Sentiment Classification over Opinionated Data Streams through Informed Model Adaptation

Vasileios Iosifidis[1], Annina Oelschlager[2] and Eirini Ntoutsi[1]

[1] Leibniz University Hannover & L3S Research Center, Hannover, Germany
{iosifidis, ntoutsi}@l3s.de
[2] Ludwig-Maximilians-University, Munich, Germany
annina.oelschlaeger@gmail.com

Abstract. Opinionated data streams are very popular data paradigms nowadays as more and more users share their opinions online about almost everything from products to persons, brands and ideas. One of the key challenges for opinionated stream mining is dealing with concept drifts in the underlying stream population by building learners that adapt to such concept changes. Ageing is a typical way of adapting to change in a stream environment as it potentially allows us to discard outdated information from the learning models and focus on the most recent information. Most of the existing approaches follow a fixed ageing strategy which remains the same over the whole stream; for example, a fixed window size in the sliding window model or a fixed ageing factor in the damped window model. This implies that we forget at the same rate over the whole course of the stream, which is counterintuitive given the volatile nature of the stream. What is more intuitive is to forget faster in times of change so as to adapt to new data and to forget slower, or in other words, to remember more, in times of stability. In this work, we propose an informative-adaptation-to-change approach where we first detect changes in the underlying data stream and then we tune the ageing factor of the ageing-based Multinomial Naive Bayes (MNB) classifier based on the detected change. Except for the up-to-date classifier our method also outputs the points of change in the stream, therefore offering more insights to the final users.

1 Introduction

A huge amount of opinions is available nowadays, as a result of the widespread usage of the social media and the Web. Opinions are valuable for consumers, who benefit from the experiences of others, in order to make better buying decisions [13] but also for vendors, who can get insights on what customers like or dislike [16]. Such sort of data are freely available, however due to their amount and complexity a proper analysis is required in order to gain insights.

Opinions are accumulated over time, building what we call opinionated streams, i.e., streams of documents which convey sentiment. The accumulating opinionated documents are subject to different forms of drift: the topics discussed in the stream change, the attitude of people towards certain topics might change, words used to describe topics or sentiment might change and so on and so forth.

In this work, we address the issue of polarity learning over opinionated streams. That is, we want to build classifiers that can cope with the volatile nature of the stream. There are two different directions for adaptation in a stream environment [5]: *blind adaptation* methods that update the underlying models constantly over the stream and *informed adaptation* methods that adapt the model only if change has been detected. The later are computationally more expensive methods as except for the adaptation step they typically include a change detection step that looks explicitly for changes in the stream. Those methods though are more informative as except for the up-to-date classification model they provide additional information on the points of change, which comprises important knowledge for the end user and it allows the user to react to changes. We propose informed adaptation over ageing-based Multinomial Naive Bayes classifiers, which incorporate ageing through the damped window model, and in particular, an approach for the online tuning of the ageing factor *lambda* of the dumped window model based on the dynamics of the underlying stream.

The rest of the paper is organized as follows: Related work is discussed in Section 2. The basic concepts and motivation are presented in Section 3. Our informed adaptation approach is presented in Section 4. Experimental results are shown in Section 5. Conclusions and open issues are discussed in Section 6.

2 Related work

Change is a key concept in data streams and refers to the fact that the distribution that generates the stream is non-stationary, rather it changes with time, causing the so-called concept drifts [18]. The ability to adapt to changes is a key property of data stream mining algorithms. There are two ways of adaptation: i) by including new instances from the stream and updating accordingly the learning model and ii) by discarding outdated information from the model, also known as forgetting. The forgetting mechanisms can be categorized into: abrupt forgetting and gradual forgetting. The former ones take into consideration only recent instances within a sliding window, whereas the latter ones assume that all instances can potentially contribute to the model but with a weight that is regulated by their age. The concept of drift adaptation and state-of-the-art techniques and algorithms for dealing with drift in data stream mining is nicely covered in [6], whereas forgetting has been the subject of many research works, e.g., [4, 9–12, 15, 17, 22] just to mention a few.

Multinomial Naive Bayes (MNB) [14] is a popular classifier due to its simplicity and good performance in practice, despite its naive assumption on the class-conditional independence of the features [3, 20]. Its simplicity and efficient online maintenance makes it particularly suitable for streams. Bermingham et al. [1] compared the performance of Support Vector Machines (SVM) and MNB classifiers on microblog data and reviews (not streams) and showed that MNB performs well on short-length, opinion-rich microblog messages (rather than on long texts). In [8], popular classification algorithms were studied such as MNBs, Random Forest, Bayesian Logistic Regression and SVMs using sequential minimal optimization for the classification in Twitter streams while building classifiers at different samples. Across tested classifiers, MNBs showed the best performance for all applied data sets. In [2], MNB has been compared to Stochastic Gradient

Descend (SGD) and Hoeffding Trees for polarity classification on streams. MNB approach which was used in this study is incremental, i.e., it accumulates information on class appearances and word-in-class appearances over the stream, however, it does not forget anything. Their experiments showed that MNB had the largest difficulty in dealing with drifts in the stream population, although its performance in times of stability was very good. Regarding runtime, MNB was the fastest model due to its simplicity in predictions but also due to the easy incorporation of new instances in the model. The poor performance of MNB [2] motivated the ageing-based MNB approach [21] which also considers the recency of the class and words-in-classes observations and uses this information to regulate the class priors and class-conditional word probabilities. Their approach though is a blind adaptation approach, i.e., the model is constantly tuned based on a fixed ageing factor λ without explicitly counting for change. In this work, we follow an informed adaptation approach by tuning λ upon (data) change.

3 Basic concepts

Before we proceed we introduce some notation:

- $\mathcal{S}$: the (accumulated) stream up to current timepoint.
- V: the vocabulary of $\mathcal{S}$.
- $\mathcal{S}_{sl}$: the current sliding window of the most recent w instances.
- V_{sl}: the vocabulary of $\mathcal{S}_{sl}$

We observe a stream $\mathcal{S}$ of opinionated documents arriving at distinct timepoints $t_0, \ldots, t_i, \ldots$. An opinionated document $d\ in\ \mathcal{S}$ is a document associated with a polarity label $c \in C$, where C is the class attribute for the polarity. In the simplest case, the polarity class has two values, positive and negative. The document d is represented through the bag-of-words model as a set of words, $d = \{w_i\}$.

Our goal is to build a polarity classifier for the prediction of the polarity of new arriving documents. Our base model is the ageing-based Multinomial Naive Bayes (*ageingMNB*) classifier [21], an MNB classifier that forgets based on the damped window model with a constant ageing factor λ. Our goal is to tune the ageing factor λ according to the dynamics of the underlying stream. That is, in times of change, the ageing should be more drastic to allow for fast adaptation to the new content received from the stream, whereas in times of stability the ageing should be kept low in order to exploit the so far learned model.

MNB is one of the most popular classifiers due to its efficiency and modest performance. The original MNB classifier works in a static setting (*staticMNB*), where the whole dataset is provided as input to the algorithm. The MNB model consists of a set of class priors and class conditional word probabilities, which are estimated from the training set. The straightforward extension of the static MNB to streams is by extending the definition of the training set to the (theoretically) never-ending stream case. In particular, the training set keeps growing by including new documents that continuously arrive from the stream. Due to its simplicity it is easy to maintain the MNB model in a stream setting; the probabilities of classes and word-class combinations are updated based on the new documents and their class labels. We refer to this model as *accumulativeMNB* [2].

The *accumulativeMNB* model includes new observations but *does not forget*. Therefore, it is difficult to adapt to changes in the stream, a fact which has been already observed in previous works [2], [21]. The reason for poor adaptation is that the historical data dominate the decisions of the classifier. To overcome this issue, the *ageingMNB* model that forgets was proposed in [21].

The *ageingMNB* classifier extends the *accumulativeMNB* by including information on the recency of the observations (classes and words-in-classes observations). The recency information is derived from the original documents, which are associated with timestamps. Each class and word-class combination in the model is associated with a timestamp, the most recent timestamp where the specific class or word-class entity was observed in the stream. The recency entries are used during classification of new instances from the stream in order to downgrade the contribution of outdated observations in the model, so as more recent observations contribute more and incur model adaptation.

The (temporal) class prior for class $c \in C$ at timepoint t is [21]:

$$\hat{P}^t(c) = \frac{N_c^t * e^{-\lambda \cdot (t - t_{lo}^c)}}{|\mathcal{S}^t|} \tag{1}$$

where N_c^t is the number of documents in the stream up to timepoint t belonging to class c and $|\mathcal{S}^t|$ is the total number of document in the stream up to t. The t_{lo}^c is the most recent observation of class c in the stream and $(t - t_{lo}^c)$ denotes the time lag between the last occurrence of the class label c in the stream and the current timepoint t.

The (temporal) class conditional word probability for a word $w_i \in d$ at t is given by [21]:

$$\hat{P}^t(w_i|c) = \frac{N_{ic}^t * e^{-\lambda \cdot (t - t_{lo}^{(w_i, c)})}}{\sum_{j=1}^{|V^t|} N_{jc}^t * e^{-\lambda \cdot (t - t_{lo}^{(w_j, c)})}} \tag{2}$$

Again, the word-class counts N_{ic} are weighted by the recency of the observations of the specific word w_i in documents of class c. Old observations will be downgraded so their effect during classification is limited.

The *ageingMNB* approach is a *blind adaptation* method [6] as it applies a constant ageing factor λ in the MNB model over the whole course of the stream without considering whether there is an actual change or not. In the next section, we propose an adaptive ageing MNB model that tunes the ageing factor λ and therefore, the MNB model, online based on changes in the underlying stream population. There are two advantages of such an approach over the blind adaptation approach of *ageingMNB* [21]: first, it allows for ageing at different rates, which as already mentioned is more intuitive in a stream setting and second, except for the classification model, it provides additional information on the points of change, which is valuable for decision making and allows the end user to react to changes. For example, if a negative sentiment starts developing for a brand as a result of bad customer experiences, the brand can quickly address customer concerns and classify misconceptions thus transforming the negative sentiment into a winning customer experience.

4 Informed Adaptation of Multinomial Naive Bayes Classifiers over Data Streams

Our solution consists of two steps: (i) a change detection step that detects changes in the underlying stream population (Section 4.1), and (ii) a tuning step that adjusts the ageing factor λ, and therefore the classifier, upon detection of change (Section 4.2).

4.1 Detecting change

There are several approaches for change detection, which are presented in detail in [5]. Since our focus in this work is on the adaptation of the ageing factor λ and due to lack of space, we present here the detector we used in our experiments, which showed the best performance among several methods we tried. Our detector falls into the category of monitoring the distributions in two different time-windows: such detectors compare the decision model built upon a *reference window* of past data to the decision model built over a *current window* of the most recent data points. In this work, we monitor the distance between the vocabularies of the most recent window $\mathcal{S}_{sl}$ and the reference window $\mathcal{S}$, i.e., V_{sl} vs V, for both the negative and the positive class. For the comparison, we employ precision, which equals to the fraction of the reference vocabulary words that also appear in the current vocabulary.

$$precision = \frac{V_{sl} \cap V}{|V|} \tag{3}$$

A high precision means that the current vocabulary comprises a large part of the reference vocabulary. Intuitively, this implies that the reference model, built over the reference vocabulary, could still be valid. Otherwise, the reference model is not well reflecting the current developments in the stream.

Change points are detected by comparing current precision to the moving average precision plus/minus α times the standard deviation, as follows:

$$\begin{aligned} precision &< \mu - \alpha * \sigma \\ precision &> \mu + \alpha * \sigma \end{aligned} \tag{4}$$

where σ is the standard deviation, μ is the average precision and α is a user defined threshold that controls the trade-off between earlier detecting true alarms by allowing some false alarms. Low values of α allow faster detection, at the cost of increasing the number of false alarms.

Except for the final change points, often is also useful to detect warning points when the monitored difference between the current precision and moving average prediction exceeds some threshold $\beta \times \sigma$, with $\beta < \alpha$. Warning points are more frequent comparing to change points. Moreover, once a warning is detected a buffer of instances is maintained for model rebuild once the warning turns into an actual change point. Otherwise, the buffer is emptied.

4.2 Adapting to change

Once a change is detected, the classifier should be updated to reflect the changing population. The most abrupt way of reacting to change is by building a new classifier over the recent data and demolishing the old one. Following a more conventional approach, one can affect the statistics of the model over the stream by tuning appropriately the ageing factor λ. We present hereafter different strategies for model adaptation to change.

Let λ_0 be an initial value of λ, set at the beginning of the stream. In the simplest case, $\lambda_0 = 0$, i.e., there is no-ageing. If $\lambda_0 > 0$, there is a constant ageing over the stream.

- ***SlowIncreaseUpToALimit*** - *Gradually increase λ by a constant value c up to a limit λ_{max}:*
  When a change is detected, λ is increased by a constant value c, i.e., it is set to $\lambda_i + c$, where λ_i is the value of λ before change. If there is still change, *lambda* will be further increased by c. Increasing λ after change is beneficial as the model will focus on more recent instances and the effect of old instances will be downgraded. However, the constant increase of λ might lead to high values and the total discard of historical data. To prevent this, we set an upper limit λ_{max} for the highest value of λ. If limit is reached, λ_{max} ageing is applied for the rest of the stream. Note that for efficiency issues we check for change not after each instance but after a certain number of instances, denoted by w. This implies that each λ value has an effect for at least w instances.
- ***SlowIncreaseFastReset*** - *Gradually increase λ by a constant value c and reset to λ_0 after λ_{max} is reached:*
  The constant increase of λ in the previous strategy implies more and more data forgetting as more changes are detected in the stream. Typically though in a stream periods of change are followed by periods of stability, therefore such a forgetting is very harsh. To count for this effect, we reset λ to its initial value λ_0 when the max value λ_{max} is reached and after a certain period at this ageing level; this period is implemented in terms of a fixed number of instances w (one could use timepoints alternatively).
- ***FastSetFastReset*** - *Fast set to λ_{max} upon change and fast reset to initial λ_0 after a certain period:*
  When a change is detected, λ is instantly increased to an upper bound λ_{max}, i.e., $\lambda = \lambda_{max}$. The λ is reset to its initial value λ_0 after a certain period of w instances. The intuition is to forget fast (with λ_{max}) in times of change and slow (with λ_0) in times of "stability".
- ***FastSetSlowDecrease*** - *Fast set to λ_{max} upon change and slow reset to initial λ_0 by $\delta\lambda\%$ decrease at each step:*
  When a change is detected, λ is instantly set λ_{max}, i.e., $\lambda = \lambda_{max}$. The λ is reset to its initial value λ_0 gradually with a $\delta\lambda\%$ step. That is, at each step, *lambda* is decreased by $\delta\lambda\%$ until it reaches λ_0. The duration of each step is w instances. This offers a more gradual adaptation of λ comparing to the previous strategy.

The above strategies aim at tuning the ageing factor λ and indirectly the MNB classifier. There are other ways to affect the classifier, which do not involve direct λ tuning though. We overview them below.

– ***Rebuild*** - *Constant* λ_0 *and model rebuild upon change:*
  A constant λ, $\lambda = \lambda_0$, is applied over the whole stream but once a change is detected the classifier is rebuilt upon the most recent w instances. The constant ageing over the whole stream should, in times of relative stability, reduce the effect of noise and in case of drastic changes, the rebuilding implies an abrupt forgetting of old, outdated information. Rebuilding incurs the fastest adaptation to change, however it completely ignores any old knowledge.

Depending on the value of λ_0 we can, for all the above strategies, distinguish two cases: i) $\lambda_0 > 0$ and ii) $\lambda_0 = 0$. The former applies a constant ageing λ_0 in the stream, whereas the later does not consider ageing. Moreover, we also include the following strategies as baselines.

– ***fadingMNB*** - *Constant ageing, no change detection:* This is the blind adaptation approach (*fadingMNB*) [21]. There is a constant ageing, $\lambda = \lambda_0 > 0$, over the stream, but there is no change detection.
– ***accumulativeMNB*** - *No-ageing, no change detection:* This is the accumulative MNB approach [2], discussed in Section 3. It does not forget, neither invokes some change detection mechanism. The model is accumulative as it considers all instances from the beginning of the stream.

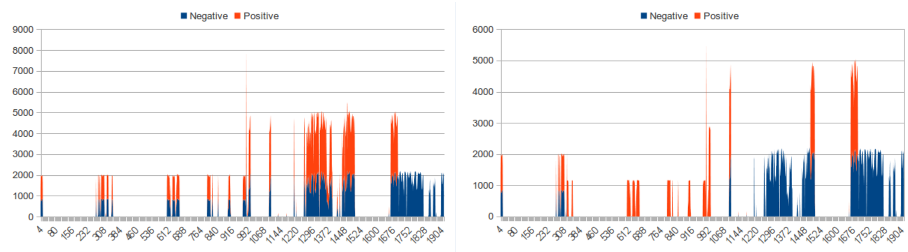
5 Experiments

5.1 Dataset

We use the TwitterSentiment dataset [19], introduced in [7]. The dataset was collected by querying the Twitter API for tweets between April 6, 2009 and June 25, 2009. The sentiment labels were derived by a Maximum Entropy classifier that was trained on emoticons [7]. The final stream consists of 1,600,000 opinionated tweets, 50% of which are positive and 50% negative. We aggregate the tweets hourly, the class distribution is shown in Figure 1(a). The class distribution is quite stable in the beginning of the stream with the positive class slightly dominating the stream. The class distribution changes drastically towards the end of the stream as only instances of the negative class are present. The change point is instance number 1,326,000. We refer to this dataset as $DS1$.

To experiment with a more volatile stream setting, we introduced some more changes to the original stream by removing certain fractions of instances. The new dataset, denoted as $DS2$, is depicted in Figure 1(b). The dataset is no longer balanced: it contains 1,073,065 tweets with 378,288 positive and 694,777 negative instances.

For the evaluation, we used prequential evaluation, where each instance of the stream is first used for testing and then for training the model. As quality measures we used accuracy over an evaluation window, $evalW$. For the detection of the change points, we used $\alpha = 1.8$. We used $\beta = 0.334$ for the detection of warning points.



(a) $DS1$ (b) $DS2$

Fig. 1. Hourly aggregated class distribution for streams $DS1$, $DS2$.

5.2 Classifiers performance

We report here on the performance of the different adaptation techniques listed in Section 4, for both $DS1$ and $DS2$[3].

Overall performance The overall results for $DS1$ are depicted in Figure 5.2 (left). The *accumulativeMNB* that does not forget achieves the worse performance, whereas *SlowIncreaseUpToALimit* with $Init-\lambda$ [4] achieves the best performance, followed by *SlowIncreaseFastReset* with $Init-\lambda$. Also, for all different strategies, a constant ageing over the stream (i.e., the $Init - \lambda$ strategies where $\lambda_0 > 0$), is better than no-ageing (i.e., the $Zero - \lambda$ strategies with $\lambda_0 = 0$).

The overall results for DS_2 are depicted in Figure 5.2 (right). Similarly to DS1, *accumulativeMNB*achieves the worse performance, whereas *Rebuild* with Init-λ achieves the best performance, followed by *FastSetFastReset* with Init-λ, *FastSetSlowDecrease*with Init-λ and *SlowIncreaseFastReset* with Init-λ. Again, having an init λ (i.e., $\lambda_0 > 0$ is better than no-ageing (i.e., $\lambda_0 = 0$), for all cases.

We should note that $DS2$ is a very volatile stream; this might explain why rebuild ranks first for $DS2$.

Overtime performance In Figures 3(a), 3(b) we show the performance over time, for the different strategies for $DS1$,$DS2$, respectively.

Table 1. Best parameter setting per strategy.

Strategies	λ_0		**instances** w **(*1,000)**		λ_{max}		**decrease** λ **ratio** ($\delta\lambda$%)		**increase** λ **value** (c)	
	DS1	DS2	DS1	DS2	DS1	DS2	DS1	DS2	DS1	DS2
fadingMNB	0.2	0.3	-	-	-	-	-	-	-	-
Rebuild-Zero-λ	-	-	-	-	-	-	-	-	-	-
Rebuild-Init-λ	0.2	0.25	-	-	-	-	-	-	-	-
FastSetFastReset-Zero-λ	-	-	100	100	0.5	0.4	-	-	-	-
FastSetFastReset-Init-λ	0.1	0.15	24	22	0.5	0.5	-	-	-	-
FastSetSlowDecrease-Zero-λ	-	-	100	100	0.5	0.4	5%	5%	-	-
FastSetSlowDecrease-Init-λ	0.1	0.15	24	22	0.5	0.5	5%	5%	-	-
SlowIncreaseUpToALimit-Zero-λ	-	-	-	-	-	-	-	-	0.6	0.1
SlowIncreaseUpToALimit-Init-λ	0.2	0.2	-	-	-	-	-	-	0.1	0.1
SlowIncreaseFastReset-Zero-λ	-	-	-	-	-	-	-	-	0.4	0.2
SlowIncreaseFastReset-Init-λ	0.2	0.2	-	-	-	-	-	-	0.4	0.2

As expected, the different strategies have an effect only after change. Before change, we can comment on the difference between approaches with an $Init - \lambda$, i.e., with ageing, and approaches with $Zero - \lambda$, i.e., no-ageing. *fadingMNB* and

[3] Parameters for DS1, DS2 are listed in Table 1.

[4] $Init - \lambda$ is the case of $\lambda_0 > 0$.

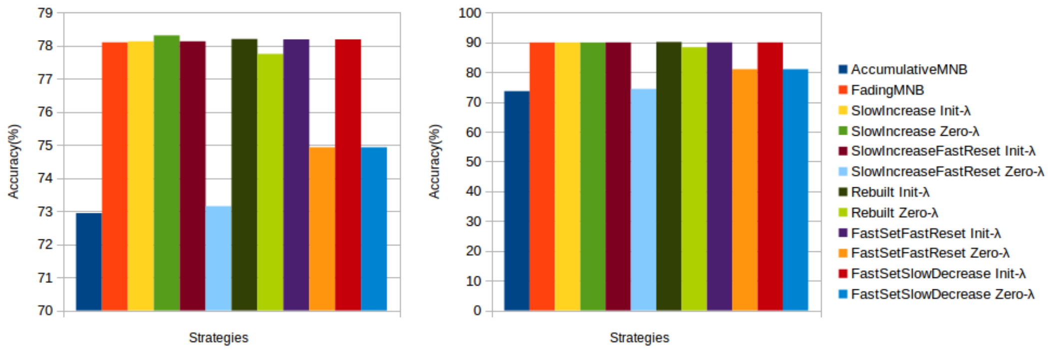
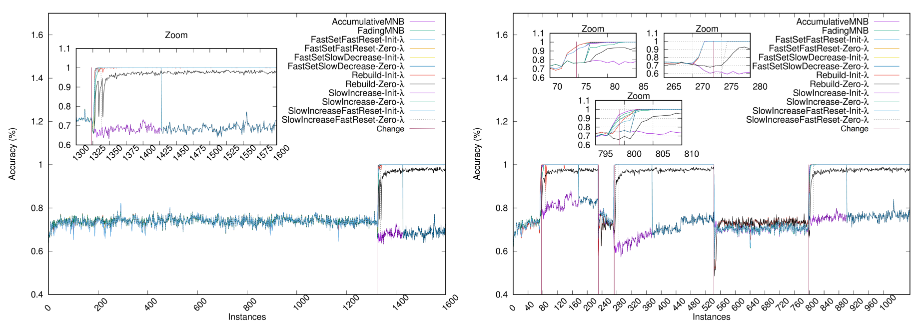


Fig. 2. Overall accuracy of different strategies $DS1$ (left), $DS2$ (right) ($Init - \lambda$ corresponds to λ_0).



(a) $DS1$ (b) $DS2$

Fig. 3. Accuracy over time for the different strategies for $DS1$ and $DS2$

all the strategies with $\lambda_0 > 0$ perform better than *accumulativeMNB*, during the "stable" period. Upon change, the differences between the different methods are better manifested: The *accumulativeMNB* has the lowest performance for both datasets as it does not manage to recover after change. Methods that reset reach the poor performance of *accumulativeMNB* after a while, i.e., when they reset to initial lambda. Again, the $init - \lambda$ approaches perform better; this is clearly depicted from the performance of the rebuild method after change (see zoom-in figures) for both datasets (red for $\lambda_0 > 0$ vs black for $\lambda_0 = 0$).

5.3 Qualitative evaluation

To qualitatively evaluate the change detector and the interplay with the classifier adaptation, we also experimented with a third focused dataset, collected from Twitter's public streaming API [5] for two specific entities, namely "Obama" and "Adele", during 2015. Our intention was to use very different entities, which will probably generate different words. "Obama"'s vocabulary, for example, will be related to politics, whereas "Adele"' vocabulary will be related to music with no

[5] https://dev.twitter.com/streaming/overview

much overlap between them. Out of the total 71,124 tweets, the majority (66,012) refers to "Obama" and the remaining (5,112) to "Adele". Figure 4 depicts the class distribution for both entities.

In the beginning of the stream, only "Obama" is present, "Adele" is introduced on instance 28,000 and remains up to instance 43,000, after that only "Obama" is present again. The vocabulary-based change detector is sensing a change at point 28,321 (recall "Adele" was introduced on instance 28,000) and raises an alarm. At point 30,000 a real change is detected and the classifier adaptation strategies take effect. The change detector starts sensing a new change at point 46,778 (recall "Adele" is removed after instance 43,000) and detects the actual change on instance 48,000. The alarms, detected changes and performance of the classifier are depicted in Figure 5. In both cases, the change detector

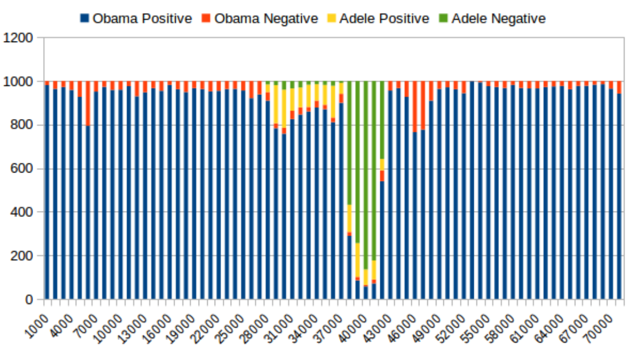


Fig. 4. Class distribution for both entities

managed to detect the changes in the underlying stream, though with delay. The delay is due to the detector itself as even for different entities like "Adele" and "Obama" coming from different areas, the vocabulary is not completely disjoint rather common words are used in both cases.

What is interesting is that the performance of the classifier started dropping before the first actual change point. A possible explanation is that even within a single topic, like "Obama" there might be changes which affect the classifier and therefore, we observe the drop. Those changes could not be detected by our vocabulary-based detector, because for example the alarm threshold α was too high or because the change itself cannot be captured by a vocabulary-based detector. The same behavior is observed after the second change point. What these incidents might indicate is that a single change detector type might not be adequate to deal with all different types of change that can occur in a stream. In practice, change might be due to different reasons like change in the class distribution, different topics discussed in the stream, internal changes within a topic etc. This calls for different types of change detectors that can be activated under different conditions. We plan to undertake this challenge of building a change detection framework of different detectors in our future work.

6 Conclusions and Outlook

We presented an informed adaptation approach for ageing-based Multinomial Naive Bayes classifiers in order to allow adaptation at different rates over the

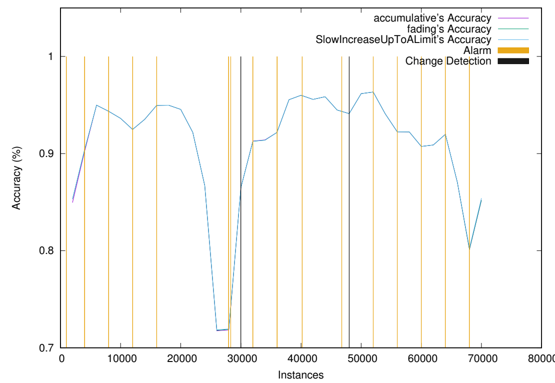


Fig. 5. Informed adaptation: change points and accuracy

stream based on the dynamics of the underlying stream population. Our motivation is that in times of change, ageing should be more harsh to allow for a faster adaptation of the model, however in times of stability the ageing factor should be lowered to allow for model exploitation. We proposed several adaptation techniques for the ageing factor λ. The experimental results showed that different strategies perform similarly but all of them outperform techniques that use no ageing. The same holds for harsh forgetting techniques, like model rebuild. In our experiments informed adaptation performed similarly to blind adaptation approaches. However, we should stress that the informed adaptation methods, expect for the adapted classification model, also provide the user with the points of change, which is valuable for decision making and reactions to change.

Thus far, we tune the model indirectly through the ageing factor λ. In our future work, we will also discard outdated parts of the model, i.e., outdated class priors and class conditional word probabilities to allow for faster adaptation to change and re-learning of outdated parts of the model. Moreover, as our qualitative experiment revealed, a change detector can detect a single type of change, although in practice, change might occur due to different reasons. We plan to investigate the possibility of a framework of different detectors that can be activated under different conditions and might call for different model update strategies.

Acknowledgements

The work was partially funded by the European Commission for the ERC Advanced Grant ALEXANDRIA under grant No. 339233 and by the German Research Foundation (DFG) project OSCAR (Opinion Stream Classification with Ensembles and Active leaRners).

References

1. A. Bermingham and A. F. Smeaton. Classifying sentiment in microblogs: Is brevity an advantage? In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1833–1836, New York, NY, USA, 2010. ACM.

2. A. Bifet and E. Frank. Sentiment knowledge discovery in Twitter streaming data. In *Discovery Science*, 2010.
3. P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.*, 29(2-3):103–130, nov 1997.
4. G. Forman. Tackling concept drift by temporal inductive transfer. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 252–259. ACM, 2006.
5. J. Gama. *Knowledge Discovery from Data Streams*. CRC Press, 2010.
6. J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, mar 2014.
7. A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6, 2009.
8. B. Gokulakrishnan, P. Priyanthan, T. Ragavan, N. Prasath, and A. S. Perera. Opinion mining and sentiment analysis on a Twitter data stream. In *Proceedings of 2012 International Conference on Advances in ICT for Emerging Regions (ICTer)*, ICTer '12, pages 182 – 188. IEEE, 2012.
9. R. Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8(3):281–300, 2004.
10. Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.
11. I. Koychev. Gradual forgetting for adaptation to concept drift. Proceedings of ECAI 2000 Workshop on Current Issues in Spatio-Temporal Reasoning,, 2000.
12. I. Koychev. Tracking changing user interests through prior-learning of context. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 223–232. Springer, 2002.
13. Y. Liu, X. Yu, A. An, and X. Huang. Riding the tide of sentiment change: Sentiment analysis with evolving online reviews. *World Wide Web*, 16(4):477–496, jul 2013.
14. A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*, pages 41–48. AAAI Press, 1998.
15. M. Pechenizkiy, J. Bakker, I. Žliobaitė, A. Ivannikov, and T. Kärkkäinen. Online mass flow prediction in cfb boilers with explicit detection of sudden concept drift. *ACM SIGKDD Explorations Newsletter*, 11(2):109–116, 2010.
16. L. Plaza and J. Carrillo de Albornoz. *Sentiment Analysis in Business Intelligence: A survey*, pages 231–252. IGI-Global, 2011.
17. M. Salganicoff. Tolerating concept and sampling shift in lazy learning using prediction error context switching. *Artificial Intelligence Review*, 11(1-5):133–155, 1997.
18. J. C. Schlimmer and R. H. Granger. Beyond incremental processing: Tracking concept drift. In *AAAI*, pages 502–507, 1986.
19. Sentiment140. Sentiment140 - a Twitter sentiment analysis tool. `http://help.sentiment140.com/`.
20. P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
21. S. Wagner, M. Zimmermann, E. Ntoutsi, and M. Spiliopoulou. Ageing-based multinomial naive bayes classifiers over opinionated data streams. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, Porto, Portugal*, pages 401–416, 2015.
22. G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.