On Using Novel “Anti-Bayesian” Techniques for the Classification of Dynamical Data Streams

Hugo Lewi Hammer, Anis Yazidi, B. John Oommen

***Abstract*—The classification of dynamical data streams is among the most complex problems encountered in classification. This is, firstly, because the distribution of the data streams is non-stationary, and it changes without any prior “warning”. Secondly, the manner in which it changes is also unknown. Thirdly, and more interestingly, the model operates with the assumption that the correct classes of previously-classified patterns become available at a juncture after their appearance. This paper pioneers the use of unreported novel schemes that can classify such dynamical data streams by invoking the recently-introduced “Anti-Bayesian” (AB) techniques. Contrary to the Bayesian paradigm, that compare the testing sample with the distribution’s central points, AB techniques are based on the information in the distant-from-the-mean samples.**

Most Bayesian approaches can be naturally extended to dynamical systems by dynamically tracking the mean of each class using, for example, the exponential moving average based estimator, or a sliding window estimator. The AB schemes introduced by Oommen *et al*., on the other hand, work with a radically different approach and with the non-central *quantiles* of the distributions. Surprisingly and counter-intuitively, the reported AB methods work equally or close-to-equally well to an optimal supervised Bayesian scheme on a host of accepted PR problems. This thus begs its natural extension to the unexplored arena of classification for dynamical data streams. Naturally, for such an AB classification approach, we need to track the non-stationarity of the *quantiles* of the classes. To achieve this, in this paper, we develop an AB approach for the online classification of data streams by applying the efficient and robust quantile estimators developed by Yazidi and Hammer [3], [13].

Apart from the methodology itself, in this paper, we compare the Bayesian and AB approaches. The results demonstrate the intriguing and counter-intuitive results that the AB approach shows competitive results to the Bayesian approach. Furthermore, the AB approach is much more robust against outliers, which is an inherent property of *quantile* estimators [3], [13], which is a property that the Bayesian approach cannot match, since it rather tracks the mean.

Keywords: *Anti-Bayesian Classification, Data Streams, Classification With delay, Incremental Quantile Estimation*

I. Introduction

A. Problem Statement

The Pertinence of Data Streams: Traditionally, Machine Learning (ML) methods are assumed to deal with static data stored in memory, which can be read several times. On the contrary, streaming data grows at an unlimited rate and arrives continuously in a single-pass manner that can only be read once. Further, there are space and time restrictions in analyzing streaming data. Consequently, one needs methods that are “automatically adapted” to update the training models based on the information gathered over the past observations whenever a change in the data is detected. In addition, a typical challenge in analyzing data streams is that the properties of the stream varies dynamically with time, where traditional static analysis tools cannot be applied.

The classification of such dynamical data streams is among the most complex problems encountered in Pattern Recognition (PR) and ML. This is primarily because the data stream’s class conditional distribution is non-stationary. It changes to a new unknown distribution, i.e., the distribution of the new stream, without any indication that such a switch is going to occur. And the most interesting facet of this is that the model operates with the assumption that the correct classes of previously-classified patterns become available at a juncture after their initial appearance.

This scenario is more pertinent today than ever before. Indeed, in the past few years, due to the advances in computer hardware technology, large amounts of data have been generated and collected and are stored permanently from different sources. Some the applications that generate data streams are financial tickers, log records or click-streams in web tracking and personalization, data feeds from sensor applications, and call detail records in telecommunications. Furthermore, data streams could be social media feeds from Twitter or online news, network data, economic or environmental data etc. Analysis of these data streams has received a lot of attention in

Author’s status: *Associate Professor*. This author can be contacted at: Oslo and Akershus University College, Department of Computer Science, Pilestredet 35, Oslo, Norway. E-mail: hugo.hammer@hioa.no.

Author’s status: *Associate Professor*. This author can be contacted at: Oslo and Akershus University College, Department of Computer Science, Pilestredet 35, Oslo, Norway. E-mail: anis.yazidi@hioa.no.

Chancellor’s Professor ; *Fellow: IEEE* and *Fellow: IAPR*. This author can be contacted at: School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway. E-mail address: oommen@scs.carleton.ca.

978-1-5090-4601-0/17/$31.00 ©2017 IEEE

the literature [4] and is considered as one of the most important challenges in the field of ML and PR.

The Bayesian ML of Data Streams: Almost all traditional classification techniques reported to-date depend, either directly or implicitly, on the Bayesian principle which yields optimal classification rules. To be more specific, within a Bayesian paradigm, if one has to classify a testing sample by resorting to *a single* point in the feature space from each class, the *optimal* Bayesian strategy would be to achieve this based on the "distance" (for example, Euclidean or Mahalonabis) from the corresponding means or *central* points in the respective distributions. In this vein, in order to deal with the challenges that pertain to data streams, a large body of studies have focused on the idea of summarizing the characteristics of a data stream by rather tracking the properties of the stream, like its distributional moments (expectation, variance, skewness, kurtosis etc) or quantiles [3], [13]. In fact, those quantities are usually easy to compute in a incremental manner and can serve as a "footprint" of the data stream in question, whence the classification can be achieved.

From the above, the informed reader can observe that any classical ML or PR task can be re-written as a new problem within the framework of analyzing data streams. Typical examples include that of assigning arriving data samples to one of a set of classes, or to a cluster, where the true class (cluster) label is revealed subsequently – with a certain time delay. Several different methods have been suggested for these tasks, and an excellent review of this is found in [4]. Indeed, traditional clustering and classification techniques proposed for dynamic data streams, typically depend, either directly or implicitly, on the Bayesian principle of optimal classification.

The "Anti-Bayesian" ML of Data Streams: In this paper, we introduce a novel alternative to the Bayesian classification approaches by operating in a diametrically opposite way, i.e., a so-called "Anti-Bayesian" (AB) manner. Indeed, we shall show the completely counter-intuitive result that by tracking a few points from each class which are distant from the mean, one can obtain remarkable classification performances for dynamic data streams — that can even outperform the Bayesian counter-part in some situations. Although we follow the steps of traditional ML and PR, we classify the data points to classes using completely different criteria, i.e., by invoking the AB paradigm. More specifically, unlike the traditional Bayesian classification strategies which rely on classifying based on the mean/central values of the classes, our paradigm advocates the classification of points to classes based on *quantiles distant from the means of each class* [8] [10] [6], which is a concept that was previously unreported in the literature. Indeed, it is actually both un-intuitive and non-obvious.

It is fitting to mention that even though AB methods have found applications in classification and clustering, their corresponding application in dynamic streams is not consequential. This is because the samples that are "outliers" (and that represent the distant quantiles in any given distribution) may not continue to be "outliers" when the distribution changes. The fact that AB schemes are valid for even such non-stationary settings is one of the primary contributions of this paper.

***Multiplicative* Incremental Quantile Estimators**: As mentioned, the central concept of this paper involves using AB methods for dynamic streams. This, in turn, necessitates the dynamic estimation of the quantiles of a time-varying distribution. As is well known, the standard way of estimating a quantile related to some probability value p in a static system, is to sort the quantiles and to then select the data point in position $\lfloor pn \rfloor$ or $\lceil pn \rceil$ (or using an appropriate weighting factor). As we will highlight later, such an approach can be non-functional, in practice, for dynamic data streams. However, incremental quantile estimators are estimators that do small (marginal) updates of the quantile estimates every time a new sample is received from the data stream. Quantile estimators, that are incremental in principle, have been reported in [12], [2], [1]. In this paper, we will, rather, invoke the more-recently introduced estimators due to Yazidi and Hammer [3], [13]. Being *multiplicative* incremental quantile estimators, they are not only more efficient then the current state-of-the-art quantile estimators for data streams, but are also far simpler to implement. The paper utilizes the Deterministic Update Based Multiplicative Incremental Quantile Estimator (DUMIQE) and its Multiple version, the MDUMIQE, which has proven consistent properties.

An Enhanced Model of Computation: In addition to all the issues mentioned here, as alluded to above, we also adopt the recently-proposed *online* classification model, with delay, proposed by Hanane *et al.* [7]. The model is composed of three stages. In the first phase, the model learns from the available labeled samples. In the second phase, the learned model predicts the class label of the unlabeled instance(s) currently observed. In the third phase, after knowing the true class label of these recently-classified instance(s), the classification model is adjusted in an *online* manner.

Robustness against outliers: When dealing with dynamic data, classical moving average estimation methods are inefficient as they are not able to deal with outlier observations which are well known to be susceptible to corrupting the estimated mean. However, the DUMIQE quantile estimator copes with this problem in a natural manner. This is an inherent part of the estimation process for quantiles, which, as such, makes AB classification much more robust against outliers.

B. Format of the Paper

First of all, in Section II, we present a rather thorough overview of the current state-of-the-art. Then, in Section III, we discuss the fundamentals of Bayesian and AB classification in static (or stationary) systems. In this section, we shall discuss the details of the techniques involved so that a practitioner can readily implement any of these methods. Section III then describes, in fair detail, the principles of Bayesian and AB classification in a static stream. Section IV explains how we can efficiently track the quantiles and the mean value of a dynamic data stream, which leads, quite naturally to Section V, where we explain the Bayesian and AB methodologies for classification in dynamical data streams. The next two sections describe the experimental results. Section VII concludes the paper.

II. Related Work

In this section, we describe the related work both with respect to the relatively-new field of PR involving the AB paradigm. We will also briefly survey the state-of-the-art when it concerns classification in dynamic data streams.

A. Related Work on "Anti–Bayesian" PR

We first review the related work on AB classification. The review is, necessarily, very brief.

The first results on AB classification dates back to 2013, where Thomas and Oommen [8] proposed the use of the quantiles of the class conditions distributions to achieve classification, instead of using the information in the mean. They formally and experimentally showed that they could obtain optimal classification for various uni-dimensional symmetric distributions, and near-optimal accuracies for asymmetric distributions. For uni-dimensional quantile-based PR, their methodology is based on comparing the testing sample with the ${\frac{n-k+1}{n+1}}^{th}$ percentile of the first distribution and the ${\frac{k}{n+1}}^{th}$ percentile of the second distribution. These results were shown to be applicable for the distributions that are members of the symmetric and asymmetric exponential family. By considering the entire spectrum of the possible values of k, the results in [8], [10] and [6], showed that the specific value of k is usually not so crucial. These authors also confirmed that the same results were true for *multi*-dimensional features.

In [9], the authors further proposed a new border identification algorithm, namely the AB Border Identification scheme. For each class, this method selects, as the corresponding border points, a small number of data points that lie close to the discriminant function's boundary, but where these points are not within the central part of the class conditional distributions.

The results of [8], [10] and [6] were used to design numerous Prototype Reduction Schemes in [11], and an AB text classification scheme in [5].

III. Bayesian and AB Classification in a Static System

A. Basic Notation

Let $X(t)$ be a stochastic variable representing the outcome from a dynamic data stream at time t. We assume that $X(t)$ is from one of K classes $C(t) \in \{1, 2, \ldots, K\}$ and that the probability that $X(t)$ is from class $C(t) = k$ is $p_k(t)$. The conditional distribution of $X(t)$ given class $C(t) = k$, has the probability distribution $f_t(x|k)$, i.e. $X(t)|C(t) = k \sim f_t(x|k)$. Using the law of total probability, we have that the marginal distribution of $X(t)$ is given by $f_t(x) = \sum_{k=1}^{K} f_t(x|k)p_k(t)$. Finally let $(x(t), c(t))$ denote an outcome of the pair, $(X(t), C(t))$, which is the data point examined and its corresponding class label.

We assume a static data stream, i.e. $X(t) = X$, $C(t) = C$, $p_k(t) = p_k$ and so on. We also assume we have a training set of n samples with class labels, $(x_1, c_1), (x_2, c_2), \ldots, (x_n, c_n)$. We now describe the Bayesian and the AB classification methodologies.

B. Bayesian Classification

Let $\widehat{\mu}_k$ denote the mean value of the samples from class k, i.e.

$$\widehat{\mu}_k = \frac{1}{\sum_{i=1}^{n} I(c_i = k)} \sum_{i=1}^{n} I(c_i = k)x_i,$$

where $I(A)$ denote the indicator function that returns the value of unity if A is true, and the value of zero if A is false.

We now receive a new sample x_0 whose class is unknown, and the intention is to classify it to one of the classes. We assume that the distributions $f(x|k)$ and p_k are unknown, and the classification must be based on the training samples. Explained in a rather informal manner, the optimal Bayesian classification rule is to assign x_0 to the class whose class mean is is closest to x_0, i.e., assign x_0 to class k if

$$\|x_0 - \widehat{\mu}_k\| < \|x_0 - \widehat{\mu}_j\| \quad \forall j \neq k.$$

Of course, one must also consider the actual metric used to measure the distance from the means. Indeed, this need not necessarily be the simple Euclidean metric, but could rather be one based on the covariance matrices, for example, the Mahalonabis distance.

C. The Anti-Bayesian (AB) Classification

The AB paradigm is based on a radically different approach from its Bayesian counterpart, where the classification is based on quantiles *distant* from the mean, rather than the mean. The methodology is described in [8], [10] and [6], where its properties have also been proven. Let Q_{kp} denote the quantile related to a given probability value, p, for a class whose index is k, i.e. $P(X \leq Q_{kp}|C = k) = p$. Further, let $\widehat{Q}_{kp}$ denote an estimate of Q_{kp} based on the sample $(x_1, c_1), (x_2, c_2), \ldots, (x_n, c_n)$ using some quantile estimation procedure. We now define $q = 1 - p$ and assume that $p < 1/2$. Consequently, we clearly see that $Q_{kp} < Q_{kq}$.

To explain the AB approach, assume for the present that we have only two classes denoted by $k = 1$ and $k =$

2. A generalization to K classes will then be explained in the next step. In such a case, the AB classification method operates as follows:

1. Determine which of the distributions $f(x|k=1)$ or $f(x|k=2)$ is to the left by using the quantiles of the distributions. We have three possible cases:

   Case 1: If $\widehat{Q}_{1p} < \widehat{Q}_{2p}$ and $\widehat{Q}_{1q} < \widehat{Q}_{2q} \implies f(x|k=1)$ is to the left of $f(x|k=2)$.

   Case 2: If $\widehat{Q}_{1p} > \widehat{Q}_{2p}$ and $\widehat{Q}_{1q} > \widehat{Q}_{2q} \implies f(x|k=2)$ is to the left of $f(x|k=1)$.

   Case 3: Else[1], we determine their relative positions by comparing the averages of the quantiles as follows:

   If $\frac{\widehat{Q}_{1p}+\widehat{Q}_{1q}}{2} < \frac{\widehat{Q}_{2p}+\widehat{Q}_{2q}}{2} \implies f(x|k=1)$ is to the left of $f(x|k=2)$.

   Else $f(x|k=2)$ is to the left of $f(x|k=1)$.

   Figure 1 depicts the above three cases. We see that for Cases 1 and 2, $f(x|k=1)$ and $f(x|k=2)$ are the distributions to the left, respectively. In the bottom figure (Case 3), the decision is not that obvious because the classes are highly overlapping.

2. Once the relative positions of the distributions are determined, the classification rule must now be specified. For simplicity, we describe this merely for Case 1 since the rules for the "mirrored" cases are analogous. The AB rule classifies using the *right* quantile of the left distribution and the *left* quantile of the right distribution. If $B = \frac{\widehat{Q}_{1q}+\widehat{Q}_{2p}}{2}$, we classify as follows:

   If $x_0 < B$, classify x_0 to class $k=1$.

   Else, classify x_0 to class $k=2$.

   This approach works even when the distributions overlap such that $\widehat{Q}_{2p}$ is to the left of $\widehat{Q}_{1q}$ as shown in Figure 2.

If we need to classify x_0 to one of $K > 2$ classes, we simply repeat the procedure described above $K-1$ times in a "winner-takes-all" sequential, pairwise manner. First, we compute if x_0 is more likely to belong to class $k=1$ or $k=2$. Assume that class $k=2$ is the most likely one. We thereafter do an evaluation between classes $k=2$ and $k=3$, and repeat this for all the remaining classes $4,\ldots,K$. Finally, we classify x_0 to the class that is most likely to be the assigned class, after going through all the $K-1$ evaluations.

IV. Tracking Quantiles and the Mean Value of Dynamic Data Streams

We now present algorithms to track the quantiles and the mean values of a data stream. Here, we assume that samples arrive at equidistant time steps[2]

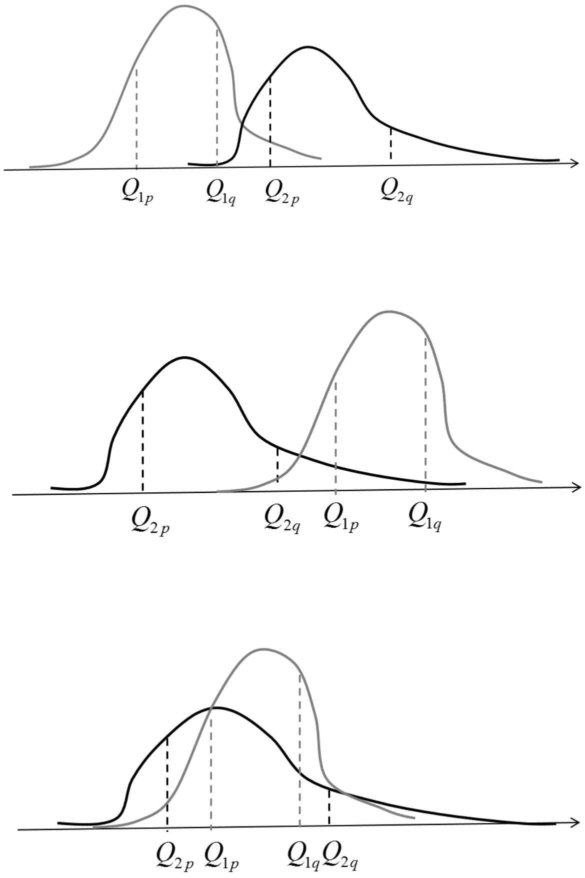

Fig. 1. This figure depicts Cases 1, 2 and 3 – arranged from top to bottom respectively.

A. Tracking Quantiles

We initiate discussions by presenting methods for tracking the quantiles of dynamic data streams. Let $Q_{kp}(t)$ denote the quantile of $X(t)|C(t)=k$ for some probability value p, i.e.,

$$P(X(t) \leq Q_{kp}(t)|C(t)=k) = p.$$

Further, let $\widehat{Q}_{kp}(t)$ be an estimator of $Q_{kp}(t)$. The standard way of estimating a quantile related to some probability value p in a static system is to sort the quantiles, and to then select the data point in position $\lfloor pn \rfloor$ or $\lceil pn \rceil$ (or using an appropriate weighting factor). Unfortunately, such an approach would not work well for dynamic data streams as the computation time and memory requirement increases linearly with the number of samples, n, arriving from the data stream.

Incremental quantile estimators are estimators that do small updates of the quantile estimates every time a new sample is received from the data stream. Incremental quantile estimators have been documented to yield a good performance for dynamical systems, as reported in [12], [2], [1]. More recently, Yazidi and Hammer

[1] This case occurs rarely in practice except when the classes are highly overlapping, in which case the classification problem is often meaningless.

[2] The methodology in this section and in Section V can easily be extended to cases when when samples are received at arbitrary time points $x(t_1), x(t_2), x(t_3), \ldots$.

suggested multiplicative incremental quantile estimators that are not only more efficient then the current state-of-the-art quantile estimators for data streams, but are also far simpler to implement [3], [13]. The AB classifications presented here are, therefore, based on these algorithms used to estimate the quantiles[3]. We now give a short description of the algorithms by Yazidi and Hammer [3], [13].

Suppose that we need to track only a single quantile of the distribution related to some probability p. The method reported in [3], [13] is as follows. We start with some initial quantile estimate $\widehat{Q}_{kp}(0)$, and update the quantile estimate for this class every time we receive a new sample $x(t)$ from class k as per Eq. (1):

$$\begin{aligned} \widehat{Q}_{kp}(t+1) &\leftarrow \widehat{Q}_{kp}(t) + \lambda p \widehat{Q}_{kp}(t), \\ &\qquad \text{if } x(t) > \widehat{Q}_{kp}(t) \\ \widehat{Q}_{kp}(t+1) &\leftarrow \widehat{Q}_{kp}(t) - \lambda(1-p) \widehat{Q}_{kp}(t), \\ &\qquad \text{if } x(t) \leq \widehat{Q}_{kp}(t). \end{aligned} \tag{1}$$

The idea in the above updating rule is quite simply the following: If the sample $x(t)$ is above (below) our current estimate, we should respectively increase (decrease) the corresponding quantile estimates. The variable λ is a parameter that controls the step size, and the weighting with p and $1-p$ is included to ensure that the estimator converges to the true quantile. A potential challenge with these simple rules is that if we start with $\widehat{Q}_{kp}(0) > 0$, every estimate will be above zero. One solution that works well in practice is to run the update rules on a right shifted quantile estimate that is known to be above zero. The estimate of $Q_{kp}(t)$ is then determined by a left shift of the right shifted estimate. For more details about this scheme, referred to as the Deterministic Update Based Multiplicative Incremental Quantile Estimator (DUMIQE), we refer the reader to [3], [13].

To now specifically apply the DUMIQE to AB classification in a dynamic environment, we observe that we need to track two quantiles for the distribution of each class, namely, for the probabilities $p < 1/2$ and $q = (1-p)$. One approach is to simply use the above DUMIQE scheme to estimate both of these quantiles. A challenge with this approach is that we may end up with unrealistic estimates in the sense that the monotone property of quantiles gets violated. This means that $\widehat{Q}_{kp}(t)$ gets a higher value than $\widehat{Q}_{kq}(t)$ even though p is less than q.

In [3], Hammer and Yazidi suggested a modification of the DUMIQE scheme to ensure that the monotone property of the quantiles are satisfied in every iteration. Suppose that at time t that the monotone property is satisfied, i.e. $\widehat{Q}_{kp}(t) < \widehat{Q}_{kq}(t)$. We may get a violation if $x(t)$ gets a value between $\widehat{Q}_{kp}(t)$ and $\widehat{Q}_{kq}(t)$. According to Eq. (1), we will obtain the following updates:

$\widehat{Q}_{kp}(t+1) \leftarrow \widehat{Q}_{kp}(t) + \lambda p \widehat{Q}_{kp}(t)$, which is an increased value, and

$\widehat{Q}_{kq}(t+1) \leftarrow \widehat{Q}_{kq}(t) - \lambda(1-q) \widehat{Q}_{kq}(t)$, which is a decreased value.

Since the lower quantile estimate gets an increased value while the upper quantile receives a reduced value, we observe that we could obtain an overlap that violates the monotone property of the quantiles. The idea suggested in [3] is to adjust the update size, λ, to ensure that this quantile monotone property is satisfied. One such value of λ (denoted $\widetilde{\lambda}$ below) can be determined by ensuring that the distance between $\widehat{Q}_{kp}(t+1)$ and $\widehat{Q}_{kq}(t+1)$ is some portion, $\alpha \in (0,1)$, of the distance from the previous iteration, i.e.,

$$\begin{aligned} &\widehat{Q}_{kq}(t+1) - \widehat{Q}_{kp}(t+1) = \alpha \left(\widehat{Q}_{kq}(t) - \widehat{Q}_{kp}(t) \right) \\ &(1 - \widetilde{\lambda}(1-q)) \widehat{Q}_{kq}(t) - (1 + \widetilde{\lambda} q) \widehat{Q}_{kp}(t) = \\ &\qquad\qquad = \alpha \left(\widehat{Q}_{kq}(t) - \widehat{Q}_{kp}(t) \right). \end{aligned} \tag{2}$$

Solving Eq. (2) with respect to $\widetilde{\lambda}$ yields:

$$\widetilde{\lambda} = \beta \frac{\widehat{Q}_{kq}(t) - \widehat{Q}_{kp}(t)}{p \left(\widehat{Q}_{kq}(t) + \widehat{Q}_{kp}(t) \right)}, \tag{3}$$

where $\beta = 1 - \alpha$. By utilizing the quantity $\widetilde{\lambda}$ for the parameter λ in Eq. (1) whenever the updating of both $\widehat{Q}_{kp}(t)$ and $\widehat{Q}_{kq}(t)$ are done, we can ensure that the monotone property is satisfied at every iteration. The parameter β, however, controls the size of the update. Using a value of β close to zero, results in small updates, while setting β close to unity, performs maximal updates without violating the monotone property. For the rest of this paper, we refer to this scheme as the Multiple DUMIQE (MDUMIQE), and mention in passing that the proof of convergence for this scheme and various other computational details are found in [3]. They are omitted here in the interest of brevity.

B. Tracking the Mean Value

The above schemes, DUMIQE and MDUMIQE, are computationally extremely light-weighted since the quantiles are tracked by only a single operation in every iteration, by resorting to Eq. (1). The natural analog when tracking the mean value, is the exponential moving average (EMA). To be more specific, let $\widehat{\mu}_k(t)$ denote the estimate of the mean value of class k at time t. In the EMA scheme we update the estimate as follows:

$$\widehat{\mu}_k(t+1) \leftarrow (1-\gamma)\widehat{\mu}_k(t) + \gamma\, x(t) \tag{4}$$

for some $\gamma \in [0,1]$.

V. Bayesian and AB Classification in Dynamical Data Streams

We now have the tools to perform classification in dynamic data streams. We first explain the methodology

[3] This approach can be seen as the AB counterpart of the Bayesian classification approach where the means of the classes are tracked by the exponential moving average.

for the Bayesian case and then proceed to the Anti-Bayesian paradigm.

A. Bayesian Classification

Bayesian classification is done in the manner explained earlier, and this has, indeed, been the basis for classification for decades. Here, in every iteration, the classification is based on the approach detailed in Section III-B. Every time we receive a new sample, we update that estimate of the mean value based on the class label, i.e., update $\widehat{\mu}_{c(s)}(s)$ as per Eq. (4):

$$\widehat{\mu}_{c(s)}(s+1) \leftarrow (1-\gamma)\widehat{\mu}_{c(s)}(s) + \gamma\, x(s),$$

for $s \leq t$. The estimates of the mean values for the other classes remain unchanged.

The reader should note that as per our model, we receive a sample $x(t+1)$ whose class is unknown. We then classify $x(t+1)$ to one of the K classes by using the Bayesian classification method described in Section III-B using the estimates of the mean values for each class at time t, namely $\widehat{\mu}_k(t)$, $k = 1, \ldots, K$. Whenever we receive the true class labels after a subsequent delay of h time steps, we follow the same procedure as described above, except that we also include the known class of *this* sample in the updated training step.

- Classify $x(t+1)$ to one of the K classes as per the Bayesian rule, and denote the result as $\widehat{c}(t+1)$.
- Update the estimate of the mean value for class $\widehat{c}(t+1)$, $\widehat{\mu}_{\widehat{c}(t+1)}(t)$ using Eq. (4).
- Classify $x(t+2)$ to one of the K classes using the estimates of the mean values at time $t+1$, namely $\widehat{\mu}_k(t+1)$, $k = 1, \ldots, K$.
- Update the estimate of the mean value of class $\widehat{c}(t+2)$ using Eq. (4).
- Repeat the above steps till time $t+h$.

B. AB Classification

To explain the AB classification, we assume that we have received samples with their respective class labels up to time t, $(x(1), c(1)), (x(2), c(2)), \ldots, (x(t), c(t))$. Every time we receive a new sample, we update that quantile estimates based on the class label $\widehat{Q}_{c(s)\,p}(s)$ and $\widehat{Q}_{c(s)\,q}(s)$ as per Eq. (1):

$$\begin{aligned}\widehat{Q}_{c(s)\,p}(s+1) &\leftarrow \widehat{Q}_{c(s)\,p}(s) + \lambda p \widehat{Q}_{c(s)\,p}(s), \\ &\qquad \text{if } x(s) > \widehat{Q}_{c(s)\,p}(s) \\ \widehat{Q}_{c(s)\,p}(s+1) &\leftarrow \widehat{Q}_{c(s)\,p}(s) - \lambda(1-p) \widehat{Q}_{c(s)\,p}(s), \\ &\qquad \text{if } x(s) \leq \widehat{Q}_{c(s)\,p}(s),\end{aligned}$$

$$\begin{aligned}\widehat{Q}_{c(s)\,q}(s+1) &\leftarrow \widehat{Q}_{c(s)\,q}(s) + \lambda q \widehat{Q}_{c(s)\,q}(s), \\ &\qquad \text{if } x(s) > \widehat{Q}_{c(s)\,q}(s) \\ \widehat{Q}_{c(s)\,q}(s+1) &\leftarrow \widehat{Q}_{c(s)\,q}(s) - \lambda(1-q) \widehat{Q}_{c(s)\,q}(s), \\ &\qquad \text{if } x(s) \leq \widehat{Q}_{c(s)\,q}(s),\end{aligned}$$

for $s \leq t$. For the MDUIQE scheme, we use $\widetilde{\lambda}$ from Eq. (3) in place of λ in the above updates. The quantile estimates for the other classes, $\widehat{Q}_{kp}(s)$ and $\widehat{Q}_{kq}(s)$ for $k \neq c(s)$, now remain unchanged.

Now suppose that we receive a new sample $x(t+1)$, whose class identity is unknown. We classify $x(t+1)$ to one of the K classes by using the AB classification method described in Section III-C using the quantile estimates for each class at time t, namely $\widehat{Q}_{kr}(t), r = p, q$, $k = 1, \ldots, K$.

We may also consider the case where we receive class labels with a delay of h time steps. In this sense, at time instant t we have samples $x(t+1), \ldots, x(t+h)$ with unknown class labels. To classify these samples, we use the following iterative procedure:

- Classify $x(t+1)$ to one of the K classes as described above, and denote the result $\widehat{c}(t+1)$.
- Update the quantile estimates of class $\widehat{c}(t+1)$, $\widehat{Q}_{\widehat{c}(t+1)\,r}(t)$, $r = p, q$ using DUMIQE/MDUMIQE.
- Classify $x(t+2)$ to one of the K classes using the quantile estimates from time $t+1$, namely $\widehat{Q}_{kr}(t+1), r = p, q$, $k = 1, \ldots, K$.
- Update the quantile estimates of class $\widehat{c}(t+2)$ using DUMIQE/MDUMIQE.
- Repeat the above steps till time $t+h$.

C. Robustness Against Outliers: AB Classification vs Bayesian Classification

We now emphasize an important property of AB classification which renders it to be superior to the Bayesian classification in dynamic environments. By virtue of the design of the quantile estimator, the AB approach is robust against outliers. This is a phenomenon that is absent in the Bayesian approach.

To clarify why this is true, we explain how the DUMIQE handles outliers. Although the magnitude of the observation is fed to the algorithm, only the fact whether the new observation is larger or smaller than the current quantile estimate is of significance. In other words, DUMIQE updates are based on the *sign* of the difference between the estimate and observation, while the EMA relies directly on the magnitude of the observations, to estimate the mean. It is thus clear that outliers might corrupt the mean estimate, while they will not have such a significant effect on the quantile estimates.

In Section VI we shall demonstrate the power of schemes for synthetic data sets.

VI. Experiments Results: Synthetic Data

We first compared the performance of the Bayesian and AB algorithms using synthetic data sets. The details of these sets and the results obtained are explained below.

A. Jump Processes

In this set of experiments, we assumed that the distribution for class k was normally distributed with expectation μ_k and standard deviation σ. We assumed a jump process with period T such that the expectations "jumped" by a value of b every half period. Formally, this is defined as:

$$\mu_k(t) = \begin{cases} ak & \text{if} \quad (t \bmod T) < T/2 \\ ak + b & \text{if} \quad (t \bmod T) > T/2, \end{cases}$$

for $k = 1, 2, \ldots, K$. Thus, the expectations for the different classes were separated by a difference a. In essence, we had the situation where $X(t)|C(t) = k \sim f_t(x|k) = N(\mu_k(t), \sigma)$ for $k = 1, 2, \ldots, K$. Finally, we assumed that $p_k(t) = \frac{1}{K}$ for $k = 1, 2, \ldots, K$ and all time steps.

In the first set of experiments, we fixed $\sigma = 1, a = 2, \sigma = 2, b = 4$ and $T = 100$. We also considered two cases where $K = 2$ and $K = 10$ classes. We then evaluated the classification performance of the AB approach using both DUMIQE and MDUMIQE to track the quantiles, and the Bayesian scheme using the EMA to track the distributions' mean values, as described in Section V. Figure 3 shows the portion of samples that were correctly classified when we were dealing with $K = 2$ classes using the three algorithms with λ being set to 0.01, and β and γ being set to 0.2.

From these results, we see that the classification performance varied periodically with a period equal to the period in the sample process, i.e., $T = 100$. The explanation is that it is easier for the schemes to predict when the delay is about one period compared to the scenario when it is only half a period. Another interesting observation is that for the Anti-Bayes multi, the classification performance was better for a delay of about 30 time steps compared to when the delay was smaller, which may seem surprising. We also observe the same phenomenon for the other two algorithms and for other choices of the tuning parameters. The explanation is that after a jump, the algorithms had to track the means and the quantiles, and it was still possible to do this in such a way that the classification improved even though we did not know the class labels of the received samples.

To demonstrate the performance of the three algorithms, we computed the portion of samples that were correctly classified when we averaged over all delays up to one period T, and used a large set of different choices for the three tuning parameters λ, β and γ. Figure 4 shows the results. For $K = 2$, the Bayesian approach performed a little better than the AB approach. The best classification was achieved using a value of γ around 0.7. For $K = 10$, the three algorithms performed about equally well, which is quite a fascinating results since the AB works in completely counter-intuitive manner.

B. Shrink/Expand Processes

In the second example, we investigated a process where the differences between the mean values shrank and expanded. We used the same setup and choice of parameters as above, but we used the following shrink/expand model for the variation of the mean values, described formally as:

$$\mu_k(t) = \begin{cases} a(k - K/2) & \text{if} \quad (t \bmod T) < T/2 \\ 2a(k - K/2) & \text{if} \quad (t \bmod T) > T/2, \end{cases}$$

for $k = 1, 2, \ldots, K$. The classification performances for the three algorithms for $K = 10$ classes are shown in left panel of Figure 5. We see that the Bayesian approach performed slightly better than the AB approach. Further, we see that the AB approach using MDUMIQE for tracking the quantiles performed better than when it used the DUMIQE.

VII. Conclusions

In this paper we have developed methods that apply the Bayesian and the recently-proposed Anti-Bayesian (AB) classification framework to perform online classifications for dynamic data streams. The classification of such dynamical data streams is among the most complex problems encountered in classification. This is, firstly, because the distribution of the data streams is non-stationary, and it changes without any prior "warning". Secondly, the manner in which it changes is also unknown. Thirdly, and more interestingly, we invoked the model with the assumption that the correct classes of previously-classified patterns become available at a juncture after their appearance. Apart from Bayesian methods, this paper pioneered the use of unreported novel schemes using AB techniques. Contrary to the Bayesian paradigm that compare the testing sample with the distribution's central points, AB techniques are based on the information in the distant-from-the-mean samples.

In this paper, the AB classification framework was based on estimating the time-varying quantiles of the distributions for the different classes. In this context, when performing AB classification for dynamic data streams, we tracked the quantiles using the DUMIQE and MDUMIQE methods developed in [3], [13]. By virtue of the design of the quantile estimator, the AB approach was shown to be more robust against outliers, which is a property absent in the Bayesian approach that tracks the mean. Both approaches were tested using synthetic data. The AB approaches performed very well, and in most cases outperformed the Bayesian analog both with respect to peak performance and the robustness with respect to the tuning parameters.

References

[1] J. Cao, L. E. Li, A. Chen, and T. Bu. Incremental tracking of multiple quantiles for network monitoring in cellular networks. In *Proceedings of the 1st ACM workshop on Mobile internet through cellular networks*, pages 7–12. ACM, 2009.

[2] F. Chen, D. Lambert, and J. C. Pinheiro. Incremental quantile estimation for massive tracking. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 516–522. ACM, 2000.

[3] H. L. Hammer and A. Yazidi. Efficient Tracking of Multiple Quantiles in Dynamic Data Streams. *Information Science (submitted)*, 2016.

[4] H.-L. Nguyen, Y.-K. Woon, and W.-K. Ng. A survey on data stream clustering and classification. *Knowledge and information systems*, 45(3):535–569, 2015.

[5] B. J. Oommen, R. Khoury, and A. Schmidt. Text classification using novel "anti-bayesian" techniques. In *Computational Collective Intelligence*, pages 1–15. Springer, 2015.

[6] B. J. Oommen and A. Thomas. "Anti–Bayesian" parametric pattern classification using order statistics criteria for some members of the exponential family. *Pattern Recognition*, 47(1):40–55, 2014.

[7] H. Tavasoli, B. J. Oommen, and A. Yazidi. On the online classification of data streams using weak estimators. In *Trends in Applied Knowledge-Based Systems and Data Science - 29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2016, Morioka, Japan, August 2-4, 2016, Proceedings*, pages 68–79, 2016.

[8] A. Thomas and B. J. Oommen. The fundamental theory of optimal "anti-bayesian" parametric pattern classification using order statistics criteria. *Pattern Recognition*, 46(1):376–388, 2013.

[9] A. Thomas and B. J. Oommen. A novel border identification algorithm based on an "anti-bayesian" paradigm. In *Computer Analysis of Images and Patterns*, pages 196–203. Springer, 2013.

[10] A. Thomas and B. J. Oommen. Order statistics-based parametric classification for multi-dimensional distributions. *Pattern Recognition*, 46(12):3472–3482, 2013.

[11] A. Thomas and B. J. Oommen. Ultimate order statistics-based prototype reduction schemes. In *Proceedings of AI'13, the 2013 Australasian Joint Conference on Artificial Intelligence*, pages 421–433. Springer, December 2013.

[12] L. Tierney. A space-efficient recursive procedure for estimating a quantile of an unknown distribution. *SIAM Journal on Scientific and Statistical Computing*, 4(4):706–711, 1983.

[13] A. Yazidi and H. L. Hammer. Multiplicative Update Methods for Incremental Quantile Estimation. *Information Science (submitted)*, 2016.

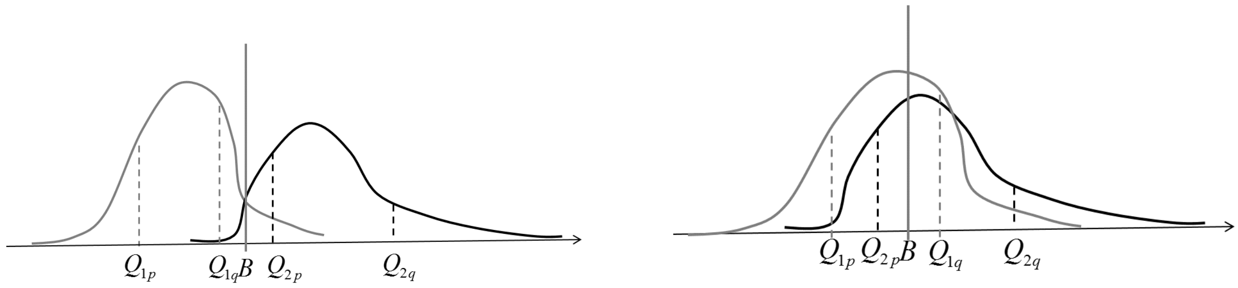

Fig. 2. The left panel shows the standard situation under Case 1, while the right panel shows a situation when $\widehat{Q}_{2p}$ is to the left of $\widehat{Q}_{1q}$.

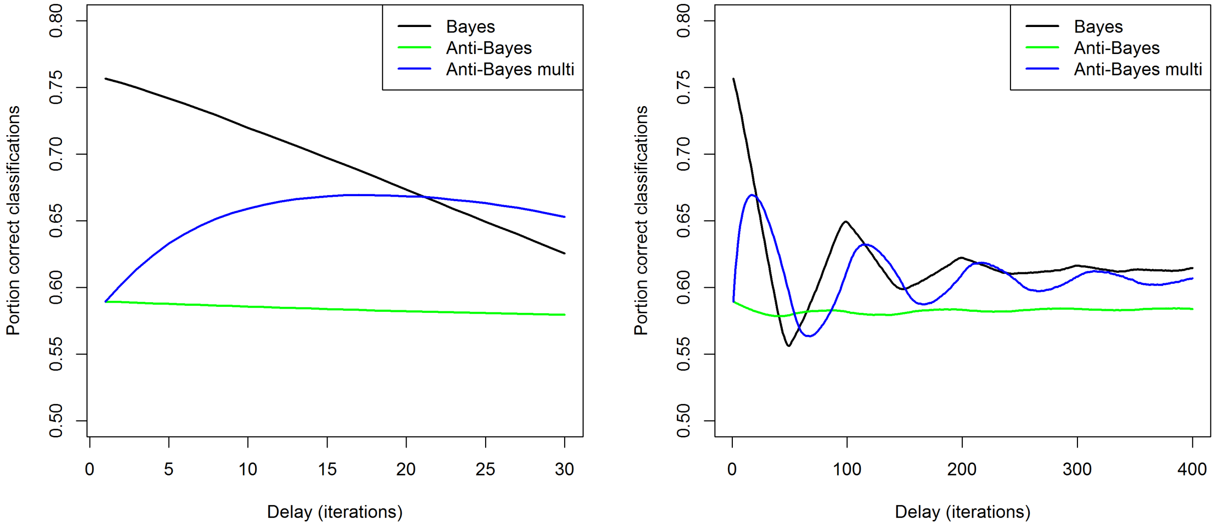

Fig. 3. Jump process: Portion of samples correctly classified for the Bayesian and AB algorithms as a function of the delay on the class label information. The right panel shows portions of correct classifications for all delays up to 400 time steps, while the left figure zooms-in on delays up to 30 time steps. ‘Anti-Bayes’ and ‘Anti-Bayes multi’ refer to the AB approaches that use the DUMIQE and MDUMIQE schemes respectively, to track the quantiles.

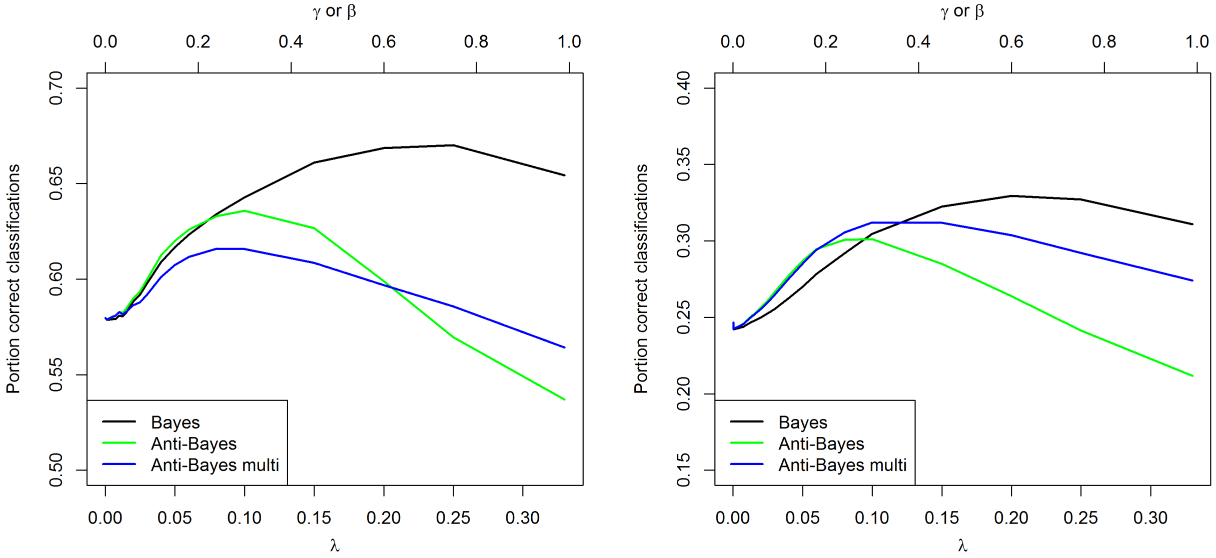

Fig. 4. Jump process: Portion of samples correctly classified for the Bayesian and AB algorithms for different choices of the tuning parameters. The left and the right panels display the cases with $K = 2$ and $K = 10$ classes, respectively. The ‘Anti-Bayes’ and ‘Anti-Bayes multi’ curves refer to the AB approaches using the DUMIQE and MDUMIQE respectively, to track the quantiles.

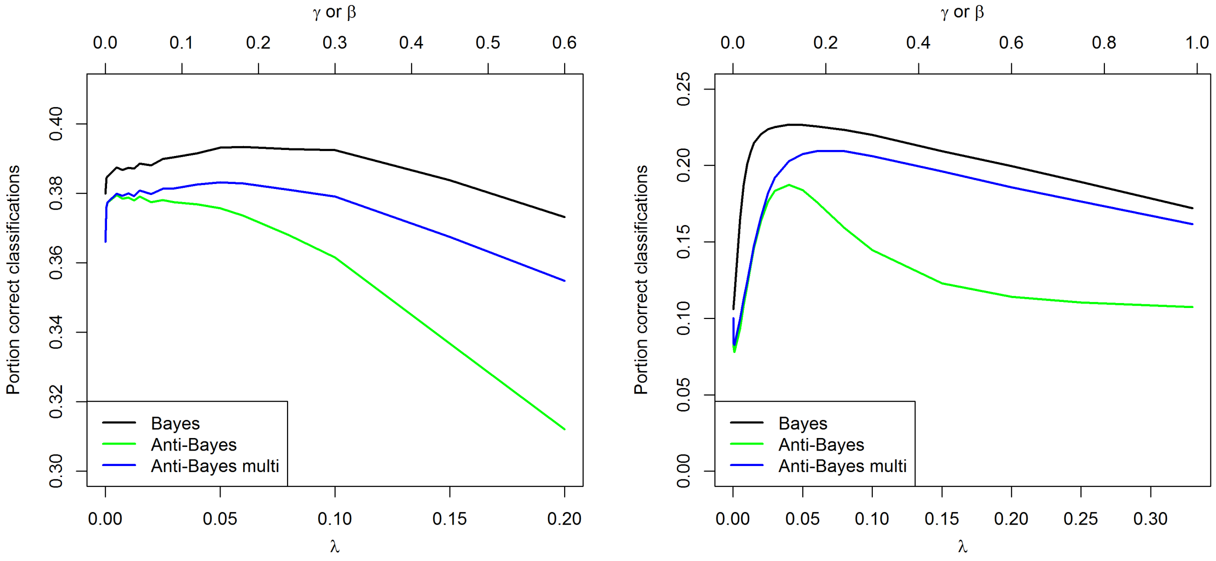

Fig. 5. Classification performance for $K = 10$ classes. The panels show the portion of samples correctly classified for the Bayesian and AB algorithms for different choices of the tuning parameters. The left and the right panels display the cases for the shrink/expand and the switch processes, respectively. ‘Anti-Bayes’ and ‘Anti-Bayes multi’ refer to the AB approaches using DUMIQE and MDUMIQE respectively, to track the quantiles.