EECE 5644 Assignment 3

Ishaan Desai (desai.is@northeastern.edu)

November 10th, 2025

Professor Erdogmus

GitHub Link:
https://github.com/ishaandesai0/EECE-5644/tree/main/Assignment%203

Question 1

This problem had me train multilayer perceptrons (aka MLPs) to approximate class posteriors using maximum likelihood parameter estimation. The goal was to examine how training set size affects classification performance and compare against the theoretically optimal classifier.

I designed a 4-class classification model in 3D space with uniform priors, with P(ω_i) = 0.25 for i = 0, 1, 2, and 3. The Gaussian distributions were created with the following parameters:

- Class 0: $[0, 0, 0]^T$
- Class 1: $[3, 0, 0]^T$
- Class 2: $[0, 2.5, 0]^T$
- Class 3: $[0, 0, 3]^T$

The covariance matrices were designed with small off-diagonal elements to introduce realistic correlations between features. The mean separation was calibrated to achieve the target error rate.

Using the true data distribution parameters, I implemented the MAP decision rule:

Decision Rule: Choose class c* = argmax_c P(ω_c|x) where P(ω_c|x) ∝ p(x|ω_c)P(ω_c)

The theoretically optimal classifier was evaluated on a test set of 100,000 samples, achieving a P(error) of 12.76%. This error rate falls within the target range of 10-20%. This baseline represents one of the best possible performances achievable.

I implemented a 2-layer MLP with 3 neurons in the input layer for 3D feature space, P neurons with ReLU activation in the hidden layer. The output layer contained 4 neurons with softmax activation.

For each training set size, I performed 10-fold cross-validation testing candidate architectures with 1, 2, 3, 4, 5, 6, 8, 20, 25, and 40 perceptrons. The results can be seen below in Table 1:

Training Samples	Optimal Perceptrons	CV Error
100	20	0.18
500	15	0.1380
1,000	6	0.12
5,000	8	0.1346
10,000	20	0.1292

Table 1: Cross Validation Results

Cross-validation showed interesting behavior. For smaller datasets, complex models were selected but still underperformed due to insufficient data. At 1,000 samples, CV correctly selected a simpler architecture to prevent overfitting. With larger datasets, more complex models are needed.

For each training set, I trained the optimal MLP architecture using the Adam optimizer with default learning rate and cross-entropy loss, which is equivalent to maximum likelihood estimation. To mitigate risk of local optima, I used 10 random initializations per training set and selected the model to achieve the highest training accuracy, ensuring good parameter estimation across all of the datasets. Results can be seen below in Table 2 and Figures 1 and 2:

Training Samples	Perceptrons	Test P(error)	Error Rate
100	20	0.1480	14.80%
500	15	0.1311	13.11%
1,000	6	0.1310	13.10%
5,000	8	0.1301	13.01%
10,000	20	0.1286	12.86%
Theoretical Optimal	-	0.1276	12.76%

Table 2: Test Set Performance

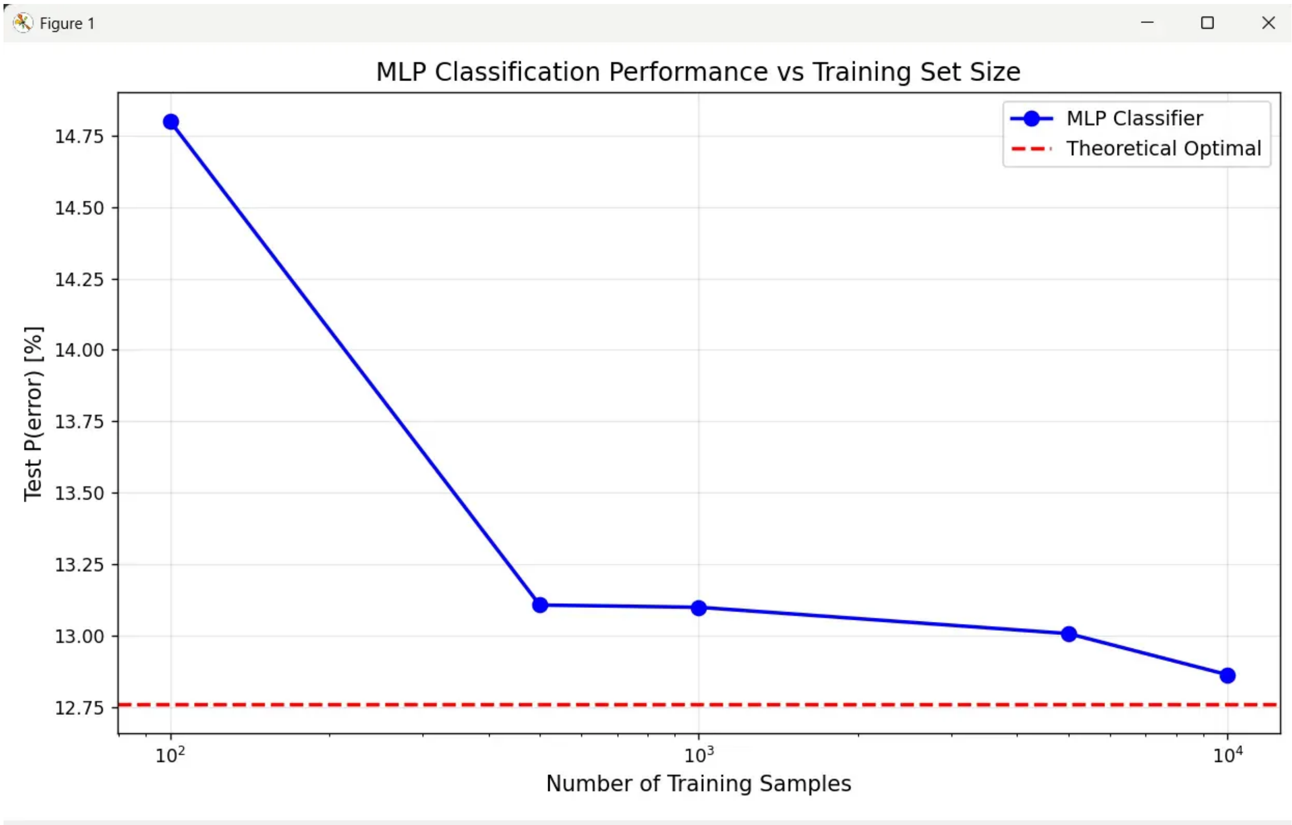

Figure 1: MLP Classification Performance vs Training Set Size

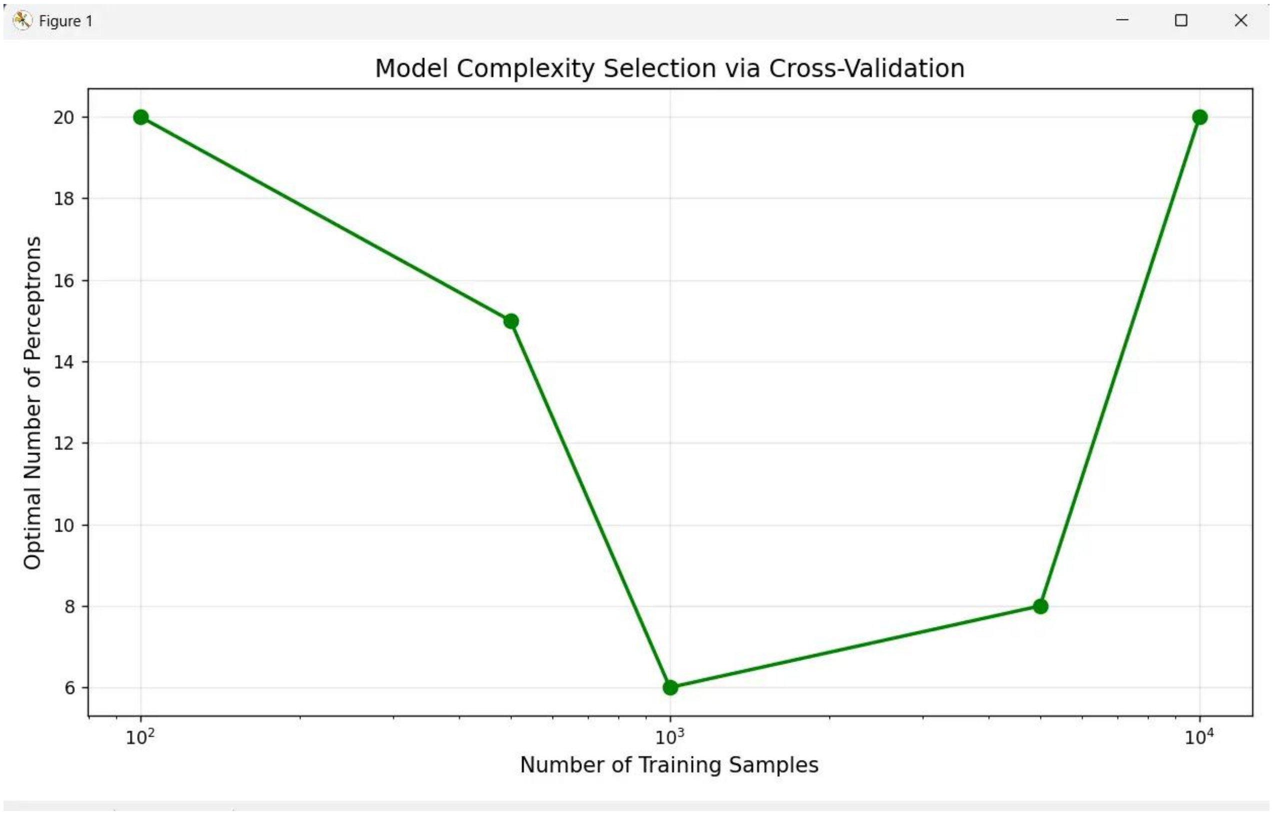

Figure 2: Model Complexity Selection via Cross-Validation

The results show a few important patterns in MLP performance as training data increases. Error drops from 14.8 to 13.11% when going from 100 to 500 samples, showing large initial improvement. Performance gains weren't as strong with more data being added, reaching an asymptote at 10,000 samples with 12.86% error. The gap at small sample sizes highlights how insufficient data leads to poor performance despite optimal model selection.

This also demonstrates that MLPs can effectively approximate optimal classifiers when provided with enough training data, requiring about 2500+ samples per class to approach optimal performance in the 3D feature space. The 10-fold cross-validation successfully prevents overfitting by choosing appropriate model complexity based on available data, while the strategy of using 10 random initializations proved to be important for smaller datasets to avoid poor local minima. There are a few limitations, however, as the ReLU activation created piecewise-linear decision boundaries not perfectly matching optimal Gaussian boundaries. Performance could improve with deeper architectures or different activation functions, and the specific random seed affects results such that multiple runs with different seeds would provide stronger confidence intervals.

Question 2

This problem had me examine how cross-validation performs for model order selection in Gaussian Mixture Models as a function of dataset size. In order to do this, I designed a 4-component GMM for 2D data with uniform mixture weights (α = 0.25 for all 4 components). The component parameters were as follows:

- Component 1: $[0, 0]^T$, moderate covariance
- Component 2: $[1.5, 1.5]^T$, significant overlap with Component 1
- Component 3: $[6, 0]^T$, well separated
- Component 4: $[3, 6]^T$, well separated

Components 1 and 2 were intentionally positioned close together relative to their covariance eigenvalues, creating significant overlap making true model order difficult to identify from limited data. The True GMM Distribution scatterplot can be seen below in Figure 3:

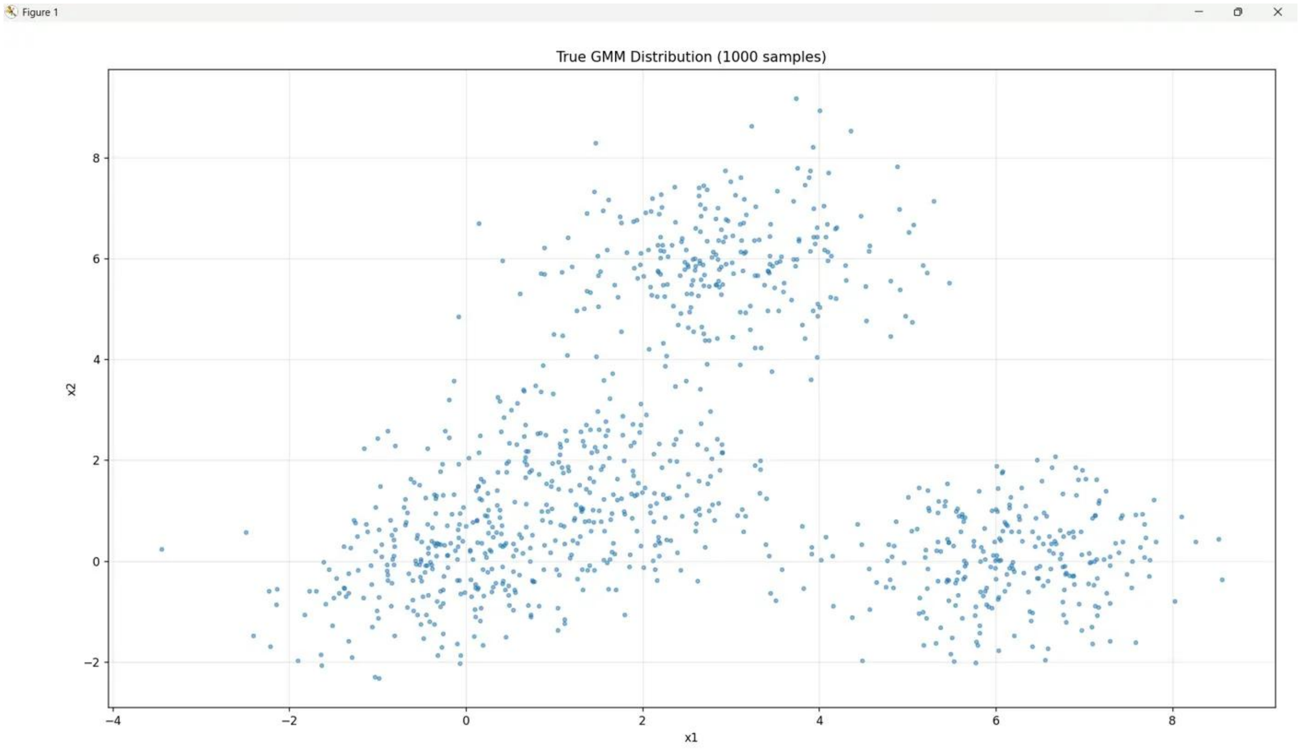

Figure 3: True GMM Distribution

This shows 1000 samples from this distribution, clearly showing the four clusters with visible overlap between the bottom left clusters.

The experiment evaluated GMM model order selection across three dataset sizes (10, 100, and 1000 samples) by testing candidate models with 1 through 10 components. I used 10-fold cross-validation, adjusted to min(10, N) for small datasets to have valid fold sizes. Parameter estimation was performed using the EM algorithm via sklearn's GaussianMixture with full covariance matrices, 5 random initializations per fit to avoid local optima, and a maximum of 200 iterations for convergence. The performance metric used here was log-likelihood on validation folds, as this directly measures how well each candidate model explains held-out data. To get statistically meaningful results, I repeated the experiment 100 times for each dataset, generating independent datasets and recording which model order was selected by cross-validation in each trial. The results can be seen in Table 3 and Figure 4 below:

Model Order	10 Samples	100 Samples	1,000 Samples
1	96.0%	0.0%	0.0%
2	4.0%	2.0%	0.0%
3	0.0%	94.0%	5.0%
4 (True)	0.0%	4.0%	87.0%
5	0.0%	0.0%	7.0%
6	0.0%	0.0%	1.0%
7-10	0.0%	0.0%	0.0%

Table 3: Model Order Selection Frequencies

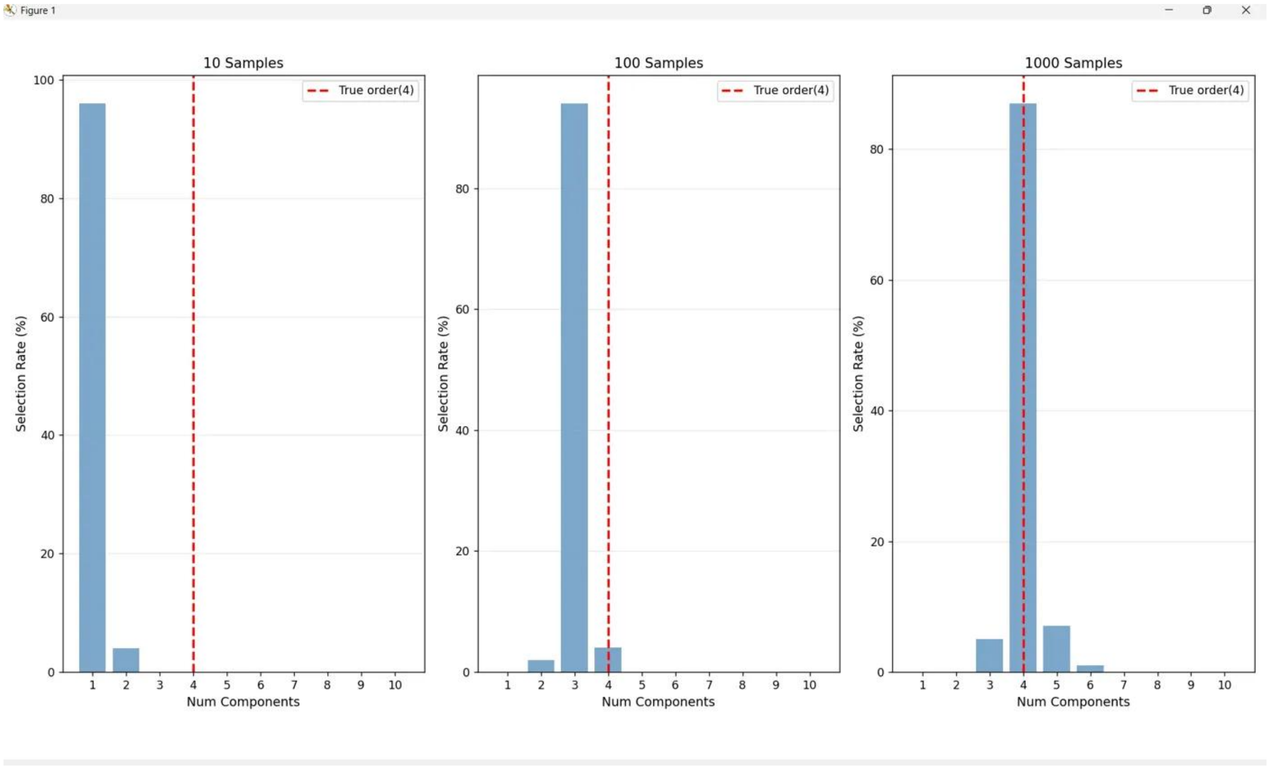

Figure 4: Visual Results of Model Order Selection Frequencies

The experimental results reveal clear patterns across different dataset sizes. With only 10 samples, the procedure almost exclusively selects 1 component, indicating underfitting due to the insufficient data. Cross-validation did prevent overfitting by choosing the simplest model. At 100 samples, the selection predominantly favors 3 components, coming close to the true order but underestimating it by a little since overlapping components are difficult to distinguish with limited data. With 1,000 samples, the procedure selected the correct 4 components most frequently, with small probabilities of selecting 3 or 5. Sufficient data allows for accurate model order selection.

A few key findings emerged from this experiment. Sample size has a strong effect on model selection, with selection distribution shifting to the right with higher complexity and sample size. Cross-validation shows a bias toward simplicity with limited data, showing the bias-variance tradeoff in action. The overlapping components significantly increase selection difficulty, as even with 1,000 samples, cross-validation only identified the correct order 87% of the time with the closeness of components 1 and 2. For this 2D, 4-component exercise, about 250 samples per component are needed for reliable model order identification.

These results also show that cross-validation performs correctly across all sample sizes. With 10 samples, insufficient data prevents supporting complex models, so the procedure selects simple models to minimize validation error. With 100 samples, the method can distinguish major structure but not fine details, leading to the selection of 3 components. With 1,000 samples, sufficient data exists to identify true complexity, resulting in correct selection of 4 components in the majority of cases. The slightly incorrect accuracy at 1,000 samples comes from three factors: random sampling variation may produce datasets with overlapped samples, the genuine challenge that overlapping components pose for model selection, and the noise in cross-validation's performance estimates based on limited validation data. The selection frequency distribution becomes more concentrated around true order as sample size increases, also aligning with theoretical results on consistent model selection and the expected logarithmic relationship between sample size and selection accuracy for mixture models.

These findings have important practical implications for mixture modeling applications. 100-250 samples per expected component is a good target to achieve reliable model order selection. While cross-validation provides a principled approach to model selection, it needs adequate data to be accurate. When components overlap significantly, model order selection becomes inherently harder and may require more data or alternative selection criteria like AIC or BIC to supplement cross-validation results.

Citations

- Lecture Notes
- Code Folder
- Repo Link: https://github.com/ishaandesai0/EECE-5644/tree/main/Assignment%203
- Claude AI – Mathematical Concepts
- Duda, R.O., Hart, P.E., and Stork, D.G. Pattern Classification, 2nd Edition