Classifiers for Concept-drifting Data Streams: Evaluating Things That Really Matter

Dariusz Brzezinski* and Jerzy Stefanowski

Institute of Computing Science, Poznan University of Technology
{dariusz.brzezinski,jerzy.stefanowski}@cs.put.poznan.pl

1 Problem statement

When evaluating the performance of a classifier for concept-drifting data streams, two factors are crucial: *prediction accuracy* and *the ability to adapt*.

The first factor could be analyzed by a simple error-rate, which can be calculated using a holdout test set, chunks of examples, or incrementally after each example [1]. More recently, Gama [2] proposed prequential accuracy as a means of evaluating data stream classifiers and enhancing drift detection methods. For imbalanced data streams, Bifet and Frank [3] proposed the use of the Kappa statistic with a sliding window to assess the classifier's predictive abilities. However, all of the aforementioned measures, when averaged over an entire stream, loose information about the classifier's reactions to drifts. For example, an algorithm which has very high accuracy in periods of concept stability, but drastically looses on accuracy when drifts occur can still be characterized by higher overall accuracy than an algorithm which has lower accuracy between drifts, but reacts very well to changes. If we want our algorithm to react quickly to, e.g., market changes, we should choose the second algorithm, but to do so we would have to analyze the entire graphical plot of the classifier's prequential accuracy, which cannot be easily automated and requires user interaction.

To evaluate the second factor, the ability to adapt, separate methods are needed. Some researchers evaluate the classifier's ability to adapt by comparing drift reaction times [4]. It is important to notice that in order to calculate reaction times, usually a human expert needs to determine moments when drifts start and stop. To automate the assessment of adaptability, Shaker and Hullermeier [5] proposed an approach, called recovery analysis, which uses synthetic datasets to calculate a classifier's reaction time. A different evaluation method, which uses artificially generated datasets was proposed by Zliobaite [6]. The author put forward three controlled permutation techniques that create datasets which can help inform about the robustness of a classifier to variations in changes. However, approaches such as [5], which calculate absolute or relative drift reaction times, require external knowledge about drifts in real streams or the use of synthetic datasets and, therefore can only be used offline. Furthermore, reaction times are always calculated separately from accuracy which makes choosing the best

* This work was partly supported by the Polish National Science Center under Grant No. DEC-2011/03/N/ST6/00360.

classifier a difficult task. Similarly, controlled permutations require generating artificial datasets and, thus, are limited to use offline, during model selection rather than on deployed models working online on real streams.

The number of discussed approaches shows that the evaluation of data stream classifiers is an important topic and there is a need to develop new methods specifically for non-stationary environments. However, all of the proposed evaluation measures concentrate on a single factor instead of combining information about accuracy and adaptability. Furthermore, many methods require the creation of artificial datasets or complex user interaction, which makes these methods difficult to use online, on a deployed data stream classifier. With these challenges in mind, we propose a new aggregated measure, which:

- combines information about accuracy and adaptability
- works online and does not require the creation of artificial datasets
- can be averaged over an entire dataset
- can be parametrized according to application-related costs, which define the importance of accuracy compared to drift reactions

2 Method

Methods for evaluating drift reaction times are calculated based on moments in the stream when a classifier starts to recover or fully recovers after a drift. It is worth noticing that, although the value being measured is usually time, it is the classifier's predictive ability that determines the moment of recovery. This shows that a single predictive measure like accuracy can be used to simultaneously evaluate the classifier's predictions and ability to adapt.

The main idea of the proposed approach is to differentiate the importance of predictions made directly after the appearance of a concept drift and predictions during periods of stability. This can be done by applying a user-defined weight to predictions during periods after a detected drift. The higher the weight, the more important predictions of drift concepts will be in the overall evaluation. This approach is partially inspired by cost-based learning for imbalanced datasets, where errors made on a minority class example cost more than errors made on examples from the majority class. In our approach, we treat examples directly after a detected drift as "minority" examples and assign a higher weight. With consecutive examples, the concept drift slowly becomes a "majority" class and the weight of examples converges back to a default value. As in cost-sensitive learning, we assume that the user can estimate the cost of not reacting to changes, i.e., the average weight of minority examples. By applying different weights to examples in times of drift and stability, we can combine information about accuracy and adaptability in a single user-controlled measure, which works online and can be averaged over the entire stream without the need of creating artificial datasets.

To implement this approach we need to determine the start and end of a new concept. To detect the start of a new concept we propose to use a drift detector, which analyzes the stream independently from the classifier(s) which will be evaluated. It is worth noting that, depending on the detector being used the evaluation method will be more appropriate for sudden or gradual changes. Concerning the end of a new concept, instead of detecting it we propose to identify the period in which a new concept is still new. In other words, we want to determine a time window of width d, in which a new concept gradually transforms into a "majority" concept. We propose to define d as the average time between previously detected drifts as this value gives the best estimate of how many examples are necessary to tag a "majority" concept. After determining periods when predictions should be weighted, we can define a weighting function.

Initially, when first examples arrive and no drift is detected each example has a weight of 1. When the drift detector signals a drift, the weight of consecutive examples is changed according to function $w(t)$. Since predictions directly after drift detection are the most important, we propose to use a non-linear function, which monotonously decreases to 1 after d examples. There are several functions that fulfill these requirements, but for the purposes of this paper we propose a logarithmic function defined as follows:

$$w(t) = max(-log_{e^{\frac{1}{w_{avg}-1}}} t + 1 + log_{e^{\frac{1}{w_{avg}-1}}} d, 1),$$

where t is the number examples after the detected drift, d is the average time between drifts, and w_{avg} is the average weight of examples during the d period.

One of our goals is to let the user adjust the proposed evaluation procedure to costs connected with slow reactions to drifts. In the proposed function, this is done by defining the base of the logarithmic function using a user-defined parameter w_{avg}. Assuming b is the base of our logarithmic function and w_{avg} defines the average weight of the d "drift" examples, we calculate the area under the curve of our logarithmic function and divide it by the number of examples, which gives us:

$$\frac{1}{lnb} + 1 = w_{avg},$$

which allows us to calculate:

$$b = e^{\frac{1}{w_{avg}-1}}$$

An example of the proposed weighting function, implemented using the Drift Detection Method [1], is illustrated in Fig. 1. The plot presents prequential accuracies of two classifiers, Dynamic Weighted Majority (DWM) and Online Bagging (Bag), on a dataset created using the Waveform generator [1]. The average prequential accuracies for these algorithm are $Acc_{DWM} = 87.37$ and $Acc_{Bag} = 88.59$, which could suggest that Bag is the better algorithm. However, if we know that the cost of not reacting to changes is three times higher than a short-term loss in accuracy ($w_{avg} = 3$), we get $Acc^{*}_{DWM} = 87.38$ and $Acc^{*}_{Bag} = 87.12$. Indeed, if we analyze the plot, we can see that DWM reacts better to drifts, therefore, if reactions to drifts are of more importance than overall accuracy, we should choose DWM instead of Bag. As this example shows, the proposed weighted accuracy does not alter the average accuracy of algorithms with consistent error rates, but those that deteriorate during concept drifts.

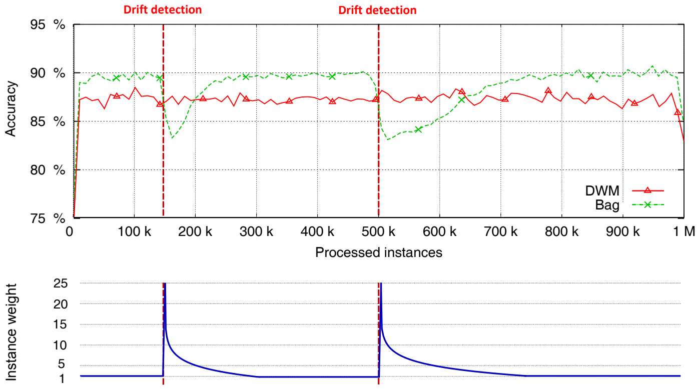

Fig. 1. Prequential accuracy and corresponding instance weight changes on an example data stream with two drifts and $w_{avg} = 3$

3 Discussion

Although the proposed approach seems to satisfy the goals stated in this paper, several aspects of evaluating classifiers for concept-drifting data streams still require discussion and examination. What are the possibilities of detecting not drift, but moments of recovery? Such information would help researchers “catch” full drifts (even on real streams), analyze them more thoroughly, and propose better classifier evaluation measures. Another problem lies in the automatic assessment of a captured reaction to drift. What functions are best for modeling the cost of slow reactions to changes? Finally, to what extent are we capable of automating the model selection process of classifiers working on real data streams? As the data gets bigger and faster, are we closer to fully self-monitoring systems or are we in need of more human intervention than before?

References

1. Gama, J.: Knowledge Discovery from Data Streams. Chapman and Hall (2010)
2. Gama, J., Sebastião, R., Rodrigues, P.P.: On evaluating stream learning algorithms. Machine Learning **90**(3) (2013) 317–346
3. Bifet, A., Frank, E.: Sentiment knowledge discovery in twitter streaming data. In Pfahringer, B., Holmes, G., Hoffmann, A.G., eds.: Discovery Science. Volume 6332 of Lecture Notes in Computer Science., Springer (2010) 1–15
4. Grossi, V., Turini, F.: Stream mining: a novel architecture for ensemble-based classification. Knowledge and Information Systems (2011) 1–35
5. Shaker, A., Hüllermeier, E.: Recovery analysis for adaptive learning from non-stationary data streams. In: Proc. CORES 2013. Volume 226 of Advances in Intelligent Systems and Computing. Springer (2013) 289–298
6. Zliobaite, I.: Controlled permutations for testing adaptive learning models. Knowledge and Information Systems (2013) 1–14