Continual Learning with Drift: Assignment Report

Introduction

This report summarizes the application of different continual learning methods for predicting `target_10_val` in a synthetic dataset. The dataset exhibits concept drift and varying noise levels, with the data being provided in a sequential, day-wise manner. We explored four different approaches: Basic KNN, KNN with a Sliding Window, KNN with Concept Drift Detection, and the Weighted Majority Algorithm.

1. Basic KNN Approach

The first approach involved using a basic K-Nearest Neighbors (KNN) algorithm. This method was straightforward, relying on the nearest neighbors for predictions without additional mechanisms to handle concept drift or noise.

Results

- **Testing Accuracy on Non-Noise Data**: 77.50%
- **Testing Accuracy on Noise Data**: 43.26%

This approach yielded reasonable accuracy in the absence of noise but performed poorly with noisy data, highlighting its sensitivity to data quality.

2. KNN with a Sliding Window

In the second approach, a sliding window was implemented in conjunction with the KNN algorithm. This technique aimed to address the issue of concept drift by focusing on a window of the most recent data, assuming it would be more representative of the current data distribution.

Results

- **Training Accuracy on Non-Noise Data**: 70.64%

- **Testing Accuracy on Noise Data**: 22.38%

With a window size of 1000, this method did not perform as well as the basic KNN, especially in the presence of noise, suggesting that the window size or the method itself might not be optimal for handling noisy, drifting data.

3. KNN with Concept Drift Detection

The third approach integrated concept drift detection into the KNN model. A custom `ConceptDriftDetector` class was employed to monitor the error rate and detect significant shifts, indicating a concept drift.

Results

- **Testing Accuracy on Non-Noise Data**: 70.22%
- **Testing Accuracy on Noise Data**: 22.71%

The concept drift detector provided a means to adapt to changing data distributions dynamically. However, the improvement in accuracy was marginal, indicating a need for more sophisticated methods or fine-tuning of the drift detection mechanism.

4. Weighted Majority Algorithm

The final approach used the Weighted Majority Algorithm, which involved training multiple KNN models and updating their weights based on their performance at each update.

Results

- **Accuracy before Weight Update on Non-Noise Data**: 77.57%
- **Accuracy after Weight Update on Non-Noise Data**: 79.38%
- **Accuracy before Weight Update on Noise Data**: 45.76%
- **Accuracy after Weight Update on Noise Data**: 47.23%

This method showed improvement after weight updates in both non-noise and noise scenarios, suggesting its effectiveness in adapting to changes in data distribution and noise levels.

Summary Table

Method	Non-Noise Data (Before Update)	Noise Data (Before Update)
Basic KNN	77.50%	43.26%
KNN with Sliding Window	70.64%	22.38%
KNN with Concept Drift	70.22%	22.71%
Weighted Majority Algorithm	77.57%	45.76%

Conclusion

The experimentation with different methods for continual learning in the presence of concept drift and noise demonstrated varying levels of effectiveness. The Weighted Majority Algorithm showed the most promise, adapting well to both concept drift and noise. Further research and refinement of these methods could lead to more robust models in dealing with real-world, dynamic data scenarios.