Active learning in data stream classification with concept drift

Michał Dams, Maciej Nalepa

Wrocław University of Science and Technology

Abstract. Real-world data streams are rarely stationary. Data can change over time on several different ways. This phenomenon called "drift concept" has very negative influence on performance of machine learning algorithms. To deal with this problem, algorithms called drift detectors have been developed. In this paper we compare the performance of 3 popular drift detector methods: DDM, EDDM and Kolmogorov-Smirnov algorithms.

Keywords: Machine learning · Active learning · Concept drift · Drift detection.

1

Literature

1 Introduction

This section provides introduction to the project by explaining the basic concepts needed to understand the topic of the work. The list of these concepts include active learning, data stream, data stream classification and conception drift.

1.1 Active learning (in machine learning)

Active learning is well described in the article written by Burr Settles [4]. For more structured and broader knowledge on this topic it is recommended to refer to this article. Active learning is a subfield of machine learning. Its main assumption is that if the learning algorithm is allowed to choose the data from which it learns, it will perform better with less training. For many sophisticated supervised learning tasks, labeled instances are very difficult, time-consuming, or expensive to obtain. Active learning helps to lower the cost of teaching the model, by reducing the number of labels needed. "It asks queries in the form of unlabeled instances to be labeled by an oracle (e.g. a human annotator). In this way, the active learner aims to achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data. Active learning is well-motivated in many modern machine learning problems where data may be abundant but labels are scarce or expensive to obtain. Note that this kind of active learning is related in spirit, though not to be confused, with the family of instructional techniques by the same name in the education literature. " [4] Examples of active learning include:

- Classification and filtering
- Speech recognition
- Information execution

Possible Scenarios There are three main scenarios where active learning has been studied. In all scenarios, at each iteration a model is fitted to the current labeled set and that model is used to decide which unlabeled example we should label next. The three main active learning scenarios are:

- membership query synthesis - active learner is expected to produce an example that it would like us to label. This scenario requires that the model will be able to capture the data distribution well enough to create examples which are reasonable and that would have a clear label

- stream-based selective sampling - the learner gets a stream of examples from the data distribution and decides if a given instance should be labeled or not.
- pool-based sampling - In this scenario the learner has access to a large pool of unlabeled examples and chooses an example to be labeled from that pool. This scenario is most relevant for when gathering data is simple (scraping images/text from the web for instance), but the labeling process is expensive.

Query strategies There are 2 main query strategies in active learning that have been used to decide which instances are most informative: "Uncertainty Sampling" and "Query by Committee" In the first one the learner query the example which it is least certain about. "Query by Committee" is an approach to select samples in which disagreement amongst an ensemble of hypotheses is used to select data for labeling

1.2 Data stream

Data stream is an ordered sequence of objects continuously flowing over time. A collection equal to or exceeding one hundred thousand objects can be considered a data stream. In many cases, when analyzing a data stream, objects can be read only once or a small number of times. Keeping the entire stream in memory is usually impossible and highly undesirable, due to limited computational and memory resources , as the prediction made should happen in a fairly fast time. In some streams, objects may arrive so fast that the acquisition of all labels may be delayed and sometimes even impossible. There are two categories of data streams: stationary and non-stationary. Stationary streams have no variability over time. In other words, this is the case where there are no changes in the distribution of data throughout the stream. Non-stationary streams are closely related to a phenomenon called concept drift. Concept drift is characterized by the fact that the distribution of the feature space undergoes some changes over time. These changes are rather indeterminate and significantly degrade the classification quality of a given classifier. These changes can occur: abruptly, incrementally, recursively and gradually.

1.3 Data stream classification

The definition from "Stream Classification" article [5] - "Stream classification is a variant of incremental learning of classifiers that has to satisfy requirements specific for massive streams of data: restrictive processing time, limited memory, and one scan of incoming examples. Additionally, stream classifiers often have to be adaptive, as they usually act in dynamic, non-stationary environments where data and target concepts can change over time. To fulfill these requirements new solutions include dedicated data management and forgetting mechanisms, concept drift detectors that monitor the underlying changes in the stream, effective online single classifiers, and adaptive ensembles that continuously react to changes in the streams."

1.4 Conception drift

A phenomenon called concept drift, means changes in the distribution of data as it flows in. This phenomenon can degrade the quality of classification. Data stream in which concept drift appears is called a non-stationary data stream. There are 2 methods of classification of non-stationary data streams

- Online - data is collected object by object and this is how the method is learned and makes predictions
- Data chunk - data is collected as a dozen or a few hundred objects at once.

Drift detector (for data chunk classification) - an algorithm that does some analysis of the data or the quality of the classifier to determine if drift has occurred. Most often, when a drift is detected, the model learned on that data is reset and re-learned - rebuilt. It's worth to mention that during evaluation of a data stream we cannot use cross validation Assuming that we want to process our stream as data blocks, we can use the test-then-train approach - we divide the entire stream into data blocks of a certain size. We then use the nth block of data, first to test the method, and only then with this data do we train the method. The exception is the first block of data.

2 Literature survey

2.1 Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning [3]

Authors of this article noted that the issues of concept drift and class imbalance have been studied separately, but the joint problem is still underexplored. They emphasized that, most of the existing techniques have ignored the influence of complex data distribution on learning imbalanced data streams. To overcome these issues, they proposed an ensemble-based model for learning concept drift from imbalanced data streams with complex data distribution, called selection-based resampling ensemble (SRE). The article explains how the algorithm works, presents it characteristics like accuracy, recall or G-mean and compares its performance with several other popular classifier algorithms. Empirical studies demonstrate the effectiveness of SRE in learning nonstationary imbalanced data streams.

2.2 The Gradual Resampling Ensemble for mining imbalanced data streams with concept drift [2]

The article focuses on enhancing the data mining of class imbalanced data streams with concept drifts. The basic oversampling methods generally are easily affected by data difficulty factors. To overcome these issues, authors have proposed an ensemble classifier called Gradual Resampling Ensemble (GRE). GRE can handle data streams which exhibit concept drifts and class imbalance. It can

quickly adapt to a new conditions, regardless types of concept drifts. Through the gradual oversampling of previous chunks using the current minority events, the class distribution of past chunks can be balanced. Favorable results in comparison to other algorithms suggest that GRE can maintain good performance on minority class, without sacrificing majority class performance.

2.3 A Systematic Study of Online Class Imbalance Learning With Concept Drift [6]

This paper provides a review of current research progress in the field of online class imbalance learning, as well as in-depth experimental study, with the goal of understanding how to best overcome concept drift in online learning with class imbalance. The topic is challenging due to the combination of difficulties related to concept drifts and class imbalance. This paper first provided a comprehensive investigation of current research progress in this field, including current research focuses and open challenges. The article entirely dedicated to online learning and includes the problem description and definitions, the individual learning issues and solutions in class imbalance and concept drift, the combined challenges and existing solutions in online class imbalance learning with concept drift, and example applications. It reveals research gaps in the field of online class imbalance learning with concept drift.

2.4 Stream-learn — open-source Python library for difficult data stream batch analysis [1]

This article is oriented around "stream-learn" – a python package dedicated to imbalanced data stream and drifting analysis. This document describes motivation and significance of the software, its description, comparison with existing solutions, as well as the impact it will have on the future research. Main component of the package is a stream generator, which allows producing a synthetic data stream that can embrace a concept drift. It is capable of generating 3 most commonly occurring drifts: sudden, gradual or incremental. The package allows conducting experiments following established evaluation methodologies (i.e., Test-Then-Train and Prequential). Besides, estimators adapted for data stream classification have been implemented, including both simple classifiers and state-of-the-art chunk-based and online classifier ensembles. The package utilizes its own implementations of prediction metrics for imbalanced binary classification tasks to improve computational efficiency.

2.5 Classifier selection for imbalanced data stream classification [7]

The dissertation focuses on the use of Dynamic Classifier Ensemble Selection algorithms combined with preprocessing methods in the task of classifying static and streaming unbalanced data. The aim of the paper was to demonstrate the inherent ability of classifier selection algorithms to deal with unbalanced data

and to propose new efficient solutions to the problem of classifying highly unbalanced data streams, rarely addressed in the literature. The aim was achieved by proposing a strategy for querying labels, called "Budget Active Labeling Strategy", which is a combination of random approach to labelling with the approach used in active learning. Additionally, the dissertation mentions as an objective the development of the python library allowing analysis of difficult data streams. The library is called "stream-learn" and it is described and used for conducting experiments related to data streams. It is worth to mention that the dissertation cover broader range of problems and the solutions for them, but it does not coincide with the topic of this project

2.6 Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams [8]

This work connects two combined research directions, i.e., non-stationary data stream classification and data analysis with skewed class distributions. Authors of this article proposed a novel framework employing stratified bagging for training base classifiers to integrate data preprocessing and dynamic ensemble selection methods for imbalanced data stream classification. The proposed approach has been evaluated based on computer experiments carried out on 135 artificially generated data streams with various imbalance ratios, label noise levels, and types of concept drift as well as on two selected real streams. Four preprocessing techniques and two dynamic selection methods, used on both bagging classifiers and base estimators levels, were considered. Experimentation results showed that, for highly imbalanced data streams, dynamic ensemble selection coupled with data preprocessing could outperform online and chunk-based state-of-art methods.

2

Experiment Plan

This paper will focus on the implementation of an active learning model with concept drift detection on an artificially generated dataset. The research questions and targets are described below.

Target 1.

Measure quality and performance of 3 popular concept drift detectors. How accurately is it possible to detect different types of concept drift in the data streams.

Target 2.

Active learning in data stream classification with concept drift.

Experiment 1: comparative evaluation of DDM, EDDM and Komolgorov-Smirnoff drift detectors.

This experiment aims to test the quality of the drift detector by comparing it with 2 other popular algorithms based on following measures : number of examples until a change is detected, false detection rates, and miss detection rates. All the detection algorithms are implemented with the use of scikit-multiflow library.

Dataset

The data for the purposes of this work will be generated using a StreamGenerator delivered by the stream-learn library. It will be fed to the pipeline in chunks of a constant size of 200 elements. There will be 3 data streams generated, each containing a different concept drift: sudden, incremental and gradual drifts. The selected data streams characteristics are as follows:

- 3 classes
- 20 features
- 3 drifts
- chunk size 200
- chunks number 1000
- classes number 3
- concept sigmoid spacing 5.0 / 10.0 (depends on the type of drift)

Classifier

A Perceptron linear model is used as a classifier. For the detector model Gaussian Naive Bayes had been chosen. The Concept Drift Detection Technique can be any of the 3 algorithms (DDM, EDDM and Komolgorov-Smirnoff).

Drift generator

To evaluate the drift detectors there will be 3 types of concept drifts generated: sudden, gradual, and incremental.

Every concept drift triggered during the experiment is a real concept drift - this type of drift will always cause a drop in model performance.

Drift detection warning and detection thresholds are set to the default values from scikit-multiflow library (warning level default=2.0, detection level default=3.0)

Evaluator

Test-Then-Train is the method used to evaluate the model. It is one of two main techniques in which each individual data chunk is first used to test the classifier updating the existing model. The evaluator uses following metrics to measure the performance of the model: accuracy, precision and recall.

Detectors' performance metrics

The research consist of measuring the robustness of algorithms in detecting data changes in the streams. Assessing measures are: number of examples until a change is detected, false detection rates, and miss detections rates.

3

Results of the experiments and statistical analysis

This chapter describes experiments and analysis carried out in this study. In order to correctly compare different drift detectors the same datasets have to be used as an input. To achieve that several synthetic datasets had been generated with the "Streamlearn" library with the use of the same random seed value.

This analysis is used to get the measure of examined algorithms for detecting changes in the streams appearing over time . The benchmark of drift detectors will involve 3 different common drift detection methods. The methods mentioned are:

- DDM - Drift Detection algorithm
- EDDM - Early Drift Detection method.
- Komolgorov-Smirnoff algorithm

There are two types of horizontal lines on the following graphs: yellow and red ones. Yellow lines represent concept drift warning. Red lines symbolise concept drift detection.

Sudden concept drifts

Following figure visualize performance of the model on the data stream with 3 sudden concept drifts. The shift between one concept to a new one happens suddenly. There are 3 model's metrics: accuracy score, precision and recall. Each drift has negative influence of the performance of the model, which needs a lot of consecutive data chunks to adjust to the new data.

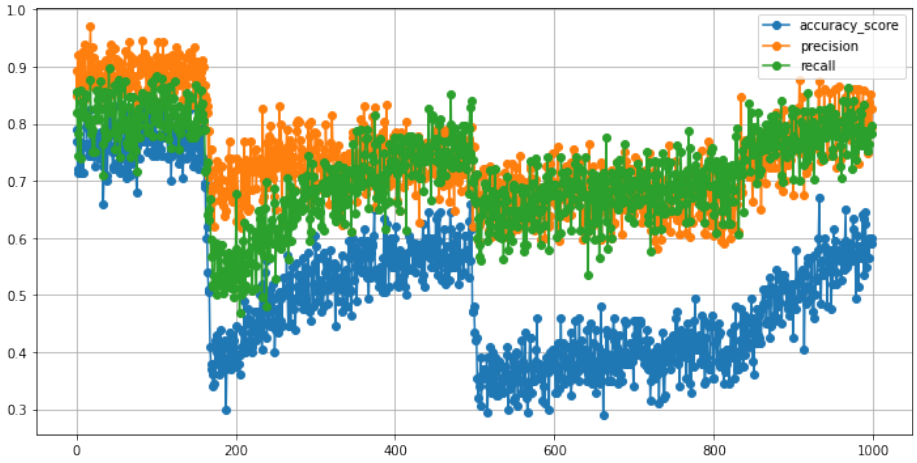

Fig. 1. Metrics score visualization

DMM

- average number of examples until a change is detected: 20
- false detection rates: 1
- miss detection rates: 0

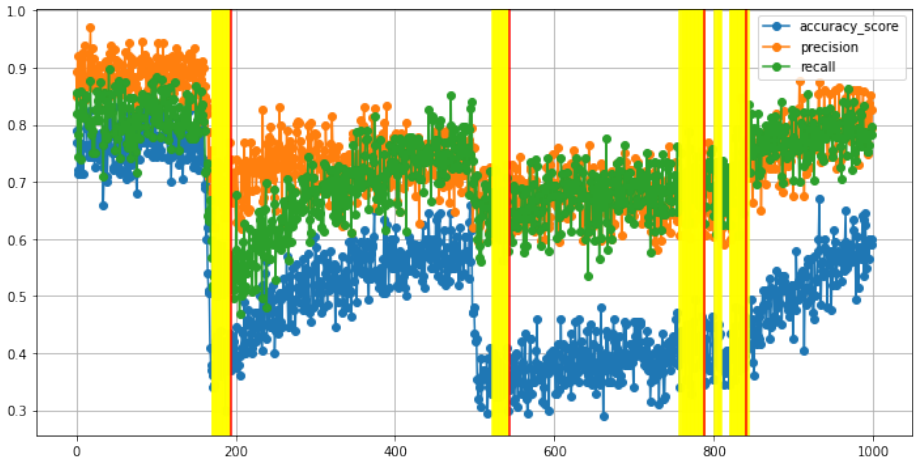

Fig. 2. DDM drift warnings and detections (Sudden drifts)

EDDM

- average number of examples until a change is detected: none
- false detection rates: 1
- miss detection rates: 3

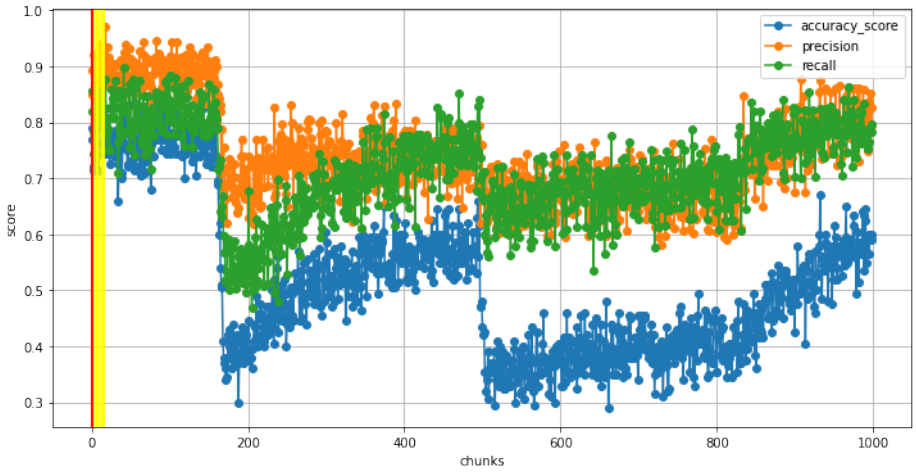

Fig. 3. EDDM drift warnings and detections (Sudden drifts)

Komolgorov-Smirnoff algorithm

- average number of examples until a change is detected: 42
- false detection rates: 2
- miss detection rates: 1

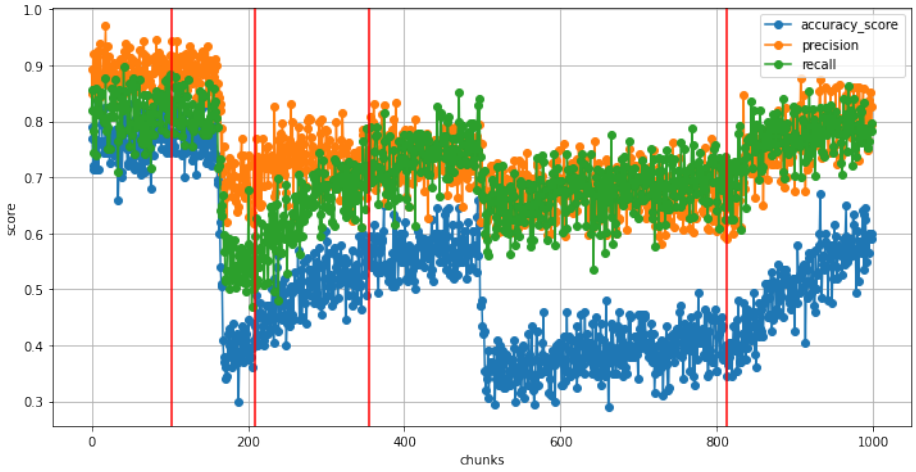

Fig. 4. Komolgorov-Smirnoff algorithm drift warnings and detections (Sudden drifts)

Gradual concept drifts

Next figure visualize performance of the model on the data stream with 3 gradual concept drifts. The target distribution changes progressively from one concept to another. The old concept starts to phase out and is to be replaced with the new one increasingly. As earlier, there are 3 model's metrics: accuracy score, precision and recall.

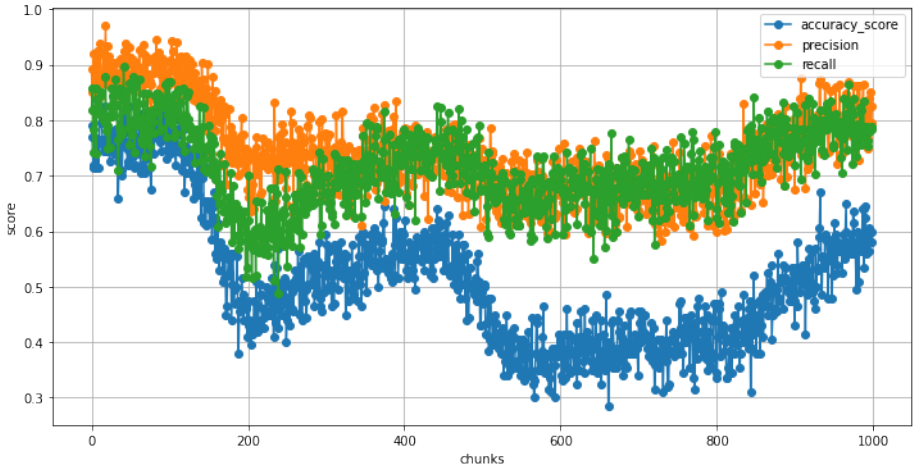

Fig. 5. Metrics score visualization

DMM

- average number of examples until a change is detected: none
- false detection rates: 1
- miss detection rates: 3

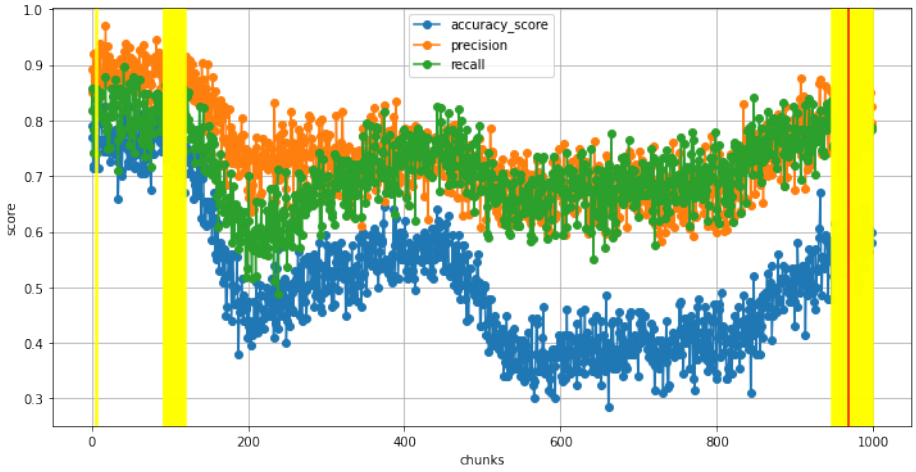

Fig. 6. DDM drift warnings and detections (Gradual drifts)

EDDM

- average number of examples until a change is detected: 86
- false detection rates: 1
- miss detection rates: 1

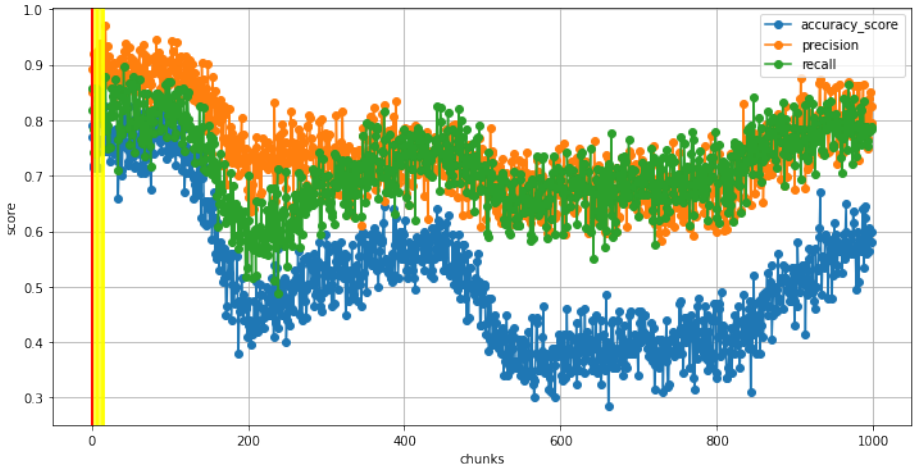

Fig. 7. EDDM drift warnings and detections (Gradual drifts)

Komolgorov-Smirnoff algorithm

– average number of examples until a change is detected: 20

– false detection rates: 3

– miss detection rates: 1

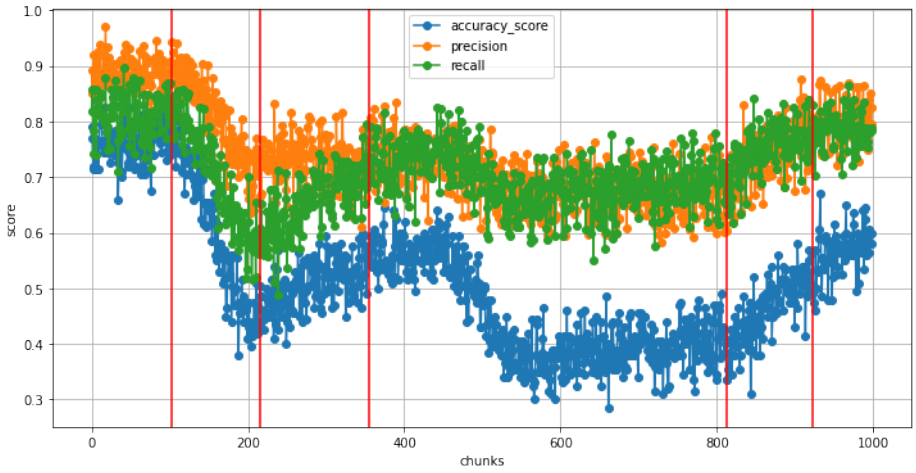

Fig. 8. Komolgorov-Smirnoff algorithm drift warnings and detections (Gradual drifts)

Incremental concept drifts

Next figure visualize performance of the model on the data stream with 3 incremental concept drifts. Occurs when a new concept replaces the old one slowly in a continuous manner. The shift between one concept to a new one is smooth. There are 3 model's metrics: accuracy score, precision and recall to assess it's performance.

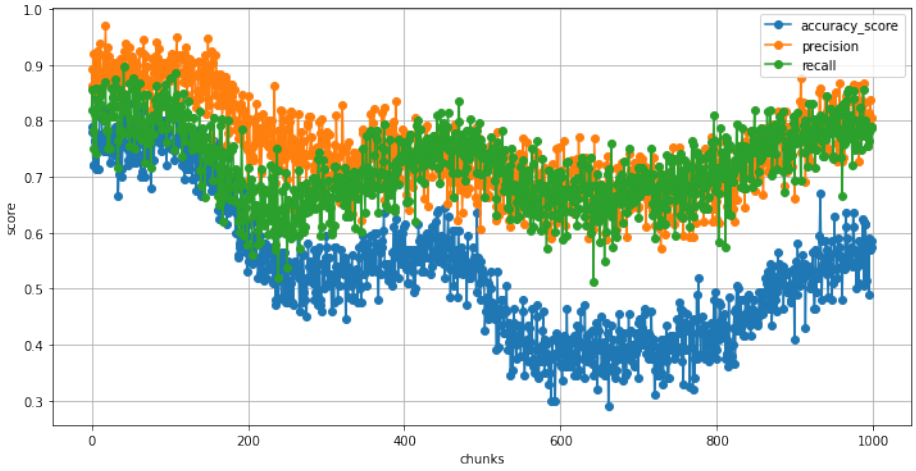

Fig. 9. Metrics score visualization

DMM

- average number of examples until a change is detected: none
- false detection rates: 0
- miss detection rates: 3

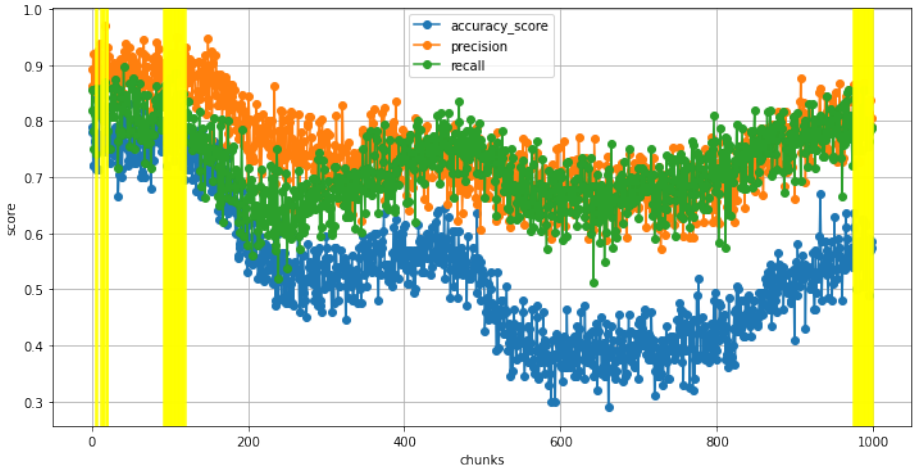

Fig. 10. DDM drift warnings and detections (Incremental drifts)

EDDM

- average number of examples until a change is detected: none
- false detection rates: 1
- miss detection rates: 3

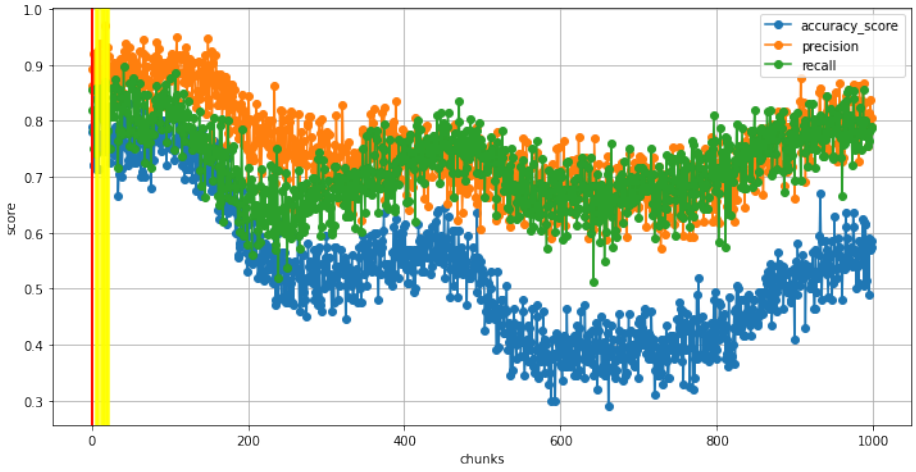

Fig. 11. EDDM drift warnings and detections (Incremental drifts)

Komolgorov-Smirnoff algorithm

- average number of examples until a change is detected: 20
- false detection rates: 2
- miss detection rates: 1

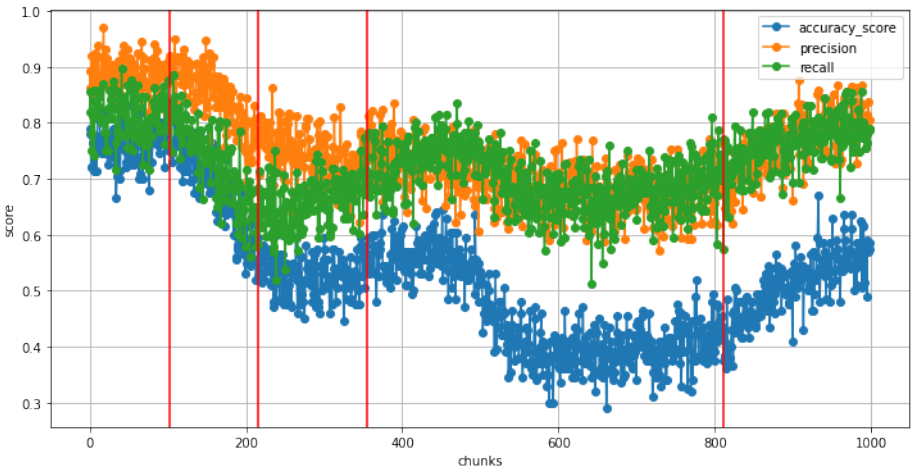

Fig. 12. Komolgorov-Smirnoff algorithm drift warnings and detections (Incremental drifts)

4

Comparative evaluation

The last chapter provides a summary of the findings in this work

DDM shows good performance for sudden drifts, but has difficulties detecting drift when the change is gradual or incremental. The algorithm need to store many data chunks for a long time, before the drift level is activated and there is the risk of overflowing the sample storage space.

EDDM aims to improve the detection rate of gradual concept drift in DDM, while keeping a good performance against sudden drifts. Additionally, beside keeping track of the error rate, this method monitors the average distance between two errors. Contrary to expectations, this detector is ineffectual for every type of drift. In every test EDDM indicates warning and drift detection at the beginning of the stream, to later miss every real concept drift. This behaviour is probably due to an implementation error and should not be considered trustworthy.

Komolgorov-Smirnoff algorithm shows altogether mediocre performance across the tests. However, when compared to DDM and EDDM it handles much better with gradual and incremental concept drifts. Intererestingly it doesn't emit any drift warnings.

Future work

This paper describes efforts and conclusions on the first target of the project: To create a drift detection algorithm and measure its quality. It examine how accurately it is in detecting different types of concept drifts.

Future work involves implementing active learning methods to the existing project, as well as conducting research on the influence they have on stream classification with concept drift.

References

1. Ksieniewicz, P., Zyblewski, P.: stream-learn - open-source python library for difficult data stream batch analysis. CoRR **abs/2001.11077** (2020), bluehttps://arxiv.org/abs/2001.11077
2. Ren, S., Liao, B., Zhu, W., Li, Z., Liu, W., Li, K.: The gradual resampling ensemble for mining imbalanced data streams with concept drift. Neurocomput. **286**(C), 150–166 (apr 2018), bluehttps://doi.org/10.1016/j.neucom.2018.01.063
3. Ren, S., Zhu, W., Liao, B., Li, Z., Wang, P., Li, K., Chen, M., Li, Z.: Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning. Knowledge-Based Systems **163**, 705–722 (2019)
4. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009)
5. Stefanowski, J., Brzezinski, D.: Stream classification. (2017)
6. Wang, S., Minku, L.L., Yao, X.: A systematic study of online class imbalance learning with concept drift. IEEE transactions on neural networks and learning systems **29**(10), 4802–4821 (2018)
7. Zyblewski, P.: Classifier selection for imbalanced data stream classification (2021)
8. Zyblewski, P., Sabourin, R., Woźniak, M.: Preprocessed dynamic classifier ensemble selection for highly imbalanced drifted data streams. Information Fusion **66**, 138–154 (2021)