Mach Learn (2015) 98:455–482
DOI 10.1007/s10994-014-5441-4

Evaluation methods and decision theory for classification of streaming data with temporal dependence

Indrė Žliobaitė · Albert Bifet · Jesse Read · Bernhard Pfahringer · Geoff Holmes

Received: 6 October 2013 / Accepted: 29 March 2014 / Published online: 26 April 2014

Abstract Predictive modeling on data streams plays an important role in modern data analysis, where data arrives continuously and needs to be mined in real time. In the stream setting the data distribution is often evolving over time, and models that update themselves during operation are becoming the state-of-the-art. This paper formalizes a learning and evaluation scheme of such predictive models. We theoretically analyze evaluation of classifiers on streaming data with temporal dependence. Our findings suggest that the commonly accepted data stream classification measures, such as classification accuracy and Kappa statistic, fail to diagnose cases of poor performance when temporal dependence is present, therefore they should not be used as sole performance indicators. Moreover, classification accuracy can be misleading if used as a proxy for evaluating change detectors with datasets that have temporal dependence. We formulate the decision theory for streaming data classification with temporal dependence and develop a new evaluation methodology for data stream classification that takes temporal dependence into account. We propose a combined measure for classification performance, that takes into account temporal dependence, and

Editor: Joao Gama.

I. Žliobaitė (✉) · J. Read
Department of Information and Computer Science, Aalto University and Helsinki Institute for Information Technology (HIIT), Espoo, Finland
e-mail: indre.zliobaite@aalto.fi

J. Read
e-mail: jesse.read@aalto.fi

A. Bifet
Huawei Noah's Ark Research Lab, Hong Kong, China
e-mail: bifet.albert@huawei.com

B. Pfahringer · G. Holmes
University of Waikato, Hamilton, New Zealand
e-mail: bernhard@waikato.ac.nz

G. Holmes
e-mail: geoff@waikato.ac.nz

we recommend using it as the main performance measure in classification of streaming data.

Keywords Data streams · Evaluation · Temporal dependence · Classification

1 Introduction

Data recording capabilities in our urban and natural environment is rapidly increasing. Sensors, cameras, counters are installed in many places, our mobile devices are equipped with sensors and the range of things we can record is increasing. All these devices generate data that arrives in a stream and needs to be analyzed in real time. Predictive models, built on such data, have wide application in monitoring of the environment (e.g. detecting traffic jams), urban planning (e.g. on demand bus transport), personal assistance and recommendation (e.g. smart homes), industrial production (e.g. quality control), and many other applications.

Predictive models on data streams differ from standard predictive modeling in several key aspects (Hulten et al. 2001; Gaber et al. 2005). First, instead of a fixed size data sample we have a continuous flow of data, hence, models need to be built and updated on the fly, they need to fit into limited memory and use fixed processing time. Second, the data distribution is expected to evolve over time, hence, models need to be equipped with diagnostic mechanisms and be able to update themselves over time in order to maintain accuracy.

Although there is much research in the data stream literature on detecting concept drift and adapting to it over time (Gama et al. 2004; Kolter and Maloof 2007; Ross et al. 2012), most work on stream classification assumes that data is distributed not identically, but still *independently*. Let X_t be an observation at time t and y_t its true label. *Identical* distribution means that the joint probability of an observation and its label is the same at any time $P(X_{t_1}, y_{t_1}) = P(X_{t_2}, y_{t_2})$, when $t_1 \neq t_2$. *Independent* distribution means that the probability of a label does not depend on what was observed earlier $P(y_t) = P(y_t|y_{t-1})$.

Temporal dependence (also known as serial correlation or autocorrelation) is often encountered in other fields, such as control theory, statistical analysis, or traditional time series analysis (Box et al. 1994), where regression modeling is the main task, and the previous values of the signal present the main (or often the only) source of predictive information. In the data streams setting typically multi-dimensional input variables, not the past values of the target variable, contain the main predictive information. Machine learning considers two classification scenarios in similar settings (Dietterich 2002), which are also different from the data streams scenario. Firstly, in sequence classification, the task is to predict a single label that applies to an entire input sequence, while in data streams the task is to predict a label for each observation. Secondly, in sequential supervised learning the entire sequence is available before making any predictions about the labels, whereas in data streams observations come in portions, predictions need to be made immediately, the entire sequence is never available and predictive models are continuously updated. Table 1 summarizes the main differences in the settings in the related problem areas.

Temporal dependence is very common in data streams coming from data recording devices, such as video surveillance, environment sensors, mobile sensors (accelerometers), consumption data (e.g. electricity, food sales). Overall, any smart sensing applications are very likely to produce temporally dependent data streams. On the other hand, in behavioral domains where each observation is a person coming from different locations and contexts (e.g. web site visitors, web searches) the problem of temporal dependence is not that prominent. The majority of data streams classification research (see e.g. Gama et al. 2014); however,

Table 1 Different settings considering temporal dependence

Problem	Operation mode	Prediction task	Instances and labels	Main predictive information
Sequence classification	Offline	Classification	Per sequence	Other than target
Sequential supervised learning	Offline	Classification	Per observation	Same as target
Time series forecasting	Real time	Regression	Per observation	Same as target
Classification of streaming data	Real time	Classification	Per observation	Other than target

has advanced with the assumption (often implicit) that data does not contain temporal dependence.

This paper focuses on evaluation peculiarities of streaming data classification with temporal dependence, accompanied with the decision theory, which explains, what optimization criteria should be used for building classifiers, why they need to be built this way, and which baselines should be used under such conditions. Except for our brief technical report (Zliobaite 2013) and a conference publication (Bifet et al. 2013), we are not aware of any work in data stream classification analyzing the effects *temporal dependence* can have on model evaluation. This paper extends the above mentioned work. A recent publication (Gama et al. 2013) presented a study on evaluating stream algorithms focusing on error estimation and comparing two alternative classifiers. The aspect of temporal dependence was mentioned, but the effects of temporal dependence have not been analyzed and not included in the evaluation, the proposed evaluation implicitly assumes independent distributions.

This paper presents two main contributions: a decision theory for predictive modeling on data streams with temporal dependence, and a methodology for evaluating classifiers on data streams with temporal dependence. We argue, that firstly, the optimization criteria needs to be correct, and secondly, the evaluation and comparison needs to be complete. The paper presents the methodology for achieving that. New contributions with respect to our conference paper (Bifet et al. 2013), which is being extended, are as follows: decision theory and associated theoretical arguments, Temporal Correction classifier, large parts of the theoretical arguments on evaluation and all the material on drift detection with temporal dependence. In addition, the experimental evaluation has been largely revised and extended.

The paper is organized according to different issues related to temporal dependence in predictive modeling on data streams: classification decision making, evaluation of classifiers, drift detection, and availability of past labels. In Sect. 2 we formulate decision theory for data streams with temporal dependence and in Sect. 3 we propose temporal classifiers. In Sect. 4 we discuss the issues of evaluation of classifiers with respect to baselines when temporal dependence is present in the data. Section 5 focuses on change detection under temporal dependence. Section 6 presents experimental analysis. In Sect. 7 we give recommendations for practitioners with respect to predictive modeling on data streams with temporal dependence. Section 8 concludes the study.

2 Decision theory for data streams with temporal dependence

2.1 Problem setting for data stream classification

A classification problem in the classical (non data stream setting) is: given a previously unseen r-dimensional observation vector X predict its class $y \in \{1, \ldots, k\}$ using a classification

model $y = h(X)$. The classification model h is constructed beforehand using a training dataset consisting of pairs of observations with known labels (X, y). It is assumed that the data is identically independently distributed (iid), which means that the joint probability $P(X, y)$ is the same for any observation and that each observation is sampled from this distribution independently from other observations.

Classification in the data stream setting has several key differences. Observations arrive in a sequence over time and this sequence is open-ended $X_1, X_2, \ldots, X_t, \ldots$. A prediction needs to be made for each observation X_i individually as soon as it arrives. The true label y_i arrives some time later after casting the prediction.

In the data stream setting there is no separate training set for constructing a model h beforehand, the model needs to be constructed and updated on the fly along with incoming data. Therefore, we have a sequence of models $h_1, \ldots, h_i \ldots$. A model is constructed incrementally taking into account all or a subset of the previous model, previous observations, and true labels $h_i = f(h_{i-1}, X_1, \ldots, X_{i-1}, y_1, \ldots, y_{i-1})$. Here f is the algorithm for model update.

Finally, in the data stream setting, data is expected to evolve over time, the data distribution is not identical at different times (not iid). Thus, the relationship between an observation and its label $y = h(X)$ may change over time. Therefore, the algorithm for model update f needs to include some forgetting mechanisms such that the model can adapt to the new data distribution over time.

In the last decade many such adaptive learning algorithms have been developed (see e.g. an overview Zliobaite 2010). The majority of existing works implicitly or explicitly assume that data in a stream is distributed not identically but still independently, i.e. observations X_i and X_{i+1} are sampled independently. This study offers an extension to data stream classification theory and practice when the independence assumption is relaxed.

2.2 Bayesian decision theory

Bayesian decision theory (Duda et al. 2001) suggests to classify an observation X such that the expected loss is minimized. Let $\lambda(i, j)$ be the loss function specifying the loss of predicting class i when the true class is j. Then the expected loss of predicting $\hat{y}$ is $L(\hat{y}) = \sum_{y=1}^{k} \lambda(\hat{y}, y) P(y|observation)$ where k is the number of classes. The optimal prediction is the one that minimizes L.

For simplicity in the following analysis we assume a zero-one loss function, where the costs of misclassification are $\lambda(\hat{y}, y) = 0$ if $\hat{y} = y$ and 1 otherwise. In that case the expected loss of predicting $\hat{y}$ reduces to $L(\hat{y}) = 1 - P(\hat{y}|observation)$.

The loss L is minimized if we predict the $\hat{y}$ that has the maximum posterior probability given the observation. Hence, if we observe an r-dimensional observation vector X, our best strategy is to predict

$$\hat{y} = \arg\max_i P(y = i|X). \tag{1}$$

This is how predictions are typically made in the classical classification setting as well as the streaming data classification scenario. The posterior probability $P(y|X)$ is estimated directly using discriminative classification models, such as a decision tree, SVM, logistic regression, or alternatively, the likelihood $P(X|y)$ is estimated using generative classification models, such as Naive Bayes or linear discriminant, and the posterior probability is computed using Bayes' theorem of inverse probability $P(y|X) = P(X|y)P(y)/P(X)$.

2.3 Decision theory for streams with temporal dependence

Temporal dependence in data streams means that observations are not independent from each other with respect to time of arrival.

Definition 1 First order temporal dependence is present when an observation is not independent from the previous observation, i.e. $P(y_t, y_{t-1}) \neq P(y_t)P(y_{t-1})$, where t is the time index, $y_t, y_{t-1} \in \{1, \ldots, k\}$, where k is the number of classes. An ℓ^{th} order temporal dependence is present if $P(y_t|y_{t-1}, \ldots, y_{t-\ell}) \neq P(y_t|y_{t-1}, \ldots, y_{t-1-\ell})$.

The temporal dependence for class i is positive if $P(y_t, y_{t-1}) > P(y_t)P(y_{t-1})$, in this case labels are likely to follow the same labels more often than the prior probability. A negative temporal dependence $P(y_t, y_{t-1}) < P(y_t)P(y_{t-1})$ makes the labels alternate. This study focuses on positive temporal dependence, which is often observed in real world data streams.

Suppose we need to make a prediction $\hat{y}_t$ at time t. By that time we will have already seen observations $X_1, \ldots, X_{t-1}$ and after casting the predictions we will have seen their labels $y_1, \ldots, y_{t-1}$, assuming immediate arrival of the true labels after casting predictions, which is a standard assumption in data stream classification. As we observe the observation vector X_t, our best strategy is to use all the available evidence and predict

$$\hat{y}_t = \arg\max_i P(y_t = i|X_t, y_{t-1}, \ldots, y_1). \tag{2}$$

If there is no temporal dependence in the data, then Eq. (2) reduces to Eq. (1), since then $P(y_t = i|X_t, y_{t-1}, \ldots, y_1) = \frac{P(y_t=i|X_t)P(y_{t-1})\cdots P(y_1)}{P(X_t)P(y_{t-1})\cdots P(y_1)} = P(y_t = i|X_t)$.

In practice the order of temporal dependence to be considered is often manually restricted to the ℓth order. Then the prediction becomes $\hat{y}_t = \arg\max_i P(y_t = i|X_t, y_{t-1}, \ldots, y_{t-\ell})$, where ℓ is the length of the history taken into account. This study primarily focuses on first order temporal dependence.

3 Classifiers for taking into account temporal dependence

We propose two approaches for incorporating temporal information into data stream classification. The first assumes a model behind temporal dependence and introduces a correction factor to the predictive model, which allows a probabilistic treatment. The second is based on data preprocessing and does not require any modification in the predictive models; hence, can be used with any off the shelf tools.

3.1 Temporal Correction classifier

One way to estimate $P(y_t = i|X_t, y_{t-1}, \ldots, y_{t-\ell})$ for all $i \in \{1, \ldots, k\}$, which is needed for classification decision making, is to assume a model on how temporal dependence happens and then use that model for estimating the posterior probabilities. Considering only first order temporal dependence we propose to model this dependence and estimate $P(y_t = i|X_t, y_{t-1})$ as illustrated in Fig. 1.

Figure 1(a) presents a standard data stream classification model, where y_t is assumed to be independent from y_{t-1}, hence $P(y_t = i|X_t, y_{t-1}) = P(y_t = i|X_t)$. The dependence is modeled from label y to observation vector X (not the other way around), since we suppose that data is generated as follows: first an object belonging to a certain class is sampled and then the observations about this object are made.

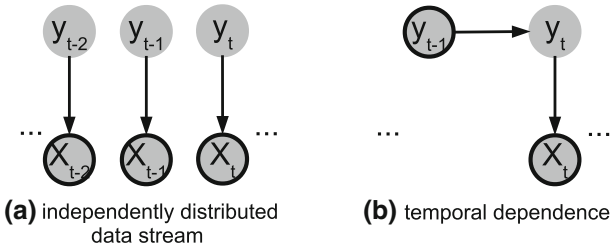

Fig. 1 Data stream classification models for: **a** data streams without temporal dependence, **b** with temporal dependence and known previous labels. *Black circles* denote the observed variables

Figure 1(b) presents our model for classification with temporal dependence when the previous label y_{t-1} is assumed to be known. This is a common assumption in data stream classification, since the previous label is required for error estimation and change detection, as well as model update, which are often executed at every time step. The classification decision is $\hat{y}_t = i$ for $i \in \{1, \ldots, k\}$, which gives the maximum posterior probability that can be expressed according to the proposed model as

$$P(y_t = i|X_t, y_{t-1}) = \frac{P(y_t = i, X_t, y_{t-1})}{P(X_t, y_{t-1})} = \frac{P(y_{t-1})P(y_t = i|y_{t-1})P(X_t|y_t = i)}{P(X_t)P(y_{t-1})}$$
$$= \frac{P(y_t = i|y_{t-1})}{P(y_t = i)} P(y_t = i|X_t). \quad (3)$$

Bayes' theorem is used to achieve the final step. Given the resulting expression, $P(y_t|X_t)$ can be estimated using an ordinary classifier that does not take into account temporal dependence and $\frac{P(y_t|y_{t-1})}{P(y_t)}$ is the term that corrects for temporal dependence, $P(y_t|y_{t-1})$ and $P(y_t)$ can be estimated incrementally from the streaming data.

3.2 Temporally Augmented classifier

The model approach is theoretically elegant, but limited in assuming first order temporal dependence and the directions of the dependencies between the observed vector X and the label y. We propose an alternative heuristic approach that can incorporate a higher order temporal dependence into a predictive model by augmenting the observation vector X_t with the previous class labels $y_{t-1}, \ldots, y_{t-\ell}$ and training a classifier on the augmented input vectors. The prediction for the observation X_t becomes a function of the original input attributes and the recent class labels

$$\hat{y}_t = h_t(X_t, y_{t-1}, \ldots, y_{t-\ell}). \quad (4)$$

The larger ℓ, the longer temporal dependence is considered. h_t is a trained classification model that can output an estimate of the posterior probability, index t indicates that the classifier can be updated over time. Any data stream classifier can be used as a base classifier with this strategy. Depending on the base classifier used, the Temporally Augmented classifier can take into account dependences between input features, the dependence between the input features and the past labels, as well as the dependence between past labels at different times. This approach is not new, it is common in time series forecasting, particularly using neural networks (e.g. Rodrigues and Gama 2009), where the main predictive information is given by the past values of the target variable.

By taking into account the dependence of the previous class label, the process can be seen as a discrete-time Markov chain, where the prediction for y_t is dependent on step y_{t-1}. If we take into account other labels, y_{t-2}, y_{t-3}, this becomes a second-order, third-order (and so on) Markov chain. The Temporally Augmented classifier is therefore conceptually related to the *filtering* task of Hidden Markov Models (Rabiner 1990; Dietterich 2002) (indeed a strong similarity is seen with Fig. 1b), where the probability of a classification is estimated, given historical and current evidence. In scenarios where the predictive variables y_t are continuous, then instead there is an analogous conceptual connection to the Kalman filter (Kalman 1960) (where it is possible to assume linear and normally-distributed variables) and particle filter (for other distributions). The typical *prediction* task of these models is that of *time series forecasting*, see Table 1, i.e., predicting $p(y_t|x_{t-1}, y_{t-1})$ where x_t is not yet available.

4 Baselines for performance evaluation over streaming data

In this section we discuss evaluation of classification performance with respect to baselines. A baseline classifier is a classifier that does not use any information about observations X, only the class labels y. When designing an intelligent classifier it is important to establish performance baselines for the minimum meaningful performance, otherwise a lot of design and computational effort may be wasted. It may happen that we compare several intelligent classifiers, find one to be significantly better than the others, but if all are worse than naive baselines, then none is good. In this section we discuss baselines for evaluating classification performance over streaming data when temporal dependence is present.

This section does not cover comparing the performance of several intelligent classifiers, which has been the subject of several recent studies. The interested reader is referred to Demsar (2006) and Gama et al. (2013) for guidelines.

4.1 Baseline classifiers

The following baseline classifiers can be established using different information about the probabilities of class labels:

1. classification with no information about data distribution;
2. classification with prior probabilities of the classes (Majority Class classifier);
3. classification with transition probabilities of the classes (Persistent classifier).

If we do not have any information about the data at all and we know that the task is to classify an observation into one of k classes, our best strategy is to assign a label at random $\hat{y} \in \{1, \ldots, k\}$, $P(\hat{y} = i) = \frac{1}{k}$. The accuracy of such classification would be

$$p_0 = \sum_{i=1}^{k} P(y = i) P(\hat{y} = i) = \frac{1}{k} \sum_{i=1}^{k} P(y = i) = \frac{1}{k}. \tag{5}$$

Most often we have at least some sample observations before putting a classifier in operation, and we can estimate at least the prior probabilities. If we have no other information about a given observation at all, our best strategy is to predict the class that has the maximum prior probability $\hat{y} = \arg\max_i P(y = i)$, where i is a class.

Definition 2 The Majority Class classifier is a classifier that predicts $\hat{y}_t = \arg\max_i P(y = i)$ for any observation X_t.

Let M denote the majority class. Then the accuracy of the Majority Class classifier is equal to the prior probability of class M

$$p_{\text{maj}} = P(y = M)1 + \sum_{i \neq M} P(y = i)0 = P(y = M). \tag{6}$$

If a temporal dependence is expected, we need a baseline that takes into account the temporal information. If no information about the observation is available, our best strategy is to predict $\hat{y}_t = \arg\max_i P(y_t = i|y_{t-1})$.

Definition 3 The Persistent classifier is a classifier that predicts the same label as previously observed, i.e. $\hat{y}_t = y_{t-1}$, for any observation X_t.

The accuracy of the Persistent classifier is equal to the prior weighted probability of observing the same class in two consecutive observations

$$p_{\text{per}} = P(y_t = y_{t-1}) = \sum_{i=1}^{k} P(y_t = i)P(y_t = i|y_{t-1} = i). \tag{7}$$

In the case when there is no temporal dependence in the data, then $P(y_t = i|y_{t-1} = i) = P(y_t = i)$ and the accuracy becomes

$$p_{\text{per}} = \sum_{i=1}^{k} P(y_t = i)^2. \tag{8}$$

The Persistent classifier is based on the same principle that is often used as a baseline in time series forecasting: the next forecast value is equal to the last observed value. In autoregressive time series it can be expressed as an ARMA(1,0) model (Box et al. 1994).

Observe that the three baseline accuracies p_0, p_{maj} and p_{per} take as input only the true labels of the underlying dataset. There is one more baseline that is sometimes considered (e.g. in the Kappa statistic Cohen 1960), that takes as input the true labels of the underlying dataset as well as the prior probabilities of the predictions produced by an intelligent classifier, that is being assessed (a reference classifier). This baseline is a random permutation of the predictions of an intelligent classifier.

Definition 4 The Random classifier is a classifier that predicts a label at random from the probability distribution of predictions of a reference classifier h, i.e. $P(\hat{y} = i) = P_h(\hat{y} = i)$ for any observation X_t.

The accuracy of the Random classifier is

$$p_{\text{ran}} = \sum_{i=1}^{k} P(y = i)P_h(\hat{y} = i). \tag{9}$$

While p_0, p_{maj} and p_{per} depend only on the dataset, p_{ran} depends on the dataset and the predictions of the classifier under evaluation.

4.2 Theoretical analysis of baseline accuracies

In this section we analyze how the baseline accuracies compare theoretically to each other. For brevity we denote the prior probability $P(y_t = i)$ as $P(i)$ and the probability $P(y_t = i|y_{t-1} = i)$ of observing class i immediately after observing class i as $P(i|i)$. Let $M \in \{1, \ldots, k\}$ denote the majority class, such that $P(M) \geq 1/k$.

Proposition 5 *The accuracy of the Majority Class classifier is greater or equal to the accuracy of the Random classifier and is greater or equal to the accuracy of classification with no information, i.e.* $p_{maj} \geq p_{ran}$ *and* $p_{maj} \geq p_0$ *and these accuracies are the same whether there is temporal information in the data or not.*

The proof can be found in Appendix .

Hence, we do not need to compare to all the baselines p_0 and p_{per} if we find that a classifier under consideration outperforms p_{maj}.

Proposition 6 *If data is distributed independently, then the accuracy of the Majority Class classifier is greater or equal to the accuracy of the Persistent classifier, i.e.* $p_{maj} \geq p_{per}$. *The accuracies are equal when the prior probabilities of the classes are equal.*

Proof Since data is distributed independently, $P(i|i) = P(i)$ for all $i \in \{1, \ldots, k\}$. Then the accuracy of the Persistent classifier is $p_{\text{per}} = \sum_{i=1}^{k} P(i)^2$. The accuracy of the Majority Class classifier is $p_{\text{per}} = P(M)$. Substituting in the expressions for accuracies gives $p_{\text{maj}} - p_{\text{per}} = P(M) - \sum_{i=1}^{k} P(i)^2 = P(M) - P(M)^2 - \sum_{i \neq M} P(i)^2 = P(M) \sum_{i \neq M} P(i) - \sum_{i \neq M} P(i)^2 = \sum_{i \neq M} P(i)(P(M) - P(i)) \geq 0$. The inequality follows from the definition of the majority class, where $P(M) \geq \frac{1}{k}$, which implies that $P(i) \leq \frac{1}{k}$ for all $i \neq M$. The equality holds only if $P(A) = P(B)$. □

From Proposition 6 we can conclude that if data is distributed independently, then we can safely use the majority class classifier as a baseline.

Proposition 7 *If data has a temporal dependence such that* $\sum_{i=1}^{k} P(i, i) > P(M)$, *where* k *is the number of classes and* M *is the majority class, then the Persistent classifier is more accurate than the Majority Class classifier, i.e.* $p_{per} > p_{maj}$.

Proof For brevity denote $P(y_t = i)$ as $P(i_t)$. Then $p_{\text{per}} - p_{\text{maj}} = \sum_{i=1}^{k} P(i_t) P(i_t|i_{t-1}) - P(M) = \sum_{i=1}^{k} P(i_t) P(i_t, i_{t-1}) / P(i_{t-1}) - P(M_t) = \sum_{i=1}^{k} P(i) P(i_t, i_{t-1}) / P(i) - P(M) = \sum_{i=1}^{k} P(i_t, i_{t-1}) - P(M) > 0$. The inequality follows from the theorem condition. □

Table 2 summarizes the performance of the Majority Class and Persistent classifiers under different conditions. We conclude that none of the baselines alone can take all aspects of performance into account, therefore if nothing is known about the data we need to compare at least to p_{per} and p_{maj}.

4.3 Cohen's Kappa statistic

The Kappa statistic due to Cohen (1960) is a popular measure for benchmarking classification accuracy under class imbalance and is used in static classification scenarios as well as streaming data classification.

Table 2 Summary of theoretical performance of the baselines

		Data distribution	
		Independent	Temporal dependence
Classes	Balanced	$p_{\text{maj}} = p_{\text{per}}$	$p_{\text{per}} > p_{\text{maj}}$, if $\sum P(i_t, i_{t-1}) > P(M)$
	Imbalanced	$p_{\text{maj}} > p_{\text{per}}$	

The Kappa statistic κ is defined as

$$\kappa = \frac{p - p_{\text{ran}}}{1 - p_{\text{ran}}}, \tag{10}$$

where p is the accuracy of the classifier under consideration (reference classifier) and p_{ran} is the accuracy of the Random classifier, as defined in Eq. (9). If the predictions of the classifier are perfectly correct then $\kappa = 1$. If its predictions coincide with the correct ones as often as by chance, then $\kappa = 0$. Note that κ can theoretically be negative, this may happen if, for instance, an adversary on purpose tries to make errors.

An approximation to the standard error of the Kappa statistic is given by Cohen (1960)

$$\delta_\kappa = \sqrt{\frac{p(1-p)}{N(1-p_{\text{ran}})^2}}, \tag{11}$$

where N is the testing sample size. With large N the sampling distribution of κ will be approximately normal.

To test the obtained κ for significance, i.e. to test the null hypothesis that any correct prediction is due to chance (true $\kappa = 0$), we need to replace p with p_{ran} in Eq. (11)

$$\delta_0 = \sqrt{\frac{p_{\text{ran}}}{N(1-p_{\text{ran}})}}. \tag{12}$$

The significance test is then a Z-test with the test statistic $z = \kappa/\delta_0$. For example, at 5 % level of significance the null hypothesis is rejected if $z > 1.65$.

In practice the κ statistic is often used without significance testing, even relatively low values of κ can be significantly different from zero but, on the other hand, not of sufficient magnitude for an application at hand.

Next, let us analyze the Kappa statistic for the baseline Majority Class and Persistent classifiers. The Majority Class classifier predicts the class with maximum prior probability for any observation, hence $p = P(M)$. Since all the predictions are the same, there is nothing to permute, hence, $p_{\text{ran}} = p$. Thus, $\kappa = \frac{p-p}{1-p} = 0$. This indication ($\kappa = 0$) corresponds to our expectations, that the Majority Class classifier achieves its accuracy merely by chance rather than as a result of informative input features and a good model.

Next we analyze the values of the Kappa statistic for the Persistent classifier in two cases. First, suppose that there is no temporal dependence in the data, then $p = \sum_{i=1}^{k} P(i)^2$. Observe that in this case $P_h(i) = P(i)$, hence $p_{\text{ran}} = \sum_{i=1}^{k} P(i)^2 = p$, and therefore $\kappa = \frac{p-p}{1-p} = 0$.

If there is positive temporal dependence such that $\sum_{i=1}^{k} P(i_t, i_{t-1}) > P(M)$, then $p = \sum_{i=1}^{k} P(i_t)P(i_t|i_{t-1}) > p_{\text{maj}}$ (Proposition 7), and $p_{\text{maj}} \geq p_{\text{ran}}$ (Proposition 5). Therefore, by the property of transitivity $\kappa = \frac{p-p_{\text{ran}}}{1-p_{\text{ran}}} > \frac{p_{\text{ran}}-p_{\text{ran}}}{1-p_{\text{ran}}} = 0$. In this case we may observe a positive κ, while a reference classifier would be performing equally badly as a naive baseline Persistent classifier. This is not a desired behavior of the κ indicator, hence we need another indicator to capture the effects of temporal dependence.

4.4 New evaluation measure—Kappa-Temporal statistic

Considering the presence of temporal dependencies in data streams we propose a new measure the Kappa-Temporal statistic, defined as

$$\kappa_{\text{per}} = \frac{p - p_{\text{per}}}{1 - p_{\text{per}}}, \tag{13}$$

where p_{per} is the accuracy of the Persistent classifier.

The Kappa-Temporal statistic may take values from 1 down to $-\infty$. The interpretation is similar to that of κ. If the classifier is perfectly correct then $\kappa_{\text{per}} = 1$. If the classifier is achieving the same accuracy as the Persistent classifier, then $\kappa_{\text{per}} = 0$. Classifiers that outperform the Persistent classifier fall between 0 and 1. Sometimes it may happen that $\kappa_{\text{per}} < 0$, which means that the reference classifier is performing worse than the Persistent classifier baseline.

We want the measures to capture the performance with respect to the baseline classifiers. Let us analyze the values of the Kappa-Temporal statistic for the baseline Majority Class and Persistent classifiers.

The Kappa-Temporal statistic for the Persistent classifier would be $\kappa_{\text{per}} = \frac{p_{\text{per}} - p_{\text{per}}}{1 - p_{\text{per}}} = 0$, as desired, independently of whether there is temporal dependence in the data or not.

However, the Kappa-Temporal statistic for the Majority Class classifier would be different, depending on the data:

- if there is temporal dependence such that $\sum_{i=1}^{k} P(i, i) > P(M)$, then $p_{\text{per}} > p_{\text{maj}}$ and thus $\kappa_{\text{per}} < 0$ (Proposition 7);
- if there is no temporal dependence and the prior probabilities of the classes are equal, then $p_{\text{per}} = p_{\text{maj}}$ and thus $\kappa_{\text{per}} = 0$ (Proposition 6);
- if there is no temporal dependence and the prior probabilities of the classes are *not* equal, then $p_{\text{maj}} > p_{\text{per}}$ and thus $\kappa_{\text{per}} > 0$ (Proposition 6).

Therefore, using κ_{per} instead of κ, we will be able to detect misleading classifier performance for data that has temporal dependence. For highly imbalanced, but independently distributed data, the majority class classifier may beat the Persistent classifier and thus using κ_{per} will not be sufficient. Overall, κ_{per} and κ measures can be seen as orthogonal, since they measure different aspects of performance. Hence, for a thorough evaluation we recommend measuring and combining both.

4.5 The Combined measure

To evaluate both aspects of the performance we propose to combine the κ and κ_{per} by taking the geometric average as follows

$$\kappa^{+} = \sqrt{\max(0, \kappa) \max(0, \kappa_{\text{per}})}. \tag{14}$$

This way if any measure is zero or below zero, the combined measure will give zero. This is to avoid the situation, when both input measures are negative, but their product is positive, suggesting that the classifier performs well, while actually it performs very badly.

Alternatively, an arithmetic average of the two measures could be considered. However, in such a case a good performance in one criteria could fully compensate for a bad performance in other criteria. The desired performance is that a good classifier should perform well on both. Taking the geometric average punishes large differences in the two input measures, therefore it is more suitable.

4.6 Computing statistics over a data stream

For estimating κ and κ_{per} we need to compute the accuracy of the evaluated classifier p, and the reference accuracies p_{ran} and p_{per} over streaming data.

For estimating p_{ran} we need to store the prior probabilities of the predictions $P_h(i)$ for $i = 1, \ldots, k$, and the prior probabilities of the data $P(i)$ for $i = 1, \ldots, k$. For estimating

p_{per} we need to store the joint probabilities of the classes $P(i, i)$ for $i = 1, \ldots, k$, and the prior probabilities of the data $P(i)$ for $i = 1, \ldots, k$ (which are already stored for estimating p_{ran}). Hence, to calculate both statistics for a k class problem, we need to maintain only $3k + 2$ estimators, where $+2$ is for storing the accuracy of the classifier p and storing the previous true label.

In the data stream setting p can be estimated recursively following the prequential protocol (Gama et al. 2013). The same protocol can be used for calculating the reference statistics. The idea is at every time step to weigh the estimators using a decay factor $\alpha \in (0, 1)$. Large α implies fast forgetting. From our practical experience, for smooth estimation we recommend $\alpha = 0.01$ for binary classification tasks with more or less equal class priors. The larger the number of classes and the larger the expected class imbalance, the smaller α should be to ensure slower forgetting to produce smooth estimates. Algorithm 1 describes the estimation procedure.

Data: $\alpha \in (0, 1)$
Result: up-to-date estimate of $p = 0$, $P(i)$, $P_h(i)$, $P(i|i)$ for all i
initialization $p = 0$, $P(i)$, $P_h(i)$, $P(i|i) = \frac{1}{k}$ for all i, $y_{prev} = 1$;
for *every instance in the stream* **do**
  make a prediction $\hat{y}$, receive the true label y **if** $\hat{y} = y$ **then**
    $p \leftarrow \alpha + p(1 - \alpha)$
  end
  else
    $p \leftarrow p(1 - \alpha)$
  end
  for $i = 1 \rightarrow k$ **do**
    if i=y **then**
      $P(i) \leftarrow \alpha + P(i)(1 - \alpha)$;
      if $i = y_{prev}$ **then**
        $P(i|i) = \alpha + P(i|i)(1 - \alpha)$
      end
      else
        $P(i|i) = P(i|i)(1 - \alpha)$
      end
    end
    else
      $P(i) \leftarrow P(i)(1 - \alpha)$
    end
    if $i = \hat{y}$ **then**
      $P_h(i) \leftarrow \alpha + P_h(i)(1 - \alpha)$
    end
    else
      $P_h(i) \leftarrow P_h(i)(1 - \alpha)$
    end
  end
  $y_{prev} \leftarrow y$
end

Algorithm 1: Computing performance estimators.

5 Performance evaluation with change detection

Many classification algorithms for data streams use change (drift) detection tests (e.g. Gama et al. 2004; Baena-Garcia et al. 2006; Bifet and Gavalda 2007; Ross et al. 2012) that signal

when the data distribution changes and it is time to update the predictive model. In this section we discuss two important issues with change detection to be aware of when there is a temporal dependence in data.

First, we show that when there is a temporal dependence, it is very likely that the assumptions of current drift detection methods are violated, hence the statistical tests are applied incorrectly. In practice this means that at least a different confidence interval is applied than is assumed. In many cases drift can still be detected with reasonable accuracy, but the theoretical guarantees of the tests (if any) are not valid anymore. We give indications on how to correct the tests, leaving development of actual algorithmic solutions, out of the scope of this paper, to be taken as separate future work.

Second, in this section we show that independent of whether a change detection test is applied correctly or not, false alarms may actually increase classification accuracy. This happens if the temporal dependence is not taken into account directly by a classifier. We give theoretical arguments why this happens. The implication is that one should take this effect into consideration when evaluating drift detectors and overall classification performance.

5.1 Change detection with temporal dependence

Current drift detection methods including Gama et al. (2004), Baena-Garcia et al. (2006), Bifet and Gavalda (2007), Ross et al. (2012) make an assumption that input data is independent from each other, the goal is to detect a change in data distribution. Typically, drift detection methods operate on a binary stream of prediction errors. Next we demonstrate that if the observations have a temporal dependence, then the streaming error resulting from predicting the labels for those observations, also have a temporal dependence, unless certain specific conditions are satisfied by the predictor. We will consider a binary classification case, since it is enough to make the point while the math is simpler.

Proposition 8 *The errors produced by a classifier on a streaming data binary classification task are distributed independently in a stream if*

1. *the observations in a stream are distributed independently,* or
2. *the probabilities of an error given a class are equal (i.e.* $P(error|A) = P(error|B)$*, where* A*,* B *are the classes),* or
3. *the ratio between the error probabilities given the class is equal to the ratio between temporal dependencies of the classes (i.e.* $\frac{P(error|A)}{P(error|B)} = \frac{P(B_t|B_{t-1})-P(B_t)}{P(A_t|A_{t-1})-P(A_t)}$*, here* $P(B_t)$ *denotes the probability of class* B *at time* t*).*

The proof can be found in Appendix .

The implication of this proposition is that the statistical tests in current drift detection methods operate under conditions where their assumptions are violated. As a result, if the sample for performing a statistical test is small, false alarms may be raised. We have noticed, however, that in practice the impact of violation of this assumption is small, especially if 50 or more observations are used to perform the tests.

Change detection taking into account temporal dependence has been studied in statistics and related disciplines (see e.g. Knoth and Schmid 2004; Wieringa 1999; Lavielle 1999), which could be used as a starting point in developing change detection tests that take into account temporal dependence.

5.2 The effect of false alarms on classification accuracy

In this section we demonstrate that false alarms in drift detection may actually increase classification accuracy if there is a temporal dependence in the data. False alarms may happen due to various reasons, for instance, if alarm thresholds in the change detection tests are set too low.

If a drift alarm is raised, adaptive learning algorithms would typically replace an old classifier with a new one built on recent data (see e.g. Gama et al. 2004; Baena-Garcia et al. 2006). Suppose a data stream is stationary (there is no true drift). In such a case a false alarm is beneficial if the classifier trained on a smaller dataset is in expectation more accurate than a classifier trained on a larger training set. This can happen if data has a temporal dependence, as the following proposition illustrates.

Proposition 9 *If a data stream has a positive temporal dependence, for small training sample sizes the accuracy of the Majority Class classifier approaches the accuracy of the Persistent classifier, i.e.* $\lim_{n\to 1} p_{per} - p_{maj} = 0$*, where n is the training sample size for the Majority Class classifier.*

The proof can be found in Appendix .

6 Experimental analysis

This experimental evaluation has two major goals. The first goal is to demonstrate how current evaluation practices may be misleading and how they can be improved using the proposed measures. The second goal is to assess the performance of the proposed Temporally Augmented and Temporal Correction classifiers that take into account temporal dependence.

6.1 Datasets

We experiment with four real datasets often used in evaluating data stream classification.

The Electricity dataset (Elec2) (Harries 1999) is a popular benchmark for testing adaptive classifiers.

A binary classification task is to predict a direction of electricity price change with respect to the moving average of the last 24 h in the Australian New South Wales Electricity Market. Input variables are recorded every 30 min and cover the period from 1996 May to 1998 December (45,312 instances in total).

The data has 5 numeric input variables: day of the week, hour, NSW demand, Victoria demand and scheduled transfer. The data is subject to concept drift due to changing consumption habits, unexpected events and seasonality. For instance, during the recording period the electricity market was expanded to include adjacent areas, which allowed production surpluses from one region to be sold to another.

The Forest Covertype (Cover) (Bache and Lichman 2013) records cartographic variables in four wilderness areas located in the Roosevelt National Forest of northern Colorado, US. The classification task is to predict the type (out of seven types) of forest cover for a given observation (30×30 meters cell). This dataset has no time stamps, but it is originally presented in the geographical (spatial order), which can be considered as a stream; this dataset has been a popular benchmark for data stream classification. The dataset contains 581,012 instances with 54 attributes.

Table 3 Characteristics of stream classification datasets. $P(M)$ is the prior probability of the majority class, $P(T) = \sum_{i=1}^{k} P(i,i)$ characterizes temporal dependence as in Proposition 7

Dataset	# instances	# attributes	# classes	$P(M)$	$P(T)$
Elec2	45,312	5	2	0.58	0.85
Cover	581,012	54	7	0.49	0.95
KDD99	494,020	41	23	0.56	0.99
Ozone	2,536	72	2	0.97	0.95

The KDD cup intrusion detection dataset (KDD99) (Bache and Lichman 2013) records intrusions simulated in a military network environment. The task is to classify network traffic into one of 23 classes (normal or some kind of intrusion) described by 41 features. The dataset contains 494,020 instances. The problem of temporal dependence is particularly evident here. Inspecting the raw dataset confirms that there are time periods of intrusions rather than single instances of intrusions.

The Ozone dataset (Ozone) (Bache and Lichman 2013) records daily temperature, humidity and windspeed measurements (72 numeric variables), the goal is to predict high ozone days (binary classification task). The data is collected from the Houston, Galveston and Brazoria areas, US, and covers the period from 1998 to 2004 (2,536 instances in total). This dataset is very highly imbalanced, ozone days make up only 3 %, the rest are normal. There is no temporal dependence in this data, we include it for benchmarking in order to illustrate what happens when classes are highly imbalanced.

The characteristics of the datasets are summarized in Table 3. As we see from $P(T) > P(M)$, the first three datasets exhibit strong temporal dependence, while there is no temporal dependence in Ozone and this dataset has a high class imbalance.

6.2 Classifiers

Along with the baseline classifiers we test five intelligent classifiers, out of which the first two are non-adaptive, and the remaining three have adaptation mechanisms. Here non-adaptive classifiers learn from data streams incrementally with new incoming data, however, they do not have forgetting mechanisms. Our goal is to illustrate the issue of selecting proper baselines for evaluation, and potential improvement in accuracy of intelligent classifiers due to taking into consideration temporal dependence. The theoretical findings of this study and the proposed κ_{per} measure are not base classifier specific, hence we do not aim at exploring a wide range of classifiers. We select several representative data stream classifiers representing different models and adaptation mechanisms for experimental illustration, summarized in Table 4.

Table 4 Classifiers used in the experiments

	Adaptation	Base classifier	Number of models
Naive Bayes (NB)	Non-adaptive	Naive Bayes	One
Hoeffding tree (HT) Domingos and Hulten (2000)	Non-adaptive	Hoeffding tree	One
Drift detection (DDM) Gama et al. (2004)	Adaptive	Naive Bayes	One
Hoeffding adaptive tree (HAT) Bifet and Gavalda (2009)	Adaptive	Hoeffding tree	One
Leveraged bagging (LBAG) Bifet et al. (2010)	Adaptive	Hoeffding tree	Ensemble

6.3 Experimental protocol

We run all experiments using the MOA software framework (Bifet et al. 2010) that contains implementations of several state-of-the-art classifiers and evaluation methods and allows for easy reproducibility.

We use the test-then-train experimental protocol, which means that every instance is first used for testing and then as a training instance for updating the classifier. For estimation of parameters (e.g. the prior probabilities) we use exponential moving average. The higher the number of classes and the larger the class imbalance, the lower the estimation weight needs to be in order to achieve sufficiently smooth estimates. We used the following smoothing parameters, which were selected via visual inspection of the resulting prior probability estimates: for Elec data $\alpha = 0.001$, for Cover data $\alpha = 0.0001$, for KDD99 data $\alpha = 0.00001$.

6.4 Limitations of the current benchmarking practices: an illustrative example

The Electricity dataset has been perhaps the most popular benchmark for evaluating stream classifiers. It has been used in over 70 experiments on data stream classification (according to Google scholar as of December 2013), for instance, Gama et al. (2004), Kolter and Maloof (2007), Bifet et al. (2009), Ross et al. (2012). To illustrate the importance of using proper baselines, we retrospectively survey new stream classifiers reported in the literature that were tested on the Electricity dataset.

Table 5 presents the accuracy results reported in papers on this dataset (sorted according to the reported accuracy). Only 6 out of 18 reported accuracies outperformed a naive baseline Persistent classifier. This suggests that current evaluation and benchmarking practices need to be revised.

Table 5 Accuracies of adaptive classifiers on the Electricity dataset reported in the literature

Algorithm name	Accuracy (%)	Reference
DDM	89.6[a]	Gama et al. (2004)
Learn++.CDS	88.5	Ditzler and Polikar (2013)
KNN-SPRT	88.0	Ross et al. (2012)
GRI	88.0	Tomczak and Gonczarek (2013)
FISH3	86.2	Zliobaite (2011)
EDDM-IB1	85.7	Baena-Garcia et al. (2006)
Persistent classifier	*85.3*	
ASHT	84.8	Bifet et al. (2009)
bagADWIN	82.8	Bifet et al. (2009)
DWM-NB	80.8	Kolter and Maloof (2007)
Local detection	80.4	Gama and Castillo (2006)
Perceptron	79.1	Bifet et al. (2010)
AUE2	77.3	Brzezinski and Stefanowski (2014)
ADWIN	76.6	Bifet and Gavalda (2007)
EAE	76.6	Jackowski (2013)
Prop. method	76.1	Martinez-Rego et al. (2011)
Cont. λ-perc.	74.1	Pavlidis et al. (2011)
CALDS	72.5	Gomes et al. (2010)
TA-SVM	68.9	Grinblat et al. (2011)

[a] Tested on a subset

6.5 New evaluation measures and benchmarking practices

In this section we compare the accuracies of five intelligent classifiers (NB, DDM, HT, HAT, LBAG) with two established baselines Majority Class and Persistent classifiers, which give important indications about the performance of intelligent classifiers with respect to class imbalance and temporal dependence in the data, as argued in Sect. 4. The goal of this experiment is to analyze, how indicative the currently used Kappa statistic and the new evaluation measures Kappa-Temporal statistic and Combined measure are about classifier performance.

We experiment with two versions of the datasets: the original datasets that potentially contain temporal dependence and randomly shuffled datasets. Random shuffling makes datasets independently and identically distributed (iid) over time. Based on our theoretical considerations, we expect the currently used statistics to be indicative in the case of iid data, but not informative in the case of temporally dependent data (the original datasets).

Figure 2 plots the accuracies of the intelligent classifiers, the baselines and the three statistics of interest. We see that the Kappa statistic is high and indicates good performance for all datasets except Ozone, which is highly imbalanced and the Kappa statistic captures that the high accuracy in Ozone is mainly due to class imbalance, as expected. We see that the Kappa statistic fails to capture the fact that in the original datasets Elec2, Cover and KDD99, where temporal dependence is present, the naive baseline Persistent classifier performs better than any intelligent classifier. On the other hand, the proposed Kappa-Temporal statistic captures this aspect of the performance and shows negative indications in all these cases.

However, as demonstrated theoretically in Sect. 4, using the Kappa-Temporal statistic alone is not enough to benchmark the performance of data stream classifiers, since the Kappa-Temporal statistic does not capture the effects of class imbalance. Such a situation can be observed in Ozone shuffled, where there is no temporal dependence, while the class imbalance is very high such that the intelligent classifiers can hardly outperform the Majority Class classifier. We see that the Kappa-Temporal statistic gives good indications and the Kappa statistic signals poor performance, as expected.

We see that the Combined measure that combines both aspects of the performance (class imbalance and temporal dependence) gives a good summary indication about the performance in a single number.

Two conclusions can be made from this experiment. First, the proposed statistic captures the characteristics of classifier performance with respect to naive baselines as expected. Second, the state-of-the-art data stream classifiers fail and perform worse than the baselines on the data streams that contain temporal dependence, since they do not have mechanisms for taking into account temporal information even though this information is available in a stream (these data stream classifiers use previous labels for incrementally updating themselves). Hence, there is a need for our proposed approaches for taking into account temporal dependence, which we experimentally analyze next.

6.6 Performance of proposed approaches for taking into account temporal dependence

We compare the performance of the proposed Temporal Correction and Temporally Augmented classifiers with the performance of ordinary stream classifiers (that do not take temporal dependence into account) and with the Persistent classifier that takes into account *only* temporal dependence on the four original datasets. Recall that Temporal Correction and Temporally Augmented classifiers can be used as wrappers to any data stream classifier. We

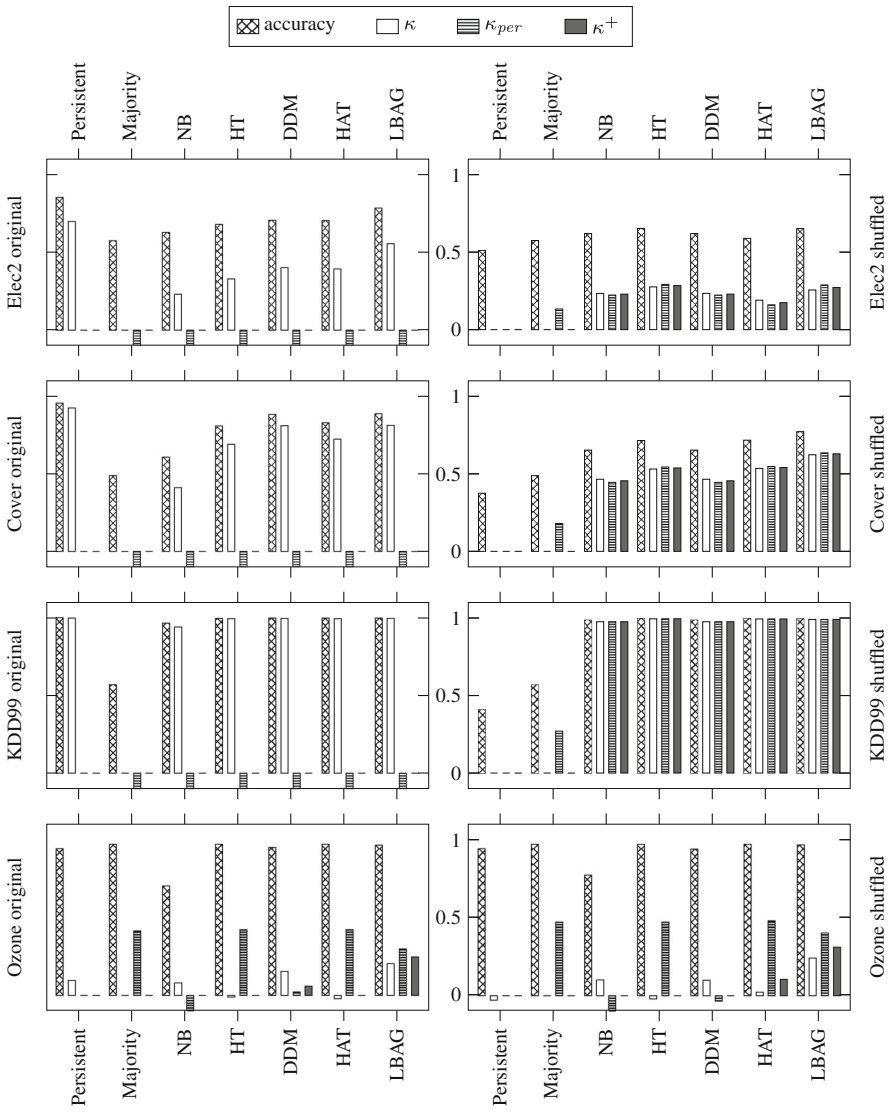

Fig. 2 Accuracy and performance statistics on the original and shuffled (iid) datasets

test the same five state-of-the-art data stream classifiers as in the previous experiments (NB, DDM, HT, HAT, LBAG).

Figure 3 presents the resulting accuracies. We see that both the Temporal Correction and the Temporally Augmented classifiers strongly outperform the ordinary classifiers on Elec2 and Cover datasets, and to some extent on the KDD99 dataset. These two classifiers are clearly benefiting from leveraging the temporal dependence in these datasets ($p(y_t|y_{t-1})$). The relatively smaller improvement on KDD99 dataset can be explained by the already-high accuracy of the ordinary classifiers. The performance on the Ozone dataset of the ordinary classifiers and the new classifiers is very similar, since the Ozone dataset does not contain much temporal dependence, but rather very high class imbalance. Thus, the absolute

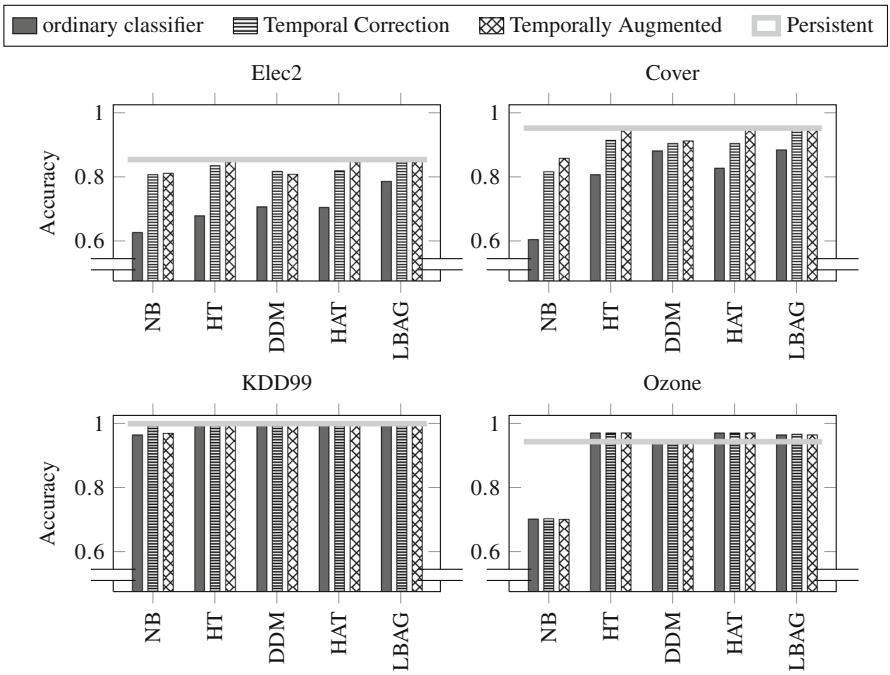

Fig. 3 Predictive performance of the classifiers taking into account temporal dependence

accuracy is high (estimating $p(y_t)$ is easy), but the lack of temporal dependence means that Temporal Correction and Temporally Augmented lose their advantage by modeling it (i.e., $p(y_t) \approx p(y_t|y_{t-1})$ in this case).

The Temporally Augmented classifier in most cases performs slightly better than Temporal Correction. This can be explained by the fact that Temporal Correction is modeled using certain independence assumptions (see Sect. 3.1), which may not always hold in reality.

A major problem, however, is that in the event that the proposed approaches offer an obvious improvement for the state-of-the-art data stream classifiers, the improvement is often not enough to significantly outperform the naive baseline Persistent. On the Ozone dataset the improvement over the baseline Persistent classifier is generally large. This is expected, since Ozone does not have strong temporal dependence, hence the Persistent classifier should not perform better than the Random classifier. However, on Elec2, Cover and KDD99 datasets that contain strong temporal dependence the performance of classifiers taking into account temporal dependence (Temporally Augmented classifier and Temporal Correction classifier) is close to or just slightly better than that of Persistent classifier. This is extremely problematic, it means that the effort of building sophisticated data stream classifiers in these situations may not be worth it. A simple Persistent classifier can do as well. On the other hand, this points out the current situation and offers an opportunity for researchers to improve over state-of-the-art classifiers.

6.7 Performance curves

In Fig. 3 we see that LBAG achieves the best performance in the ordinary data stream classification setting, when no temporal dependence is taken into account. Figure 4 plots

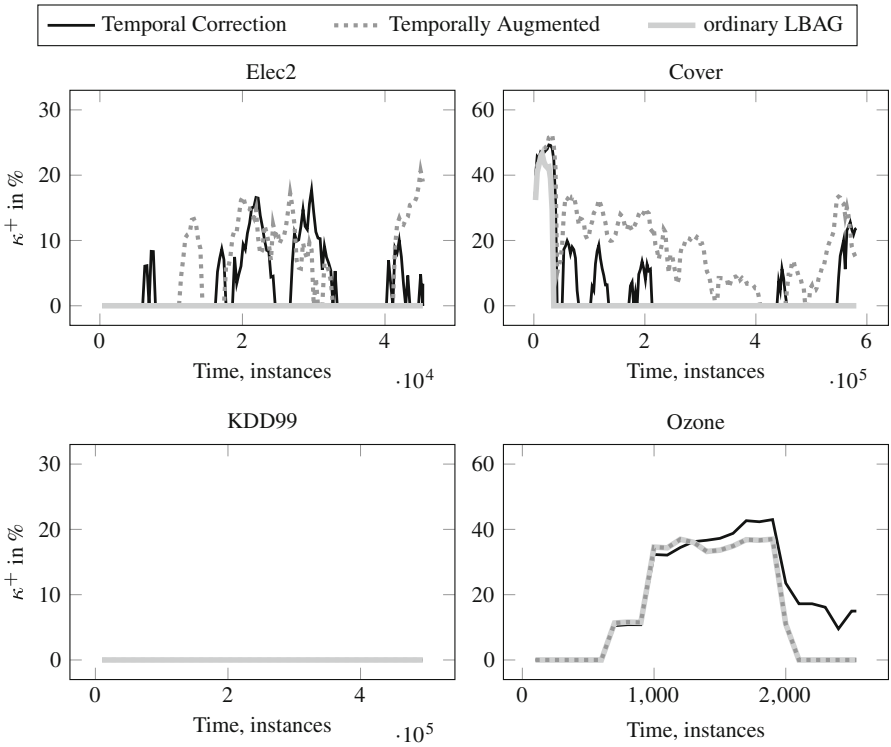

Fig. 4 Performance of LBAG (Combined measure κ^+) over time

the new Combined measure κ^+ of the performance over time on all four datasets. The plots present accuracies computed over a sliding window.

Several interesting observations can be made. In the Elec2, Cover and KDD99 datasets (that have strong temporal dependence) the ordinary LBAG performs poorly and almost never reaches any significant results as indicated by the Combined measure $\kappa^+ = 0$. On Elec2 and Cover both proposed approaches for taking into account temporal dependence substantially improve performance reaching positive κ^+, which means that the naive baseline Persistent classifier is outperformed, and the good accuracy is not due to class imbalance at random. The KDD99 dataset is a special case, where the accuracy of the baseline Persistent classifier is already so high (99.9 % accuracy) that it becomes nearly impossible to outperform.

Recall that the Ozone dataset is very highly imbalanced (97 %), but contains no positive temporal dependence, therefore we can expect the ordinary classifier LBAG to perform well, which happens to be the case as can be seen from the plot. We see that Temporally Augmented classifier has no advantage in performance on this dataset, as expected. However, we see Temporal Correction performing slightly better. This reveals an interesting advantage of Temporal Correction. We can see from Table 3 that the Ozone dataset has slightly negative temporal dependence (the proper probability of the majority class is more than the probability of a majority class instance following a majority class in a sequence). Temporal Correction classifier estimates the sequential probabilities and successfully captures this dependence.

Note, that NB and DDM use Naive Bayes as the base classifier. Naive Bayes assumes independence of inputs. When temporal dependence is present, the labels that are close in

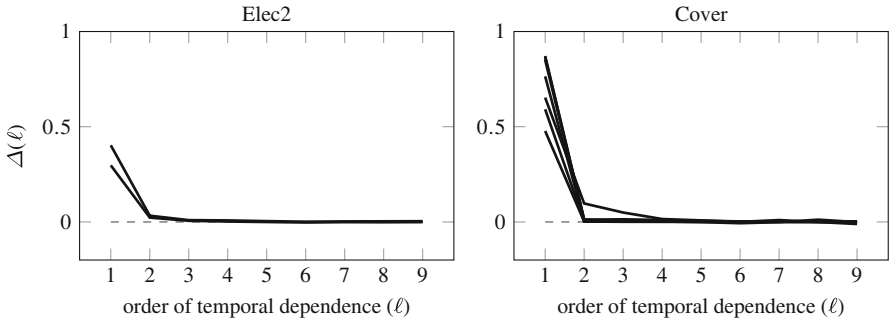

Fig. 5 Evidence for temporal dependence in *Elec2* and *Cover* datasets (*each line* represents one class)

time are not independent from each other. In turn, the input features are not independent from the past labels. The Temporally Augmented classifier with Naive Bayes technically violates this assumption. Many studies have shown that Naive Bayes can perform well in cases when the independence assumption is violated (e.g. Domingos and Pazzani 1997). We see from the plots that in practice the accuracy is not affected noticeably, Temporally Augmented in most of the cases still outperforms Temporal Correction, which has no violation, but uses a simplified model of temporal dependence.

6.8 Sensitivity analysis to the order of temporal dependence

In the previous experiments we considered only first order temporal dependence. Next we analyze the performance of the Temporally Augmented classifier taking into account higher order temporal dependences. In this analysis we use the Elec2 and Cover datasets, since on these datasets we saw large improvements due to taking into account first order temporal dependence, we investigate if incorporating higher order temporal dependence can improve the performance further.

From Definition 1 it follows that positive dependence of order ℓ is present in data if adding information about one more past label changes the conditional probability of observing some of the classes now. To check whether Elec2 and Cover actually contain higher order temporal dependence, in Fig. 5 we plot the difference between conditional probabilities of the classes when taking one more past label into account $\Delta(\ell) = P(y_t = i | y_{t-1} = i, \ldots, y_{t-\ell} = i) - P(y_t = i | y_{t-1} = i, \ldots, y_{t-\ell-1} = i)$. If $\Delta(\ell) \neq 0$ it means that ℓ^{th} order temporal dependence is present.

We see that both datasets have strong first order dependence and some second order dependence, while there is almost no higher order dependence. Therefore, we do not expect to see any major improvements due to taking into account higher than second order dependence. Figure 5 confirms this expectation. It depicts accuracies of Temporally Augmented with different base classifiers taking into account different windows of past labels (ℓ). We see small improvement in classifiers, particularly DDM, when second order dependence is taken into account; however, we see no further improvement.

It is interesting to note that Elec2 data has a seasonal component, the consumption patterns tend to recur every 24 h. However, the added value of taking such a long history into account is not necessarily worthwhile, for instance, $\Delta(48) = 0.02$. Even though a label 48 observations ago (24 h ago) may be strongly correlated with the current label, this does not necessarily provide extra predictive information, since this information may already be in the input features or labels at other lags. Experiments with the Temporally Augmented classifier

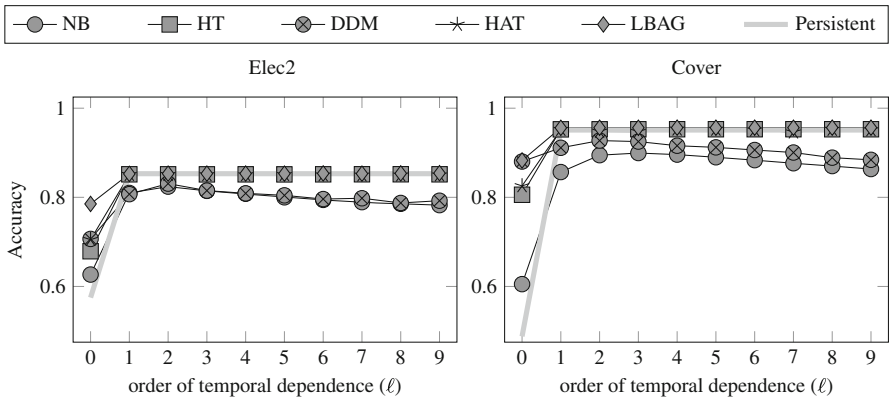

Fig. 6 Accuracies of Temporally Augmented classifier as a function of the order of considered temporal dependence ℓ

including long history, reported in Fig. 6, confirmed this observation. No substantial benefits in accuracy due to including longer history were observed.

6.9 Problems with drift detection

In the last experimental section we analyze drift detection when temporal dependence is present, as discussed in Sect. 5. We use the DDM classifier for analysis of the performance, the same as in our preceding experiments. This classifier is equipped with a change detection mechanism, that drops the old portion of data when a change is detected, and starts training from scratch. To test the effect of false positives, we use DDM-random where instead of an intelligent change detection we put a random change detector, that does not consider any data and simply alarms a change at every time step with probability p. Note that DDM is using a warm-up period of 30 instances, during which change alarms are not allowed. We keep this constraint. It means that for DDM-random if $p = 1$ change is alarmed at every 30th instance. Our goal is to analyze if increasing probability of an alarm gives a higher accuracy, as theoretically argued in Sect. 5. For comparison we also plot NB. The difference between NB and DDM is only in the fact that DDM uses change detection and NB does not. We expect NB and DDM to perform the same on the identically distributed datasets where no change detections should occur (all the changes detected on such datasets would be false alarms).

We experiment with two datasets, Elec2 and Cover, that contain temporal dependence as well as concept drift. We use three versions of these datasets. The first version is the original dataset. The second dataset is shuffled in such a way that the order of the labels (and thus the temporal dependence) is preserved, but within each class data is randomly permuted such that the class conditional distribution becomes uniform over time. This way we expect to get rid of concept drift, but preserve the original temporal dependence. The third dataset is a random permutation of the original dataset over time, making the distribution uniform and dataset itself iid. This procedure was used previously in our experiment with performance statistics.

Figure 7 plots the results, note the log scale on the horizontal axis. The plots with original datasets and datasets with temporal dependence show clear trends of increasing accuracy when the probability of false alarms is increasing. This confirms the theoretical results that false alarms make a classifier that does not take temporal dependence into account behave like the Persistent classifier. In Elec2 shuffled, Cover shuffled, and Elec2 temporal the accuracies

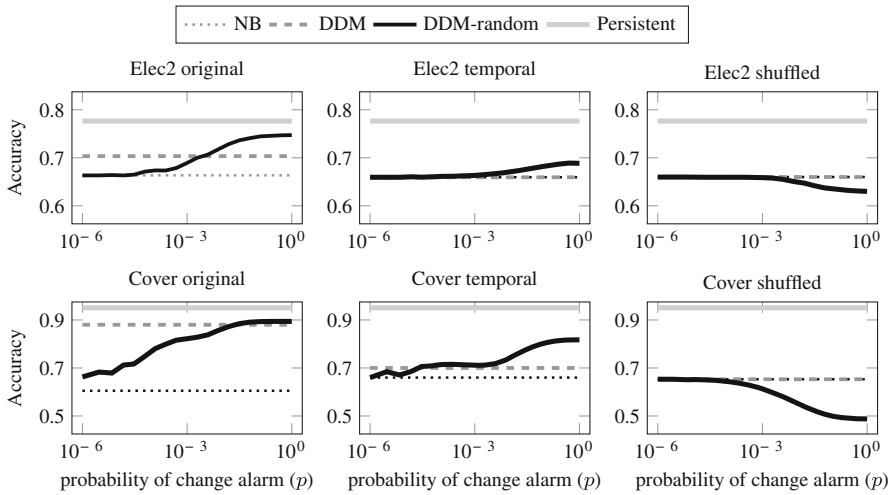

Fig. 7 Effects to change detection; x is plotted in log

of NB and DDM are overlapping, which means that no changes are detected. The accuracies in the plots do not quite reach the theoretical limit of the Persistent classifiers, since the training set size cannot approach its minimum (1) in this experiment due to the warm-up constraint (30 instances) imposed by DDM, but the original DDM, which is supposed to detect only true changes and minimize the number of false alarms, is clearly outperformed when temporal dependence is present.

The plots with randomly shuffled datasets (no temporal dependence and no concept drift) confirm that false alarms come at a cost. False alarms reduce the average training set size for the models. It is well known from statistical learning theory that the generalization performance of a predictive model depends on the training sample size (see e.g. Duda et al. 2001).

An important conclusion follows from this experiment. Classification accuracy should not be used as a proxy for evaluating change detectors with datasets that have temporal dependence. Furthermore, if data contains temporal dependence, false alarms may improve observed classification accuracy. However, this improvement is not meaningful taking into consideration the naive baseline Persistent classifier, which presents the theoretical limit for such an improvement.

7 Recommendations for practitioners

Two main recommendations follow from our analysis. First, one should try to utilize two sources of information when building predictive models: information contained in descriptive input features, and temporal information contained in past labels. The proposed approaches Temporal Correction and Temporally Augmented present simple means for taking temporal information into account.

In the data stream setting running an online experiment just to test whether there is a temporal dependence may be impractical or sometimes even infeasible. An easy test whether there is a temporal dependence (and whether it is worth considering taking it into account) is to compare the accuracy of the Majority Class classifier to the accuracy of the Persistent

classifier on a small sample of data (100 observations or so). If temporal dependence is present, then consider wrapping your favorite classifiers into Temporally Augmented and Temporal Correction classifiers.

Second, we recommend using the Combined measure in any case for data stream classification (instead of the Kappa statistic), as there is no need to know if there is a temporal dependence in the data. The Combined measure evaluates the performance of a classifier with respect to two aspects: whether it is close to random guessing of labels and whether it is close to a persistent naive prediction always predicting the last seen label. If there is no temporal dependence in the data, the Combined measure will give the same results as the Kappa statistic.

8 Conclusion

As researchers, we may have not considered temporal dependence in data stream mining seriously enough when evaluating stream classifiers. We presented a decision theory for classification and proposed two generic classification approaches that can be used with any existing classifiers for taking temporal information into account. We also theoretically analyzed classifier evaluation peculiarities when temporal dependence is present in the data and proposed a new evaluation statistic to take temporal dependence into account. Finally, we pointed out that change detection results should be interpreted with caution when there is a temporal dependence. We showed that signaling a lot of false positives actually leads to better prediction accuracy than a correct detection.

This study opens interesting directions for future research. Firstly, we see that the proposed approaches Temporal Correction and Temporally Augmented, while performing much better than current state-of-the-art approaches, still have a lot of room for improvement in accuracy. More sophisticated approaches for taking into account temporal dependence could be investigated. Secondly, in reality previous labels may arrive with a delay, in such a case classifier update will be delayed. If we take temporal dependence into the predictive model, there are several non-trivial options of how to make a prediction if labels are delayed. One could use the previous predicted label, an older label or a combination of both. This calls for a thorough investigation and is left out of the scope of the present paper for future research.

Acknowledgments I. Žliobaitė's research has been supported by the Academy of Finland Grant 118653 (ALGODAN).

Appendix

Proofs

Proof (of Proposition 5) The accuracies p_{maj}, p_{ran} and p_0 do not include conditional probabilities with respect to the sequence of the data, hence, they are the same whether there is a temporal dependence in the data or not.

Next we prove that $p_{\text{maj}} \geq p_0$. For brevity denote $P(y = i)$ as $P(i)$ and $P_h(\hat{y} = i)$ as $P_h(i)$. Let $P(M)$ be the prior probability of the majority class, which implies that $P(M) \geq \frac{1}{k}$. Since $p_0 = \frac{1}{k}$, we get that $p_{\text{maj}} \geq p_0$.

Finally, we prove that $p_{\text{maj}} \geq p_{\text{ran}}$. To prove the proposition we need to demonstrate that $p_{\text{maj}} - p_{\text{ran}} \geq 0$. Substituting in the expressions for accuracies gives $p_{\text{maj}} - p_{\text{ran}} =$

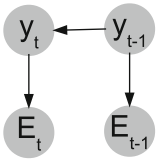

Fig. 8 The model for joint probability in the proof of Proposition 8

$P(M) - \sum_{i=1}^{k} P(i)P_h(i) = P(M) - P(M)P_h(M) - \sum_{i\neq M} P(i)P_h(i) = P(M)(1 - P_h(M)) - \sum_{i\neq M} P(i)P_h(i) \geq \frac{1}{k}\sum_{i\neq M} P_h(i) - \sum_{i\neq M} P(i)P_h(i) \geq \frac{1}{k}\sum_{i\neq M} P_h(i) - \sum_{i\neq M} \frac{1}{k} P_h(i) = 0$. The second inequality follows from the observation that for the minority classes $i \neq A$ the prior probabilities $P(i) \leq \frac{1}{k}$. □

Proof (of Proposition 8) Let $P(E_t)$ denote the probability of an error at time t. If the errors are distributed independently in a stream, then $P(E_t|E_{t-1}) = P(E_t)$, we need to prove this under the theorem conditions.

The probability of an error can be expressed as $P(E_t) = P(E_t, A_t) + P(E_t, B_t) = P(A_t)P(E_t|A_t) + P(B_t)P(E_t|B_t) = P(A_t)P(\hat{B}_t|A_t) + P(B_t)P(\hat{A}_t|B_t)$, where A and B are the true classes (binary classification task), and $\hat{A}$ and $\hat{B}$ are the predicted classes. Similarly, $P(E_{t-1}) = P(A_{t-1})P(\hat{B}_{t-1}|A_{t-1}) + P(B_{t-1})P(\hat{A}_{t-1}|B_{t-1})$. Assuming there is no concept drift and the classifier predicts using a fixed rule we have $P(i_t) = P(i_{t-1}) = P(i)$ and $P(\hat{i}_t|j_t) = P(\hat{i}_{t-1}|j_{t-1}) = P(\hat{i}|j)$, for $i, j \in \{A, B\}$. Therefore, $P(E_t) = P(E_{t-1})$. Hence, $P(E_t|E_{t-1}) = P(E_t)$ can be rewritten as $P(E_t, E_{t-1}) = P(E_t)^2$.

The right side is $P(E_t)^2 = P(A)^2P(\hat{B}|A)^2 + 2P(A)P(B)P(\hat{A}|B)P(\hat{B}|A) + P(B)^2P(\hat{A}|B)^2$. The left side can be expressed as $P(E_t, E_{t-1}) = \sum_{i\in\{A,B\}}\sum_{j\in\{A,B\}} P(E_t, E_{t-1}, y_t = i, y_{t-1} = j)$, where y denotes the true class label. Since the error at time t only depends on the true label at time t, but not the true label at time $t-1$, we can express the joint probability following the graphical model in Fig. 8 as $P(E_t, E_{t-1}) = \sum_{i\in\{A,B\}}\sum_{j\in\{A,B\}} P(y_{t-1} = j)P(E_{t-1}|y_{t-1} = j)P(y_t = i|y_{t-1} = j)P(E_t|y_t = i) = P(A_{t-1})P(\hat{B}_{t-1}|A_{t-1})P(A_t|A_{t-1})P(\hat{B}_t|A_t) + P(B_{t-1})P(\hat{A}_{t-1}|B_{t-1})P(A_t|B_{t-1})P(\hat{B}_t|A_t) + P(A_{t-1})P(\hat{B}_{t-1}|A_{t-1})P(B_t|A_{t-1})P(\hat{A}_t|B_t) + P(B_{t-1})P(\hat{A}_{t-1}|B_{t-1})P(B_t|B_{t-1})P(\hat{A}_t|B_t) = P(A)P(A_t|A_{t-1})P(\hat{B}|A)^2 + P(B)P(A_t|B_{t-1})P(\hat{A}|B)P(\hat{B}|A) + P(A)P(B_t|A_{t-1})P(\hat{A}|B)P(\hat{B}|A) + P(B)P(B_t|B_{t-1})P(\hat{A}|B)^2$. Having both expressions now we can analyze the difference $P(E_t, E_{t-1}) - P(E_t)^2 = P(A)P(\hat{B}|A)^2\big(P(A_t|A_{t-1}) - P(A)\big) + P(B)P(\hat{A}|B)P(\hat{B}|A)\big(P(A_t|B_{t-1}) - P(A)\big) + P(A)P(\hat{A}|B)P(\hat{B}|A)\big(P(B_t|A_{t-1}) - P(B)\big) + P(B)P(\hat{A}|B)^2\big(P(B_t|B_{t-1}) - P(B)\big) = P(A)P(\hat{B}|A)^2(P(A_t|A_{t-1}) - P(A)) + P(B)P(\hat{A}|B)P(\hat{B}|A)(P(B) - P(B_t|B_{t-1})) + P(A)P(\hat{A}|B)P(\hat{B}|A)(P(A) - P(A_t|A_{t-1})) + P(B)P(\hat{A}|B)^2(P(B_t|B_{t-1}) - P(B)) = \big(P(A_t|A_{t-1}) - P(A)\big)P(A)P(\hat{B}|A)\big(P(\hat{B}|A) - P(\hat{A}|B)\big) + \big(P(B_t|B_{t-1}) - P(B)\big)P(B)P(\hat{A}|B)\big(P(\hat{A}|B) - P(\hat{B}|A)\big) = \big(P(\hat{B}|A) - P(\hat{A}|B)\big)\big(P(A)P(\hat{B}|A)\big(P(A_t|A_{t-1}) - P(A)\big) - P(B)P(\hat{A}|B)\big(P(B_t|B_{t-1}) - P(B)\big)\big)$. We can see that this expression is equal to *zero* if $P(\hat{B}|A) = P(\hat{A}|B$, which is the proposition condition #2, *or* if $P(A_t|A_{t-1}) = P(A)$ and $P(B_t|B_{t-1}) = P(B)$, which means that there is no temporal dependence in data, which is the proposition condition #1, *or* if $P(A)P(\hat{B}|A)\big(P(A_t|A_{t-1}) - P(A)\big) = P(B)P(\hat{A}|B)\big(P(B_t|B_{t-1}) - P(B)\big)$, which transforms to $\frac{P(A_t|A_{t-1}) - P(A)}{P(B_t|B_{t-1}) - P(B)} = \frac{P(B)P(\hat{A}|B)}{P(A)P(\hat{B}|A)}$, which is the proposition condition #3. □

Proof (of Proposition 9) Persistent classifier does not depend on training sample size, since only the previous label is used for making predictions. Its accuracy is given in Eq. (7) as

$p_{\text{per}} = \sum_{i=1}^{k} P(y_t = i)P(y_t = i|y_{t-1} = i)$. Majority Class classifier requires knowing the prior probabilities of the classes, which depend on the sample size used for estimation, as follows. Temporal dependence in data can be represented as a Markov chain with the $k \times k$ transition matrix $R = (r_{i|j})$, where $r_{ij} = P(y_t = j|y_{t-1} = i)$, and k is the number of classes. The transition matrix for a finite state Markov chain is a stochastic matrix[1]. An irreducible aperiodic stochastic matrix converges to a stationary distribution $\lim_{n\to\infty}$ and the convergence rate is exponential in the order of the second largest eigenvalue (see e.g. Schmitt and Rothlauf 2001). Hence, $\frac{P(i_t)-P(i)}{P(i_{t-1})-P(i)} \approx \lambda_2$, here $P(i_t)$ is the prior probability of seeing class i at time t from the start of sampling and $p(i)$ is the prior probability of class i after seeing infinitely many samples. The prior probability of class i in the first n samples is $P(\bar{i}_n) = \sum_{t=1}^{n} P(i_t)/n$. The sum can be modeled as a geometric progression with ratio λ_2, which is $\sum_{t=1}^{n} P(i_t) \approx \big(P(i_1) - P(i)\big)\frac{\lambda_2^n - 1}{\lambda_2 - 1} + nP(i)$.

If a detection alarm is fired, there has been an observation at time 0 immediately before restarting training of the classifier. This observation may have belonged to any class i with a probability $P(i)$. Therefore, at time 1 after restarting the training the observation is class i with the probability $P(i_t|i_{t-1})$. If $P(i_t|i_{t-1}) > \frac{1}{k}$, then i is the majority class at time 1. Then at time n the probability of the class i is $P(\bar{i}_n) \approx \big(P(i_t|i_{t-1}) - P(i)\big)\frac{\lambda_2^n - 1}{n(\lambda_2 - 1)} + P(i)$. The overall probability of the majority class at time n is then $p_{\text{maj}} = \sum_{i=1}^{k} P(i)P(\bar{i}_n) \approx \sum_{i=1}^{k} \Big(P(i)P(i_t|i_{t-1})\frac{\lambda_2^n - 1}{n(\lambda_2 - 1)} + P(i)\big(1 - \frac{\lambda_2^n - 1}{n(\lambda_2 - 1)}\big)\Big)$. Substituting in the expression for p_{maj} at time n into the proposition statement gives $\lim_{n\to 1} p_{\text{maj}} - p_{\text{per}} = \sum_{i=1}^{k} P(i)P(i_t|i_{t-1}) - \sum_{i=1}^{k} P(i)P(i_t|i_{t-1}) = 0$. □

References

Bache, K., & Lichman, M. (2013). *UCI machine learning repository*. http://archive.ics.uci.edu/ml. Accessed Oct 2013.

Baena-Garcia, M., del Campo-Avila, J., Fidalgo, R., Bifet, A., Gavalda, R., & Morales-Bueno, R. (2006). Early drift detection method. In *Proceedings of the 4th ECMLPKDD International Workshop on Knowledge Discovery from Data Streams* (pp. 77–86).

Bifet, A., & Gavalda, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 7th SIAM International Conference on Data Mining, SDM*.

Bifet, A., & Gavalda, R. (2009). Adaptive learning from evolving data streams. In *Proceedings of the 8th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VIII, IDA '09* (pp. 249–260).

Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2010). Moa: massive online analysis. *The Journal of Machine Learning Research*, *11*, 1601–1604.

Bifet, A., Holmes, G., & Pfahringer, B. (2010). Leveraging bagging for evolving data streams. In *Proceedings of the 2010 European conference on Machine Learning and Knowledge Discovery in Databases, ECMLPKDD* (pp. 135–150).

Bifet, A., Holmes, G., Pfahringer, B., & Frank, E. (2010). Fast perceptron decision tree learning from evolving data streams. In *Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD* (pp. 299–310).

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavaldà, R. (2009). New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD* (pp. 139–148).

Bifet, A., Read, J., Zliobaite, I., Pfahringer, B., & Holmes, G. (2013). Pitfalls in benchmarking data stream classification and how to avoid them. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECMLPKDD* (pp. 465–479).

Box, G., Jenkins, G. M., & Reinsel, G. C. (1994). *Time series analysis forecasting and control* (3rd ed.). Englewood Cliffs, NJ: Prentice-Hall.

[1] A stochastic matrix is a square matrix whose entries are non-negative and whose rows sum to 1.

Brzezinski, D., & Stefanowski, J. (2014). Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE Transactions on Neural Networks and Learning Systems, 25*(1), 81–94.
Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*(1), 37–46.
Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.
Dietterich, T. G. (2002). Machine learning for sequential data: A review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical, Pattern Recognition* (pp. 15–30).
Ditzler, G., & Polikar, R. (2013). Incremental learning of concept drift from streaming imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 25*(10), 2283–2301.
Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 71–80).
Domingos, P., & Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, *29*(2–3), 103–130.
Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York: Wiley.
Gaber, M. M., Zaslavsky, A., & Krishnaswamy, S. (2005). Mining data streams: a review. *SIGMOD Record*, *34*(2), 18–26.
Gama, J., & Castillo, G. (2006). Learning with local drift detection. In *Proceedings of the 2nd International Conference on Advanced Data Mining and Applications, ADMA* (pp. 42–55).
Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In *Proceedings of the 7th Brazilian Symposium on Artificial Intelligence, SBIA* (pp. 286–295).
Gama, J., Sebastião, R., & Rodrigues, P. (2013). On evaluating stream learning algorithms. *Machine Learning*, *90*(3), 317–346.
Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys, 46*(4) (in press).
Gomes, J., Menasalvas, E., & Sousa, P. (2010). CALDS: Context-aware learning from data streams. In *Proceedings of the 1st International Workshop on Novel Data Stream Pattern Mining Techniques, StreamKDD* (pp. 16–24).
Grinblat, G., Uzal, L., Ceccatto, H., & Granitto, P. (2011). Solving nonstationary classification problems with coupled support vector machines. *IEEE Transactions on Neural Networks*, *22*(1), 37–51.
Harries, M. (1999). *SPLICE-2 comparative evaluation: Electricity pricing*. Technical report, University of New South Wales.
Hulten, G., Spencer, L., & Domingos, P. (2001). Mining time-changing data streams. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD* (pp. 97–106).
Jackowski, K. (2013). Fixed-size ensemble classifier system evolutionarily adapted to a recurring context with an unlimited pool of classifiers. *Pattern Analysis and Applications*. doi:10.1007/s10044-013-0318-x.
Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, *82*, 35–45.
Knoth, S., & Schmid, W. (2004). Control charts for time series: a review. In H. J. Lenz & P. T. Wilrich (Eds.), *Frontiers in statistical quality control* (Vol. 7, pp. 210–236). Heidelberg: Physica-Verlag.
Kolter, J., & Maloof, M. (2007). Dynamic weighted majority: an ensemble method for drifting concepts. *The Journal of Machine Learning Research*, *8*, 2755–2790.
Lavielle, M. (1999). Detection of multiple changes in a sequence of dependent variables. *Stochastic Processes and Their Applications*, *83*(1), 79–102.
Martinez-Rego, D., Perez-Sanchez, B., Fontenla-Romero, O., & Alonso-Betanzos, A. (2011). A robust incremental learning method for non-stationary environments. *Neurocomputing*, *74*(11), 1800–1808.
Pavlidis, N., Tasoulis, D., Adams, N., & Hand, D. (2011). Lambda-perceptron: an adaptive classifier for data streams. *Pattern Recognition*, *44*(1), 78–96.
Rabiner, L. R. (1990). A tutorial on hidden Markov models and selected applications in speech recognition. In A. Waibel & K.-F. Lee (Eds.), *Readings in speech recognition* (pp. 267–296). San Francisco, CA: Morgan Kaufmann Publishers Inc.
Rodrigues, P. P., & Gama, J. (2009). A system for analysis and prediction of electricity-load streams. *Intelligent Data Analysis*, *13*(3), 477–496.
Ross, G., Adams, N., Tasoulis, D., & Hand, D. (2012). Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognition Letters*, *33*, 191–198.
Schmitt, F., & Rothlauf, F. (2001). On the importance of the second largest eigenvalue on the convergence rate of genetic algorithms. In *Technical Report, Procedings of the 14th Symposium on Reliable Distributed Systems*.
Tomczak, J., & Gonczarek, A. (2013). Decision rules extraction from data stream in the presence of changing context for diabetes treatment. *Knowledge and Information Systems*, *34*(3), 521–546.

Wieringa, J. E. (1999). *Statistical process control for serially correlated data*. Ph.D. thesis, Groningen University.

Zliobaite, I. (2010). Learning under concept drift: An overview. *CoRR* abs/1010.4784.

Zliobaite, I. (2011). Combining similarity in time and space for training set formation under concept drift. *Intelligent Data Analysis*, *15*(4), 589–611.

Zliobaite, I. (2013). How good is the electricity benchmark for evaluating concept drift adaptation. *CoRR* abs/1301.3524.