On the Need of Class Ratio Insensitive Drift Tests for Data Streams

André Maletzke ANDREGUSTAVO@USP.BR
Denis dos Reis DENISMR@USP.BR
Everton Cherman ECHERMAN@USP.BR
Gustavo Batista GBATISTA@ICMC.USP.BR
Universidade de São Paulo, São Carlos, Brazil

Abstract

Early approaches to detect concept drifts in data streams without actual class labels aim at minimizing external labeling costs. However, their functionality is dubious when presented with changes in the proportion of the classes over time, as such methods keep reporting concept drifts that would not damage the performance of the current classification model. In this paper, we present an approach that can detect changes in the distribution of the features that is insensitive to changes in the distribution of the classes. The method also provides an estimate of the current class ratio and use it to adapt the threshold of a classification model trained with a balanced data. We show that the classification performance achieved by such a modified classifier is greater than that of a classifier trained with the same class distribution as the current imbalanced data.

Keywords: Class imbalance, concept drift, quantification

1. Introduction

Class imbalance is an omnipresent problem. Researchers and practitioners have addressed the issue of learning from skewed data in a large number of application domains. In batch learning, an imbalanced dataset has one or more minority classes heavily outnumbered by the majority classes.

Such discrimination between minority and majority classes is not so clear for sequence data, as in online learning. In a data stream, the examples are not provided at once, but they arrive during the stream. Additionally, concept drifts can change the data distribution, requiring the online learner to update its model.

Concept drifts can occur in the features or the class attribute. Drifts in the class attribute may lead to imbalanced class ratios. Therefore, the relative frequency of the classes may depend on the span of the stream that is being processed. For this reason and differently from the batch setting, in online learning, a class can be both minority and majority within the same dataset.

To make this discussion more concrete, consider a real application of an insect surveillance sensor (Silva et al., 2015). The objective of such a sensor is to predict the species of a flying insect using data obtained from the wings movement. The class distribution may be affected by several factors, including the circadian rhythm of the insects. The circadian rhythm is a biological process that governs peaks of activity and periods of resting. For many insect species, these peaks occur at dawn and dusk, as shown in Figure 1.

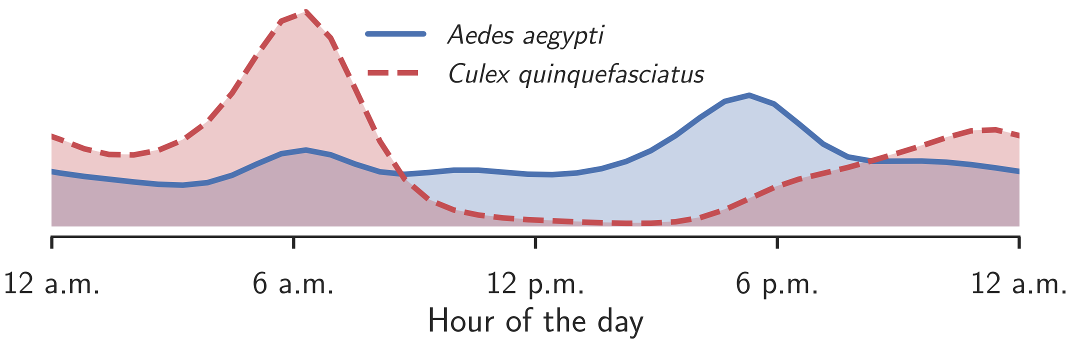

Figure 1: Histograms representing the circadian rhythm of *Aedes aegypti* and *Culex quinquefasciatus* mosquitoes. At dawn, *Culex* is the dominant class, but it is outnumbered by *Aedes* at dusk (dos Reis et al., 2018b).

A conventional approach to deal with concept drifts is to use drift tests. These tests can be divided into two categories: supervised and unsupervised. The difference lies in the need of class labels after deployment. The supervised tests assume that one examples' true class labels are available as soon as the classifier issues a prediction for this example. Such an assumption does not hold for a large number of applications. In the case of the insect sensor, the actual classes of the insects that passed in front of the sensor are expensive to obtain. In field conditions, this information could only be obtained by having an expert next to the sensor.

On the other hand, the unsupervised tests detect changes in the data distribution to flag concept drifts while not requiring class labels after deployment. Two examples are the nonparametric incremental drifts tests proposed by Kifer et al. (2004) and dos Reis et al. (2016). These approaches compare two samples: the first is the one used to train the classifier and the second is the latest stream data. The test indicates a concept drift if these samples come from different data distributions.

The main point of this paper is that the unsupervised drift tests available in the literature are sensitive to changes in the features as well as in the class attribute. It means that a significant difference in the class distribution will cause a drift flag. Moreover, such tests do not distinguish flags that were caused by changes in the features and the class attribute. However, we advocate that an ideal unsupervised detector should either be insensitive to drifts in the class attribute or, preferably, explicit which is the type of change that occurred.

We show a simple precursory experiment to make our point clearer. Figure 2 shows a comparison between three *hypothetical* Random Forest classifiers assessed with three performance measures: AUC, F1, and accuracy. The horizontal axis represents the class distribution in the test set for a binary classification problem. The red and green curves represent classifiers that were trained with a fixed balanced training data. The first o them, how-

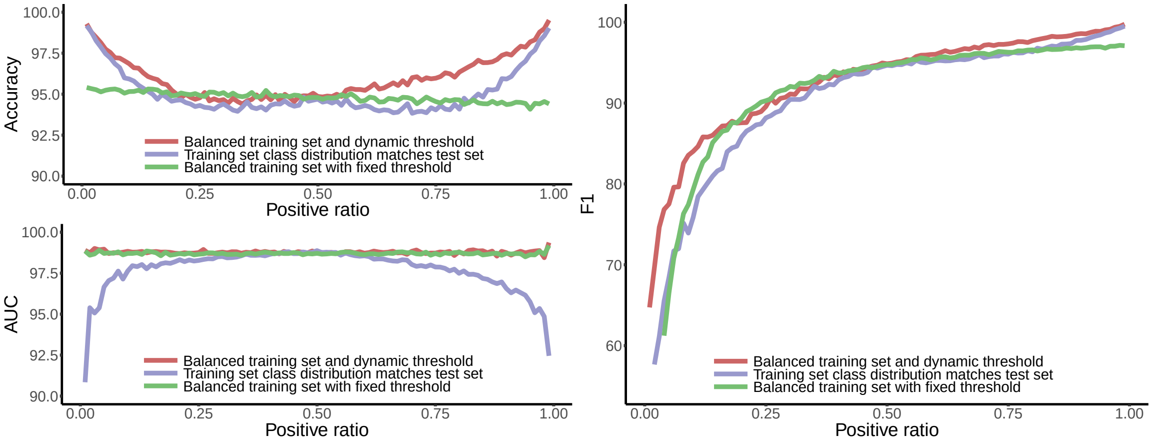

Figure 2: A comparison between two Random Forest classifiers on insect data. The classifier trained with a fixed and balanced class distribution outperforms the classifier trained with the same class distribution as the test set.

ever, has its threshold adjusted according to the expected class proportion at each test set[1], while the latter's threshold remains unchanged. The blue curve represents a classifier that is trained with data that share the same class distribution as each test set. Both classifiers trained with a fixed balanced class distribution performed better than the third classifier. This can be explained by the latter's need of extrapolating further with fewer examples for the minority class. Finally, among those classifiers trained upon a fixed balanced data, the one that has its threshold adjusted presented the best performance, which is expected.

From this experiment, we can advocate that if we have a classifier trained in a fairly balanced distribution, possibly with the aid of sampling techniques, there is no gain in retraining the classifier if the class distribution changes and the feature distribution remains the same, provided that the new class distribution is known. A better approach would be to keep the same classifier and adjust its decision threshold. Therefore, a concept drift test should preferably tell apart changes in the class distribution and changes in the distribution of the features.

In this paper, we adapt a recently proposed technique for learning with recurrent concepts (dos Reis et al., 2018b) as an unsupervised concept drift test. The proposed test is insensitive to changes in the class distribution and provides an estimate for the class distribution in the test set. In other words, the proposal is also a quantification method. We use such quantification estimate to adjust the decision threshold of the classifiers dynamically. We show that our test is more accurate than the state-of-the-art for recognizing concept drifts and, as a consequence, leads to more accurate classifiers.

This paper is organized as follows: Section 2 summarizes the related work on data streams, concept drift, and quantification methods. Section 3 details the proposed method

[1] The decision threshold is a scalar value that separates positive examples from negatives ones in a rank. Examples with a score greater than the threshold are labeled as positive. The remaining examples are marked as negative.

for detection of concept drifts. Section 4 explains the experimental setup and list the datasets that are used in our evaluation. Section 5 analyzes our experimental results and compare them against the state-of-the-art. Finally, Section 6 presents our conclusions and directions for future work.

2. Related Work

In this section, we summarize the relevant related work in concept drift detection and quantification in data streams.

2.1. Data Streams and Concept Drift Detection

A data stream is an ordered sequence of instances, $E = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_t, \ldots)$, where $\boldsymbol{e}_t \in \mathbb{R}^p$ is an example in a p-dimensional feature space. In supervised problems, each instance $\boldsymbol{e_t}$ has an associated label. Thus, a supervised data stream can be represented by an ordered sequence of pairs $E_s = ((\boldsymbol{e}_1, y_1), (\boldsymbol{e}_2, y_2), \ldots, (\boldsymbol{e}_t, y_t), \ldots)$, where $y_t \in C = \{c_1, c_2, \ldots, c_k\}$.

Data stream mining faces several challenges, such as high volume, velocity, and volatility (Nguyen et al., 2015). In several application domains, data distributions are nonstationary, leading to concept drifts. A concept drift is a significant change in the data distribution that may impact the classification performance of a system. Therefore, data streams require identifying and reacting to concept drifts as well as updating the models to incorporate those changes (Gama et al., 2004).

Supervised methods that deal with concept drifts assume the presence of actual labels throughout the entire stream. One example's true label is usually made available as soon as the model issues a prediction for this example. The true labels are used to retrain the model periodically or to detect drifts by monitoring the model's classification performance and retraining it when necessary (Kuncheva et al., 2008; Bifet and Gavaldà, 2009; Masud et al., 2011; Wu et al., 2012). However, such supervised strategy strongly depends on the constant availability of labels to rebuild the model or to detect the concept drift. This dependence can rarely be fulfilled in real-world applications.

Some recent papers have explored the use of statistical tests as unsupervised triggers for concept drift. Kifer et al. (2004) applied hypothesis tests to detect concept drifts without labels. The hypothesis test verifies whether the distributions of training and test sets are similar. Žliobaitė (2010) has employed a similar strategy over data streams. A hypothesis test compares if two consecutive sliding windows present the same behavior. Maletzke et al. (2017) and dos Reis et al. (2016) explore similar mechanisms, applying hypothesis tests to compare classification scores rather than feature values. The scores come from the training set and the most recent data stream sample. Such unsupervised detection approaches need current true labels only to retrain the model, similarly to an active learning approach, reducing the dependence on true label availability.

However, the statistical tests employed in these four papers are sensitive to changes in both features and the class attribute. More importantly, those changes are not distinguished. Therefore, if the only difference between two sets of data is the proportion of the classes, then those methods generically indicate a concept drift, regardless of the change in the class distribution not incurring the need of training a new classification model. In the previous section, we argued and showed empirical evidence that knowing how the class attribute changes suffice to adapt the threshold of a model and keep comparable or superior performance to a retrained model. Although classification performance can be affected by changes in the proportions of the classes, training a new classification model is not a better option than adapting the current model without requesting true labels. For this reason, detectors should not flag such changes as concept drifts but instead should provide by how much the class distribution changed.

dos Reis et al. (2018b) proposes an approach to learn in the presence of recurrent *concepts*. The main assumption is that the data stream approximately follows a small set of data distributions, called concepts. Therefore, the authors propose two methods to identify the most similar concept given the current data sample. One of these methods, the Single Most Relevant HDy (SMR-HDy), has a dual property: at the same time, it is a proxy for a distance between two data samples and a quantification method. We postpone the details of SMR-HDy to the next section. For now, let us clarify that, as a distance, SMR-HDy is not able to identify a new concept. Therefore, one of the contributions of this paper is to give a statistical interpretation of SMR-HDy distances. We want to understand how likely a given distance is and flag a concept drift for distances beyond a statistical threshold.

As a quantification method, SMR-HDy provides an estimate of the positive class distribution. Our approach uses this estimate to adjust the decision threshold of a classifier. Additionally, in the process of estimating the positive class ratio, SMR-HDy does an internal search for the minimum distance between all assessed class ratios. We hypothesize that, due to this search, such distance is invariant to the class distributions and is sensitive just to feature drifts.

2.2. Quantification

Quantification is a supervised task, recently formalized as a machine learning problem (Forman, 2005). This task shares similarities with classification. For instance, both consider the same representation for examples and a nominal output feature describing the class. However, quantification is not particularly interested in predicting the label for each instance. Instead, it is interested in the overall quantity of elements of a specific class. Consequently, a quantifier issues an output for a set of examples, rather than one for each instance. The output is an estimate for the class distribution, which consists of a sequence of real values that are the estimates of the proportions of the individual classes in the set.

A straightforward, but usually inaccurate quantifier, is the Classify and Count (CC) approach. It consists of directly using a classifier to quantify a test set, i.e., this method simply counts the number of examples predicted in each class. CC has a series of limitations. Although, it can provide optimal results for test sets with a class distribution that makes FPR equal to FNR[2], CC introduces a systematic error as the class distribution changes (González et al., 2017). This is a severe issue for quantification problems since quantification is only interesting when the class distribution is nonstationary. Otherwise, we could directly use the training set class distribution to predict the test set distribution.

The literature in quantification is quite vast, and we do not intend to do a comprehensive review in this paper. We point the interested reader to (González et al., 2017) for a thorough

[2] FPR and FNR are the false positive and negative ratios, respectively.

survey on the subject. We are particularly interested in a class of methods known as Mixture Models (MM) (Forman, 2006), and one representative of this class named HDy (González-Castro et al., 2013).

The HDy builds two normalized (unit area) histograms, H^+ and H^-, for the scores obtained by the classifier on two validation sets with exclusively positive and exclusively negative instances, respectively. When presented with an unlabeled test set, the algorithm builds a histogram H^T with the scores obtained by the same classifier. These histograms, H^+, H^-, and H^T, represent the distributions of the training set for each class and of the test set, respectively. Finally, HDy estimates the positive proportion rate as follows:

$$\mathrm{HDy}(H^+, H^-, H^T) = \operatorname*{arg\,min}_{0 \leq \alpha \leq 1} \left\{ \mathrm{HD}\left(\alpha H^+ + (1-\alpha) H^-, H^T\right) \right\}$$

where HD is the Hellinger Distance (Pollard, 2002), and each histogram, with B bins, is represented as a vector in the $\mathbb{R}^B$. Hellinger Distance is a function that estimates the dissimilarity between two probability distributions. Equation 1 defines Hellinger Distance for discrete distributions, as a histogram with B bins, and Figure 3 illustrates this process.

$$\mathrm{HD}(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{B} \left(\sqrt{p_i} - \sqrt{q_i}\right)^2} \tag{1}$$

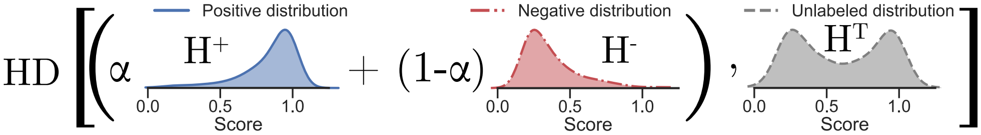

Figure 3: HDy uses the Hellinger Distance (HD) to measure the dissimilarity between a mixture of positive and negative score distributions and an unlabelled score distribution, where α is the proportion of the positive class. HDy searches for an α that minimizes the Hellinger Distance (dos Reis et al., 2018c).

The original intent of HDy was solely to estimate the proportion of the classes in a binary quantification problem. However, dos Reis et al. (2018b) observe that HDy provides, as a byproduct, the distance between the test sample and the most similar linear combination of positive and negative training samples. Therefore, such byproduct distance can be taken as a dissimilarity measure between the training and test samples, while being insensitive to changes in the proportions of the classes.

SMR-HDy (dos Reis et al., 2018b) assumes that a known set of concepts can approximately describe a data stream. Therefore, this approach stores the positive and negative training scores for all concepts and uses the byproduct distance of HDy to search for the most similar concept to the test sample.

The Hellinger Distance suits SMR-HDy since we can compare different distances and decide which one is smaller. In other words, we do not infer that a distance is either small or large, but instead, we verify which concept has the most similar data distribution to the current data by choosing the concept with the smallest HDy distance.

In this paper, we do not assume the existence of a set of known concepts and develop SMR-HDy as a concept drift test. However, this distance lacks statistical meaning, and we are unable to affirm that it is big enough to infer that a concept drift has occurred with high probability. In the next section, we provide a statistical meaning for this single distance and offer a way of estimating a threshold that, once surpassed, indicates the occurrence of concept drift with high probability.

3. Proposal

As mentioned, the SMR-HDy algorithm consists in identifying, among a set of known concepts, which one is the most similar to the current stream data. In this paper, we are not interested in calculating the similarity to known concepts. Instead, we want to identify when the latest data significantly differs from the sample used to train the current classifier.

To this task, we need to set a threshold to the HDy distances so that distances greater than this value indicate a concept drift. We empirically observed that distributions of HDy distances are approximately normally distributed around a mean when the histograms H^+, H^-, and H^T come from the same concept, even when the test set is imbalanced.

Figure 4 presents the results for ten benchmark binary datasets from UCI (Dheeru and Karra Taniskidou, 2017) and OpenML (Vanschoren et al., 2013) repositories. We uniformly and randomly split each dataset into two disjoint halves. The first half is reserved for training and validation, while the second for testing. To estimate the scores output by classification models, we used 10-fold cross-validation with the first half of the dataset. After obtaining the validation scores for H^+ and H^-, we induced classifiers in the first half.

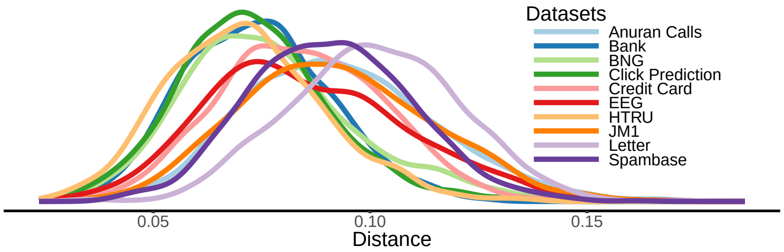

Figure 4: Distributions of the HDy distances without concept drifts.

To make the best use of our limited data, we restricted the test set to 1,000 examples so that there would be enough examples to vary the proportion of the positive class from 0 to 1, and also provide variability across samples in the extremes of this range. We changed the positive class proportion from 0% to 100%, with increments of 1% at each time. For each positive ratio, we created 10 random samples of 1,000 events. No event appears more than once in each sample, even though it can appear on multiple samples.

Considering that the HDy distances are normally distributed, we can define an adequate threshold, measured in standard deviations from the mean, to flag when two samples A and B are significantly different. We propose a simple criterion to create a threshold (θ) based on the distributions of the distances. If the distance reported by HDy differs more than two standard deviations from the mean, we consider with high probability that there is a concept drift. Therefore, we propose a threshold θ as a decision boundary for whether a drift is reported. According to our proposal, named **C**oncept **D**istance **T**hreshold (CDT), θ is defined as follows:

$$\theta = \mu + 2\sigma \tag{2}$$

Subsequently, the decision $\mathcal{D}$ of whether a drift between two samples A and B is reported is defined as follows:

$$\mathcal{D}(A, B) = \begin{cases} \text{drift}, & \text{HDy}(A, B) \geq \theta \\ \text{no drift}, & \text{otherwise} \end{cases}$$

We note that to the purpose of concept drift detection, only distances that significantly deviate from the mean on the upward side are reported, as they mean a greater measured distance between the data distributions than it is expected.

In the following section, we present our experimental setup and evaluation procedure to validate our proposal. We start the next section with a preliminary experiment to check the viability of our assumptions. Later on, we describe the four real datasets that were used to evaluate our method.

4. Experimental Evaluation

We preliminarily evaluate our design decision with real-world data gathered with an insect sensor. According to literature, there is a broad set of factors that influence the flying behavior of insects (and, therefore, cause concept drifts) such as temperature, humidity, air pressure, age, availability of water and food and so on (Maletzke et al., 2018). However, the temperature is a prominent factor. Figure 5 illustrates the influence of temperature on the wing-beat frequency, one of the features we extract from the signals.

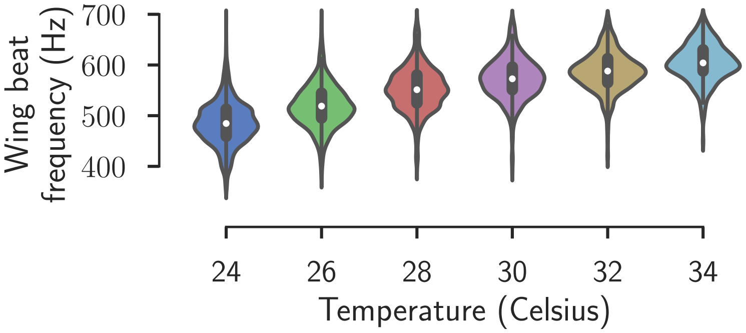

Figure 5: Influence of temperature on the wing-beat frequency of female *Aedes aegypti* mosquitoes (dos Reis et al., 2018b).

The insect data were gathered in the laboratory using insectaries with sensors attached. Each insectary maintains insects of a single species and sex, leading to labeled data. We can also control temperature, humidity, and other ambient conditions maintaining the insectaries in climatized chambers.

We created a scenario to test our threshold. Let the ranges of temperature be the concepts and consider as current (or known) concept the data collected with under lower temperature ($\sim 24°$C) and as new (or unknown) concept the data gathered under higher temperature ($\sim 34°$C). We split the data described by the known concept ($\sim 24°$C) into two disjoint halves, and we have repeated the same experimental design previously described (Section 3) to obtain the distribution of the distances. Afterward, we apply the classification model and get the scores for the positive and negative classes (H^+ and H^-). In the case of this experiment, the positive class is composed of events of female *Aedes aegypti* and the negative class by male *Aedes aegypti* mosquitoes.

Figure 6 shows the distributions of the distances when the test set comes from the known and unknown concepts. There is a clear separation between the distances from different concepts, and we explore this difference to create a trigger to flag when data of an unknown concept arrives. The dashed line represents an upper threshold defined according to Equation 2. We observe that the distances computed between samples of the known concept remain mostly below the limit set by $\theta = \mu + 2\sigma$.

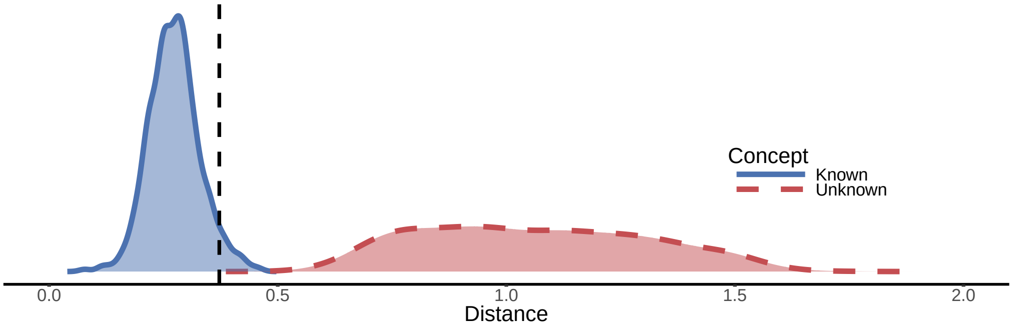

Figure 6: Distance distribution obtained from test sets composed by examples from two different concepts. The known concept is insect data obtained at $\sim 24°$C. The unknown concept was obtained at $\sim 34°$C. The vertical line represents two standard deviations from the known concept mean.

This preliminary evaluation indicates that our method is invariant to changes in the class distributions yet being sensitive to changes in the feature space. We further evaluate our proposal in the next experiments. For these, we use four real datasets:

Aedes-Culex A modified version of *Aedes aegypti-Culex quinquefasciatus* dataset, described by dos Reis et al. (2018a). This dataset contains $8,000$ events described by features that register the passage of female *Aedes aegypti* and female *Culex quinquefasciatus* mosquitoes in front of an optical sensor. Besides wing beat frequency, there are other 25 numerical features obtained from the signal. Two ranges of temperature define the concepts, and the classification task is to distinguish between the two species. $W_{tr} = 2,000$ and $W_{ts} = 1,000$;

Aedes-Sex A portion of the dataset *Aedes aegypti*-sex, described in (dos Reis et al., 2018a) was used. This version contains only two concepts and $8,000$ entries. Each event is described by the wing beat frequency, and other 25 numerical features obtained from the passage of female and male *Aedes Aegypti* mosquitoes in front of the optical sensor. $W_{tr} = 2,000$ and $W_{ts} = 1,000$;

Arabic-Digit A modified version of the Arabic-Digit dataset (Hammami and Bedda, 2010; Lichman, 2013), which contains $8,800$ entries described by a fixed number of MFCC values for the human speech of Arabic digits (among 10). The spoken digit defines the concept, and the task is to predict the sex of the speaker. $W_{tr} = 400$ and $W_{ts} = 200$;

Wine Wine Quality (Cortez et al., 2009; Lichman, 2013) contains $6,498$ entries described by 11 features. The classification task is to differentiate between two ranges of quality of wines. The concept is given by the type of the wine (red or white). $W_{tr} = 800$ and $W_{ts} = 300$.

Each dataset is divided into two parts of equal size. The examples of one of them intended exclusively for training a classification model and estimating a drift detection threshold according to CDT. The other part is intended to create test samples of size W_{ts}. Each part is composed of two well-known concepts. For each concept, from the training part, we select W_{tr} events with balanced class distribution to train the classification model used throughout our experimental evaluation. Every example in the same test sample belong to the same concept, and while an example does not appear more than once in the same sample, it can occur in different samples.

We select one of them to be the known concept (Tr_{kn}) and the other one to be the unknown concept (Tr_{un}). For each training set (Tr_{kn} and Tr_{un}), a classifier is induced (named δ_{kn} and δ_{un}, respectively). The rationale for using two training sets (Tr_{kn} and Tr_{un}) is straightforward: we want to evaluate at first the accuracy for detecting drifts. Based on the answer of our trigger, we want to select which classifier is used for the classification task (δ_{kn} or δ_{un}), and evaluate the impact of the decision of the trigger on classification accuracy.

Additionally, we use the Tr_{kn} to learn the threshold (θ) according to the procedure described in Section 3. The only difference from that procedure is that we only use 100 instances to simulate test samples with different class distributions. This change is due to the limited number of entries in Tr_{kn}. Figure 7 (a) shows a schematic representation of the threshold learning process, and how the classifiers for each concept were created (δ_{kn} or δ_{un}).

Given an estimated threshold and the classifier for each concept, we measure the performance of our proposal regarding concept identification accuracy. We compare our proposal with an unsupervised method that applies a hypothesis test over the scores output by the classifier to identify drifts. For that, we use the Kolmogorov-Smirnov test (KS). This is a non-parametric test to compare distributions and is a state-of-the-art approach to identify concept drifts (Kifer et al., 2004; Žliobaitė, 2010; dos Reis et al., 2016, 2018a). We compare, using the KS test, the distributions of the scores obtained through cross-validation on the training set with output scores of the classifier δ_{kn} for each test sample. Similarly to (dos Reis et al., 2016), we apply the KS test with a significance level of 0.001 to detect drifts.

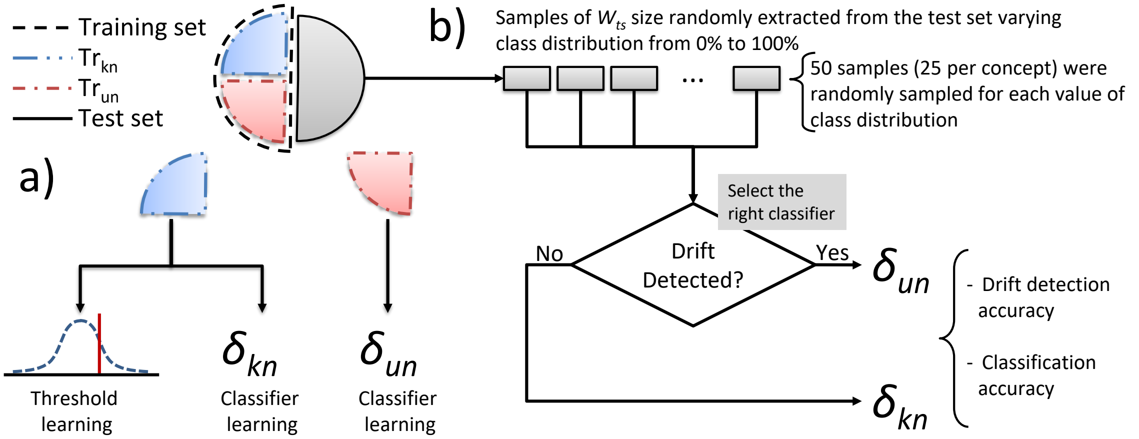
Training set
Tr$_{kn}$
Tr$_{un}$
Test set
a)
b)
Samples of W_{ts} size randomly extracted from the test set varying class distribution from 0% to 100%
50 samples (25 per concept) were randomly sampled for each value of class distribution
...
Select the right classifier
Drift Detected?
No
Yes
δ_{un}
δ_{kn}
- Drift detection accuracy
- Classification accuracy
Threshold learning
δ_{kn}
Classifier learning
δ_{un}
Classifier learning

Figure 7: Experimental setup used to evaluate our proposal.

To evaluate the capability of identifying when a concept drift occurs, we randomly extract test samples of size W_{ts} from the test set, according to Figure 7(b). For each sample, we vary the positive class proportion from 0% to 100%, with increments of 1%. For each positive ratio, we create 50 random samples of W_{ts} events (25 per concept). No event appears more than once in each sample, though it can occur on multiple samples.

Our proposal is suitable for binary classification and quantification problems. We assume that each processed chunk of data (referred to as sample) is entirely generated by only one concept. Although the algorithm receives a sample as input, this chunk can represent a sliding window in a data stream.

All classification models are Random Forests with 200 trees, using randomForest[3] package from R with default parameters.

In our experimental setup, a reliable concept drift trigger will only flag changes in the feature space and will ignore the simulated imbalanced class test sets.

5. Results

Table 1 presents the accuracy rates for concept drift detection obtained by CDT and KS algorithms, as well as the classification accuracies.

The CDT algorithm performed consistently better than the KS algorithm regarding drifts detection. We note that CDT achieved perfect concept identification in three out of four datasets. Additionally, when our proposal was used as a trigger to detect concept drifts, we obtained better classification accuracy, except for the Aedes-Culex dataset. This indicates that for Aedes-Culex dataset the concept has no relevance for the classification task. On the other hand, for the Aedes-Sex, Arabic-Digit, and Wine datasets, the drift detection has led to the best accuracy rates with the lowest standard deviations.

As expected, the KS approach has shown to be severely affected by imbalanced class distributions in the test sets. For all datasets, the drift identification accuracy of the KS

[3] https://cran.r-project.org/web/packages/randomForest/index.html

Table 1: Mean accuracy rates for concept identification and classification. Standard deviations are in parentheses.

Datasets	Drift detection accuracy		Classification accuracy	
	CDT	KS	CDT	KS
Aedes-Culex	0.785 (0.239)	0.601 (0.192)	0.722 (0.158)	0.758 (0.151)
Aedes-Sex	1.000 (0.000)	0.567 (0.167)	0.982 (0.005)	0.802 (0.142)
Arabic-Digit	1.000 (0.000)	0.643 (0.220)	0.981 (0.005)	0.959 (0.019)
Wine	1.000 (0.000)	0.679 (0.222)	0.840 (0.061)	0.669 (0.143)

was lower than our proposal. This evaluation highlights the high sensibility of the KS test to the class distribution, restricting the KS test applicability. This result corroborates with the main point argued at the beginning, regarding the sensibility to changes in the class distribution of the existing unsupervised drifts tests.

Figure 8 (right) shows the sensitivity of the KS to class distribution. The drift detection accuracy is quite lower for extreme values of class distribution. Conversely, as shown in Figure 8 (left), the CDT behavior seems to be insensitive regarding the class distribution, except to the Aedes-Culex dataset.

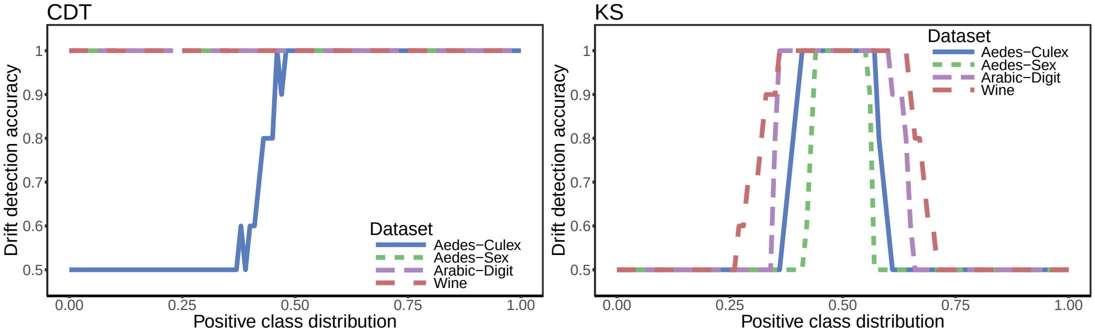

Figure 8: Impact of class distribution over the drift detection accuracy. CDT (on the left) and KS (on the right).

Furthermore, regarding classification accuracy, we note an important relationship with the drift detection performance. For all datasets, except Aedes-Culex, the classification accuracy rates were higher when the CDT was used as a trigger. Figure 9 illustrates the classification accuracy for each positive class proportion.

For the Aedes-Culex dataset, the KS approach presented better classification accuracy even when the correct concept was not detected. We observed that the classifier δ_{un} performed well even on samples from the known concept. Contrarily, when the classifier δ_{kn} was applied to the samples from the unknown concept, the classification rate declined. Additionally, for this dataset, KS test chose the classifier δ_{un} for all samples, regardless of true concept. This fact explains the classification rate for the Aedes-Culex dataset.

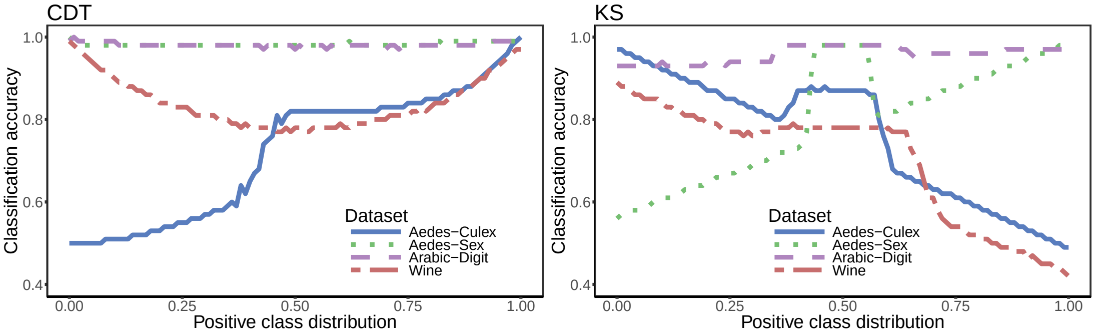

Figure 9: Impact of class distribution over the classification accuracy using CDT (on the left) and KS (on the right).

The CDT results corroborate with the initial assumption argued in Figure 2, except for the dataset Aedes-Culex, where our proposal has a limited drift detection accuracy. This can be explained by the lower drift detection accuracy of our proposal for this dataset, selecting the wrong classifier mainly for the lower positive class distributions.

Two distinct aspects impact the CDT performance. First, our proposal has a high drift detection accuracy. For this reason, we use the adequate classifier for most of the test sets (δ_{kn} or δ_{un}). Second, the CDT provides the class distribution of each test set as a byproduct, and therefore we can use this information to adjust the decision threshold of the classifiers dynamically.

Our results suggest that in many real-world applications where concept drifts are expected, our proposal is the first method to detect drift with class distribution insensitivity.

Finally, our proposal has some limitations that are worth mentioning. The first limitation is that we only address binary problems. This limitation is related to the underlying method used in our proposal, the HDy algorithm. A second limitation refers to the threshold learning phase, performed to estimate the value of θ for each dataset. This phase may be time-consuming, and it is a mandatory step for our method. Conversely, given the estimated threshold our proposal has a straightforward and fast application over data streams.

6. Conclusion

We present a method that accurately flags concept drift occurrences in the feature space. Our approach is the first unsupervised drift detection method to target at being insensible to class imbalance explicitly. Additionally, our proposal provides as a byproduct an estimate for the class distribution, which is used to adjust the decision threshold of the classifiers dynamically. We have empirically shown that our proposal outperformed the most used strategy to flag concept drifts in an unsupervised setup.

CDT is limited to binary classification. One possibility of extending our proposal to multi-class problems consists in aggregating multiple binary models, as is done in the one-vs-rest approach.

Acknowledgments

The authors would like to thank CAPES (grant PROEX-6909543/D), CNPq (grant 306631 /2016-4), FAPESP (grant 2016/04986-6). This material is based upon work supported by the United States Agency for International Development under Grant No AID-OAA-F-16-00072.

References

Albert Bifet and Ricard Gavaldà. Adaptive learning from evolving data streams. In *IDA*, pages 249–260. 2009.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Denis dos Reis, André Maletzke, and Gustavo Batista. Unsupervised context switch for classification tasks on data streams with recurrent concepts. In *ACM/SIGAPP*, volume 33, Pau, France, 2018a. ACM.

Denis dos Reis, André Maletzke, Diego F. Silva, and Gustavo E. A. P. A. Batista. Classifying and counting with recurrent contexts. In *ACM SIGKDD*, KDD '18, pages 1983–1992, 2018b. ISBN 978-1-4503-5552-0. doi: 10.1145/3219819.3220059.

Denis M. dos Reis, Peter Flach, Stan Matwin, and Gustavo E.A.P.A. Batista. Fast unsupervised online drift detection using incremental kolmogorov-smirnov test. In *ACM SIGKDD*, pages 1545–1554, San Francisco, 2016. ISBN 9781450342322. doi: 10.1145/2939672.2939836.

Denis M. dos Reis, André Gustavo Maletzke, Everton Cherman, and Gustavo E.A.P.A. Batista. One-class quantification. In *ECML-PKDD*, 2018c.

George Forman. Despite Inaccurate Classification. In *ECML*, pages 564–575, Porto, 2005.

George Forman. Quantifying trends accurately despite classifier error and class imbalance linear. In *ACM SIGKDD*, pages 20–23, Philadelphia, 2006. ISBN 1595933395.

João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. *Learning with Drift Detection*, pages 286–295. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-28645-5.

Pablo González, Alberto Castaño, Nitesh V Chawla, and Juan José Del Coz. A review on quantification learning. *ACM Computing Surveys (CSUR)*, 50(5):74, 2017.

Víctor González-Castro, Roco Alaiz-Rodrguez, and Enrique Alegre. Class distribution estimation based on the hellinger distance. *Information Sciences*, 218:146 – 164, 2013. ISSN 0020-0255. doi: http://dx.doi.org/10.1016/j.ins.2012.05.028.

Nacereddine Hammami and Mouldi Bedda. Improved tree model for arabic speech recognition. In *ICCSIT*, volume 5, pages 521–526, 2010.

Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 180–191. VLDB Endowment, 2004.

Ludmila I Kuncheva et al. Nearest neighbour classifiers for streaming data with delayed labelling. In *ICDM*, pages 869–874. IEEE, 2008.

M. Lichman. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2013.

Andre Maletzke, Denis Reis, and Gustavo Batista. Quantification in data streams: Initial results. In *BRACIS*, pages 43–48, Uberlndia, 2017. doi: 10.1109/BRACIS.2017.74.

André Maletzke, Claudia Milaré, Barbara Nadai, Jesse Saroli, Shailendra Singh, Juliano Corbi, Agenor Mafra-Neto, Eamonn Keogh, and Gustavo Batista. Automatic insect recognition with optical sensors with variability of temperature and humidity. In *AMCA 84th Annual Meeting*, 2018.

Mohammad M Masud, Jing Gao, Latifur Khan, Jiawei Han, and Bhavani Thuraisingham. Classification and novel class detection in concept-drifting data streams under time constraints. *TKDE*, 23(6):859–874, 2011.

Hai-Long Nguyen, Yew-Kwong Woon, and Wee-Keong Ng. A survey on data stream clustering and classification. *KAIS*, 45:535–569, 2015. ISSN 0219-1377. doi: 10.1007/s10115-014-0808-1.

David Pollard. *A user's guide to measure theoretic probability*, volume 8. Cambridge University Press, 2002.

Diego F. Silva, Vinícius M. A. Souza, Daniel Ellis, Eamonn Keogh, and Gustavo E.A.P.A. Batista. Exploring low cost laser sensors to identify flying insect species. *J Intell Robot Syst*, 80(1):313–330, 2015. ISSN 1573-0409. doi: 10.1007/s10846-014-0168-9.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198.

Xindong Wu, Peipei Li, and Xuegang Hu. Learning from concept drifting data streams with unlabeled data. *Neurocomputing*, 92:145–155, 2012.

Indre Žliobaitė. Learning under concept drift: an overview. *Computing Research Repository*, abs/1010.4784, 2010. URL http://arxiv.org/abs/1010.4784.

Indre Žliobaitė. Change with delayed labeling: when is it detectable? In *ICDMW*, pages 843–850. IEEE, 2010.