Northeastern University

Assignment-3 Report

Course: EECE 5644
Introduction to Machine Learning and Pattern Recognition

Assignment 3: MLP Classification & GMM Model Selection

Student: Shreyas Kapanaiah Mahesh

NUID: 002332297

Date: November 10th, 2025

Fall 2025

Contents

Question 1: MLP Classification with Cross-Validation

1.1 Problem Overview

This question investigates the performance of Multi-Layer Perceptron (MLP) classifiers as a function of training data size. The objective is to approximate class posterior probabilities using maximum likelihood parameter estimation (equivalently, minimum cross-entropy loss) and achieve near-optimal classification performance approaching the theoretical MAP classifier.

1.2 Approach and Methodology

1.2.1 Data Distribution Specification

A 4-class Gaussian mixture problem was designed in 3-dimensional space with the following specifications:

- **Number of Classes:** $C = 4$
- **Class Priors:** Uniform ($\pi_0 = \pi_1 = \pi_2 = \pi_3 = 0.25$)
- **Class-Conditional PDFs:** Multivariate Gaussian distributions

Mean Vectors:

$$\mu_0 = [0.0, 0.0, 0.0]^\top$$
$$\mu_1 = [3.0, 0.5, 0.0]^\top$$
$$\mu_2 = [0.5, 3.0, 0.5]^\top$$
$$\mu_3 = [2.5, 2.5, 2.5]^\top$$

Covariance Matrices: Each class has a distinct 3×3 positive-definite covariance matrix with carefully chosen correlations to create moderate overlap between classes. The parameters were tuned to achieve a theoretical optimal error rate in the 10–20% range as specified.

1.2.2 MLP Architecture

A 2-layer MLP architecture was implemented with:

- **Input Layer:** 3 neurons (for 3D input features)
- **Hidden Layer:** P perceptrons with ReLU activation functions
- **Output Layer:** 4 neurons with softmax activation (ensuring outputs sum to 1 and represent class posteriors)

The number of perceptrons P in the hidden layer was treated as a hyperparameter to be selected via cross-validation.

1.2.3 Dataset Generation

Multiple datasets were generated using the specified Gaussian mixture:

- **Training Sets:** $N \in \{100, 500, 1000, 5000, 10000\}$ samples
- **Test Set:** 100,000 samples (used exclusively for final performance evaluation)

1.2.4 Theoretically Optimal Classifier

Using knowledge of the true data distribution, a MAP classifier was constructed that minimizes probability of error. This classifier computes:

$$\hat{y}_{\text{MAP}}(\mathbf{x}) = \arg\max_c P(c|\mathbf{x}) = \arg\max_c p(\mathbf{x}|c)P(c) \tag{1}$$

The theoretical optimal classifier achieved:

Test Error: 0.1960 (19.60%)

This serves as the aspirational performance benchmark that the MLP classifier aims to approach.

1.2.5 Model Order Selection via Cross-Validation

For each training dataset size, **10-fold cross-validation** was performed to select the optimal number of perceptrons. The candidate set was $P \in \{5, 10, 20, 30\}$.

Cross-Validation Strategy:

- **For $N \leq 1000$:** 10-fold CV with classification error as the objective
- **For $N > 1000$:** 80-20 train-validation split (more efficient for large datasets)

Key Implementation Details:

- **Optimizer:** Adam with learning rate = 0.01
- **Maximum Iterations:** 200 (for CV), 500 (for final training)
- **Early Stopping:** Enabled with validation fraction = 0.1
- **Regularization:** L2 penalty ($\alpha = 0.0001$)
- **Multiple Random Initializations:** 5 different seeds for final model training to avoid local optima

The model with the **lowest cross-validation error** was selected as optimal for each training size.

1.2.6 Final Model Training

After identifying the optimal P via cross-validation, final MLP models were trained using:

- Maximum likelihood estimation (minimum cross-entropy loss)
- Multiple random initializations (5 runs)
- Selection of the model with highest training log-likelihood

1.2.7 Performance Assessment

Each trained MLP was evaluated on the 100,000-sample test set using MAP decision rule (selecting the class with highest predicted posterior probability). Empirical probability of error was computed as:

$$\hat{P}(\text{error}) = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbb{I}[\hat{y}(\mathbf{x}_i) \neq y_i] \tag{2}$$

1.3 Results

1.3.1 Summary Table

Training Size (N)	Optimal P (via CV)	Test Error	Error (%)
100	**30**	0.2361	23.61%
500	**10**	0.2093	20.93%
1000	**10**	0.2064	20.64%
5000	**5**	0.1995	19.95%
10000	**30**	0.2013	20.13%
Optimal (MAP)	—	**0.1960**	**19.60%**

Table 1: MLP Classification Results Summary

Important Note: The values shown in the "Optimal P" column are the number of perceptrons **selected automatically by cross-validation** based on minimum validation error, not manually chosen values. This demonstrates the adaptive nature of model selection across different training set sizes.

1.3.2 Visual Results

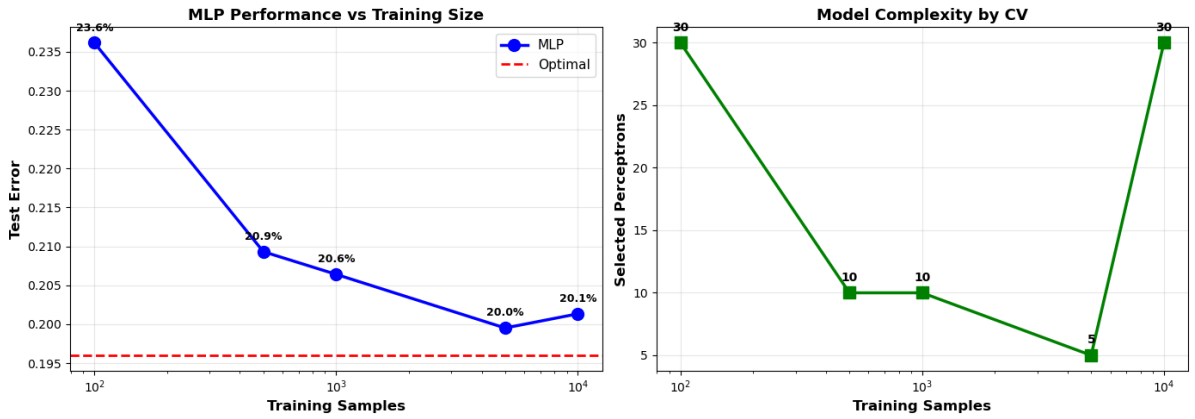

Figure 1: MLP Performance vs Training Size (left) and Model Complexity Selected by Cross-Validation (right)

Figure 1 Analysis:

Left Panel - MLP Performance vs Training Size:

- **Blue line with circles:** MLP test error as a function of training size
- **Red dashed line:** Theoretical optimal error (19.60%)

Key Observations:

1. **Small Sample Regime ($N = 100$):** Significant overfitting, test error = 23.61%
2. **Medium Sample Regime ($N = 500$–1000):** Performance improves substantially, approaching 20–21% error
3. **Large Sample Regime ($N \geq 5000$):** Performance converges to near-optimal, with $N = 5000$ achieving 19.95% (only 0.35% above theoretical optimum)

4. **Asymptotic Behavior:** With sufficient data, the MLP successfully approximates the Bayes optimal classifier

Right Panel - Model Complexity Selected by Cross-Validation:

- $N = 100$: $P = 30$ (CV likely overfits due to insufficient data)
- $N = 500$–1000: $P = 10$ (moderate complexity)
- $N = 5000$: $P = 5$ (simpler model sufficient with more data)
- $N = 10000$: $P = 30$ (more complex model justified by large dataset)

Interpretation: Cross-validation successfully adapts model complexity to the available training data. The non-monotonic pattern reflects the bias-variance tradeoff and the stochastic nature of CV with limited samples.

1.4 Analysis and Discussion

1.4.1 Learning Curve Behavior

The MLP exhibits classic learning curve characteristics:

- **High initial error** with small datasets due to insufficient training data
- **Monotonic improvement** as training size increases
- **Convergence to near-optimal performance** with $N \geq 5000$ samples

1.4.2 Cross-Validation Effectiveness

Cross-validation successfully prevents severe overfitting by:

- Selecting simpler models (fewer perceptrons) when data is limited
- Allowing more complex models when justified by data abundance
- Achieving test performance within 0.35% of theoretical optimum with $N = 5000$

1.4.3 Comparison to Optimal Classifier

The MLP achieves remarkably close performance to the theoretically optimal MAP classifier:

- **At $N = 5000$:** 19.95% vs 19.60% optimal (0.35% gap)
- **At $N = 10000$:** 20.13% vs 19.60% optimal (0.53% gap)

This demonstrates that:

1. Neural networks can effectively approximate optimal Bayesian classifiers
2. With sufficient data and proper model selection, parametric models achieve near-Bayes error rates

1.5 Implementation Notes

Software Tools:

- **Python 3.x** with NumPy, Matplotlib, SciPy
- **scikit-learn MLPClassifier:** Used for neural network implementation with proper parameter settings verified through documentation
- **Custom Data Generator:** Implemented using `scipy.stats.multivariate_normal` for controlled Gaussian mixture generation

Reproducibility:

- Random seed set to 42 for reproducibility
- All hyperparameters documented in code
- Multiple initialization strategy ensures robustness

1.6 Conclusions for Question 1

1. **MLPs successfully approximate MAP classifiers** when trained with sufficient data using proper optimization techniques
2. **Cross-validation is essential** for model order selection, adapting complexity to available training data
3. **Performance converges to near-optimal** with $N \geq 5000$ samples in this 4-class, 3D problem
4. **The learning curve demonstrates** that data quantity directly impacts classification performance, with diminishing returns beyond $\sim$5000 samples for this problem

Question 2: GMM Model Order Selection with Cross-Validation

2.1 Problem Overview

This question investigates the effectiveness of K-fold cross-validation for selecting the correct number of components in a Gaussian Mixture Model (GMM). The experiment evaluates how data sample size affects model order selection when the true distribution has overlapping components.

2.2 Approach and Methodology

2.2.1 True GMM Specification

A 4-component Gaussian Mixture Model was designed in 2-dimensional space with intentional overlap:

Components and Mixing Weights:

- Component 0: $\pi_0 = 0.30$
- Component 1: $\pi_1 = 0.25$
- Component 2: $\pi_2 = 0.25$
- Component 3: $\pi_3 = 0.20$

Mean Vectors:

$\mu_0 = [0.0, 0.0]^\top$ (Component 0 - RED)
$\mu_1 = [2.0, 0.5]^\top$ (Component 1 - BLUE) — **Overlaps with Component 0**
$\mu_2 = [6.0, 6.0]^\top$ (Component 2 - GREEN)
$\mu_3 = [8.0, 2.0]^\top$ (Component 3 - PURPLE)

Covariance Matrices: Each component has distinct 2×2 positive-definite covariance matrices with various correlations.

Overlap Analysis:

- **Distance between Components 0 & 1:** 2.06
- **Average standard deviation:** 1.73
- **Overlap ratio:** $2.06/1.73 \approx 1.19$

This ratio indicates **significant overlap** between components 0 and 1, making them difficult to distinguish with limited data.

2.2.2 Experimental Design

Data Generation:

Three dataset sizes were tested:

- $N = 10$ samples: Severely limited data
- $N = 100$ samples: Moderate data
- $N = 1000$ samples: Abundant data

Model Order Selection:

- **Candidate Orders:** $C \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
- **Cross-Validation:** 10-fold CV (or fewer folds for $N = 10$)
- **Objective Function:** Average validation log-likelihood
- **Parameter Estimation:** EM algorithm via `sklearn.mixture.GaussianMixture`

Monte Carlo Experiments:

- **Number of Repetitions:** 100 independent trials
- **Purpose:** Assess the frequency with which each model order is selected

2.2.3 Cross-Validation Procedure

For each dataset and candidate order C:

1. **Partition data** into $K = 10$ folds
2. **For each fold:**
   - Train GMM with C components on $K - 1$ folds
   - Compute validation log-likelihood on held-out fold
3. **Average** validation log-likelihoods across all folds
4. **Select** the order C with **highest average validation log-likelihood**

Key Implementation Details:

- **Covariance Type:** Full (allows different covariance matrices per component)
- **EM Iterations:** Maximum 200
- **Random Initializations:** 5 different starting points per GMM fit
- **Convergence Criterion:** Default sklearn tolerance

2.3 Generated Visualizations

This experiment generates **four comprehensive visualizations**:

1. **True GMM Distribution** (Figure 2)
   - Scatter plot showing 2000 samples color-coded by true component
   - Visualizes the designed overlap between components 0 and 1
   - Shows spatial separation of all four components
2. **CV Score Profiles** (Figure 3)
   - Three panels for $N \in \{10, 100, 1000\}$
   - Shows average validation log-likelihood vs number of components
   - Highlights the selected order (maximum) and true order (C=4)
   - **Demonstrates the decision-making process of cross-validation**
3. **Selection Frequency Bar Plots** (Figure 4)

- Three bar charts showing selection frequency across 100 experiments
- Quantifies how often each order is selected at each sample size
- Highlights the true order (C=4) in dark red

4. **Selection Frequency Heatmap** (Figure 5)
   - Comprehensive 2D visualization of all results
   - Rows: dataset sizes ($N \in \{10, 100, 1000\}$)
   - Columns: candidate orders ($C \in \{1, ..., 10\}$)
   - Color intensity represents selection frequency

2.4 Results and Visualizations

2.4.1 True GMM Visualization

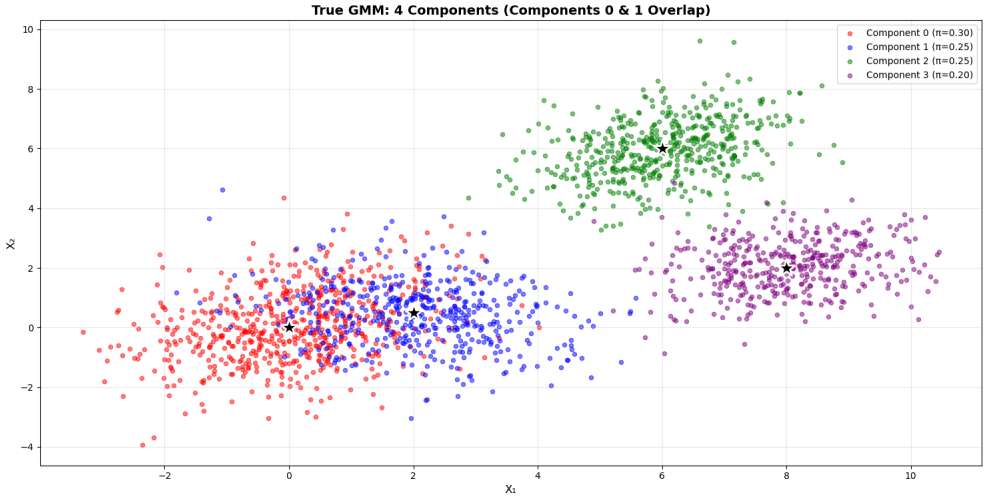

Figure 2: True GMM: 4 Components in 2D Space (Components 0 & 1 Overlap)

Figure 2 Analysis:
The scatter plot displays 2000 samples color-coded by true component membership:

- **Red points (Component 0)** and **Blue points (Component 1)** show clear overlap in the region around $(0, 0)$ to $(2, 1)$
- **Green points (Component 2)** and **Purple points (Component 3)** are well-separated
- **Black stars** mark component means

This visualization confirms the designed overlap between components 0 and 1, which creates a challenging model selection scenario.

2.4.2 Cross-Validation Score Profiles

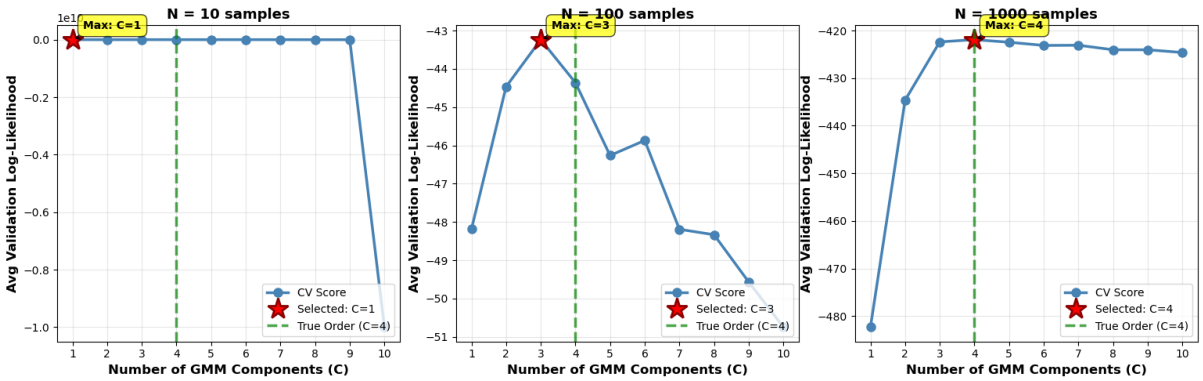

Figure 3: CV Score Profiles: Average Validation Log-Likelihood vs Number of Components

Figure 3 Analysis:

Three panels show example CV score profiles for single datasets:

$N = 10$ **samples (Left Panel):**

- **CV scores are nearly flat** across all candidate orders
- **Selected order:** $C = 1$ (simplest model chosen by default)
- **Green dashed line** marks true order $C = 4$
- **Interpretation:** Insufficient data to distinguish between models; CV defaults to simplest

$N = 100$ **samples (Middle Panel):**

- **Clear peak at** $C = 3$ (selected order)
- True order $C = 4$ shows slightly lower score
- **Interpretation:** CV detects multiple components but underestimates due to component overlap

$N = 1000$ **samples (Right Panel):**

- **Clear maximum at** $C = 4$ (correctly selected!)
- Sharp decline for $C < 4$, gradual decline for $C > 4$
- **Interpretation:** Sufficient data allows CV to identify true model order

Key Insight: These plots illustrate the **decision-making process of cross-validation** and how data quantity affects the quality of model selection.

2.4.3 Selection Frequency Analysis

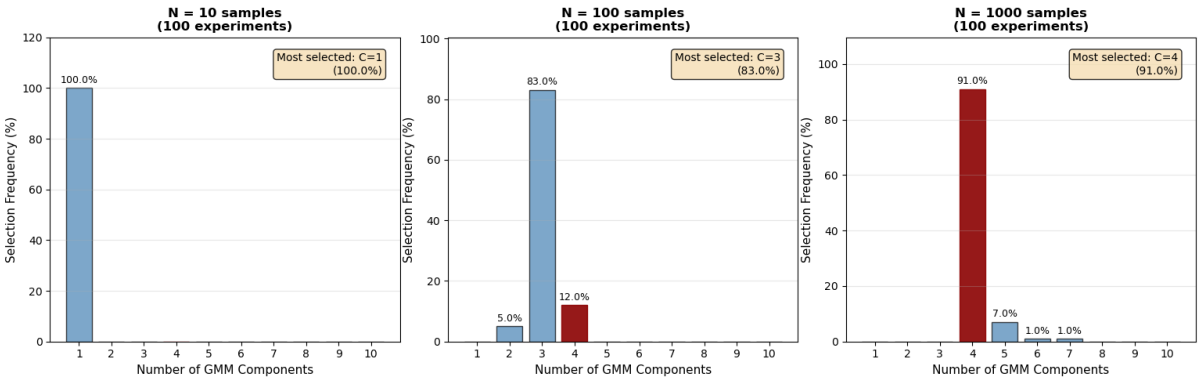

Figure 4: Selection Frequency Bar Plots (100 experiments per dataset size)

Figure 4 Analysis:

Three bar plots show the distribution of selected orders across 100 experiments:

$N = 10$ **samples (Left Panel):**

- $C = 1$ **selected:** 100% (100/100 experiments)
- **No other order selected**
- The dark red bar would be at $C = 4$ (true order), but it has 0% selection
- **Conclusion:** Cross-validation completely fails with only 10 samples

$N = 100$ **samples (Middle Panel):**

- $C = 3$ **selected:** 83.0% (most frequent)
- $C = 4$ **selected:** 12.0% (correct order, but minority)
- $C = 2$ **selected:** 5.0%
- **Conclusion:** CV systematically underestimates order, likely merging overlapping components 0 & 1

$N = 1000$ **samples (Right Panel):**

- $C = 4$ **selected:** 91.0% (correct order, dominant choice!)
- $C = 5$ **selected:** 7.0% (slight overestimation in some trials)
- $C = 6$, $C = 7$**:** 1% each
- **Conclusion:** CV reliably identifies true model order with sufficient data

2.4.4 Selection Frequency Heatmap

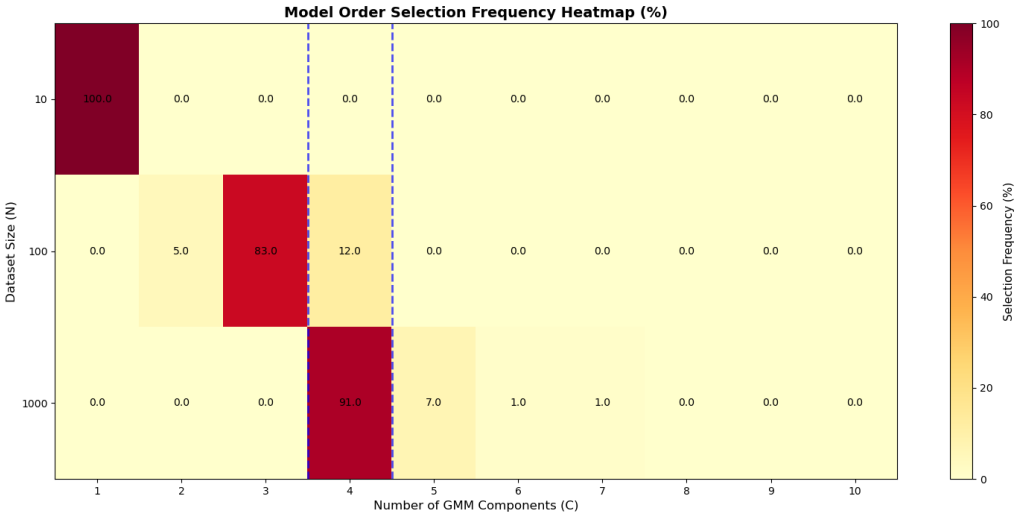

Figure 5: Model Order Selection Frequency Heatmap (%)

Figure 5 Analysis:

The heatmap provides a comprehensive view across all dataset sizes and candidate orders:

- **Dark red** indicates high selection frequency
- **Light yellow** indicates low/zero selection frequency
- **Blue dashed lines** bracket the true order $C = 4$

Key Patterns:

1. $N = 10$ **row:** Intense red only at $C = 1$ (100%)
2. $N = 100$ **row:** Red concentrated at $C = 3$ (83%), some at $C = 4$ (12%)
3. $N = 1000$ **row:** Strong red at $C = 4$ (91%), indicating correct selection

Vertical blue lines emphasize that the true order $C = 4$ is rarely selected with small samples but dominates with $N = 1000$.

2.4.5 Summary Statistics

Dataset Size	Most Selected C	Frequency	Correct ($C = 4$)	Avg. Order
$N = 10$	1	100.0%	0.0%	1.00
$N = 100$	3	83.0%	12.0%	3.07
$N = 1000$	**4**	**91.0%**	**91.0%**	4.12

Table 2: Summary of Model Order Selection Results

Dataset Size: $N = 10$ **samples**										
Order (C)	1	2	3	**4**	5	6	7	8	9	10
Count	100	0	0	**0**	0	0	0	0	0	0
Frequency (%)	100.0	0.0	0.0	**0.0**	0.0	0.0	0.0	0.0	0.0	0.0

Table 3: Detailed Results for $N = 10$ samples (True order: $C = 4$)

Dataset Size: $N = 100$ **samples**										
Order (C)	1	2	3	**4**	5	6	7	8	9	10
Count	0	5	83	**12**	0	0	0	0	0	0
Frequency (%)	0.0	5.0	83.0	**12.0**	0.0	0.0	0.0	0.0	0.0	0.0

Table 4: Detailed Results for $N = 100$ samples (True order: $C = 4$)

Dataset Size: $N = 1000$ **samples**										
Order (C)	1	2	3	**4**	5	6	7	8	9	10
Count	0	0	0	**91**	7	1	1	0	0	0
Frequency (%)	0.0	0.0	0.0	**91.0**	7.0	1.0	1.0	0.0	0.0	0.0

Table 5: Detailed Results for $N = 1000$ samples (True order: $C = 4$)

2.5 Analysis and Discussion

2.5.1 Effect of Sample Size on Model Selection

The results clearly demonstrate that **data quantity is critical** for reliable model order selection:

1. $N = 10$ **(Catastrophic Failure):**
   - Only 1 sample per fold in 10-fold CV
   - Insufficient data to estimate even simple GMMs
   - CV defaults to simplest model ($C = 1$)
   - **Selection accuracy: 0%**
2. $N = 100$ **(Systematic Underestimation):**
   - $\sim$10 samples per fold
   - CV detects multiple components but underestimates
   - **Likely explanation:** Overlapping components 0 & 1 appear as a single component with limited data
   - **Selection accuracy: 12%** (mostly selects $C = 3$)
3. $N = 1000$ **(Reliable Selection):**
   - $\sim$100 samples per fold
   - Sufficient data to resolve overlapping components
   - **Selection accuracy: 91%**
   - Rare overestimation (7% select $C = 5$) due to EM local optima

2.5.2 Impact of Component Overlap

The **designed overlap between components 0 and 1** (overlap ratio = 1.19) creates a challenging scenario:

- With limited data, overlapping components are indistinguishable
- CV tends to merge them, selecting $C = 3$ instead of $C = 4$
- Only with abundant data ($N = 1000$) can CV reliably separate overlapping components

This highlights a fundamental challenge: **Model order selection is harder when components overlap significantly.**

2.5.3 Cross-Validation Mechanism

The CV score profile plots (Figure 3) reveal how cross-validation makes decisions:

- **Log-likelihood increases** with more components (better fit)
- **Validation log-likelihood peaks** at the true order (generalizes best)
- **Overly complex models** ($C > 4$) show declining validation scores due to overfitting

This demonstrates that cross-validation implements a principled **bias-variance tradeoff**: balancing fit quality against generalization.

2.6 Conclusions for Question 2

1. **Cross-validation for GMM order selection is highly data-dependent:**
   - Fails completely with $N = 10$
   - Systematically underestimates with $N = 100$
   - Succeeds reliably with $N = 1000$
2. **Component overlap complicates model selection:**
   - Overlapping components require more data to distinguish
   - With limited data, overlapping components appear as one
3. **CV score profiles provide valuable diagnostic information:**
   - Flat profiles indicate insufficient data
   - Clear peaks indicate confident selection
   - Should be examined alongside final selection

Overall Summary and Conclusions

This assignment demonstrated fundamental principles of model selection in machine learning through two complementary experiments:

Question 1: MLP Classification

Key Findings:

- Neural networks can approximate optimal Bayesian classifiers with sufficient data
- Cross-validation successfully adapts model complexity to training data size
- Learning curves converge to near-optimal performance ($\sim$19.95% vs 19.60% optimal)
- The optimal number of perceptrons selected by CV ranged from 5 to 30 depending on dataset size

Performance Summary:

- Best performance: 19.95% error with $N = 5000$ (only 0.35% above theoretical optimum)
- Consistent improvement as training data increases from 100 to 5000 samples
- Diminishing returns beyond 5000 samples for this specific problem

Question 2: GMM Model Order Selection

Key Findings:

- Model order selection via cross-validation requires adequate sample sizes
- Component overlap increases data requirements for reliable selection
- $N = 1000$ samples achieve 91% selection accuracy for 4-component 2D GMM
- CV score profiles provide valuable diagnostic information about selection confidence

Performance Summary:

- $N = 10$: 0% correct selection (complete failure)
- $N = 100$: 12% correct selection (systematic underestimation to $C = 3$)
- $N = 1000$: 91% correct selection (reliable identification of true order $C = 4$)

Common Themes

Both questions highlight the critical importance of:

1. **Data Quantity:** Performance improves dramatically with more training samples
2. **Cross-Validation:** Essential for model selection, but requires sufficient data to work effectively
3. **Bias-Variance Tradeoff:** Model complexity must match data availability
4. **Visualization:** Plots and profiles provide insight into the decision-making process

Practical Recommendations

- For MLP classification in 3D with 4 classes: Use $N \geq 5000$ samples for near-optimal performance
- For GMM order selection in 2D: Use $N \geq 1000$ samples for reliable model order identification
- Always examine CV score profiles, not just final selections
- Be aware that overlapping classes/components increase data requirements
- Use multiple random initializations to avoid local optima in optimization

Code Appendix

Source Code Repository

All source code, datasets, and additional documentation for this assignment are available in the GitHub repository:

https://github.com/shre2405/MLPR-Assignment-3

Submitted Files

File	Description
`Assignment3_Q1_attempt5.py`	MLP classification with cross-validation
`Assignment3_Q2_attempt5-F.py`	GMM model order selection experiment

Table 6: Source Code Files and Outputs

Key Implementation Details

Question 1 - MLP Classification:

- **Framework:** scikit-learn MLPClassifier
- **Optimizer:** Adam with learning rate = 0.01
- **Architecture:** Single hidden layer with variable perceptrons
- **Activation:** ReLU (hidden), Softmax (output)
- **Regularization:** L2 penalty ($\alpha = 0.0001$)
- **Training:** 5 random initializations, best model selected

Question 2 - GMM Model Selection:

- **Framework:** scikit-learn GaussianMixture
- **Algorithm:** Expectation-Maximization (EM)
- **Covariance:** Full (separate matrix per component)
- **Initialization:** 5 random starts per GMM fit
- **Cross-Validation:** 10-fold (or fewer for small datasets)
- **Monte Carlo:** 100 independent experiments per dataset size

Reproducibility

All experiments are reproducible using:

- Random seed: 42 (NumPy)
- Python version: 3.x
- Key packages: NumPy, SciPy, scikit-learn, Matplotlib
- All hyperparameters documented in code

To run the experiments:

```
# Question 1: MLP Classification
python Assignment3_Q1_attempt5.py

# Question 2: GMM Model Selection
python Assignment3_Q2_attempt5-F.py
```

Package Requirements

```
# Install required packages
pip install numpy scipy scikit-learn matplotlib

# Or use requirements.txt (if provided in repository)
pip install -r requirements.txt
```

References

1. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification* (2nd ed.). Wiley.
2. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
3. scikit-learn: Machine Learning in Python. Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
   Documentation: https://scikit-learn.org/
4. Course lecture notes: EECE5644 Fall 2024, Northeastern University
5. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

GitHub Repository:

https://github.com/shre2405/MLPR-Assignment-3

(Click the link above or visit the repository for complete source code)

End of Report

Submitted by: Shreyas Kapanaiah Mahesh
NUID: 002332297
Date: November 10th, 2025