

Fast Incremental Naïve Bayes with Kalman Filtering

Giacomo Ziffer
DEIB - Politecnico di Milano
Milano, Italy
giacomo.ziffer@mail.polimi.it

Emanuele Della Valle
DEIB - Politecnico di Milano
Milano, Italy
emanuele.dellavalle@polimi.it

Alessio Bernardo
DEIB - Politecnico di Milano
Milano, Italy
alessio.bernardo@polimi.it

Albert Bifet
University of Waikato, New Zealand
LTCI, Télécom ParisTech, France
abifet@waikato.ac.nz

Abstract—In recent years an increasing number of applications, IoT sensors and websites have produced endless streams of data. These data streams are not only unbounded, but their characteristics dynamically change over time, generating a phenomenon called concept drift. The standard machine learning models do not work properly in this context and new techniques have been developed in order to tackle these challenges. In this paper we present a new Naïve Bayes algorithm that exploits Kalman Filter, namely KALMANNB, to manage automatically concept drift. Furthermore, we want to investigate when this new approach, which directly follows the values of data’s attributes, is better than the standard strategy, which monitors the performance of the model in order to detect a drift. Extensive experiments on both artificial and real datasets with concept drifts reveal that KALMANNB is a valid alternative to the state-of-the-art algorithms, outperforming the latter especially in case of recurring concept drifts.

Index Terms—Naïve Bayes, Kalman Filter, Incremental Learning, Concept Drift Management, Recurring Drift

I. INTRODUCTION

We are living in a digital era in which we are surrounded by devices and applications that produce a high amount of data at a significant rate. Data are coming from many different sources, like online websites, social media platforms, hardware sensors (industrial equipment or Internet of Things devices) and transactions occurring both online and offline. Mining and analyzing these heterogeneous and boundless data is crucial for many applications (e.g. spam filtering systems, traffic control and intrusion detection).

The exponential growth of data brings with it even other challenges. Data streams are not only unbounded, but their characteristics can evolve over time, often in unforeseen ways. Consider, for example, a business that uses socio-economic factors to classify customers into different categories. Over the time the demographics of the customer base may change, leading to a shift in the probability of each demographic factor. In this situation the model must adapt its acquired knowledge in order to shape the new trend it is analyzing. This phenomenon is known as concept drift and depending on the various rates at which it occurs, it can be gradual or abrupt. To

solve this problem several solutions have been developed, most of which combine a concept drift detector with an incremental approach. The former monitors the performance of the latter and when it gets drastically worse, the model is adjusted so that only the data after the concept drift are learned. The algorithm can thus rapidly recover when drifts happen.

Moreover, in real-world data streams, the environment is subject to variations which unpredictably repeat over time, leading to a type of concept drift known as recurring drift. Unfortunately, only few approaches take into consideration this issue, despite the fact it can happen in a lot of different situations. For example, it often occurs in fashion that previously seen concepts reappear after some time. Similarly, the interests of a user may evolve over time, since he can change preferences or rediscover old passions. As a consequence, apparently obsolete data or learned concepts could be relevant again in the future.

Algorithms able to handle concept drift are distinguished in *blind* and *informed* approaches, accordingly to how they manage the concept drift. The former manage the concept drift in an implicit way, while the latter manage it in an explicit way. To clarify this distinction, our work aims to investigate the following **research questions**: *which are the advantages of using a blind approach, rather than an informed approach with an external concept drift detector? In which situations the former performs better than the latter?*

The main contribution of this study is KALMANNB, a novel Naïve Bayes method able to automatically manage concept drift thanks to the use of the Kalman Filter. Furthermore, we evaluate our solution against both artificial and real-world datasets with gradual, abrupt and recurring concept drifts and we show when it outperforms the state-of-the-art models, highlighting the best context for each algorithm.

The remainder of this paper is organized as follows. In Section II we present some existing techniques on top of which we build our contribution and experiments. In Section III we motivate and explain the proposed method. Section IV describes the models and the artificial and real-world datasets

used in our experiments and shows the evaluation results. Section V discusses the conclusions we reached based on the experiments and outlines directions for future research.

II. RELATED WORK

In the last decade different techniques have been developed to handle the concept drift phenomenon. Instead of keeping static models, these algorithms can learn in an incremental fashion, by updating their models whenever there is a new sample, and without the need to recreate them from scratch using all the data. As pointed out in [1], we can identify two main categories of adaptation strategy: blind or informed. Blind approaches consist in algorithms that automatically manage the concept drift without any explicit detection. A straightforward example of this category is the set of algorithms that forget the older data regardless of whether there is a concept drift or not (e.g. fixed-size window algorithm or gradual forgetting mechanisms with a fading factor). On the contrary, informed adaptation strategies combine an incremental algorithm with a concept drift detector, which notifies the model about the presence of the concept drift. The underlying idea is to monitor a specific performance indicator of the algorithm, such as accuracy or error rate, and when it gets drastically worse, the model is adjusted so that only the information after the concept drift is learned. The algorithm can therefore rapidly recover when drifts happen. Some of the most used concept drift detectors are ADWIN, DDM and EDDM.

The Adaptive Windowing (ADWIN) [2] is an approach that exploits an adaptive size sliding window in order to detect changes in data streams. Whenever two different sub-windows are distinct enough, it assumes the presence of the concept drift and it starts removing the oldest element from the window. The Drift Detection Method (DDM) [3] is based on statistical control process. It assumes the probability distribution of the error-rate to be normal, and it uses two different confidence levels to fire respectively the warning and the out-of-control alarm. The Early Drift Detection Method (EDDM) [4] has been developed as an improvement of the DDM, aiming to tackle also the gradual drift. Opposed to the latter that monitors the error-rate, EDDM keeps track of the average distance between two errors.

Finally, we introduce one of the most common estimation methods in automatic control theory, the Kalman Filter [5]. It is an efficient recursive filter that evaluates the state of a dynamic system from a series of measurements and provides a prediction of the future state, based on past estimations. It considers the state as a linear combination of its previous value with a control signal and a process noise with variance Q , and each measurement as a linear combination of the signal value and the measurement noise with variance R .

The concept drift detectors previously introduced look for a change in the classification error of the model. The main difference in applying Kalman Filter instead of those concept drift detectors is that the former directly follows the data distribution and adapts automatically to its changes, by

Algorithm 1 KALMANNB training update pseudo code

```

1: Input: data stream  $D$ 
2: Initialize model
3: for each instance  $X, y$  in  $D$  do
4:   for each attribute  $x_i$  in  $X$  do
5:     for each attribute's value  $v_j$  in model do
6:       if  $v_j = x_i \wedge y_k = y$  then
7:         Kalman Filter  $KF_{i,j,k} \leftarrow$  state update(1)
8:       else if  $v_j \neq x_i \wedge y_k = y$  then
9:         Kalman Filter  $KF_{i,j,k} \leftarrow$  state update(0)
10:      end if
11:    end for
12:  end for
13: end for

```

Algorithm 2 Kalman Filter $KF_{i,j,k}$ state update pseudo code with Q and R predefined values

```

1: Input: value  $v$ 
2:  $x_0 \leftarrow KF_{i,j,k}$  state
3:  $P_0 \leftarrow KF_{i,j,k}$  variance
4:  $K \leftarrow P_0 / (P_0 + R)$ 
5:  $x_1 \leftarrow x_0 + K * (v - x_0)$ 
6:  $P_1 \leftarrow P_0 * (1 - K) + Q$ 
7:  $KF_{i,j,k}$  state  $\leftarrow x_1$ 
8:  $KF_{i,j,k}$  variance  $\leftarrow P_1$ 
9: End

```

comparing the uncertainty of the current state and the one of the new measurement. This distinction plays an important role in determining the different situations each approach can handle better, which is also one of the objective of this work.

The advantages of Kalman Filter's application in the context of concept drift management were firstly investigated in [6]. Bifet and Gavaldà proposed different approaches exploiting the CUSUM test, the Kalman Filter and ADWIN. Furthermore, they suggested the combination of the Kalman Filter with ADWIN, named K-ADWIN, with the objective of exploiting both methods' strengths. Although the obtained results were interesting, the versatility of ADWIN made the latter prevail over Kalman Filter models in later developments.

III. PROPOSED METHOD

There is always a constant need for adaptive models that can be applied in time-changing streaming contexts. As pointed out in [7], prior research has thoroughly investigated rule based and tree based streaming data classification algorithms. On the contrary, only few methods exploit statistical approaches.

Our proposal is called KALMANNB and consists in an incremental Naïve Bayes algorithm for categorical feature that exploits Kalman Filter to handle concept drift in data streams without the help of a concept drift detector.

Naïve Bayes classifier (NB) is one of the simplest machine learning models, yet it performs surprisingly well in many situations, outperforming even more complex algorithms. Being a fast and light approach, it is widely used in many

real-time problems, like spam or news filtering. Especially in those applications, concept drifts occur frequently, increasing the need for fast, simple and robust solutions. NB classifier is a “probabilistic classifier”, which reformulates the prediction problem by using the Bayes’ theorem, in order to calculate the probability. Finally, NB combines this probabilistic model with a decision rule, called Maximum A Posteriori (1), to obtain a classifier. Upon receiving an unlabelled sample X , it computes the conditional probability for each class and predicts the one whose value is the maximum among those calculated.

$$y = \operatorname{argmax}_{k \in \{1, \dots, K\}} \frac{p(y_k)}{p(X)} \prod_{i=1}^n p(x_i | y_k) \quad (1)$$

In case of discrete attributes, which is the condition analyzed in this work, each probability is computed as the ratio between two different frequencies. Indeed, $p(x_i | y_k)$ can be seen as the number of instances with attribute’s value x_i and class y_k divided by the total number of training instances with class y_k . Assuming that each attribute x_i can take up to j different values, the Naïve Bayes classifier has just to store a counter $N_{i,j,k}$ for each attribute i , each attribute’s value j and each available class k . This representation is naturally incremental, since the algorithm needs only to update the respective counter for each new incoming instance.

KALMANNB extends the model described above, it incorporates the concept drift management by replacing each counter $N_{i,j,k}$ with a Kalman Filter $KF_{i,j,k}$ as the estimator of the specific frequency. Instead of having just the counter, KALMANNB keeps a tuple (x, P) which constitutes the state of the Kalman Filter. x represents the estimated frequency, while P the variance of the process (i.e., a measure of the approximated accuracy of the estimation).

As shown in Alg. 1, for each incoming labelled sample (X, y) , the update involves all the estimators of the class y . It inserts 1 in the state update of the Kalman Filter where $x_i = v_j \wedge y_k = y$ (Lines 6-7) and 0 in the state update of all the other estimators, where $x_i \neq v_j \wedge y_k = y$ (Lines 8-9). Alg. 2 shows the update of the state (x, P) of a Kalman Filter $KF_{i,j,k}$. In particular, the parameter *value* is 1 or 0 depending on the conditions explained above for Alg. 1. Concerning the parameter Q and R , they are predefined and assumed to be fixed. Nevertheless, they can also be set dynamically by exploiting for instance the width of ADWIN’s window, as pointed out in [6]. The state update shown in Alg. 2 computes the Kalman gain K (Line 4), which balances the variance of the previous state P_0 with Q . Then, it updates its frequency estimation using K with a convex combination of the previous state x_0 and the new value v (Lines 5 and 7). Finally, it updates P using both K and Q (Lines 6 and 8).

Thanks to this approach, KALMANNB is able to include directly the Kalman Filter in its structure, without compromising the general functioning of the model. Indeed, this new algorithm maintains the simplicity of the Naïve Bayes approach and also its speed (which is fundamental in real-time streaming contexts), with the addition of the ability to manage automatically the concept drift.

IV. EXPERIMENTAL EVALUATION

In the experiments we compare the behaviours of the following algorithms:

- KALMANNB with $Q = 1$ and $R = 1000$
- KALMANNB with $Q = 10$ and $R = 2000$
- ADWINBUFFER with $size = W$ ($max = 1000$), where W is the ADWIN window’s width
- K-ADWIN with $R = W^2/50$ and $Q = 200/W$, where W is the ADWIN window’s width
- EXTERNAL-ADWIN
- EXTERNAL-DDM
- EXTERNAL-EDDM

Concerning the KALMANNB models, the chosen values are those that obtained the best results among the different models tested with fixed parameters. The initial state of each Kalman Filter in the model is $x_0 = 0$, $P_0 = 0$. ADWINBUFFER is an approach that implements a buffer, i.e. a sliding window with variable size, to store the last received samples, and adjusts its size to the ADWIN window’s width in order to handle the concept drift. K-ADWIN is a combination of the Kalman Filter and ADWIN, as proposed in [6]. Finally, EXTERNAL-ADWIN, EXTERNAL-DDM and EXTERNAL-EDDM all implement the structure taken from the *DriftDetectionMethodClassifier* available in MOA¹. EXTERNAL-ADWIN exploits a modified version of ADWIN, which enables it to signal the warning level. All these three variants use the standard incremental version of the Naïve Bayes as the BaseClassifier. Note that with the exception of the two KALMANNB models, which are blind approaches, all the other algorithms are informed approaches.

Both artificial and real datasets with concept drift are used to test the algorithms. The evaluation methodology chosen is the prequential mode [8], where each incoming sample is firstly used to test the model and then to train it. This methodology is the most suitable in context with concept drift.

Concerning the artificial datasets, two stream generators with categorical features taken from scikit-multiflow² are used:

- LED StreamGenerator, with $noise = 0.1$ and 24 *drifted features*, that simulates a gradual drift (5000 samples generated).
- STAGGER StreamGenerator, combined with the ConceptDriftStream to artificially injects either gradual, abrupt or recurring concept drift (2000 samples generated). Drift is generated by changing the classification function. Gradual and abrupt drifts are simulated inserting the drift after 1000 samples, with respectively a *width* of 500 and 1; while in the recurring drift change occurs every 500 samples (i.e. 3 drifts in total).

Real datasets represent spam and news filtering problems with concept drift. In particular, Usenet datasets simulate a news filtering system with the presence of recurring concept

¹<https://moa.cms.waikato.ac.nz/>

²<https://scikit-multiflow.github.io>

TABLE I
ACCURACY EVALUATED IN PREQUENTIAL MODE WITH ARTIFICIAL AND REAL DATASETS

Data Sets	Algorithms							
	STANDARD NB	KALMANNB $Q=1 R=1000$	KALMANNB $Q=10 R=2000$	ADWIN BUFFER	K-ADWIN	EXTERNAL ADWIN	EXTERNAL DDM	EXTERNAL EDDM
LED <i>gradual</i>	75.64 ± 0.60	69.95 ± 0.65	68.63 ± 0.91	75.36 ± 0.50	61.26 ± 6.38	75.64 ± 0.60	75.64 ± 0.60	75.49 ± 0.62
STAGGER <i>gradual</i>	79.79 ± 0.61	93.97 ± 0.46	93.07 ± 0.72	88.59 ± 0.79	88.75 ± 1.01	93.69 ± 0.49	92.22 ± 1.81	93.38 ± 0.45
STAGGER <i>abrupt</i>	80.07 ± 0.60	97.58 ± 0.17	98.41 ± 0.17	95.91 ± 0.66	95.41 ± 0.26	97.62 ± 0.33	99.27 ± 0.13	98.24 ± 0.08
STAGGER <i>recurring</i>	72.21 ± 4.55	94.38 ± 0.83	96.85 ± 0.38	85.76 ± 1.15	88.36 ± 0.54	90.94 ± 0.70	92.53 ± 2.15	95.12 ± 0.13
USENET <i>recurring</i>	53.73	62.12 (3)	63.56 (1)	60.30 (5)	55.60 (7)	56.10 (6)	62.51 (2)	60.35 (4)
USENET1 <i>recurring</i>	63.31	74.52 (2)	77.79 (1)	64.24 (6)	68.91 (5)	62.51 (7)	70.18 (3)	69.91 (4)
USENET2 <i>recurring</i>	72.11	75.78 (2)	76.12 (1)	71.98 (6)	73.27 (5)	71.85 (7)	74.25 (4)	75.18 (3)
SPAM <i>gradual</i>	90.59	94.05 (2)	94.85 (1)	93.07 (3)	91.87 (4)	90.52 (6)	91.11 (5)	89.96 (7)
ELIST <i>abrupt</i>	83.70	74.39 (5)	69.95 (7)	71.29 (6)	77.84 (4)	83.70 (1)	83.70 (1)	83.23 (3)
Rank		2.8	2.2	5.2	4.6	5.4	3	4.2

TABLE II
MODEL COMPLEXITY (KB) EVALUATED IN PREQUENTIAL MODE WITH ARTIFICIAL DATASETS

Data Sets	Algorithms							
	STANDARD NB	KALMANNB $Q=1 R=1000$	KALMANNB $Q=10 R=2000$	ADWIN BUFFER	K-ADWIN	EXTERNAL ADWIN	EXTERNAL DDM	EXTERNAL EDDM
LED <i>gradual</i>	94.29 ± 0.03	118.58 ± 0.13	118.59 ± 0.13	815.84 ± 2.95	133.72 ± 1.66	106.01 ± 0.04	105.7 ± 22.8	142.6 ± 37.7
STAGGER <i>gradual</i>	5.29 ± 0.00	5.70 ± 0.01	5.70 ± 0.01	204.14 ± 2.96	16.05 ± 0.03	16.75 ± 0.05	10.39 ± 1.25	9.42 ± 1.66
STAGGER <i>abrupt</i>	5.27 ± 0.00	5.64 ± 0.00	5.66 ± 0.00	192.21 ± 0.53	15.97 ± 0.01	16.62 ± 0.03	7.97 ± 0.15	7.91 ± 0.00
STAGGER <i>recurring</i>	5.22 ± 0.00	5.72 ± 0.00	5.73 ± 0.01	145.2 ± 15.1	15.16 ± 0.08	15.46 ± 0.08	7.88 ± 0.10	7.80 ± 0.01

drifts related to the change of interest of a user over time, while SPAM represents gradual concept drift and it is based on the Spam Assassin Collection [9]. Usenet contains 5,931 instances and it is available here³. Usenet1 and Usenet2 datasets are both introduced in [10] and they contain 100 features and 1500 samples each with different behaviours over time. SPAM dataset consists of 9,324 instances with 500 attributes, while Elist has 6,000 samples with 28,000 features. Except for Usenet, all the other real datasets can be obtained here⁴.

Tab. I shows the accuracy calculated in prequential mode for each dataset described above. Note that with artificial data we performed 10 runs for each setting, therefore both mean and standard deviation are shown, in order to analyze how stable each model is. Since in real datasets there is no randomness to

take into account, we performed only one run for each dataset. Hence, tests on real data show only one value per algorithm, corresponding to the evaluated metric.

The results of Tab. I can be summarized as follows. In case of recurring drifts, KALMANNB proves to be the best approach with both artificial and real datasets. It outperforms all the other tested models, obtaining an improvement in the accuracy ranging from 1% to 7%. At the same time, with artificial data it performs well also in presence of gradual or abrupt concept drifts, where its performances are close to those of ADWIN, DDM and EDDM. In all the tests with real-world datasets, KALMANNB performs well: except for ELIST, KALMANNB with $Q=10$, $R=2000$ scores always the best accuracy and it results in the best approach. Moreover, as shown in Tab. I, both KALMANNB models obtain overall the best average ranking on real-world datasets. It is worth

³<http://www.liaad.up.pt/kdus/products/datasets-for-concept-drift>

⁴http://mlkd.csd.auth.gr/concept_drift.html

TABLE III
TIME (S) EVALUATED IN PREQUENTIAL MODE WITH ARTIFICIAL DATASETS

Data Sets	Algorithms							
	STANDARD NB	KALMANNB $Q=1$ $R=1000$	KALMANNB $Q=10$ $R=2000$	ADWIN BUFFER	K-ADWIN	EXTERNAL ADWIN	EXTERNAL DDM	EXTERNAL EDDM
LED <i>gradual</i>	4.16 ± 0.02	2.48 ± 0.02	2.32 ± 0.02	143.23 ± 0.56	6.04 ± 0.46	8.31 ± 0.04	7.95 ± 0.04	7.75 ± 0.04
STAGGER <i>gradual</i>	0.59 ± 0.10	0.53 ± 0.12	0.32 ± 0.02	8.83 ± 1.61	1.87 ± 0.31	1.63 ± 0.30	1.17 ± 0.23	0.93 ± 0.16
STAGGER <i>abrupt</i>	0.45 ± 0.01	0.39 ± 0.01	0.31 ± 0.02	6.58 ± 0.22	1.44 ± 0.05	1.27 ± 0.06	0.90 ± 0.03	0.73 ± 0.02
STAGGER <i>recurring</i>	0.47 ± 0.03	0.42 ± 0.02	0.34 ± 0.01	6.77 ± 0.41	1.58 ± 0.07	1.30 ± 0.07	0.92 ± 0.05	0.76 ± 0.03

pointing out that the only two cases in which KALMANNB is performing significantly worse than the other models, are situations in which the best models are STANDARD NB, EXTERNAL ADWIN and EXTERNAL DDM. In these situations they have exactly the same performances, meaning that the two algorithms with ADWIN and DDM are not able to detect any concept drifts. Finally, ADWIN BUFFER and K-ADWIN do not show any significant results in these analyses. They always achieve average performances without outperforming the other models.

Tab. II and Tab. III report the evaluation of two other important metrics: the *model complexity* and the *speed* of the algorithms. In real-time streaming contexts, strict time and memory constraints are frequent, thus they must be taken into account in the assessment. The evaluation has been made only with artificial data since the results are significant enough to highlight the behaviour of each model. The settings are the same of the first evaluation. As for the accuracy, the model complexity’s test with LED data shows unusual results, indicating again a time-changing concept not caught by the algorithms. Furthermore, by analyzing both model complexity and time, ADWIN BUFFER becomes unfeasible due to the amount of memory it requires and its slowness.

Concerning KALMANNB, the most important conclusion we can draw is that it appears to be the least complex and the fastest model among the approaches that handle the concept drift. Indeed, it maintains the same complexity of STANDARD NB, outperforming also its speed. Furthermore, compared to the other models that change their structure depending on the context, it has a fixed complexity that does not depend on whether there is or not a concept drift. These results make KALMANNB the best model in contexts where there is low available storage or data arrive at a high rate, so the algorithm cannot take much time to update its model. The reason behind the differences in both model complexity and time with respect to the other algorithms lies in their model’s inner structure. Indeed, they are all informed approaches, since they exploit a concept drift detector. Despite the versatility, this approach adds complexity to the model, slowing down the entire update process. On the other hand, a blind solution like

KALMANNB does not affect the general structure, keeping it simple and fast.

To deepen the differences of the various approaches we performed further evaluations, aiming at assessing the *recovery ability* of each method. With this analysis we want to highlight the capacity of each algorithm to recover in terms of performance metric (e.g. accuracy score) after a concept drift occurs. Only data generated by the STAGGER StreamGenerator are used in the evaluation, since it is possible to inject different types of concept drift depending on the width used. Each test is performed with a data stream of 1000 samples, split in batches of 20 samples each (50 batches in total). The concept drift is inserted at the 40th batch, therefore the algorithms do not have too many samples to adapt to the new concept. As a consequence, the measured accuracy is affected by the algorithm’s ability to recover quickly after the concept drift. Ten runs have been performed for each test and results show their means and standard deviations. The column *width* indicates the number of samples present in the transition phase between the two concepts. Both gradual and abrupt concept drifts are simulated by using different widths.

Three metrics are evaluated in order to assess the recovery ability of each model: *accuracy* (Tab. IV), *low peak* (Tab. V), and *number of iterations for recovery* (Tab. VI). In particular, low peak represents the least accuracy score obtained by the model after the concept drift (i.e. between batch 40th and 50th). The third metric indicates the number of iterations needed by each algorithm to achieve competitive performances after the concept drift (i.e. at least 95% of the accuracy obtained by the algorithm before the drift). For instance, we can observe that KALMANNB with $Q=1$, $R=1000$ takes in average 6.20 iterations to recover from an abrupt concept drift with $width = 1$.

By looking at Tab. IV, Tab. V, and Tab. VI we can acknowledge that, with abrupt concept drift, the best model is EXTERNAL DDM for all the evaluated metrics. On the other hand, with a more gradual drift, the best algorithm is KALMANNB. It is also worth pointing out that, opposed to EXTERNAL DDM, the performances of our solution are not highly affected in passing from abrupt to gradual concept drift.

TABLE IV
ACCURACY EVALUATED IN PREQUENTIAL MODE WITH STAGGER DATASET FOR THE RECOVERY SPEED ANALYSIS

Data Sets	Width	Algorithms						
		KALMANNB $Q=1 R=1000$	KALMANNB $Q=10 R=2000$	ADWIN BUFFER	K-ADWIN	EXTERNAL ADWIN	EXTERNAL DDM	EXTERNAL EDDM
STAGGER <i>abrupt</i>	1	95.09 ± 0.35	97.04 ± 0.30	91.85 ± 0.75	91.60 ± 0.88	94.78 ± 0.86	98.35 ± 0.31	96.44 ± 0.32
STAGGER <i>gradual</i>	100	95.16 ± 0.56	96.37 ± 0.87	89.92 ± 0.46	89.78 ± 0.51	93.73 ± 0.59	95.36 ± 1.07	95.84 ± 0.59
STAGGER <i>gradual</i>	400	90.92 ± 0.70	90.25 ± 0.67	88.79 ± 0.64	87.97 ± 0.83	89.89 ± 0.50	89.84 ± 0.62	90.53 ± 0.69

TABLE V
LOW PEAK OF THE ACCURACY EVALUATED IN PREQUENTIAL MODE WITH STAGGER DATASET FOR THE RECOVERY SPEED ANALYSIS

Data Sets	Width	Algorithms						
		KALMANNB $Q=1 R=1000$	KALMANNB $Q=10 R=2000$	ADWIN BUFFER	K-ADWIN	EXTERNAL ADWIN	EXTERNAL DDM	EXTERNAL EDDM
STAGGER <i>abrupt</i>	1	41.00 ± 6.99	42.50 ± 5.40	38.00 ± 4.22	37.00 ± 3.50	39.00 ± 4.59	43.00 ± 8.56	40.00 ± 5.77
STAGGER <i>gradual</i>	100	52.00 ± 8.88	59.50 ± 9.92	39.00 ± 5.16	36.50 ± 5.30	41.50 ± 4.12	55.00 ± 9.72	47.50 ± 6.35
STAGGER <i>gradual</i>	400	59.50 ± 7.25	57.00 ± 4.22	53.50 ± 3.37	51.00 ± 3.94	53.50 ± 5.30	56.50 ± 7.84	55.00 ± 9.72

TABLE VI
NUMBER OF ITERATIONS FOR RECOVERY EVALUATED WITH STAGGER DATASET FOR THE RECOVERY SPEED ANALYSIS

Data Sets	Width	Algorithms						
		KALMANNB $Q=1 R=1000$	KALMANNB $Q=10 R=2000$	ADWIN BUFFER	K-ADWIN	EXTERNAL ADWIN	EXTERNAL DDM	EXTERNAL EDDM
STAGGER <i>abrupt</i>	1	6.20 ± 0.42	3.30 ± 0.48	11.00 ± 3.30	9.10 ± 1.10	5.00 ± 1.15	1.70 ± 0.67	3.50 ± 0.53
STAGGER <i>gradual</i>	100	6.00 ± 0.94	4.20 ± 0.42	14.80 ± 2.15	11.50 ± 1.58	6.00 ± 0.82	5.80 ± 1.62	4.50 ± 0.97
STAGGER <i>gradual</i>	400	10.80 ± 2.25	9.20 ± 3.29	19.30 ± 1.16	22.00 ± 4.55	10.30 ± 2.58	17.40 ± 5.95	11.30 ± 3.92

These results show that the new proposed approach is able to recover faster, in terms of samples needed, than the other compared methods. Therefore, we can empirically motivate why it is performing better in case of recurring drift. Indeed, being less affected by the concept drift, its overall performances are better.

V. CONCLUSION

We have presented KALMANNB, a novel Naïve Bayes model able to manage automatically the concept drift in data streams with categorical features by exploiting the Kalman Filter. We have evaluated it against some synthetically generated and real-world datasets in the context of news and spam filtering, using prequential testing mode. All the results empirically prove that KALMANNB is a valid alternative compared to the approaches tested in this work, especially in case of recurring drift and with strict time and memory requirements. We also investigated the advantages of using a

blind methodology rather than an informed one, highlighting the situations in which it is better to give up the versatility of the informed model for a better, simpler and faster approach, like KALMANNB. Finally, we tested the ability to recover after a concept drift, and our solution proved to be the best one, requiring in average less samples to obtain competitive performances after a concept drift. Therefore, the proposed algorithm is a valid solution to learn with fast time-changing data streams.

In case of abrupt concept drift, KALMANNB achieved worse results compared to the other algorithms. As pointed out by the tests, distinct couples of Q and R values can generate KALMANNB models with different performances depending on the type of concept drift. Thus, to extend this approach, we want to investigate the benefits of using an ensemble method of multiple KALMANNB with different Q and R values. This new model could be able to overcome the limitation mentioned above by combining the strengths of each model built.

Moreover, the second main disadvantage of this approach is its applicability only to categorical data. In this case, indeed, we need to estimate one dimension (i.e. the frequency), while with numerical features we would need two different quantities (i.e. the mean and the variance), in order to approximate the data with a normal distribution. Hence, the next step of this research will be to extend KALMANNB to the other contexts, aiming at making it a more robust model.

As a final point, future studies should also analyze the improvements resulting from the combination of KALMANNB with other algorithms. In this perspective, we want to investigate the advantages of using KALMANNB at the leaves of the Hoeffding Tree and the Adaptive Random Forest models and to compare this approach with the current versions of the Hoeffding Adaptive Tree and Adaptive Random Forest algorithms that use the standard Naïve Bayes method at their leaves.

ACKNOWLEDGMENT

We would like to thank Dr. Malik Souded and the **Innovation Lab of ALTEN SA** located at Sophia Antipolis (Provence-Alpes-Côte d’Azur, France) for creating the conditions to work on this topic and for the support during the development of the project.

REFERENCES

- [1] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM Comput. Surv.*, vol. 46, no. 4, pp. 44:1–44:37, 2014.
- [2] A. Bifet and R. Gavaldà, “Learning from time-changing data with adaptive windowing,” in *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA*. SIAM, 2007, pp. 443–448.
- [3] J. Gama, P. Medas, G. Castillo, and P. P. Rodrigues, “Learning with drift detection,” in *Advances in Artificial Intelligence - SBIA 2004, 17th Brazilian Symposium on Artificial Intelligence, São Luis, Maranhão, Brazil, September 29 - October 1, 2004, Proceedings*, ser. Lecture Notes in Computer Science, A. L. C. Bazzan and S. Labidi, Eds., vol. 3171. Springer, 2004, pp. 286–295.
- [4] M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, “Early drift detection method,” in *Fourth international workshop on knowledge discovery from data streams*, vol. 6, 2006, pp. 77–86.
- [5] G. Welch, G. Bishop *et al.*, “An introduction to the kalman filter,” 1995.
- [6] A. Bifet and R. Gavaldà, “Kalman filters and adaptive windows for learning in data streams,” in *Discovery Science, 9th International Conference, DS 2006, Barcelona, Spain, October 7-10, 2006, Proceedings*, ser. Lecture Notes in Computer Science, L. Todorovski, N. Lavrac, and K. P. Jantke, Eds., vol. 4265. Springer, 2006, pp. 29–40.
- [7] S. Mehta *et al.*, “Concept drift in streaming data classification: Algorithms, platforms and issues,” *Procedia computer science*, vol. 122, pp. 804–811, 2017.
- [8] A. Bifet, G. D. F. Morales, J. Read, G. Holmes, and B. Pfahringer, “Efficient online evaluation of big data stream classifiers,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, L. Cao, C. Zhang, T. Joachims, G. I. Webb, D. D. Margineantu, and G. Williams, Eds. ACM, 2015, pp. 59–68.
- [9] I. Katakis, G. Tsoumakas, and I. P. Vlahavas, “Tracking recurring contexts using ensemble classifiers: an application to email filtering,” *Knowl. Inf. Syst.*, vol. 22, no. 3, pp. 371–391, 2010.
- [10] I. Katakis, G. Tsoumakas, and I. P. Vlahavas, “An ensemble of classifiers for coping with recurring contexts in data streams,” in *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*, ser. Frontiers in Artificial Intelligence and Applications, M. Ghallab, C. D. Spyropoulos, N. Fakotakis, and N. M. Avouris, Eds., vol. 178. IOS Press, 2008, pp. 763–764.