Expert Systems with Applications 41 (2014) 8144–8156

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Review

A comparative study on concept drift detectors

Paulo M. Gonçalves Jr. [a,*], Silas G.T. de Carvalho Santos [b], Roberto S.M. Barros [b], Davi C.L. Vieira [b]

[a] Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco, Brazil
[b] Centro de Informática, Universidade Federal de Pernambuco, Brazil

ARTICLE INFO

Article history:
Available online 19 July 2014

Keywords:
Data streams
Time-changing data
Concept drift detectors
Comparison

ABSTRACT

In data stream environments, drift detection methods are used to identify when the context has changed. This paper evaluates eight different concept drift detectors (DDM, EDDM, PHT, STEPD, DOF, ADWIN, Paired Learners, and ECDD) and performs tests using artificial datasets affected by abrupt and gradual concept drifts, with several rates of drift, with and without noise and irrelevant attributes, and also using real-world datasets. In addition, a 2^k factorial design was used to indicate the parameters that most influence performance which is a novelty in the area. Also, a variation of the Friedman non-parametric statistical test was used to identify the best methods. Experiments compared accuracy, evaluation time, as well as false alarm and miss detection rates. Additionally, we used the Mahalanobis distance to measure how similar the methods are when compared to the best possible detection output. This work can, to some extent, also be seen as a research survey of existing drift detection methods.

1. Introduction

Learning from data is a topic that is addressed by several research fields, such as data mining, machine learning, and pattern recognition. Traditionally, this learning is performed in static environments, where a dataset is available for the classifier to read as many times as needed for training. Also, another characteristic is that the target concept to be learned is fixed. Several classifiers have been proposed and, nowadays, many efficient classifiers are available.

Currently, however, several expert systems have to deal with data that flow continuously. They can therefore not be stored for later analysis and must be processed as they arrive. Examples of applications that have to handle this requirement, known as *data streams*, are TCP/IP traffic, GPS data, sensor networks, and customer click streams (Gama, 2010). Compared to batch learning, data stream processing imposes restrictions on memory usage, limited testing time and learning, and one-time reading data.

Another difficulty of data streams is that the target concept may change in time, usually after a minimum stability period (Gama, 2010), a problem known as *concept drift*. The problem of concept drift has received a lot of attention over the past few years, mainly because it negatively impacts the accuracy of the classifiers that learned on the basis of past training instances. Some examples of situations where concept drifts may occur are "spam, fraud or climate change detection" (Elwell & Polikar, 2011). Concept drift may be classified in terms of the speed of change and the reason of change. Considering the speed of change, an *abrupt concept drift* occurs when a change between two contexts happens suddenly, while a *gradual concept drift* describes the case where the transition between two contexts occurs smoothly.

With respect to the reason of change, a *real concept drift* occurs "when a set of examples has legitimate class labels at one time and has different legitimate labels at another time" (Kolter & Maloof, 2007). On the other hand, a *virtual concept drift* occurs when "the target concepts remain the same but the data distribution changes" (Delany, Cunningham, Tsymbal, & Coyle, 2005). In practice, however, "virtual concept drift and real concept drift often occur together" (Tsymbal, Pechenizkiy, Cunningham, & Puuronen, 2008).

Several approaches have been proposed to deal with concept drift, including (a) adapting a classifier's internal structure and (b) using ensemble classifiers. In this paper, we focus on the methods used to identify the occurrence of a concept drift. Based on the concept drift identification, these methods can be used inside other classifiers to modify either the internal structure or the number of classifiers in an ensemble. Examples of classifiers that internally use concept drift detection methods are Diversity for Dealing with Drifts (DDD) (Minku & Yao, 2012) and Recurring Concept Drifts (RCD) (Gonçalves & Barros, 2013). In addition, they can be used in conjunction with any classifier to identify a drift. Drift detection

* Corresponding author.
E-mail addresses: paulogoncalves@recife.ifpe.edu.br (P.M. Gonçalves Jr.), sgtcs@cin.ufpe.br (S.G.T. de Carvalho Santos), roberto@cin.ufpe.br (R.S.M. Barros), dclv@cin.ufpe.br (D.C.L. Vieira).

http://dx.doi.org/10.1016/j.eswa.2014.07.019
0957-4174/

methods usually use a specific classifier (also called "base learner") to analyze its accuracy and indicate when a drift has occurred.

The objective of this paper is to compare several concept drift detection methods and analyze under which conditions they perform well. In every study that proposes a new drift detection method a number of tests are performed to verify its usefulness. Unfortunately, the base learner, the datasets, the metrics, and the types of drifts used in the experiments vary considerably, as well as the other methods with which they are compared. This makes it difficult to choose which method to use in different situations.

Thus, in this paper eight concept drift detection methods (DDM (Gama, Medas, Castillo, & Rodrigues, 2004), EDDM (Baena-García et al., 2006), PHT (Page, 1954), ADWIN (Bifet & Gavaldà, 2007), Paired Learners (Bach & Maloof, 2008), ECDD (Ross, Adams, Tasoulis, & Hand, 2012), DOF (Sobhani & Beigy, 2011), and STEPD (Nishida & Yamauchi, 2007)) were compared in terms of accuracy, evaluation time, false alarm and miss detection rates, as well as distance to the drift point. These methods were selected from the ones with the highest number of citations, provided that there was a freely available implementation or at least a detailed description of the algorithm to allow a direct implementation and avoid incorrect results and/or slow performance. The experiments were performed using the most cited datasets in the area, including artificial datasets affected by abrupt and gradual concept drifts, with variable rates of drift, and real-world ones. To the best of our knowledge, this is the broadest experiment to date comparing a wide range of concept drift detectors in datasets with different characteristics and using different metrics.

The rest of this paper is organized as follows: Section 2 presents the drift detection methods used in the experiments and the techniques they use to identify concept drifts. Section 3 describes the parameters used in the drift detectors, the datasets used in the experiments, and the adopted evaluation methodology. The results obtained in the experiments are analyzed in Section 4. Finally, Section 5 presents our conclusions.

2. Background

Concept drift detection methods use a base learner (classifier) to classify incoming instances. For each instance, it outputs a class prediction, which is usually compared to the true class label. Based on the classification result (true for correct classification and false otherwise), the drift detection method can indicate whether a drift has occurred or not. Finally, the base learner is trained on the instance. This process is repeated for each incoming instance.

Several concept drift detection methods have already been proposed, one of which is the Drift Detection Method (DDM) (Gama et al., 2004). This uses a base learner to classify incoming instances and the classification result is used to compute the online error-rate of the base learner. The classification result indicates whether the base learner classified the arriving instance correctly or not. If the base learner correctly classifies the actual instance, the error-rate decreases. DDM considers that, when the concept changes, the base learner will incorrectly classify the arriving instances that are created based on a different data distribution. Thus, if the error-rate increases, it is an indication of a concept drift.

On the other hand, while the distribution is stationary, i.e., it remains unchanged, the error rate decreases. Therefore, the error rate (p_i) and the standard deviation $\left(s_i = \sqrt{p_i(1-p_i)/i}\right)$ are computed and these values are stored when $p_i + s_i$ reaches its minimum (obtaining p_{min} and s_{min}). When $p_i + s_i \geqslant p_{min} + 2 \cdot s_{min}$, a warning level is reached and instances are stored in anticipation of a possible concept drift. If $p_i + s_i \geqslant p_{min} + 3 \cdot s_{min}$, a drift level is reached, indicating a context change. The base learner and the values of p_{min} and s_{min} are then reset and a new base learner is trained on the examples stored since the warning level.

The Early Drift Detection Method (EDDM) (Baena-García et al., 2006) is similar to DDM but, instead of using the error rate, it uses the distance-error-rate of the base learner to identify whether a drift has occurred. This metric computes the number of examples between two classification errors. When there is no concept drift, the base learner improves its predictions and the distance between errors increases. On the other hand, when a concept drift occurs, the base learner makes more mistakes and the distance between error decreases. As described in the original paper, EDDM is best suited to dealing with slow gradual concept drifts. The average distance between two errors (p_i) and its standard deviation (s_i) are computed. These values are stored when $p_i + 2 \cdot s_i$ reaches its maximum value (obtaining p_{max} and s_{max}). This value indicates that the base learner best approximates the current concept.

Like DDM, EDDM defines two thresholds. When $(p_i + 2 \cdot s_i)/(p_{max} + 2 \cdot s_{max}) < \alpha$, the warning level is reached and the instances are stored anticipating a concept drift. The drift level is reached when $(p_i + 2 \cdot s_i)/(p_{max} + 2 \cdot s_{max}) < \beta$, indicating a change in the context. The values of α and β are 0.95 and 0.9, respectively, the same chosen in the original paper after some experimentation. The base learner and the values of p_{max} and s_{max} are reset and a new base learner is trained on the examples stored since the warning level.

The Page-Hinkley Test (PHT) (Page, 1954) is a sequential analysis technique that can be used as a concept drift detector. It computes the observed values (here we used the actual accuracy of the classifier) and their mean up to the current moment. When a concept drift occurs, the base learner will fail to correctly classify incoming instances, making the actual accuracy decrease. As a result, the average accuracy up to the current moment also decreases. The cumulative difference between these two values (U_T) and the minimum difference between these two values (m_T) are computed. Higher U_T values indicate that the observed values differ considerably from their previous values. When the difference between U_T and m_T is above a specified threshold corresponding to the magnitude of changes that are allowed (λ), a change in the distribution is detected. Higher λ values result in fewer false alarms, but might miss or delay some changes.

Adaptive Windowing (ADWIN) (Bifet & Gavaldà, 2007) is another drift detection method. It uses sliding windows of variable size, which are recomputed online according to the rate of change observed from the data in these windows. The algorithm dynamically enlarges the window (W) when there is no apparent change in the context, and shrinks it when a change is detected. The algorithm attempts to find two sub-windows of W that exhibit distinct averages. If that occurs, it concludes that the corresponding expected values are different, meaning that the older portion of the window is based on a data distribution different than that of the present one, and is therefore dropped. The maximum length of the window is "statistically consistent with the hypothesis that there has been no change in the average value inside the window" (Bifet, Holmes, Pfahringer, & Frank, 2010). Additionally, ADWIN provides rigorous guarantees of its performance, in the form of limits on the rates of false positives and false negatives.

The Paired Learners (PL) (Bach & Maloof, 2008) method, as the name suggests, uses two learners: a stable and a reactive one. The stable learner predicts based on all of its experience, while the reactive one predicts based on a window of recent examples. A circular list of bits with length w (the same length as the window) stores the value *one* if an instance was incorrectly classified by the stable learner and correctly classified by the reactive learner, and *zero* otherwise. If the number of bits of this circular list set to *one* exceeds a parameterized value θ, it indicates that the reactive learner, trained on recent instances, has a better predictive