Evolving Stream Classification Challenge

1. Overview

In this assignment, you will build an evolving data stream classification model. You will use scikit-multiflow, an open-source Python library that provides tools for data stream mining and classification. You can find more information about tnis library on the official webpage.

Your task is to train data stream classification methods on the given data streams. You will use test-then-train –i.e. prequential evaluation method to compare performance of the classification models.

2. Concepts

- Data Stream: refers to an environment where data arrives continuously over time. This means we cannot assume that we have access to all the data at the beginning. Instead, we need to update our model incrementally as new data arrives. This is in contrast to traditional batch learning, where we can access all the data simultaneously and train the model on the entire dataset.
- Concept Drift: refers to a change in the underlying distribution of the data. For example, in the case of a spam email detection model, the characteristics of spam emails may change over time, which means that the system needs to adapt and update itself to identify the new characteristics of spam emails correctly. Data stream classification models need to adopt a concept drift detection and handling method to address concept drift. We refer to a data stream with concept drift as *evolving data stream*.
- Prequential Evaluation: is a commonly used evaluation approach for data stream classification tasks. In the prequential evaluation method, the model is tested on each incoming instance before it is used to update the model. It is also known as interleaved test-then-train evaluation method.

3. Requirements

You will need the following libraries in Python to complete this challenge. A Python solution is preferred. You need to find alternative libraries in other programming languages.

- numpy: a Python library for numerical computing
- scikit-learn: a library for machine learning in Python
- scikit-multiflow: a library for data stream mining and classification

4. Datasets as Data Streams

You will use two synthetic and two real datasets as data streams to compare performance of the classification models. As synthetic data streams, use *HyperplaneGenerator* and *SEAGenerator* classes from scikit-multiflow to generate 100,000 data instances for each. For future access, write the generated data instances into files named HyperplaneDataset and SEADataset. For the HyperplaneGenerator, use 10 features and 2 class labels as input parameters. Use default values for the other parameters.

As real data streams, you will experiment with the *Spam* and *Rialto* datasets. You can obtain these datasets from https://github.com/ogozuacik/concept-drift-datasets-scikit-multiflow.

5. Classification Task with Concept Drift Handling

Implement an instance of the following classification models. You can use scikit-multiflow and scikit-learn libraries to this aim.

- Adaptive Random Forest (ARF)[4]
- Streaming Agnostic Model with k-Nearest Neighbors (SAM-kNN)[3]
- Streaming Random Patches (SRP)[1]
- Dynamic Weighted Majority (DWM)[2]
- Multi-Layer Perceptron (MLP)

For the MLP model, use two hidden layers with 16 neurons in each layer. Use default parameters for the ensemble models.

6. Results and Discussion

Construct an instance of the classification models. For each dataset, use Interleaved Test-Then-Train approach to train and evaluate performance of these classifiers. Use prediction accuracy as evaluation metric. Report the following results for the classification models on each dataset:

- **Overall accuracy**: Overall prediction accuracy of the models.
- **Prequential accuracy plot**: Prequential accuracy is defined as the prediction accuracy of a model over the w most recent data instances. Use 20 sliding windows of size (*dataset size*/20) to calculate prequential accuracy values. Plot the obtained accuracy values over time for each dataset.

Compare performance of the classification models. How the MLP model performs compared to the ensemble models? What do you infer from the drops in the accuracy values of the models over time, in the prequential accuracy plots?

7. Report Format and Content

Use the ACM conference paper LaTeX or Word template for your report format. You can find the templates for LaTeX on Overleaf and Word on the ACM website.

Your report should include the following sections: Abstract, Keywords, Introduction, brief Related Work, Methodology, Experimental Results, Conclusion, and References. The Abstract provides a summary of your report, while the Introduction outlines the problem and its significance. The Methodology describes the approach used, and the Experimental Results present the findings. The Conclusion summarizes the key findings and References lists all cited sources.

Submit your code along with your report. The code should be included in a single zip file that includes a Readme file explaining how to run the code and any necessary dependencies.

8. References

[1] Gomes, H. M., Read, J., and Bifet, A. Streaming random patches for evolving data stream classification. In *2019 IEEE International Conference on Data Mining (ICDM)* (2019), IEEE, pp. 240–249.

[2] Kolter, J. Z., and Maloof, M. A. Dynamic weighted majority: An ensemble method for drifting concepts. *The Journal of Machine Learning Research 8* (2007), 2755–2790.

[3] Losing, V., Hammer, B., and Wersing, H. Knn classifier with self adjusting memory for heterogeneous concept drift. In *2016 IEEE 16th international conference on data mining (ICDM)* (2016), IEEE, pp. 291–300.

[4] Sugiyama, M., Idé, T., Nakajima, S., and Sese, J. Semi-supervised local fisher discriminant analysis for dimensionality reduction. *Machine learning 78* (2010), 35–61.