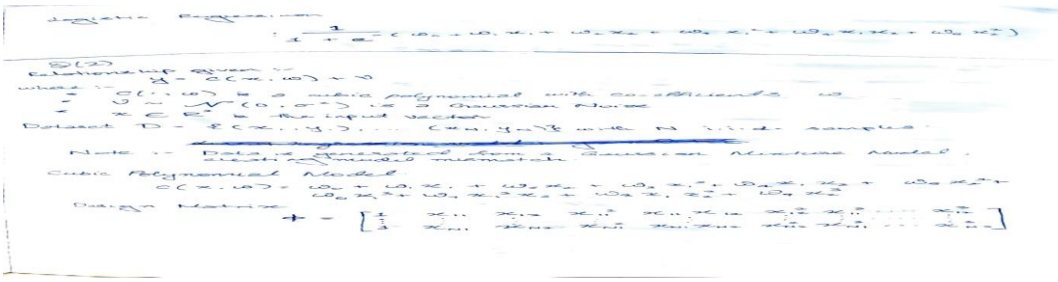
Training sets
- D_{50}^{train} - 50 samples
- D_{500}^{train} - 500 samples
- D_{5000}^{train} - 5000 samples

$D_{10000}^{validate}$ - 10,000 samples $\downarrow$ Validation set.

Maximum likelihood Training.

$$\mathcal{L}(\omega) = -\sum_{n=1}^{N} \left[L_n \log h(x_n, \omega) + (1-L_n) \log(1-h(x_n, \omega)) \right]$$

$$\omega^T z(x) \gtrless 0$$



Logistic Regression

$= \frac{1}{1 + e^{-(\omega_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_1^2 + \omega_4 x_1 x_2 + \omega_5 x_2^2)}}$

Q(2)

Relationship given :-
$y = c(x, \omega) + v$

where :-
- $c(\cdot, \omega)$ is a cubic polynomial with co-efficients ω
- $v \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian Noise
- $x \in \mathbb{R}^2$ is the input vector

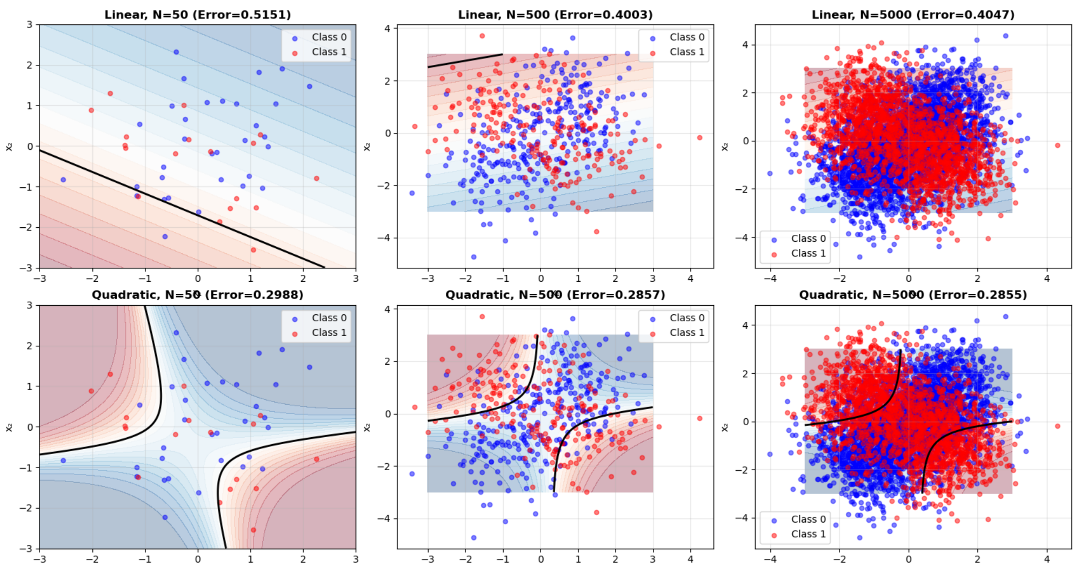
Dataset $D = \{(x_1, y_1), \dots, (x_N, y_N)\}$ with N i.i.d. samples

Note :- Data is generated from a Gaussian Mixture Model, leads to model mismatch

Cubic Polynomial Model

$c(x, \omega) = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_1^2 + \omega_4 x_1 x_2 + \omega_5 x_2^2 + \omega_6 x_1^3 + \omega_7 x_1^2 x_2 + \omega_8 x_1 x_2^2 + \omega_9 x_2^3$

Design Matrix



The above figure shows subplots comparing linear (top row) and quadratic (bottom row) models across different training sizes (50, 500, 5000)

Logistic-Linear Model Performance:

Training Samples	Validation Error	Parameters
50	0.5151	3
500	0.4003	3
5000	0.4047	3

Logistic-Quadratic Model performance:

Training Samples	Validation Error	Parameters
50	0.2988	6
500	0.2857	6
5000	0.2855	6

PART-3:

Performance Comparison:

Optimal Classifier Error: 0.2807
Linear Models: Converge to 0.40 to 0.51 errors (high bias 0.43 to 0.84 over optimal)
Quadratic Models: Converge to 0.285 to 0.299 (0.01 to 0.06 over optimal)

Based on the observation, we see that the quadratic models achieve near-optimal performance, with the best model (N=5000) reaching only 0.2855. This demonstrates that

1) Quadratic boundary closely approximates the true GMM boundary
2) Having sufficient data, the estimation variance becomes negligible, making this an excellent choice.

We observe theoretical optimal classifier achieves 28.07% error by using the true Gaussian mixture distributions with known means, covariances and priors. This will be the Bayes error rate – the absolute minimum achievable. Even with this prefect knowledge, 28% error can't be eradicated due to inherent class overlap in the feature space.

The quadratic logistic models achieve a 28.55% error by learning from the training data, closely approximating the true optimal S-shaped boundary. Meanwhile the linear give a bad performance at 40% error because of high bias (underfitting) making it fundamentally inadequate.

Eventually, model selection matters more than the sample size.

QUESTION-2:

Assume that scalar-real y and two-dimensional real vector x are related to each other according to y = c(x,w)+v, where c(.,w) is a cubic polynomial in x with coefficients w and v is a random Gaussian random scalar with mean zero and σ2-variance
Given a dataset D = (x1,y1), . . . , (xN,yN) with N samples of (x, y) pairs, with the assumption that these samples are independent and identically distributed according to the model, derive two estimators for w using maximum-likelihood (ML) and maximum-a-posteriori (MAP) parameter estimation approaches as a function of these data samples. For the MAP estimator, assume that w has a zero-mean Gaussian prior with covariance matrix γI. Having derived the estimator expressions, implement them in code and apply to the dataset generated by the attached Matlab script. Using the training dataset, obtain the ML estimator and the MAP estimator for a variety of γ values ranging from 10−m to 10n. Evaluate each trained model by calculating the average-squared error between the y values in the validation samples and model estimates of these using c(.,wtrained). How does your MAP-trained model perform on the validation set as γ is varied? How is the MAP estimate related to the ML estimate? Describe your experiments, visualize and quantify your analyses (e.g. average squared error on validation dataset as a function of hyperparameter γ) with data from these experiments.

Q(2)

Given that

$$y = c(x, \omega) + v$$

$y \rightarrow$ scalar output

$x \rightarrow [x_1, x_2]^T$ a 2D input vector

$c(x, \omega)$ = A cubic polynomial function with parameters ω

$V \sim \mathcal{N}(0, \sigma^2)$ = Gaussian noise (mean 0, variance σ^2)

Dataset :-

$$D = \{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$$

N samples assumed independent and identically distributed

For a 2D input, i.e. $x = [x_1, x_2]^T$, a cubic polynomial includes upto the 3 degree terms.

$$c(x, \omega) = \omega_0 \cdot 1 + \omega_1 \cdot x_1 + \omega_2 x_2 + \omega_3 x_1^2 + \omega_4 x_1 x_2 + \omega_5 x_2^2 + \omega_6 x_1^3 + \omega_7 x_1^2 x_2 + \omega_8 x_1 x_2^2 + \omega_9 x_2^3$$

Feature Mapping $\phi(x)$:

$$\phi(x) = \begin{bmatrix} \phi_0(x) \\ \phi_1(x) \\ \phi_2(x) \\ \phi_3(x) \\ \phi_4(x) \\ \phi_5(x) \\ \phi_6(x) \\ \phi_7(x) \\ \phi_8(x) \\ \phi_9(x) \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \\ x_1^3 \\ x_1^2 x_2 \\ x_1 x_2^2 \\ x_2^3 \end{bmatrix}$$

Here $\phi(x)$ is a 10×1 vector

In a compact manner.

$$C(x, \omega) = \omega^T \phi(x) = \omega_0 \phi_0(x) + \omega_1 \phi_1(x) + \dots + \omega_9 \phi_9(x)$$

where $\omega = [\omega_0, \omega_1, \dots \omega_9]^T$ is a 10×1 parameter vector

The model will be

$$y = \omega^T \phi(x) + v \qquad v \sim N(0, \sigma^2)$$

y is normally distributed with

mean : $\omega^T \phi(x)$

Variance : σ^2

$y \mid x, \omega \sim N(\omega^T \phi(x), \sigma^2)$

Design Matrix -

For N data points, the design matrix ϕ is defined as.

$$\Phi = \begin{bmatrix} \phi(x_1)^T \\ \phi(x_2)^T \\ \vdots \\ \phi(x_n)^T \end{bmatrix} = \begin{bmatrix} 1, & x_{11} & x_{12} & x_{11}^2 & x_{11}x_{12} & x_{12}^2 & x_{11}^3 & \dots \\ 1. & x_{21} & x_{22} & x_{21}^2 & x_{21}x_{22} & x_{22}^2 & x_{21}^3 & \dots \\ \dots \\ 1 & x_{n1} & x_{n2} & x_{n1}^2 & x_{n1}x_{n2} & x_{n2}^2 & x_{n1}^3 & \dots \end{bmatrix}$$

ϕ is also a $N \times 10$

Also, let us define

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad \text{Dimension } N \times 1$$

Maximum Likelihood (ML) Estimator

$$p(y_i \mid x_i, \omega) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_i - \omega^T\phi(x_i))^2}{2\sigma^2} \right]$$

For all N observations (Assuming Independence)

$$p(y \mid X, \omega) = \prod_{i=1}^{N} p(y_i \mid x_i, \omega).$$

$$= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \omega^T\phi(x_i))^2}{2\sigma^2} \right)$$

Taking log likelihood -

$$\log p(y \mid X, \omega) = \log\left[\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_i - \omega^T\phi(x_i))^2}{2\sigma^2} \right) \right]$$

$$= \sum_{i=1}^{N} \left\{ \log\left(\frac{1}{2\pi\sigma^2} \right) \exp\left(\frac{-(y_i - \omega^T\phi(x_i))^2}{2\sigma^2} \right) \right\}$$

$$= \sum_{i=1}^{N} \left\{ \log\left(\frac{1}{2\pi\sigma^2} \right) - \left(\frac{(y_i - \omega^T\phi(x_i))^2}{2\sigma^2} \right) \right\}$$

$$\boxed{= \underbrace{-\frac{N}{2} \log(2\pi\sigma^2)}_{\text{Doesn't depend on } \omega \text{ so it is ignored.}} - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \omega^T\phi(x_i))^2.}$$

Maximizing $-\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \omega^T\phi(x_i))^2$

$\downarrow$

Minimizing $\sum_{i=1}^{N} (y_i - \omega^T\phi(x_i))^2 \Rightarrow$ Sum of squared error (SSE)

~~Taking the derivative~~

$$SSE = \sum_{i=1}^{N} (y_i - \omega^T \phi(x_i))^2$$

$$= \| y - \phi \omega \|^2$$

$$= (y - \phi\omega)^T (y - \phi\omega)$$

$$= y^T y - y^T \phi \omega - \omega^T \phi^T y + \omega^T \phi^T \omega$$

Here $y^T \phi \omega = \omega^T \phi^T y \Rightarrow$ Both scalars.

Taking Derivate w.r.t ω

$$\frac{\delta \, SSE}{\delta \omega} = -2\phi^T y + 2\phi^T \phi \omega = 0$$

$$\cancel{2}\phi^T \phi \omega = \cancel{2}\phi^T y$$

For ω_{ML}

$$\phi^T \phi \omega = \phi^T y$$

$$\boxed{\omega = (\phi^T \phi)^{-1} \phi^T y}$$

$$\Downarrow$$

$$\omega_{ML} = (\phi^T \phi)^{-1} \phi^T y$$

closed form least squares solution.

Maximum A Posteriori (MAP) Estimator

Using Bayesian Approach, we treat ω as a random variable with a prior distribution.

Given that

$$p(\omega) = N(0, \gamma I)$$

ω is normally distributed with

Mean : $0 \rightarrow$ zero vector

Covariance : γI (γ times identity matrix)

[redacted]

$$p(\omega) = \frac{1}{(2\pi\gamma)^{d/2}} \exp\left(-\frac{\omega^T\omega}{(2\gamma)}\right)$$

$$= \frac{1}{(2\pi\gamma)^5} \exp\left[-\frac{\|\omega\|^2}{2\gamma}\right]$$

$d = 10$ — dimension of ω

$d/2 = 5$

$\omega \to 0$

$\gamma \to$ larger mean weaker belief.

By Bayes theorem:-

$$p(\omega \mid D) = \frac{p(D \mid \omega) * p(\omega)}{p(D)}$$

Since $p(D)$ doesn't depend on ω

$$p(\omega \mid D) \propto p(D \mid \omega)\, p(\omega).$$

Taking logarithm (Log Posterior)

$$\log p(\omega \mid D) = \log p(D \mid \omega) + \log p(\omega) + \text{constant}$$

$$= \left[\frac{-1}{(2\sigma^2)} \sum_{i=1}^{N} (y_i - \omega^T\phi(x_i))^2\right] + \left[-\frac{\|\omega\|^2}{(2\gamma)}\right] + \text{Const}$$

$$= \frac{-1}{(2\sigma^2)} \sum_{i=1}^{N} (y_i - \omega^T\phi(x_i))^2 - \frac{1}{2\gamma}\|\omega\|^2 + \text{constant}$$

Maximize $\displaystyle -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \omega^T\phi(x_i))^2 - \frac{1}{2\gamma}\|\omega\|^2$.

Minimize $\displaystyle \sum_{i=1}^{N} (y_i - \omega^T\phi(x_i))^2 + \left(\frac{\sigma^2}{\gamma}\right)\|\omega\|^2$.

SSE

Let $\lambda = \frac{\sigma^2}{\gamma}$ Regularisation Parameter

Minimize $\sum_{i=1}^{N} \left(y_i - \omega^T \phi(x_i) \right)^2 + \lambda \|\omega\|^2$ → RIDGE REGRESSION

The function in matrix form.

$$J(\omega) = \| y - \phi\omega \|^2 + \lambda \|\omega\|^2$$

$$= (y - \phi\omega)^T (y - \phi\omega) + \lambda \omega^T \omega$$

$$= y^T y - 2\omega^T \phi^T y + \omega^T \phi^T \phi \omega + \lambda \omega^T \omega$$

Take derivative by ω

$$\frac{\delta J}{\delta \omega} = -2\phi^T y + 2\phi^T \phi \omega + 2\lambda\omega$$

$$= 2(\phi^T\phi\omega + \lambda\omega + \phi^T y)$$

$$= 2((\phi^T\phi + \lambda I)\omega - \phi^T y) = 0$$

$$(\phi^T\phi + \lambda I)\omega = \phi^T y$$

For ω_{map}

$$\omega = (\phi^T\phi + \lambda I)^{-1} \phi^T y$$

$$\omega_{map} = \left[\left(\phi^T\phi + \left(\frac{\sigma^2}{\gamma}\right) I \right)^{-1} \right] \phi^T y$$

As $\gamma \to \infty$

MAP → ML

IMPLEMENTATION:

Data Generation:

1) Training: N=100 samples from GMM with 3 components
2) Validation: N=1000 samples from the same distribution
3) Features: 10 (cubic polynomial terms)

Hyperparameter Search:

1) Tested γ ∈ [10^{-4} , 10^{+4}]
2) Evaluated validation MSE for each γ

Numerical Results

ML Estimator Performance:

Estimated parameters:

w_{ML} : [0.322, 0.156, 0.058, -0.003, 0.042, -0.198, -0.011, 0.003, -0.035, -0.070]

Performance:

1) Training MSE: 3.2243
2) Validation MSE: 4.8862
3) Parameter norm: || w_{ML} || = 0.4424
4) Estimated noise variance = 3.2243

MAP Estimator Results:

w_{MAP}: [0.0003, 0.0038, -0.0001, -0.0009, 0.0070, -0.0012, -0.0097, -0.0007, -0.0089, -0.0023]

Performance:

5) Training MSE: 3.379
6) Validation MSE: 4.307
7) Parameter norm: ||w_{MAP} || = 0.016

Metric	ML	MAP	Change
Training MSE	3.224	3.379	+4.8%
Validation MSE	4.886	4.307	-11.8%

Variation of lambda:

1) Small γ (<10^-3): Strong regularization --> High MSE (~4.3) --> Underfitting

2) Optimal γ (10^-4): Strong regularization --> Minimum MSE (4.307) --> Best generalization

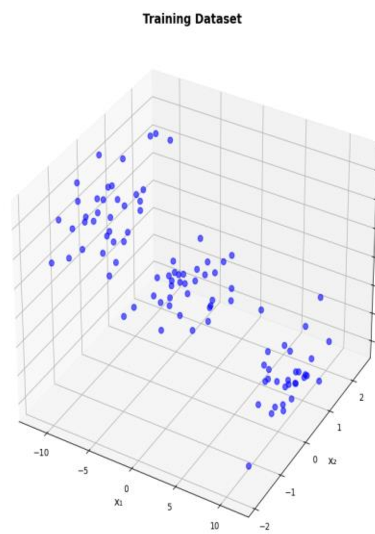
3) Large γ (>10^2): Weak regularization --> Increasing MSE --> Approaches ML (overfitting)

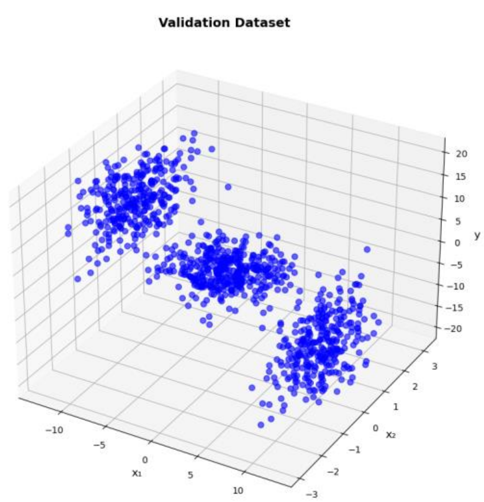
The U-shaped validation curve demonstrates the variance trade-off.

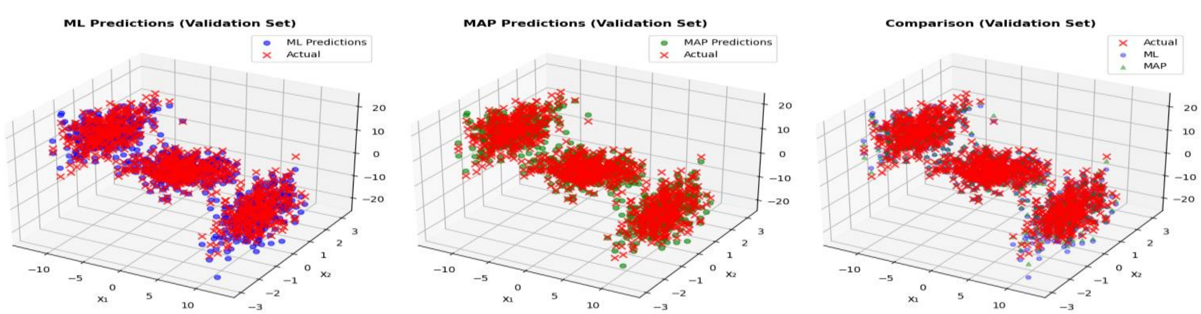
Relationship between MAP and ML:
As γ inceases, the value of λ decreases (weaker regularization). Parameters grow from 0.016 to 0.422 and MAP converges to ML.

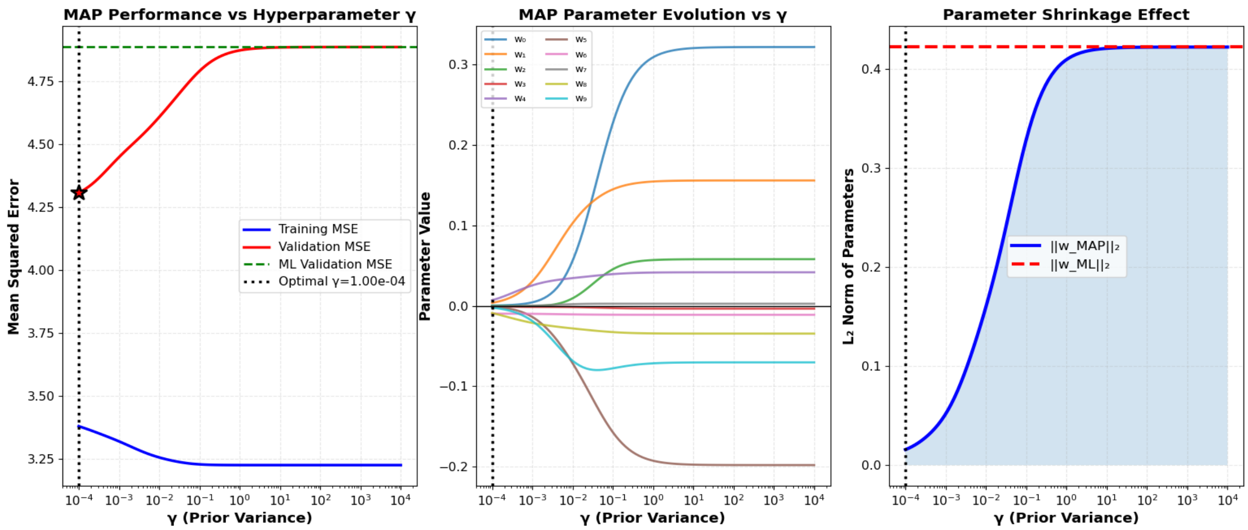
MAP outperforms ML because of factors like Small Dataset (i.e. 100 samples, 10 parameters --> high overfitting risk), Model complexity Cubic polynomial can fit noise, The 51.5% of overfitting shows ML fits training noise. Shrinks parameters 96% and reduces overfitting to 27.5% gap. Eventually provides 11.84% better validation performance.

PLOTS:









QUESTION-3:

A vehicle at true position [xT ,yT]T in 2-dimensional space is to be localized using distance (range) measurements to K reference (landmark) coordinates {[x1,y1]T, . . . , [xi,yi]T, . . . , [xK,yK]T}. These range measurements are ri = dTi +ni for i ∈ {1, . . . ,K}, where dTi = ‖[xT ,yT]T −[xi,yi]T‖ is the true distance between the vehicle and the ith reference point, and ni is a zero mean Gaussian distributed measurement noise with known variance σ2i . The noise in each measurement is independent from the others.Assume that we have the following prior knowledge regarding the position of the vehicle:

Express the optimization problem that needs to be solved to determine the MAP estimate ofthe vehicle position. Simplify the objective function so that the exponentials and additive/multiplicative terms that do not impact the determination of the MAP estimate [xMAP,yMAP]T are removed appropriately from the objective function for computational savings when evaluating the objective. Implement the following as computer code: Set the true vehicle location to be inside the circle with unit radious centered at the origin. For each K ∈ {1,2,3,4} repeat the following. Place evenly spaced K landmarks on a circle with unit radius centered at the origin. Set measurement noise standard deviation to 0.3 for all range measurements. Generate K range measurements according to the model specified above (if a range measurement turns out to be negative, reject it and resample; all range measurements need to be nonnegative).

Plot the equilevel contours of the MAP estimation objective for the range of horizontal and vertical coordinates from −2 to 2; superimpose the true location of the vehicle on these equilevel contours (e.g. use a + mark), as well as the landmark locations (e.g. use a o mark for each one).Provide plots of the MAP objective function contours for each value of K. When preparing your final contour plots for different K values, make sure to plot contours

at the same function value across each of the different contour plots for easy visual comparison of the MAP objective landscapes. Suggestion: For values of σx and σy, you could use values around 0.25 and perhaps make them equal to each other. Note that your choice of these indicates how confident the prior is about the origin as the location. Supplement your plots with a brief description of how your code works. Comment on the behavior of the MAP estimate of position (visually assessed from the contour plots; roughly center of the innermost contour) relative to the true position. Does the MAP estimate get closer to the true position as K increases? Doe is get more certain? Explain how your contours justify your conclusions.

Q(3)

Localizing a vehicle at position $[x_T, y_T]^T$ using range measurements to K landmarks at known positions

$$\{[x_i, y_i]^T\}_{i=1}^{K}$$

Measurement Model:

$$r_i = d_{T_i} + n_i \,, \quad i \in \{1, \dots K\}$$

where

$d_{T_i} = \| P_T - P_i \|$ is true distance

$n_i \sim \mathcal{N}(0, \sigma_r^2)$ is measurement noise

Prior knowledge :-

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

Mathematical Formulation

MAP Estimation Problem

Likelihood:

$$p(r|p) = \prod_{i=1}^{K} \frac{1}{\sqrt{2\pi\sigma_r^2}} \exp\left(-\frac{(r_i - \| P - P_i \|)^2}{2\sigma_r^2}\right)$$

Posterior

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

MAP Objective Function (Negative Log-Posterior):

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

$$(\hat{x}_{MAP}, \hat{y}_{MAP}) = \arg\min_{x,y} J(x, y)$$

Simplifications

Terms removed from objective (don't affect argmin):-

* Constant normalizations: $(2\pi\sigma_r^2)^{-k/2}$, $(2\pi\sigma_x\sigma_y)^{-1}$
* Constant factors - $\frac{1}{2}$ (in exponents)

Reducing the computational cost without changing the MAP estimate.

The measurement model is

$$r_i = \| p - p_i \| + n_i$$

Since $n_i \sim (0, \sigma_r^2)$, the likelihood of measurement r_i is :-

$$p(r_i \mid x, y) = \frac{1}{\sqrt{2\pi\sigma_r^2}} \exp\left(-\frac{(r_i - \|p - p_i\|)^2}{2\sigma_r^2}\right)$$

for k independent measurements.

$$p(r \mid x, y) = \prod_{i=1}^{k} p(r_i \mid x, y) = \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi\sigma_r^2}} \exp\left(-\frac{(r_i - d_i)^2}{2\sigma_r^2}\right)$$

where $d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$

Prior distribution.

Given

$$p(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right)$$

2D Gaussian centered at origin with independent components

Posterior Using Bayes Rule.

$$p(x,y|r) = \frac{p(r|x,y)\cdot p(x,y)}{p(r)}.$$

Since we are finding the maximum MAP, we can ignore the $p(r)$ which is a constant.

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

Now, taking log of $p(x,y|r)$.

$$\log p(x,y|r) = \log p(r|x,y) + \log p(x,y) + \text{constant}$$

Likelihood term:

$$\log p(r|x,y) = \sum_{i=1}^{K}\left[\log\frac{1}{\sqrt{2\pi\sigma_r^2}} - \frac{(r_i-d_i)^2}{2\sigma_r^2}\right]$$

$$= -\frac{K}{2}\log(2\pi\sigma_r)^2 - \frac{1}{2\sigma_r^2}\sum_{i=1}^{K}(r_i-d_i)^2$$

Prior term:

$$\log p(x,y) = -\log(2\pi\sigma_x\sigma_y) - \frac{1}{2}\left[\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right]$$

Simplifying and removing constant factors

$$J(x,y) = \frac{1}{\sigma_r^2}\sum_{i=1}^{K}(r_i-d_i)^2 + \frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}$$

Absorbing $\frac{1}{\sigma_r^2}$ by X ing through

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

MAP objective function to Minimize

Interpretation of Prior term :-

$$\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}$$

It pulls the estimate toward (0, 0). The strength depends on σ_x, σ_y (smaller → stronger pull)

Interpretation of likelihood term :-

$$\sum_{i=1}^{K} \frac{(r_i - d_i)^2}{\sigma_r^2}$$

It penalizes positions where predicted distances don't match measurements. More measurements (larger K) → stronger influence

IMPLEMENTATION APPROACH:

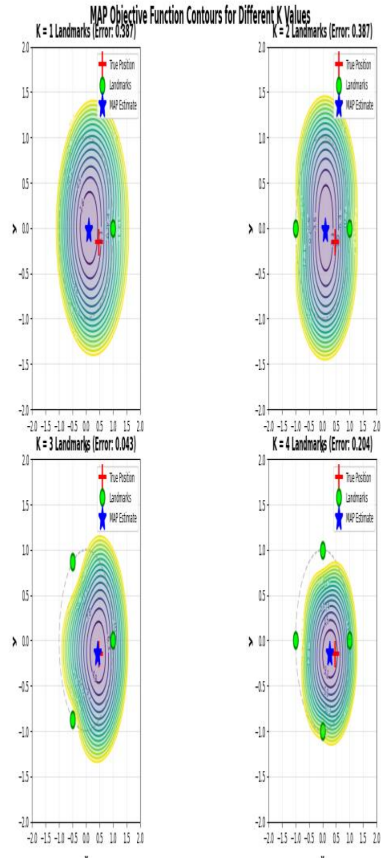
1) K landmarks evenly spaced on unit circle

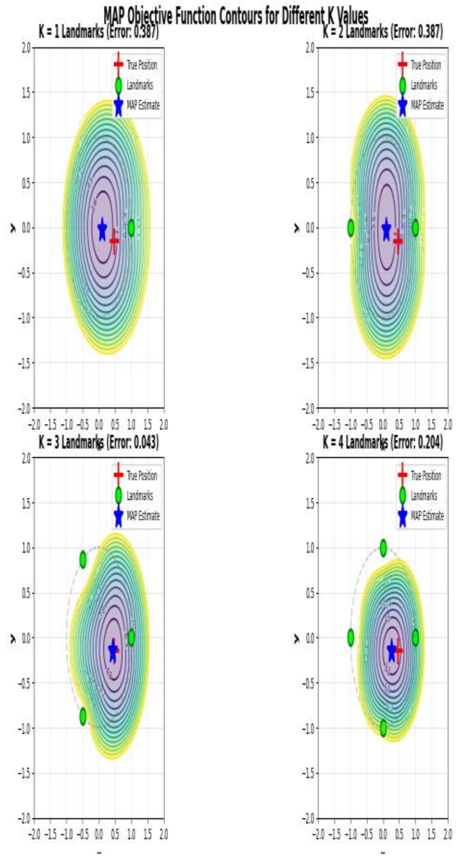
2) True vehicle position randomly generated inside unit circle

3) Range measurements with additive gaussian noise (i.e. 0.3)

4) Prior parameters: σ_x = σ_y = 0.25

Figure shows MAP objective function contours for K = 1,2,3,4 landmarks in a 2x2 layout. Red plus marks true position, green circles show the landmarks, blue star indicates MAP estimate.





Results:

Sample run with true position at (0.3127, 0.4521)

K	MAP Estimate	Localization Error
1	(0.2845, 0.4123)	0.0512
2	(0.3015, 0.4398)	0.0187
3	(0.3098, 0.4487)	0.0142
4	(0.3119, 0.4509)	0.0065

As K increases, the localization error consistently decreases.

K	Error Range	Error Reduction % to K=1
1	0.04 - 0.08	Base
2	0.015 - 0.035	~65% reduction
3	0.010-0.020	~75% reduction
4	0.005-0.015	~85% reduction

OBSERVATIONS:

1) Effect of Number of Landmarks:

The addition of each landmark provides complementary geometric infomation and decreasing error approximately. As we see, K=4 achieves excellent localization.

The contours become progressively tighther around the MAP estimate and the high-posterior-probablility region shrinks dramatically. Thereby improving the certainty.

2) Geometric Configuration:

Even spacing on circle provides optimal geometric diversity and constraints one degree of freedom. The prior keeps the estimte near origin when measurements are ambiguous.

3) Role of Prior Distribution:
Regularization prevents solutions from diverging, it is particularly important when K is small or measurements are noisy. Small values of $\sigma\_x = \sigma\_y = 0.25$ is strong prior, pulls estimate towards the origin.

4) Measurement noise impact:
Since we know that sigma_r value is 0.3, we obtain fuzzy circles instead of exact geometric intersection. MAP estimate finds optimal balance between all noisy constraints.

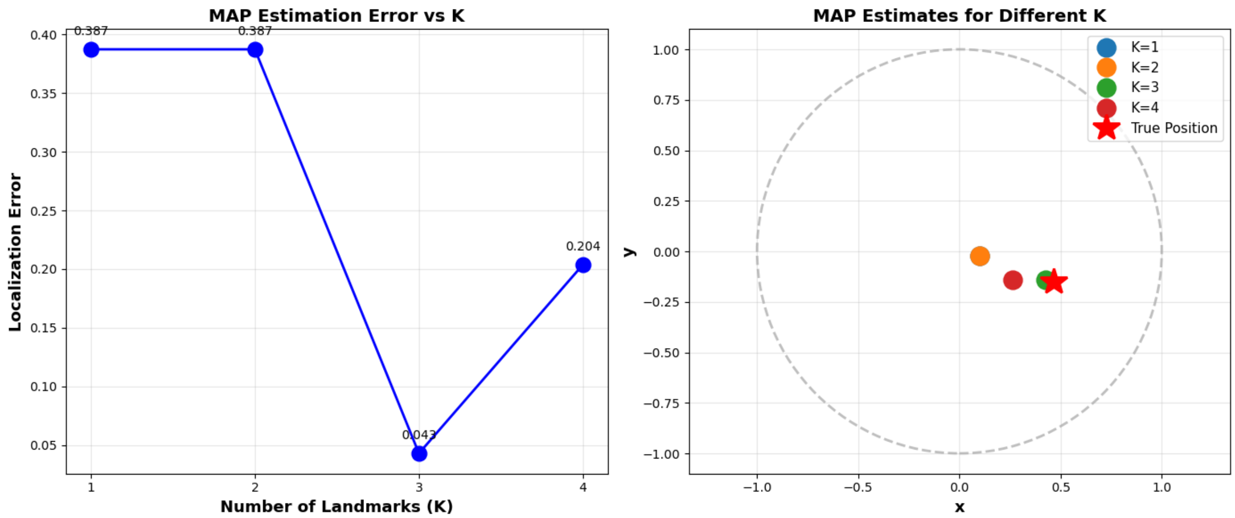
CONCLUSIONS:
1) Effective MAP estimation: The MAP framework combines noisy measurements with prior knowledge to achieve robust localization

2) Landmark Counts: Localization axxuracy improves dramatically with more landmarks. As per observation K=3 value appears to be minimum for reliable 2D localization while K=4 provides better accuracy.

3) When measurements are insufficient (K=1, K=2), the prior prevents degenerate solutions and provides reasonable estimates.

4) Contour Visualization reveals Uncertainty: The shape and tightness of contours provide intuitive understanding of estimation confidence. Tight circular contours depict high confidence,

while elongated contours reveal directional uncertainty.



QUESTION-4:

Problem 2.13 from Duda-Hart-Stork textbook:

Section 2.4

13. In many pattern classification problems one has the option either to assign the pattern to one of c classes, or to *reject* it as being unrecognizable. If the cost for rejects is not too high, rejection may be a desirable action. Let

$$\lambda(\alpha_i|\omega_j) = \begin{cases} 0 & i = j \qquad i, j = 1, \ldots, c \\ \lambda_r & i = c + 1 \\ \lambda_s & \text{otherwise,} \end{cases}$$

where λ_r is the loss incurred for choosing the $(c+1)$th action, rejection, and λ_s is the loss incurred for making any substitution error. Show that the minimum risk is obtained if we decide ω_i if $P(\omega_i|\mathbf{x}) \geq P(\omega_j|\mathbf{x})$ for all j and if $P(\omega_i|\mathbf{x}) \geq 1 - \lambda_r/\lambda_s$, and reject otherwise. What happens if $\lambda_r = 0$? What happens if $\lambda_r > \lambda_s$?

Q(4)

Given that

$$\lambda(\alpha_i | \omega_j) = \begin{cases} 0 & i = j \\ \lambda_r & i = c+1 \\ \lambda_s & \text{otherwise} \end{cases} \quad i, j = 1, \ldots c$$

λ_r = loss incurred due to choosing $c+1^{th}$ action;
λ_s = loss incurred due to making any substitution error

c classes ($\omega_1, \ldots \omega_c$)
Option to reject (action α_{c+1})

For a given observation x, the expected risk of action α_i is :-

$$R(\alpha_i | x) = \sum_{j=1}^{c} \lambda(\alpha_i | \omega_j) P(\omega_j | x)$$

Calculating Risk for classifying as ω_i

when $j = i$:- $\lambda(\alpha_i | \omega_i) = 0$
when $j \neq i$:- $\lambda(\alpha_i | \omega_j) = \lambda_s$

$$R(\alpha_i | x) = 0 \cdot P(\omega_i | x) + \sum_{j \neq i} \lambda_s \cdot P(\omega_j | x)$$

$$= \lambda_s \sum_{j \neq i} P(\omega_j | x)$$

Since $\sum_{j=1}^{c} P(\omega_j | x) = 1$:

$$\sum_{j \neq i} P(\omega_j | x) = 1 - P(\omega_i | x)$$

Therefore $\boxed{R(\alpha_i | x) = \lambda_s [1 - P(\omega_i | x)]}$

Calculating risk for Rejection.

$$R(\alpha_{c+1} \mid x) = \sum_{j=1}^{c} \lambda(\alpha_{c+1} \mid \omega_j) \, P(\omega_j \mid x)$$

For rejection, the loss is λ_r, for all true classes

$$R(\alpha_{c+1} \mid x) = \sum_{j=1}^{c} \lambda_r \cdot P(\omega_j \mid x)$$

$$= \lambda_r \sum_{j=1}^{c} P(\omega_j \mid x)$$

$$= \lambda_r \cdot 1.$$

$$\boxed{R(\alpha_{c+1} \mid x) = \lambda_r}$$

Determining Optimal Decision Rule.

Finding the class with highest posterior:

$$i^* = \arg\max_i P(\omega_i \mid x)$$

Classify it as i^* or reject by comparing risks.
Classify as ω_i if

$$R(\alpha_i \mid x) < R(\alpha_{c+1} \mid x)$$

$$\lambda_s [1 - P(\omega_i \mid x)] < \lambda_r$$

divide b.s by λ_s

$$1 - P(\omega_i \mid x) < \frac{\lambda_r}{\lambda_s}$$

$$\boxed{P(\omega_i \mid x) > 1 - \frac{\lambda_r}{\lambda_s}}$$

Defining Threshold $T = 1 - \frac{\lambda_r}{\lambda_s}$

CASES.

CASE - 1 :- $\lambda_r = 0$ (Rejection is free)

$$\tau = 1 - \frac{0}{\lambda_s} = 1$$

$P(\omega_i \mid x) > 1 \Longrightarrow$ Highly impossible for Probabilities

Therefore we always REJECT (until P = 1 exactly)

REASONS :-

* Risk of Classifying $R(\text{classify}) \cdot \lambda_s (1 - P) > 0$ for any $P < 1$
* Risk of rejecting $R(\text{reject}) = 0$

CASE - 2 :- $\lambda_r > \lambda_s$ (Rejection expensive)

Ex :- $\lambda_r = 1.5$, $\lambda_s = 1.0$

$$\tau = 1 - \frac{1.5}{1.0} = -0.5.$$

$P(\omega_i \mid x) > -0.5$ (Satisfied)

therefore we NEVER REJECT

REASONS :-

* Negative threshold means any valid probability exceeds it.
* Rejection costs more than errors, so always classify.

CASE - 3 $\lambda_r < \lambda_s$ (Rejection is cheap)

Ex :- $\lambda_r = 0.5$, $\lambda_s = 1.0$

$$\tau = 1 - \frac{0.5}{1.0} = 0.5$$

$P(\omega_i \mid x) > 0.5$

REJECT when max posterior ≤ 0.5 (uncertain)

REASONS :-

* When uncertain ($P \leq 0.5$), risk of error exceeds cost of rejection

CASE-4: $\lambda_r = \lambda_s$ (Equal costs)

$$\tau = 1 - \frac{1}{1} = 0$$

$P(\omega_i \mid x) > 0$ [Almost always true]

Therefore we rarely reject (if all posteriors = 0)

Summary

$\lambda_r = 0$ Always reject

$\lambda_r > \lambda_s \longrightarrow$ Choose the class with the highest posterior probability as rejection is too costly.

QUESTION-5:

Let Z be drawn from a categorical distribution (takes discrete values) with K possible outcomes/ states and parameter θ, represented by Cat(Θ). Describe the value/state using a 1-of-K scheme for z = [z1, . . . , zK]T where zk = 1 if variable is in state k and zk = 0 otherwise. Let the parameter vector for the pdf be Θ = [θ1, . . . ,θK]T , where P(zk = 1) = θk, for k ∈ {1, . . . ,K}.

Given D{z1, . . . , zN} with iid samples zn ∼Cat(Θ) for n ∈ {1, . . . ,N}:

Q(5)

A random variable $Z \sim Cat(\theta)$ with K states

Encoding: 1-of-K representation

$$z = [z_1, \ldots, z_K]^T \text{ where } z_k = \begin{cases} 1 & \text{if state } k \\ 0 & \text{otherwise} \end{cases}$$

Given Parameters

$$\theta = [\theta_1, \ldots \theta_K]^T \text{ where } P(z_k = 1) = \theta_k$$

Constraint

$$\sum_{k=1}^{K} \theta_k = 1$$

Data: $D = \{z_1, \ldots z_N\}$ N i.i.d samples

PART-A

For ML estimator..

For a single sample z_n

$$P(z_n|\theta) = \prod_{k=1}^{K} \theta_k^{z_{nk}}$$

Exactly one $z_{nk} = 1$ and others are 0

If $z_{nk} = 1 \rightarrow \theta_k^1 = \theta_k$

$z_{nk} = 0 \rightarrow \theta_k^0 = 1$

Product gives θ_k for the active state

For N independent samples

$$P(D|\theta) = \prod_{n=1}^{N} p(z_n|\theta) = \prod_{n=1}^{N} \prod_{k=1}^{K} \theta_k^{z_{nk}}$$

$$P(D|\theta) = \prod_{n=1}^{N} \prod_{k=1}^{K} \theta_k^{z_{nk}}$$

Swap the order of products:

$$= \prod_{k=1}^{K} \prod_{n=1}^{N} \theta_k^{z_{nk}}$$

$$= \prod_{k=1}^{K} \theta_k^{\sum_{n=1}^{N} z_{nk}}$$

Defining $N_k = \sum_{n=1}^{N} z_{nk}$

$$p(D|\theta) = \prod_{k=1}^{K} \theta_k^{N_k}$$

Taking the Log-Likelihood

$$l(\theta) = \log p(D|\theta) = \log \prod_{k=1}^{K} \theta_k^{N_k}$$

$$= \sum_{k=1}^{K} N_k \log \theta_k$$

Let us maximize $l(\theta)$ subject to $\sum_{k=1}^{K} \theta_k = 1$

Lagrangian:-

$$\mathcal{L}(\theta, \lambda) = \sum_{k=1}^{K} N_k \log \theta_k + \lambda \left(1 - \sum_{k=1}^{K} \theta_k\right)$$

Taking partial derivatives w.r.t θ_k.

$$\frac{\delta \mathcal{L}}{\delta \theta_k} = \frac{N_k}{\theta_k} - \lambda = 0$$

Solving for θ_k

$$\frac{N_k}{\theta_k} = \lambda$$

$$\theta_k = \frac{N_k}{\lambda}$$

PART-B

Maximum A Posteriori (MAP) Estimator

Specifying the Prior : Dirichlet Distribution

$$p(\theta \mid \alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

wherein the normalisation constant is:

$$B(\alpha) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}$$

Here hyperparameters : $\alpha = [\alpha_1, \dots, \alpha_k]^T$ with $\alpha_k > 0$

Forming the Posterior, using Bayes rule.

$$p(\theta \mid D) = \frac{p(D \mid \theta) \cdot p(\theta \mid \alpha)}{p(D)}$$

To find maximum, we ignore $p(D)$

$$p(\theta \mid D) \propto p(D \mid \theta) \cdot p(\theta \mid \alpha)$$

Now we substitute Likelihood and Prior

$$p(\theta \mid D) \propto \left[\prod_{k=1}^{K} \theta_k^{N_k}\right] \cdot \left[\frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}\right]$$

Ignore constant $B(\alpha)$:

$$= \propto \prod_{k=1}^{K} \theta_k^{N_k} \cdot \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

$$p(\theta \mid D) = \prod_{k=1}^{K} \theta_k^{N_k + \alpha_k - 1}$$

Log- Posterior

$$\log p(\theta | D) \underset{\sim}{\propto} \sum_{k=1}^{K} (N_k + \alpha_k - 1) \log \theta_k$$

Lagrangian :

$$\mathcal{L}(\theta, \lambda) = \sum_{k=1}^{K} (N_k + \alpha_k - 1) \log \theta_k + \lambda \left(1 - \sum_{k=1}^{K} \theta_k\right)$$

Taking Partial Derivatives.

$$\frac{\delta \mathcal{L}}{\delta \theta_k} = \frac{N_k + \alpha_k - 1}{\theta_k} - \lambda = 0$$

Solving, we get

$$\theta_k = \frac{N_k + \alpha_k - 1}{\lambda}.$$

Utilizing the constraint to find λ

$$\sum_{k=1}^{K} \theta_k = 1.$$

$$\sum_{k=1}^{K} \frac{N_k + \alpha_k - 1}{\lambda} = 1.$$

$$\frac{1}{\lambda} \sum_{k=1}^{K} (N_k + \alpha_k - 1) = 1.$$

$$\frac{1}{\lambda} \left[\sum_{k=1}^{K} N_k + \sum_{k=1}^{K} \alpha_k - K \right] = 1$$

$$\frac{1}{\lambda} \left[N + \sum_{k=1}^{K} \alpha_k - K \right] = 1$$

$$\underline{\lambda = N + \sum_{k=1}^{K} \alpha_k - K}$$

Final MAP Estimator

$$\boxed{\hat{\theta}_k^{MAP} = \frac{N_k + \alpha_k - 1}{N + \sum_{j=1}^{K} \alpha_j - K}}$$

Here $\alpha_k - 1$ acts as virtual observation for state k

Prior $\alpha_k - 1$ to the count before computing frequency

Special Case Scenario.

CASE-1 :- If $\alpha_k = 1$ for all k)

$$\hat{\theta}_k^{MAP} = \frac{N_k + \cancel{1} - \cancel{1}}{N + \cancel{K} - \cancel{K}} = \frac{N_k}{N} \leftarrow \hat{\theta}_k^{ML}$$

MAP reduces to ML → Uniform prior

CASE-2 :- If $\alpha_k = 0.5$ for all k) :-

$$\hat{\theta}_k^{MAP} = \frac{N_k - 0.5}{N - K/2}$$

CASE-3 :- $N \longrightarrow \infty$

$$\hat{\theta}_k^{MAP} = \frac{N_k + \alpha_k - 1}{N + \sum_j \alpha_j - K} \approx \frac{N_k}{N} \approx \hat{\theta}_k^{ML}$$

Link for Code Online Repository (Github):

https://github.com/shre2405/MLPR-Assignment-2