Machine Learning

Homework Assignment 5

Name: Ahmadreza Farvardin
Student ID: 610301221

July 10, 2025

Problems

Problem 1

Part (a)

To model the parameters of a Gaussian mixture model (GMM) using a multi-layer neural network for 1D i.i.d. data samples x, the network's output should represent:

1. **Mixing Coefficients (π_k):**
   - A softmax output to ensure $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k > 0$
   - Size: K units (for K components)
2. **Means (μ_k):**
   - Linear output (unconstrained real values)
   - Size: K units
3. **Variances (σ_k^2):**
   - Softplus/exp output to ensure positivity
   - Size: K units (for diagonal covariance in 1D)

The network takes x as input and outputs all GMM parameters (π_k, μ_k, σ_k^2) simultaneously. For batch processing, outputs are structured as tensors with dimensions matching the number of components K.

Total Output Units: $3K$ (for K-component GMM)

Part (b)

Recommended Activation Functions for GMM Parameters in Neural Networks

For a neural network estimating Gaussian Mixture Model (GMM) parameters, the output layers should use the following activation functions to ensure valid parameter values:

1. **Mixture Weights (π_k)**
   - **Constraint:** Must be probabilities ($0 \leq \pi_k \leq 1$) summing to 1
   - **Activation: Softmax**
     - Ensures $\sum \pi_k = 1$ and $\pi_k > 0$
     - Example: For 3 components, outputs $[0.2, 0.5, 0.3]$

2. **Means (μ_k)**
   - **Constraint:** Can be any real number
   - **Activation: Linear** (no activation)
     - Allows $\mu_k \in (-\infty, +\infty)$
     - Example: Outputs $[-1.2, 0.5, 3.1]$ for 3 components

3. **Variances (σ_k^2)**
   - **Constraint:** Must be strictly positive
   - **Recommended Activations:**
     - **Softplus:** $\sigma_k^2 = \log(1 + \exp(z))$
       (Smooth, avoids exact zeros)
     - **Exponential:** $\sigma_k^2 = \exp(z)$
       (More aggressive for small values)
     - Example: Converts network output $[-0.3, 1.2]$ to $[0.74, 3.32]$

Special Cases

For full covariance matrices (multidimensional case):

- Use **Cholesky decomposition** with:
  - Linear activation for off-diagonal elements
  - Softplus/Exp for diagonal elements

Note

The network should output all parameters simultaneously in a vector of length $3K$ (for K components):

$$[\pi_1 ... \pi_K, \mu_1 ... \mu_K, \sigma_1^2 ... \sigma_K^2]$$

These choices guarantee:

1. Valid probability distribution (π_k)

2. Unconstrained means
3. Positive-definite covariance

while remaining fully differentiable for backpropagation.

Part (c)

Loss Function for Neural Network Estimating GMM Parameters

For a neural network predicting Gaussian Mixture Model (GMM) parameters, the appropriate loss function is the negative log-likelihood (NLL) of the data under the predicted mixture distribution:

Mathematical Formulation

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\sum_{k=1}^{K}\pi_k \cdot \mathcal{N}(x_i|\mu_k,\sigma_k^2)\right)$$

Where:

- π_k, μ_k, σ_k^2 are the network's outputs (after proper activations)
- $\mathcal{N}(x_i|\mu_k,\sigma_k^2)$ is the Gaussian PDF evaluated at x_i
- N is the number of samples in the batch

Key Properties

- **Interpretation:** Minimizing this loss is equivalent to maximizing the likelihood of the data.
- **Numerical Stability:**
  - Compute the log-sum-exp of component log-probabilities to avoid underflow:

$$\log\sum_{k=1}^{K}\exp(\log\pi_k + \log\mathcal{N}(x_i|\mu_k,\sigma_k^2))$$

- **Gradient Flow:** The loss is differentiable w.r.t. all parameters (π_k, μ_k, σ_k^2), enabling backpropagation.

Problem 2

Notice: The order of α and β is different from the one in the question, but it does not affect the solution.

Part (a)

We have a random variable x generated through a two-level hierarchical process involving two independent binary variables y and z.

Given:

- $P(y = 1) = \beta$, $P(y = 0) = 1 - \beta$
- $P(z = 1) = \alpha$, $P(z = 0) = 1 - \alpha$ (independent of y)

Steps of Solution

1. Joint Distribution Decomposition

The joint distribution factors as:

$$P(x, y, z) = P(x|y, z) \cdot P(z|y) \cdot P(y)$$

Since z is independent of y, $P(z|y) = P(z)$

2. Component Evaluation

- $P(y)$: $P(y = 1) = \beta$, $P(y = 0) = 1 - \beta$
- $P(z)$: $P(z = 1) = \alpha$, $P(z = 0) = 1 - \alpha$
- $P(x|y, z)$:
  - For $y = 0$:
    - $z = 0$: $x \sim \text{Exp}(\lambda_1) \rightarrow P(x|y = 0, z = 0) = \lambda_1 e^{-\lambda_1 x}$
    - $z = 1$: $x \sim \text{Exp}(\lambda_2) \rightarrow P(x|y = 0, z = 1) = \lambda_2 e^{-\lambda_2 x}$
  - For $y = 1$:
    - $z = 0$: $x \sim N(\mu_1, 1) \rightarrow P(x|y = 1, z = 0) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu_1)^2}{2}}$
    - $z = 1$: $x \sim N(\mu_2, 1) \rightarrow P(x|y = 1, z = 1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu_2)^2}{2}}$

3. Final Joint Distribution

$$P(x,y,z) = \begin{cases} (1-\beta)(1-\alpha)\lambda_1 e^{-\lambda_1 x} & \text{if } y=0, z=0 \\ (1-\beta)\alpha\lambda_2 e^{-\lambda_2 x} & \text{if } y=0, z=1 \\ \beta(1-\alpha)\frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu_1)^2}{2}} & \text{if } y=1, z=0 \\ \beta\alpha\frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu_2)^2}{2}} & \text{if } y=1, z=1 \end{cases}$$

Part (b)

Complete Data Log-Likelihood Derivation

1. Complete Data Likelihood

For N i.i.d. observations:

$$\mathcal{L}_{\text{complete}} = \prod_{i=1}^{N} P(x_i, y_i, z_i)$$

Expanding:

$$\begin{aligned} \mathcal{L}_{\text{complete}} = & \prod_{i:y_i=0,z_i=0} \left[(1-\beta)(1-\alpha)\lambda_1 e^{-\lambda_1 x_i}\right] \\ & \times \prod_{i:y_i=0,z_i=1} \left[(1-\beta)\alpha\lambda_2 e^{-\lambda_2 x_i}\right] \\ & \times \prod_{i:y_i=1,z_i=0} \left[\beta(1-\alpha)\frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i-\mu_1)^2}{2}}\right] \\ & \times \prod_{i:y_i=1,z_i=1} \left[\beta\alpha\frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i-\mu_2)^2}{2}}\right] \end{aligned}$$

2. Log-Likelihood Derivation

Taking natural logarithm:

$$\begin{aligned} \ell_{\text{complete}} = & \sum_{i:y_i=0,z_i=0} [\ln(1-\beta) + \ln(1-\alpha) + \ln\lambda_1 - \lambda_1 x_i] \\ & + \sum_{i:y_i=0,z_i=1} [\ln(1-\beta) + \ln\alpha + \ln\lambda_2 - \lambda_2 x_i] \\ & + \sum_{i:y_i=1,z_i=0} \left[\ln\beta + \ln(1-\alpha) - \frac{1}{2}\ln(2\pi) - \frac{(x_i-\mu_1)^2}{2}\right] \\ & + \sum_{i:y_i=1,z_i=1} \left[\ln\beta + \ln\alpha - \frac{1}{2}\ln(2\pi) - \frac{(x_i-\mu_2)^2}{2}\right] \end{aligned}$$

3. Final Expression

Let $N_{ab} := \#\{i : y_i = a, z_i = b\}$. Then:

$$\begin{aligned}\ell_{\text{complete}} =& N_{00}\left[\ln(1-\beta) + \ln(1-\alpha) + \ln\lambda_1\right] \\ &+ N_{01}\left[\ln(1-\beta) + \ln\alpha + \ln\lambda_2\right] \\ &+ N_{10}\left[\ln\beta + \ln(1-\alpha) - \frac{1}{2}\ln(2\pi)\right] \\ &+ N_{11}\left[\ln\beta + \ln\alpha - \frac{1}{2}\ln(2\pi)\right] \\ &- \lambda_1 \sum_{i:y_i=0,z_i=0} x_i - \lambda_2 \sum_{i:y_i=0,z_i=1} x_i \\ &- \frac{1}{2}\sum_{i:y_i=1,z_i=0}(x_i-\mu_1)^2 - \frac{1}{2}\sum_{i:y_i=1,z_i=1}(x_i-\mu_2)^2\end{aligned}$$

where "Group ab" denotes observations where $y = a, z = b$.

Part (c)

E-Step for Hidden Variables (y_i, z_i) in EM Algorithm

1. Goal of the E-Step

Compute posterior probabilities for each observation x_i:

- $\gamma_{i,00} = P(y_i = 0, z_i = 0 \mid x_i)$
- $\gamma_{i,01} = P(y_i = 0, z_i = 1 \mid x_i)$
- $\gamma_{i,10} = P(y_i = 1, z_i = 0 \mid x_i)$
- $\gamma_{i,11} = P(y_i = 1, z_i = 1 \mid x_i)$

2. Bayes' Rule Application

For each $(a, b) \in \{0, 1\}^2$:

$$\gamma_{i,ab} = \frac{P(x_i \mid y_i = a, z_i = b) \cdot P(y_i = a, z_i = b)}{P(x_i)}$$

where:

$$P(x_i) = \sum_{a,b} P(x_i \mid y_i = a, z_i = b) \cdot P(y_i = a, z_i = b)$$

3. Component Probabilities

Joint probabilities:

$$
\begin{aligned}
P(y_i = 0, z_i = 0) &= (1-\beta)(1-\alpha) \\
P(y_i = 0, z_i = 1) &= (1-\beta)\alpha \\
P(y_i = 1, z_i = 0) &= \beta(1-\alpha) \\
P(y_i = 1, z_i = 1) &= \beta\alpha
\end{aligned}
$$

Conditional likelihoods:

$$
\begin{aligned}
P(x_i \mid y_i = 0, z_i = 0) &= \lambda_1 e^{-\lambda_1 x_i} \\
P(x_i \mid y_i = 0, z_i = 1) &= \lambda_2 e^{-\lambda_2 x_i} \\
P(x_i \mid y_i = 1, z_i = 0) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i-\mu_1)^2}{2}} \\
P(x_i \mid y_i = 1, z_i = 1) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i-\mu_2)^2}{2}}
\end{aligned}
$$

4. Posterior Probabilities

$$
\begin{aligned}
\gamma_{i,00} &= \frac{(1-\beta)(1-\alpha)\lambda_1 e^{-\lambda_1 x_i}}{P(x_i)} \\
\gamma_{i,01} &= \frac{(1-\beta)\alpha\lambda_2 e^{-\lambda_2 x_i}}{P(x_i)} \\
\gamma_{i,10} &= \frac{\beta(1-\alpha)\frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i-\mu_1)^2}{2}}}{P(x_i)} \\
\gamma_{i,11} &= \frac{\beta\alpha\frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i-\mu_2)^2}{2}}}{P(x_i)}
\end{aligned}
$$

where $P(x_i)$ is the sum of all numerators.

5. Expected Values Computation

Expected counts:

$$\mathbb{E}[N_{00}] = \sum_{i=1}^{N} \gamma_{i,00}$$
$$\mathbb{E}[N_{01}] = \sum_{i=1}^{N} \gamma_{i,01}$$
$$\mathbb{E}[N_{10}] = \sum_{i=1}^{N} \gamma_{i,10}$$
$$\mathbb{E}[N_{11}] = \sum_{i=1}^{N} \gamma_{i,11}$$

Auxiliary expected values:

$$\mathbb{E}\left[\sum_{i:y_i=0,z_i=0} x_i\right] = \sum_{i=1}^{N} \gamma_{i,00} x_i$$
$$\mathbb{E}\left[\sum_{i:y_i=0,z_i=1} x_i\right] = \sum_{i=1}^{N} \gamma_{i,01} x_i$$
$$\mathbb{E}\left[\sum_{i:y_i=1,z_i=0} (x_i - \mu_1)^2\right] = \sum_{i=1}^{N} \gamma_{i,10} (x_i - \mu_1)^2$$
$$\mathbb{E}\left[\sum_{i:y_i=1,z_i=1} (x_i - \mu_2)^2\right] = \sum_{i=1}^{N} \gamma_{i,11} (x_i - \mu_2)^2$$

Part (d)

M-Step: Updating Parameters β, μ_2, and λ_2

1. Update Rule for β

Parameter β represents $P(y_i = 1)$.

Update Formula:

$$\beta^{\text{new}} = \frac{\mathbb{E}[N_{10} + N_{11}]}{N} = \frac{\sum_{i=1}^{N} (\gamma_{i,10} + \gamma_{i,11})}{N}$$

where:

- $\gamma_{i,10} = P(y_i = 1, z_i = 0 \mid x_i)$
- $\gamma_{i,11} = P(y_i = 1, z_i = 1 \mid x_i)$

Derivation: The update maximizes the expected complete-data log-likelihood with respect to β, representing the expected proportion of observations where $y_i = 1$.

2. Update Rule for μ_2

Parameter μ_2 is the mean of the Gaussian distribution when $(y_i = 1, z_i = 1)$.

Update Formula:

$$\mu_2^{\text{new}} = \frac{\sum_{i=1}^{N} \gamma_{i,11} x_i}{\sum_{i=1}^{N} \gamma_{i,11}}$$

Derivation: This update maximizes the Gaussian component of the log-likelihood, weighted by the responsibilities $\gamma_{i,11}$. It represents the weighted average of observations assigned to $(y_i = 1, z_i = 1)$.

3. Update Rule for λ_2

Parameter λ_2 is the rate of the exponential distribution when $(y_i = 0, z_i = 1)$.

Update Formula:

$$\lambda_2^{\text{new}} = \frac{\sum_{i=1}^{N} \gamma_{i,01}}{\sum_{i=1}^{N} \gamma_{i,01} x_i}$$

Derivation: This update maximizes the exponential component of the log-likelihood, weighted by the responsibilities $\gamma_{i,01}$. It is the inverse of the weighted mean of observations assigned to $(y_i = 0, z_i = 1)$.

Summary of Dependencies

- For β: Requires $\gamma_{i,10}$, $\gamma_{i,11}$ from E-step
- For μ_2: Requires $\gamma_{i,11}$, x_i from data
- For λ_2: Requires $\gamma_{i,01}$, x_i from data

Each update maximizes the corresponding term in the expected complete-data log-likelihood while keeping other parameters fixed.

Remark on Derivation

The update rules are derived by maximizing the expected complete-data log-likelihood:

$$Q(\theta|\theta^{\text{old}}) = \mathbb{E}_{y,z|x,\theta^{\text{old}}}[\log P(x, y, z|\theta)]$$

with respect to each parameter while holding others fixed.

1. Update Rule for β

Derivation: The term involving β in the expected log-likelihood is:

$$\sum_{i=1}^{N} [\gamma_{i,10} \ln \beta + \gamma_{i,11} \ln \beta + \gamma_{i,00} \ln(1 - \beta) + \gamma_{i,01} \ln(1 - \beta)]$$

Taking derivative with respect to β and setting to zero:

$$\frac{\sum_i(\gamma_{i,10}+\gamma_{i,11})}{\beta}-\frac{\sum_i(\gamma_{i,00}+\gamma_{i,01})}{1-\beta}=0$$

Solving yields:

$$\beta^{\text{new}}=\frac{\sum_{i=1}^{N}(\gamma_{i,10}+\gamma_{i,11})}{N}$$

2. Update Rule for μ_2

Derivation: The term involving μ_2 is:

$$\sum_{i=1}^{N}\gamma_{i,11}\left[-\frac{(x_i-\mu_2)^2}{2}\right]$$

Taking derivative and setting to zero:

$$\sum_i\gamma_{i,11}(x_i-\mu_2)=0$$

Solving yields:

$$\mu_2^{\text{new}}=\frac{\sum_{i=1}^{N}\gamma_{i,11}x_i}{\sum_{i=1}^{N}\gamma_{i,11}}$$

3. Update Rule for λ_2

Derivation: The term involving λ_2 is:

$$\sum_{i=1}^{N}\gamma_{i,01}\left[\ln\lambda_2-\lambda_2x_i\right]$$

Taking derivative and setting to zero:

$$\frac{\sum_i\gamma_{i,01}}{\lambda_2}-\sum_i\gamma_{i,01}x_i=0$$

Solving yields:

$$\lambda_2^{\text{new}}=\frac{\sum_{i=1}^{N}\gamma_{i,01}}{\sum_{i=1}^{N}\gamma_{i,01}x_i}$$

Problem 3

Part (a)

Computational Complexity Comparison

General Steps in Agglomerative Clustering

Both single linkage and complete linkage follow these steps:

1. **Initial Distance Computation:** $O(n^2)$ for n points
2. **Cluster Initialization:** $O(n)$
3. **Iterative Merging:** $O(n)$ iterations

Linkage Definitions

- **Single Linkage:**
$$d(C_1, C_2) = \min_{x \in C_1, y \in C_2} d(x, y)$$
- **Complete Linkage:**
$$d(C_1, C_2) = \max_{x \in C_1, y \in C_2} d(x, y)$$

Complexity Analysis

1. **Distance Matrix Updates:**
   - When merging clusters A and B into cluster AB:
$$\begin{aligned} \text{Single:} \quad & d(AB, C) = \min(d(A, C), d(B, C)) \\ \text{Complete:} \quad & d(AB, C) = \max(d(A, C), d(B, C)) \end{aligned}$$
   - Both require $O(1)$ per update
   - Total updates: $O(n^2)$ over all iterations
2. **Finding Closest Clusters:**
   - Naive implementation: $O(n^2)$ per iteration
   - Optimized (with priority queue): $O(n \log n)$ per iteration
   - Total iterations: $O(n)$

Overall Complexity

- **Space Complexity:** $O(n^2)$ for distance matrix
- **Time Complexity:**
  - Naive implementation: $O(n^3)$
  - Optimized implementation: $O(n^2 \log n)$

Conclusion

Single linkage and complete linkage have identical asymptotic complexity:

- Both require $O(n^2)$ space
- Both require $O(n^3)$ time (naive) or $O(n^2 \log n)$ time (optimized)
- The only difference is in the min/max operation used in distance updates

Part (b)

Which method is more robust to outliers?

Complete linkage is more robust to outliers than single linkage.

Reasoning

1. Single Linkage

- Uses minimum distance between clusters:

$$d(C_1, C_2) = \min_{x \in C_1, y \in C_2} d(x, y)$$

- **Weakness to Outliers:**
  - Susceptible to "chaining effect"
  - Single outlier can cause premature cluster merging
  - Only considers closest pair of points

2. Complete Linkage

- Uses maximum distance between clusters:

$$d(C_1, C_2) = \max_{x \in C_1, y \in C_2} d(x, y)$$

- **Robustness to Outliers:**
  - Considers worst-case separation
  - Requires all points to be relatively close for merging
  - Individual outliers cannot force cluster merging

Therefore, when dealing with datasets containing potential outliers, complete linkage is the more reliable choice.

Part (c)

Complete Linkage Hierarchical Clustering

Given Points:

- A(0, 0.5)
- B(0.5, 0)
- C(3, 1)
- D(3.5, 1)
- E(3, 0.5)
- F(2, 2)

Pairwise Euclidean Distances:

	A	B	C	D	E	F
A	0	0.71	3.04	3.54	3.04	2.06
B	0.71	0	2.92	3.43	2.55	2.50
C	3.04	2.92	0	0.5	0.5	1.41
D	3.54	3.43	0.5	0	0.71	1.80
E	3.04	2.55	0.5	0.71	0	1.80
F	2.06	2.50	1.41	1.80	1.80	0

Clustering Steps:

1. **First Merge:** C, E (d = 0.5)
   - Clusters: {A}, {B}, {C,E}, {D}, {F}
   - Updated distances:

$$d(\{C,E\},D) = \max(0.5, 0.71) = 0.71$$
$$d(\{C,E\},F) = \max(1.41, 1.80) = 1.80$$
$$d(\{C,E\},A) = \max(3.04, 3.04) = 3.04$$
$$d(\{C,E\},B) = \max(2.92, 2.55) = 2.92$$

2. **Second Merge:** {C,E}, D (d = 0.71)
   - Clusters: {A}, {B}, {C,E,D}, {F}
   - Updated distances:

$$d(\{C,E,D\},F) = \max(1.41, 1.80, 1.80) = 1.80$$
$$d(\{C,E,D\},A) = \max(3.04, 3.04, 3.54) = 3.54$$
$$d(\{C,E,D\},B) = \max(2.92, 2.55, 3.43) = 3.43$$

3. **Third Merge:** A, B (d = 0.71)
    - Clusters: {A,B}, {C,E,D}, {F}
4. **Fourth Merge:** {C,E,D}, F (d = 1.80)
    - Clusters: {A,B}, {C,E,D,F}
5. **Final Merge:** {A,B}, {C,E,D,F} (d = 3.54)

Merge Summary:

Step	Merge	Distance
1	C, E	0.5
2	(C,E), D	0.71
3	A, B	0.71
4	(C,E,D), F	1.80
5	(A,B), (C,E,D,F)	3.54

Dendrogram:

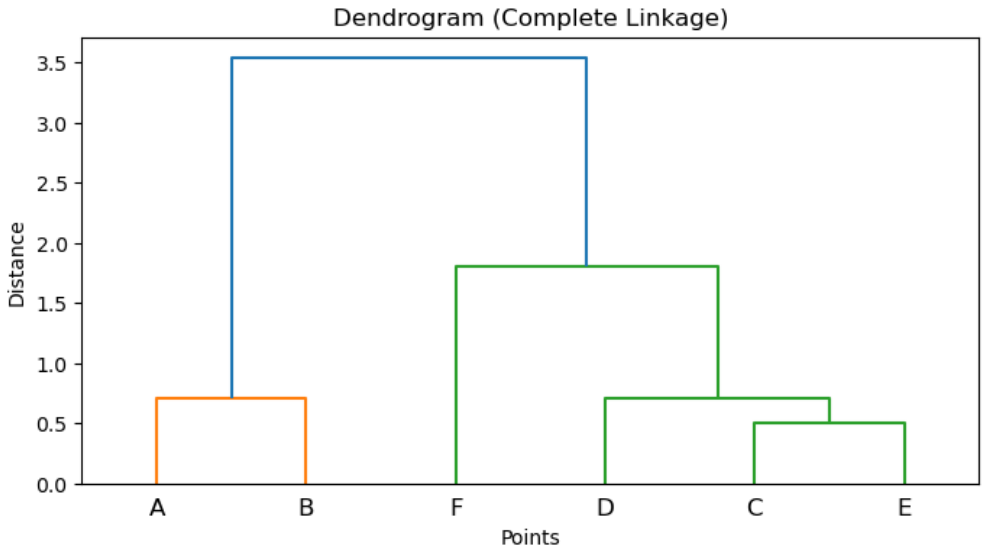

Figure 3.1: Hierarchical Clustering Dendrogram

Problem 4

Part (a)

When is choosing the right ϵ value more challenging?
Choosing ϵ in DBSCAN is particularly challenging in two main scenarios:

1. Data with Varying Densities

- **Challenge:** DBSCAN uses a single ϵ value globally
- **Problems:**
  - Small ϵ: May correctly identifies dense clusters but misses sparse ones (breaking them into noise).
  - Large ϵ: Merges distinct dense clusters or includes noise
- **Reasoning:** Fixed density threshold (ϵ, MinPts) cannot accommodate naturally varying cluster densities

2. High-Dimensional Data

- **Challenge:** Curse of dimensionality affects distance metrics
- **Problems:**
  - Too small ϵ: Most points become noise
  - Too large ϵ: All points merge into one cluster
- **Reasoning:** Euclidean distances become less meaningful in high dimensions, making ϵ harder to interpret

Part (b)

Given MinPts = 3, which plot requires larger ϵ to produce two clusters?
Plot 2 requires a larger ϵ value.

Reasoning

- **Plot 1:**
  - Both inner and outer clusters are dense
  - Smaller ϵ sufficient to form clusters
- **Plot 2:**
  - Inner points are sparsely distributed
  - Outer circle is dense
  - Larger ϵ needed to connect sparse inner points into a cluster

Conclusion: Plot 2 needs larger ϵ to connect the sparse inner points while the dense outer circle forms naturally.

Part (c)

How will ϵ change when MinPts = 1 compared to part (b)?

General Effect of MinPts = 1

With MinPts = 1:

- Each point needs only one neighbor to be a core point
- Much smaller ϵ sufficient for cluster formation
- Can form single-point clusters

Analysis by Plot

Plot 1:

- Dense clusters in both regions
- ϵ will decrease significantly
- Only needs to connect nearest neighbors

Plot 2:

- Dense outer circle: very small distances between points
- Sparse inner points: can form individual clusters
- ϵ will decrease very much compared to part (b)

Conclusion

ϵ will be much smaller in both plots because:

- Only needs to connect nearest neighbor pairs
- No requirement to form large connected components
- Especially significant decrease in Plot 2 where previously large ϵ was needed to connect sparse inner points

Part (d)

Using ϵ from part (b), which plot will have more black points labeled as noise?
Plot 1 will have more black points labeled as noise.

Reasoning

- **Plot 1:**
  - Smaller ϵ due to dense clusters
  - Smaller neighborhood radius for black points
  - Less likely to have sufficient neighbors within ϵ
  - Most black points will be labeled as noise
- **Plot 2:**
  - Larger ϵ due to sparse inner points
  - Larger neighborhood radius for black points
  - More likely to capture enough neighbors from dense outer ring
  - Fewer black points will be labeled as noise

Conclusion

Plot 1 will identify more black points as noise because:

- Smaller ϵ means smaller neighborhoods
- Black points less likely to meet MinPts requirement
- Contrast with Plot 2 where larger ϵ increases chance of cluster inclusion