Information Sciences 355–356 (2016) 127–151

Contents lists available at ScienceDirect

Information Sciences

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

Recognizing input space and target concept drifts in data streams with scarcely labeled and unlabelled instances

Edwin Lughofer[a,*], Eva Weigl[b], Wolfgang Heidl[b], Christian Eitzinger[b], Thomas Radauer[c]

[a] Department of Knowledge-based Mathematical Systems/Fuzzy Logic Laboratorium Linz-Hagenberg, Johannes Kepler University Linz, Austria
[b] Profactor GmbH, Steyr-Gleink, Austria
[c] Sony DADC, Austria

ARTICLE INFO

Article history:
Received 31 July 2015
Revised 21 January 2016
Accepted 16 March 2016
Available online 24 March 2016

Keywords:
Data stream classification
Input space and target concept drift
Drift detection
Scarcely labeled and unlabeled streams
Semi-supervised and unsupervised performance indicators
Single-pass active learning filter

ABSTRACT

In classification-based stream mining, drift detection is essential in order to (i) inform operators when unintended system changes occur and (ii) make classifier updates more flexible when changes are intentional. Current detection approaches usually rely on the assumption that fully supervised labeled streams are available for monitoring (the changes in) classifier performance. This is an unrealistic scenario in many on-line real-world applications as true class labels would have to be known, which usually requires tedious feedback efforts of operators working with the systems. We propose *two techniques to improve economy and applicability of current drift detection techniques*: (i) a semi-supervised approach that employs single-pass active learning filters to select the most interesting samples for supervising classifier performance and (ii) a fully unsupervised approach based on the degree of overlap between a classifier's output certainty distributions that can be applied to any unlabeled classification stream. For both variants, a specific handling of imbalanced class distributions in the streams is proposed, which allows also possible downtrends in classifier behavior for under-represented classes to be observed. The statistical monitoring of classifier behavior relies on a modified version of the Page-Hinkley test, where a fading factor and an automatic thresholding concept (based on the Hoeffding bound) were introduced to render it more flexible for detecting successive drift occurrences in a stream. We compared our approaches to the fully supervised variant in two real-world on-line applications, including a systematic analysis of the capabilities of our methods. The semi-supervised approach was able to detect real as well as artificially built-in drifts in these streams with a similar delay (of about 5–6 min) as the supervised variant, and this with only 20% actively selected samples. The unsupervised variant was able to detect input space drifts with reasonable delays as well, but failed to detect target concept drifts — using both approaches in tandem therefore allows us to distinguish between input space and target concept drifts.

* Corresponding author. Tel.: +43 72363343431.
E-mail address: edwin.lughofer@jku.at (E. Lughofer).

http://dx.doi.org/10.1016/j.ins.2016.03.034
0020-0255/

1. Introduction

1.1. Motivation and state of the art

In today's industrial, media, health-care and other forms of applied decision support systems, *data stream classification* [10,11,38] plays an essential role for coping with the real-time processing and modeling demands (also termed as *massive on-line analysis* [3]) and for appropriate handling of vast amounts of highly complex data. Especially, in the case of *Big Data*[1] [50], an emerging scientific field mentioned in several objectives of the Horizon 2020 European Framework Programme[2], data stream classification is a fruitful methodology for handling and processing data in a block-wise or sample-wise manner. It integrates incremental concepts for establishing classification models on-line [10] and allows only a single-pass over the data [25]: unlike many optimization and other learning techniques, it uses no iterative procedure.

Another wide field of application for data stream classification is the (fully) autonomous on-line processing of data streams [27], which are often recorded at a high frequency, in order to quickly make or support important decisions. Classifiers are then required that can adapt to the changing dynamics of the processes or can even expand their knowledge on demand and on the fly, thus, they are usually termed *(incrementally) evolving classifiers* [26,32] — for instance, their ability to include new event types in visual inspection systems results in new classes the classifier should become aware of in order to be able to correctly return them [28]; another example is their ability to automatically integrate new operation modes or previously seen system states, moving the feature space into so far unexplored regions [7,22]. Static classifiers, which are trained once at the beginning with some batch recorded and stored data from past process cycles, but which remain unchanged in new on-line processes, usually show significant downtrends in case of dynamic environments [28,38].

1.1.1. Drift problematic

One particular problem in data stream classification is the possible emergence of *drifts* over time [45]. Drifts can occur, for instance, due to dynamic changes in system behavior, environmental conditions etc. over time, which renders the older relations and concepts contained in the system obsolete [45] — e.g. consider, for example, the prediction of the behavior of customers in an on-line shop to be predicted: changes in customer type or mood usually may cause shifts in their buying behavior (thus rendering the current prediction model obsolete), or consider that a government in a country may be constituted by different politicians over several years, then different classes in the government (e.g. the middle class) may enjoy a different behavior and interpretation in different periods of time. Drifts should be distinguished from cases in which *new* operation modes or system states arise, usually during on-line operation modes and process cycles. Samples from new modes or states should be included in the models with the same weight as previously seen samples. This is important to extend the models while keeping the relations in previously seen states untouched (so, they remain valid for future predictions), to avoid catastrophic forgetting [30]. Drifts, however, usually mean that the older learned relations are no longer valid, and thus an explicit, exquisite handling within incrementally time-varying (on-line) modeling cycles is required. From a data-driven point of view, drifts are usually characterized by a (gradual or abrupt) change in the underlying data distribution, which induces changes in the data's input or target concepts over time [48]. This makes appropriate drift handling the main concern of learning processes in on-line and dynamic systems [52] and in non-stationary environments in general [38].

First, it is important to detect severe drifts in order inform the operators or users working with the system, especially when the drifts are due to *unintended changes* in the process, such as failures or system shifts. In other cases, drifts may arise due to *intended changes*, for which increased flexibility in the classifier updates should compensate: the classifier should represent the system state after the drift while making previously learnt mappings/dependencies obsolete. Hence, drift detection is an essential prerequisite for proper handling of, and feasible reaction to, drifts.

In the literature, several approaches to *drift detection* have been proposed. One prominent approach can be found in [42], which relies on statistical process control on the model error and is applicable for both, regression and classification streams. For classification case, it employs instance-based classifiers, where the reference base is updated based on some clever reservoir sampling techniques, according to the most frequent class among the youngest samples. It tracks the mean and the standard deviation of classifier's error over the latest 100 samples, which are used in a one-sided z-test to check for a drift. For both, updating the reference base as well as updating the test statistic for checking drifts, it assumes that the labels of *all* samples are given. A similar strategy for statistical process control is used in [12] and [13]. An extension of the classical statistical process control has been recently introduced in [44] by proposing a linear four rates approach which modifies the classical hypothesis test (based on the binomial distribution of the model error) to a convex combination of old and new classifier's performance, which is measured four-fold (true positives, false positives, true negatives and false negatives) instead of a single model error/accuracy value, thus suffering much less from imbalanced class problems; however, it can be only applied to the binary classification case and to fully labeled streams. Other approaches compute statistics over sliding windows, e.g. by tracking the model error in the well-known approaches ADWIN [1] and FLORA [48]. ADWIN therefore checks for a clear change within a sliding window of past model errors by dividing it into all possible feasible

[1] https://de.wikipedia.org/wiki/Big_Data.
[2] http://ec.europa.eu/programmes/horizon2020/.