# Concept Drift Detection based on Anomaly Analysis

Anjin Liu[1], Guangquan Zhang[1] and Jie Lu[1]

[1] Decision Systems & E-Service Intelligence Research Laboratory, Center for Quantum Computing and Intelligent System, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia

anjin.liu@student.uts.edu.au;

{guangquan.zhang,jie.lu} @uts.edu.au

**Abstract.** In online machine learning, the ability to adapt to new concept quickly is highly desired. In this paper, we propose a novel concept drift detection method, which is called Anomaly Analysis Drift Detection (AADD), to improve the performance of machine learning algorithms under non-stationary environment. The proposed AADD method is based on an anomaly analysis of learner's accuracy associate with the similarity between learners' training domain and test data. This method first identifies whether there are conflicts between current concept and new coming data. Then the learner will incrementally learn the non-conflict data, which will not decrease the accuracy of the learner on previous trained data, for concept extension. Otherwise, a new learner will be created based on the new data. Experiments illustrate that this AADD method can detect new concept quickly and learn extensional drift incrementally.

**Keywords:** Adaptive Intelligent Systems, Online Machine Learning, Incremental Learning, Concept Drift

## 1 Introduction

In the real world, there are a growing number of applications generating data continuously and requiring efficient machine learning algorithms to cope with this data. For example, personal assistance applications dealing with information filtering, macroeconomic forecasting, bankruptcy prediction or individual credit scoring [1]. Moreover, the fast pace of preference changing of the target customers (*concept drift*) is also a challenge to existing learning algorithms. As a result, conventional machine learning algorithms, which hold a stationary distribution assumption, will be replaced by more efficient online learning algorithms, which have the ability to adapt to new environment quickly, sooner or later.

The issue of *concept drift* refers to the change of the distribution underlying the data at different time steps [2], in which the term *concept* refers to the distribution of a problem at a certain time step. Concept drift will lead to the predictions of well-trained classifiers become less accurate as time passes. More formally, lets denote the feature vector as x and the class label as y, then an infinite sequence of (x, y) presents

the data stream, $p_t(x, y)$ is the distribution of data chuck at time step t, the term concept drift means that $p_t(x, y) \neq p_{t+1}(x, y)$ [3]. Recall the Bayesian Probability Theory, $p(x, y)$ can be decompose as $p(x, y) = p(x) \times p(y \mid x)$. In Kelly, et al. [4] publication, they concluded that concept drift can be caused by the drifting of $p(x)$ over time t (it can also be written as $p(x \mid t)$), or the drifting of $p(y \mid x)$, which is the conditional probability of feature x, or both. Virtual concept drift is neither the change of $p(x \mid t)$ nor $p(y \mid x)$. It is caused by the sampling shift of current $p(x \mid t)$ or $p(y \mid x)$, or both.

Concept drift can be categorized into different types based on different criteria as shown in the literatures. Minku, et al. [4] proposed that concept drift could be categorized into 14 types based on the drifting speed, severity, predictability, frequency and recurrence. In the real-world applications, the types of concept drift can be varied and mixed. In addition, in some special cases, virtual drift may have the same effect on learning model as concept change. For example, $p_{t+1}(x \mid t)$ is the sampling shift of $p_t(x \mid t)$ and they are not equal, they might be treated as two different concepts and assigned to two learners separately. If the data chuck at t+2 with distribution of mixed $p_{t+1}(x \mid t)$ and $p_t(x \mid t)$, neither $\text{classifier}_t$ nor $\text{classifier}_{t+1}$ would achieve a high accuracy. These issues have made concept drift even difficult to be solved. Current ensemble drift detection and handling approaches treats virtual concept drift and real concept drift as the same problem. These approaches detect drifts based on the outputs of learner at each time step without considering whether the drift is a sampling shift or a new one. As a result, their performances are limited.

Motivated by these issues, we propose a novel drift detection method for online learning algorithms, which runs anomaly analysis on the accuracy associate with the similarity between training domain and test data. In the anomaly analysis, we focus on the data that was correctly classified by existing learners. We compare the similarity of the distribution between the old correctly classified data and the new data. Under normal circumstances, including virtual concept drift, the similarity and the accuracy should stay at a stable ratio. Otherwise, it can be either a concept change or noise. Our approach is capable to high light the data with unknown distribution and identifies the conflicts instances. Therefore, both virtual drift and real concept drift could be handled well.

The organization of this paper is as follow: In the next section, we survey the state of art drift detection methods based on learners' outputs. Section 3 explains and presents the details of our new proposed AADD method. Section 4 presents the preliminary and the evaluation results of the proposed AADD method. Section 5 concludes this paper and discusses some future works.

## 2 Related work

This section formally presents the problem of concept drift, and analyzes the advantages and drawbacks of established literature with regard to concept dirt detection based on learners' outputs.

### 2.1 Problem Description: Concept Drift

We reference the definition from a most recent concept drift review which was given by Zliobaite [1], to present concept drift. In classification problems, at every time step **t** we have historical data (labelled) available $\mathbf{x}^{\mathbf{H}} = (\mathbf{x}_1, \ldots, \mathbf{x}_t)$. For each time increasing t+1, a new instance $\mathbf{x}_{t+1}$ arrive. The task is to predict a label $c_i$, where $c_i$, $c_2, \ldots, c_k$ is the set of class labels, for every new coming instance $\mathbf{x}_{t+1}$. The optimal classifier to classify $\mathbf{x} \rightarrow c_i$ is completely determined by a prior probabilities for the classes $p(c_i)$ and the class-conditional probability density functions (pdf) $p(\mathbf{x}|c_i)$, $i = 1, \ldots, k$. They define a set of prior probabilities of the classes and class-conditional pdfs as concept or data source, denote it **S**:

$$\mathbf{S} = (p(c_1), p(\mathbf{x}|c_1)), (p(c_2), p(\mathbf{x}|c_2)), \ldots, (p(c_k), p(\mathbf{x}|c_k)) \quad (1)$$

Every instance $\mathbf{x}_t$ is generated by a source $\mathbf{S}_t$. If all the data is sampled from the same source, i.e. $\mathbf{S}_1 = \mathbf{S}_2 = \ldots = \mathbf{S}_{t+1} = \mathbf{S}$ we say that the concept is stable. If for any two time points i and j exists $\mathbf{S}_i \neq \mathbf{S}_j$, we say that there is a concept drift.

However, for some special situation $\mathbf{S}_{new}$, $\mathbf{S}_i \neq \mathbf{S}_j$ and $\mathbf{S}_i \cup \mathbf{S}_j = \mathbf{S}_{new}$, if we treat them as three different concepts, we will have to train three different learners. Moreover, $\mathbf{S}_{new}$ would not be able to take the advantages of previous concept or data source $\mathbf{S}_i$ and $\mathbf{S}_j$. Therefore, we suggest learning this type of drift incrementally.

### 2.2 Drift Detection Methods by Outputs of Learners

To the best of our knowledge, explicit drift detection can be categorized into three groups, detecting drift by data distribution [5], learner outputs [6, 7] and competence model [3]. A more comprehensive literature review can be found in [3]. Comparing with other types of drift detection methods, drift detection by learners' outputs is the most intuitive and has a relative low computational cost. On the other hand, this type of drift detection method can only take reactions after drift. In this section, we will give a literature review on the drift detection by learners' outputs.

Drift Detection Method (DDM), which proposed by Gama et al. [7], detects concept drift by tracing the online error rate of the learning algorithm. It treats the error of a set of examples as a Bernoulli trail random variable. The number of errors in a sample set should follow Binomial distribution. The changes in the errors of the algorithm indicate the changes of the class distribution. Since DDM assesses a learner through its overall performance, it is more suitable for identifying concept change and rebuilding models rather than updating an existing learner, which means that it cannot handle slowly gradual drift [6]. Hence, Baena-García et al. [6] proposed a upgraded version of DDM, Early Drift Detection Method (EDDM), to improve the detection in the presence of gradual concept drift. The difference is that EDDM considers the distance between two consecutive erroneous classifications instead of the overall error rate. They assume that the change of the distance reflects the changes of the current concept. Moreover, they applied a warning system to reduce the error detections caused by noise. In spite of that, EDDM is still very sensitive to noisy examples [8].

## 3 Anomaly Analysis Drift Detection Methods

This section presents the AADD method. In Section 3.1, we give an intuitive explanation of the proposed method. In Section 3.2, a detailed description of the AADD method is explained.

### 3.1 The Idea of AADD

The main reason why concept drift would lead to well-trained learner becoming inaccurate is either the test data distribution is not learned sufficiently or the class labels changed with the same distribution. If the environment is not noise free, it can also cause the same problem. Therefore, from this point of view, we believe that if we could monitor the performance of a learner on its confident data distribution on which the learner always have a high accuracy, we would be able to quickly identify what caused the drop of accuracy. For example, if a batch of data chuck is from a new distribution and has no conflict with current learner, the similarity would be low and the accuracy would be low as well. We use a table to illustrate the differences.

**Table 1.** Concept drift similarity & accuracy analysis

| Drift or Noise | Similarity Change | Accuracy Change |
|---|---|---|
| Noise | No change | Fluctuating |
| Concept extension (Distribution extended) | Decreasing | Decreasing |
| Concept change (Conflicts under the same distribution) | No change | Decreasing |

The core idea of AADD method is that monitoring the similarity and accuracy of the new available sample data at each time step. By taking the similarity into consideration, AADD would be able to identify whether the incorrect predictions is caused by the conflicts between current learning model and new concept or they are caused by unlearned distribution or noise. In addition, with similarity functions, data can be only trained and analyzed once in an online manner.

### 3.2 The Method Description

At each time step of the data stream, we assume that there are 2 batches of data, $D_{train}(t)$, $D_{test}(t)$. The $D_{train}(t)$ can be a subset of $D_{test}(t)$ with known labels or they share similar distributions. After utilizing $D_{train}(t)$ to create and update a learner $\boldsymbol{L}$ (step 3, 8 and 11), we change the label of an instance in $D_{train}(t)$ to *TRUE* if it be classified correctly by $\boldsymbol{L}$, else to *FALSE* to form a new dataset $D`_{train}(t)$. And then we use $D`_{train}(t)$ to create or update learner $\boldsymbol{L}$'s similarity function (step 4, 9 and 12). For each new coming train data chuck, we run anomaly analysis to verify the ratio of the similarity and the accuracy, incremental learning it if no conflict with current learner, otherwise, build new learner base on it, Step 6. The AADD method is described as follows:

Anomaly Analysis Drift Detection:

**Input:**

Noise sensitive parameter θ

Updateable learning algorithm

For each time step $D_{train}$, $D_{test}$

**Output:**

Predictions of $D_{test}$

```
1.  For t = 0: numBatch
2.    If t = 0
3.      buildNewLearner(D_train(t))
4.      buildSimilarityFunctions(D`_train (t))
5.    Else
6.      conflictDetection(D_train (t))
7.      If no conflict
8.        incrementalLearning(currentLearner, D_train (t))
9.        incrementalLearning(similarityFunction, D`_train (t))
10.     Else
11.       buildNewLearner(D_train (t))
12.       buildSimilarityFunctions(D`_train (t))
13.     End
14.   End
15. End
16. classification(currentLearner, D_test(t))
```

The conflict between active learner and new coming data is identified by the following function, which indicates the abnormal data batch:

$$acc_{Dtrain} \times similarity_{Dtrain} < acc_{learner} \times similarity_{Dtrain} - \theta \times (1 - similarity_{Dtrain}) \quad (2)$$

where $acc_{Dtrain}$ is the accuracy of the training data, $similarity_{Dtrain}$ is the similarity of the training data, $acc_{learner}$ is the stable accuracy of current learner, θ is a parameter to control the sensitive to noise, the smaller value the θ is, the more sensitive to noise.

## 4 Experiments and Result Analysis

In this section, we present our experiment results of AADD method. First, in Section 4.1, we give the configuration details of the experiments. Secondly, in Section 4.2, we show the accuracy change caused by concept drift and plot the anomaly points at each time step on a graph.

### 4.1 Experiment Setup

In order to test AADD method, we applied it on the SEA Concepts [9], which has been used by many researchers as a standard to test algorithms for concept drift. This

dataset have two class values and three features, with only two features being relevant and the third one being noise. Class values are assigned based on the sum of the two relevant features. If the sum of these two features of one instance is lower than a given threshold, this instance will be assigned to class 1. Otherwise, it will be assigned to class 2. The threshold will be updated after a predefined time step to simulate an abrupt shift in the class boundary. The values of the three features are uniformly distributed between 0 and 10, and the threshold is changed three times throughout the experiment with increasing severity 8→9→7.5→9.5. For example:

**Table 2.** SEA Concepts

| attribute 1(0 - 10) | attribute 2 (0 - 10) | attribute 3 (noise) (0 - 10) | Class {1, 2} |
|---|---|---|---|
| Threshold = 8 (if attribute1 + attribute2 < 8 then class 1, otherwise class2) | | | |
| 8.498129 | 1.243221 | 5.675182 | class 2 |
| … | … | … | … |
| Threshold = 9.5 (if attribute1 + attribute2 < 9.5 then class 1, otherwise class2) | | | |
| 1.406376 | 0.738125 | 2.598439 | class 1 |
| … | … | … | … |

Our testing procedure is identical to that described in [9]: 50000 instances training data, 250 instances each time step. In our experiment, we used the weka naive bayes updateable classifier as the base learner and the similarity functions. We run the test 10 times and calculate the average accuracy as the final result. The θ we used here is 0.05. Meanwhile, we also run a test with the same classifier but manually discard old learners and create new one at each drift time step. At last, we put our implementation of NSE++ [10] to demonstrate how a concept drift may affect the performance of learning model. The parameters of NSE are same as it was suggested in their paper, a = 10, b = 0.5 data size = 250, base learner is weka naive bayes updateable classifier.

### 4.2 Experiment Results

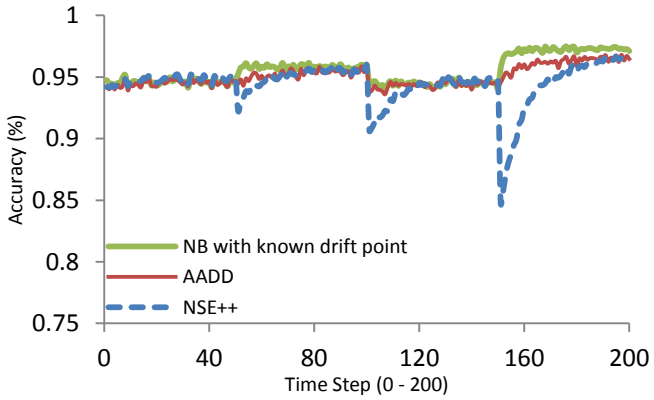


**Fig. 1.** Compare AADD method with NB and NSE++

Fig 1 is the performance of three concept drift algorithms on SEA concept. As shown above, NB with known drift point was forced to drop old learner and create new one at each drift time step. Therefore, it is barely affected by the change of concept. The difference of accuracy between each concept period is caused by the concept itself (some concepts are easier to be predicted correctly). By contrast, NSE++ with same base classifier, which does not have active drift detection method, have a significant accuracy drop at each drift time step. After that, it recovers from drift gradually. Regarding to AADD, there is no significant accuracy drop at drift time step. However, indeed, there is a recovery period for AADD to get back to normal accuracy by incremental learning. This is because of that the AADD method can identify the conflict learners once there is a drift and abandon them before they could cause any negative effect on new concept predictions.

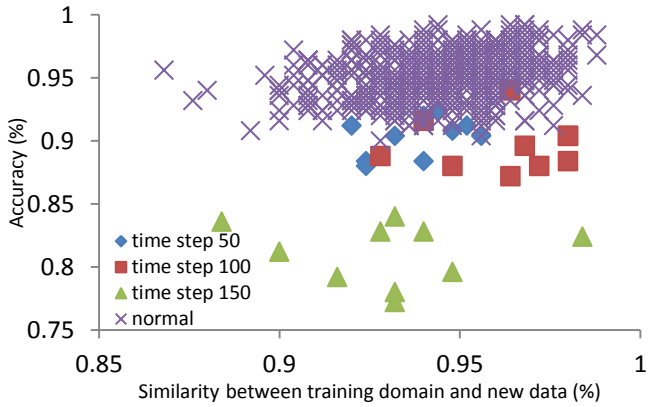


**Fig. 2.** Anomaly analysis of the accuracy associate with similarity

Fig 2 shows the distribution of the accuracy of the current learner on new coming data and the similarity returned by its similarity function. We put all the 10 runs into one figure so that we could have enough drift points to present the differences between stable points and drifting points. As described in section 4.1, time step 50 is the time when threshold change from 8→9 (presented with diamond), time step 100 is 9→7.5 (presented with square) and time step 150 is 7.5→9.5 (presented with triangle). The points at time step 150 are easy to be understood. As it has the biggest concept change *absolute*(7.5 – 9.5) = 2, the accuracy is much lower than the others with the same similarity. Regarding to time step 50 and 100, both of them have an accuracy drop compare to the normal data points. However, as time 50 has a relatively small change *absolute*(8 – 9) = 1, the shifts of similarity of these points are relatively stable. By contrast, time step 100 has a larger change *absolute* (9 – 7.5) = 1.5, so it would spread into a larger region. From fig 2, we can also see that after quick reaction to adapt to new concept, which is creating new classifier at time step 50, 100 and 150, the rest time step of these new concepts have go back to normal position. There is no other anomaly point after concept change. In other words, time step 51…99, 101 …

159 and 151… 200 are under stable concept. Regarding to those drifting points that are mixed with normal points, it is because of that the random data at that time does not have a clear drift distribution and the later time step was recognized as drift point.

From the results above, it is manifest that anomaly analysis of the accuracy and the similarity would be helpful to identifying concept drift.

## 5 Conclusion and Further Study

This paper introduces a novel drift detection method, called AADD, based on the anomaly analysis of the learner's accuracy corresponding to the similarity between its training domain and test data. It has the ability to distinguish between unknown distribution and conflict distribution and to solve them separately. The AADD method is capable to detect concept extension and change efficiently and highlights the conflict instances at the same time. It offers a great convince to drift handling.

Our next attempt will aim to combine some drift handling approaches with AADD to research how to take the advantages of AADD for improving the performance of learning under non-stationary environment.

## 6 References


1. I. Zliobaite, "Learning under concept drift: an overview," Overview", Technical report, Vilnius University, 2009 techniques, related areas, applications Subjects: Artificial Intelligence 2009.
2. I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes, "Active Learning With Drifting Streaming Data," *Neural Networks and Learning Systems, IEEE Transactions on,* vol. 25, pp. 27-39, 2014.
3. N. Lu, G. Zhang, and J. Lu, "Concept drift detection via competence models," *Artificial Intelligence,* vol. 209, pp. 11-28, 2014.
4. M. G. Kelly, D. J. Hand, and N. M. Adams, "The impact of changing populations on classifier performance," in *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 367-371.
5. T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi, "An information-theoretic approach to detecting changes in multi-dimensional data streams," in *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications*, 2006.
6. M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, "Early drift detection method," 2006.
7. J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence–SBIA 2004*, ed: Springer, 2004, pp. 286-295.
8. K. Nishida and K. Yamauchi, "Detecting concept drift using statistical testing," in *Discovery Science*, 2007, pp. 264-269.
9. W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," presented at the Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, California, 2001.
10. R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans Neural Netw,* vol. 22, pp. 1517-31, Oct 2011.