Comparative Evaluation of Data Drift Detection Algorithms: A Benchmarking Study

(Project Idea) - Group 17

(1). Introduction

In machine learning, maintaining model accuracy over time is critical, especially in environments where data distributions can change unpredictably. Concept drift, where the statistical characteristics of the target variable evolve, presents a substantial challenge to model performance. Detecting and promptly adapting to these changes is essential for ensuring the reliability and effectiveness of machine learning models in real-world applications.

Concept drift detection involves methods to identify shifts in data patterns that impact model performance. Error rate-based detection is a prominent approach that focuses on monitoring changes in predictive accuracy. When significant deviations in prediction errors occur, it indicates potential concept drift, triggering adaptive measures to recalibrate the model and maintain its accuracy.

In this project, our objective is to evaluate and compare different error rate-based concept drift detection methods across a range of machine learning algorithms. By assessing their effectiveness under varying conditions, we aim to understand their strengths, limitations, and applicability in real-world machine-learning applications. Ultimately, this exploration will equip us with insights to deploy robust concept drift detection strategies that enhance model adaptability and longevity in dynamic environments.

(2). Proposed Solution

1. Algorithm Selection: Identifying and selecting the most important and commonly used data drift detection algorithms. Generate a synthetic regression dataset that simulates a sudden concept drift.
2. Use reference dataset samples to train Machine Learning Models with these methods.
    - Linear Regression
    - Decision Tree Regression
    - Random Forest Regression
    - Support Vector Regression
3. Generate a data stream by using generated datasets. Initially, stream data, representing the reference concept, followed by data representing the drifted concept. The moment where the concepts shift marks the occurrence of data drift.
4. Test the following drift detectors on the data stream to evaluate and compare their performance with the Machine Learning Model.
    - DDM (Drift Detection Method)
    - EDDM (Early Drift Detection Method)
    - Page Hinkley

    - ADWIN (Adaptive Windowing)

5. Experiments: Conducting experiments to analyze and compare the performance of the selected algorithms.
6. Library/Tool Development: Develop a simple, extendable library or tool that can replicate the results and be used in real-case scenarios. Develop an app to monitor and manage the concept drift of a real-world scenario.
7. Comprehensive Documentation: Compiling a detailed report that includes experimental findings, comparative analyses, discussions, and conclusions.

(3). Datasets

- Synthetic Datasets: Generated to simulate controlled drift scenarios, we can get algorithm performance analysis under known conditions. We expect to use them for research purposes (Eg: Friedman Dataset generator)
- Real-world datasets: Sourced from various domains. These data will be used in the app development process. (Eg: https://www.kaggle.com/datasets/microize/newyork-yellow-taxi-trip-data-2020-2019?select=taxi%2B_zone_lookup.csv).

(4). Similar Projects

1. Learning under Concept Drift: A Review
   By Jie Lu, Fellow, IEEE, Anjin Liu, Member, IEEE, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang
2. Frouros: Sample of similar tools: https://github.com/IFCA-Advanced-Computing/frouros
3. DeepChecks: A package for comprehensively validating machine learning models and data. It includes functionalities for checking data integrity, data drift, and model performance. (URL: https://github.com/deepchecks/deepchecks)
4. Handling Concept Drifts in Regression Problems – the Error Intersection Approach
   By Lucas Baier1, Marcel Hofmann2, Niklas Kühl1, Marisa Mohr2,3 and Gerhard Satzger1