# Adaptive Semi Supervised Opinion Classifier With Forgetting Mechanism

Max Zimmermann[a]
Faculty of Computer Science
Otto-von-Guericke-University
of Magdeburg,Germany
max.zimmermann@iti.cs.uni-magdeburg.de

Eirini Ntoutsi
Institute for Informatics
Ludwig-Maximilians-University
of Munich,Germany
ntoutsi@dbs.ifi.lmu.de

Myra Spiliopoulou
Faculty of Computer Science
Otto-von-Guericke-University
of Magdeburg,Germany
myra@iti.cs.uni-magdeburg.de

## ABSTRACT

Opinion stream classification methods face the challenge of learning with a limited amount of labeled data: inspecting and labeling opinions is a tedious task, so systems analyzing opinions must devise mechanisms that label the arriving stream of opinionated documents with minimal human intervention. We propose an opinion stream classifier that only uses a seed of labeled documents as input and thereafter adapts itself, as it reads documents with unknown labels. Since the stream of opinions is subject to concept drift, we use two adaptation mechanisms: *forward adaptation*, where the classifier incorporates to the training set only those unlabeled documents that it considers informative enough in comparison to those seen thus far; and *backward adaptation*, where the classifier gradually forgets old documents by eliminating them from the model. We evaluate our method on opinionated tweets and show that it performs comparably or even better than a fully supervised baseline.

## 1. INTRODUCTION

Opinion stream classification is traditionally based on the assumption that the labels of the opinionated documents are made available to the classifier for adaptation. This assumption is rather unrealistic: the users upload their opinions but do not come back to assign labels to them, nor can human experts be engaged to mark each opinion as positive or negative. We propose an opinion stream classifier that uses a single initial seed of labeled documents for training and thereafter adapts by judiciously considering some of the new documents for re-learning and by forgetting some of the old documents.

Streams of opinionated documents show up in several applications, such as product reviewing in e-commerce. There, the streaming data capture changes in the attitude of people towards the products, e.g. whether people start disliking a popular product in response to some negative news on it. As we have shown in [18], change in attitude can also concern the product features that people consider important. Stream mining is also applied on tweets [2], aiming to discover bursts and to monitor the attitude of people towards events. While unsupervised stream learners return models that describe the data in the absence of a ground truth, stream classifiers require that the labels of the observed documents are made available soon after the documents themselves have been seen. Active stream learning methods [19] aim to reduce the number of documents that have to be labeled, but still require continuous involvement of the human expert. Our opinion stream classifier rather requires only an initial set of labeled documents.

Our contribution is an adaptive classifier that uses only an initial seed of documents for learning and later adapts with help of unlabeled documents. Our method has two aspects: learning from new documents (forward adaptation) and forgetting old ones (backward adaptation). For *forward adaptation*, we extend the method we proposed in [18] to select those new documents that are most important and most reliably classified by the model learned thus far. For *backward adaptation*, we remove some of the earlier seen documents from the model. Although most stream classifiers use a sliding window over the data, backward adaptation through active elimination of past information from the model is a rather new adaptation modality [13, 14, 1].

The paper is organized as follows. We discuss related work in Section 2. In Section 3 we discuss the forward adaptation mechanism that exploits unlabeled documents and then the backward adaptation mechanism that removes outdated documents from the model. In Section 4 we evaluate our method on streams of tweets. The last section concludes our study with a summary and open issues for future work.

## 2. RELATED WORK

The learning over a stream of opinionated documents requires adaptation to concept changes. Adaptation to concept drift is intensively investigated in stream classification literature. The core idea is to detect change by monitoring the labels of the arriving data instances [8] and then to adjust the model towards the most recent data. If the model is learned by a classifier ensemble, as e.g. in [4], then adjustment involves re-weighting the individual classifiers. More elaborate techniques have been proposed, including dedi-

[a] Max Zimmermann was partially funded by the German Research Foundation project SP 572/11-1 "IMPRINT: Incremental Mining for Perennial Objects".


*SAC'14* March 24-28, 2014, Gyeongju, Korea.

cated techniques for opinionated streams [2, 3], but they rely on the prompt arrival of new *labeled* documents.

The method of Silva et al. [15] operates on an initial seed of documents, building upon advances of semi-supervised classification. Their classifier consists of a set of sentiment rules, extracted from the initial seed. Such a rule consists of a set of terms as antecedent and the predicted sentiment label as consequent. Using the rules, the likelihood of each sentiment/label is computed for a document, and then the document is labeled with the most likely label. These newly labeled documents are included to the seed, provided that their sentiment score exceeds a threshold. To update the classifier with a new document, Silva et al. increase counts like confidence, support and cardinality of related rules and also extract new rules; extracting new rules is very costly though. No rules are discarded, although some of them may be outdated: Silva et al. [15] do not differentiate between old and new documents, so that rules describing only old documents are not forgotten. In contrast, we propose a classifier that adapts itself by fading out old documents and assigning higher weights to recent documents.

Drury et al. [5] propose a semi-supervised classification algorithm that trains the model from an initial seed and relearns the classifier with a seed extended by *self-training* (cf. self-training was introduced in [7]). As in [15], a document is added to the seed, if the label has been assigned with high confidence. However, Drury et al. [5] consider a static environment, where all text messages contribute equally to the classifier. We rather use an ageing function, so that recent documents have a greater impact on the classifier.

Adaptation to concept drift is particularly challenging for semi-supervised classifiers, since after some time the initial seed is no more representative of the class distribution. This issue has been investigated by Dyer and Polikar in [6]. Their method is promising, but it is very sophisticated and resource-demanding, and it is unclear how it scales for data with more than two dimensions. Since the feature space of opinionated documents has a substantially larger cardinality, we propose a less sophisticated method that relies on the usefulness of the arrived tweets for the classifier.

If a classifier incorporates new unlabeled instances on the basis of a small initial seed, then it will be unable to depart from the initial concept and thus will fail to respond to drift. Therefore, we propose to *forget* old data, not simply by taking old instances out of the sliding window but by eliminating them from the model. Our approach is inspired by the relational stream classifier proposed in [13], which applies an ageing function on the decision tree: if a branch receives no new instances for some time, it is discarded. We build upon this principle by using the age of the instances to decide when to discard them from the model, thereby allowing for arbitrary model learners, not just decision trees.

# 3. ADAPTIVE LEARNING WITH ONLY AN INITIAL SEED

We observe a growing collection $\mathcal{D}$ of documents in batches. The batches constitute a stream, which we monitor at distinct timepoints $t_0, t_1, \ldots, t_i, \ldots$; at each $t_i$ we receive a batch with equal number of documents. A document $d \in \mathcal{D}$ is represented by the *bag-of-words* model, i.e. the ordering of the words is ignored whereas for each word $w_i \in d$ its frequency $f_i^d$ is stored. We assume an *initial seed set* $\mathcal{S}$ of documents: for each $d \in \mathcal{S}$, the label $class(d) \in C$ is known ($C$ is the set of all labels). The set of words in $\mathcal{S}$ is the *vocabulary* $V$.

**Algorithm 1:** Adaptive Semi-Supervised Opinion Stream Classifier (ADASTREAM)

```
   Input : initial seed S, stream D, α, λ
1  t ← 1; Δ (S) ← train initial classifier on seed S
2  batch ← batch at timepoint t
3  while batch do
4      for k=1 to |batch| do
5          d ← batch_k; c ← compute label for d by Δ (S)
6          if usefulness of c ≥ α then
               // forward adaptation
7              for i=1 to |d| do
                   // update word counts based on d
8                  N_ic^aged = N_ic^aged + (f_i^d * age(d,t));
                       // see Eq.3
9              N_c^aged = N_c + age(d,t)
10             S ← S ∪ d // update the seed
11     for j=1 to |S \ batch| do
           // backward adaptation
12         d ← S_j; c ← compute label for d by Δ (S)
13         N_c^aged = N_c^aged + age(d,t) − age(d,t−1)
14         for i=1 to |d| do
15             N_ic^aged = (f_i^d * age(d,t)) − (f_i^d * age(d,t−1))
               +N_ic^aged
16     t ← t +1; batch ← batch at timepoint t
```

| Parameter | Definition |
|---|---|
| $\alpha$ | usefulness threshold: $-1 < \alpha \leq 0$ |
| $\lambda$ | decay rate: $0 < \lambda < 1$ |
| $f_i^d$ | number of occurrences of a word $w_i$ in a document $d$ |
| $N_{ic}^{aged}$ | weighted number of occurrences of the word $w_i$ in documents of class label $c$ |
| $N_c^{aged}$ | weighted number of documents per class |
| $\Delta$ | classifier |
| $V$ | vocabularly |

**Table 1: Variables**

The pseudocode of our algorithm is depicted in Algorithm 1. The seed set $\mathcal{S}$ is used to initially train a sentiment classifier $\Delta(\mathcal{S})$ (cf. Section 3.1). We apply a *usefuleness* test on each document $d \in \mathcal{S}$ (line 6) and we store/update the counts of words from useful documents (line 8) for each class. Then, we update the word counts for those words which are from documents for which the classifier derives a *useful* label (line 8), using the labels of the classifier itself (self-training, semi-supervised).

We must update the word counts and the classifier while considering concept drift. We perform two adaptation steps. In (i) *forward adaptation* (lines 7-10), arriving documents are labeled, tested for usefulness and conditionally incorporated into $\mathcal{S}$; the words in them are used to update the statistics and thus adapt $\Delta(\mathcal{S})$ (cf. Section 3.2). In (ii) *backward*

*adaptation* (lines 12-15), the contribution of words on the classifier is modified, as their weight decreases with time; newly arriving documents, and consequently the words in them are given higher weights, so that counts of newer words affect the classifier to a greater extent (cf. Section 3.3). Old words are gradually forgotten and removed from the model.

## 3.1 Basic Learner: Multinomial Naive Bayes

To learn a classifier $\Delta(\mathcal{S})$ over $\mathcal{S}$, we use Multinomial Naive Bayes (MNB) [10]. In conventional MNB, the probability of a class $c$ given a document $d$ is:

$$P(c|d) = \frac{P(c)\prod_{i=1}^{|d|} P(w_i|c)^{f_i^d}}{P(d)} \quad (1)$$

where $P(c)$ is the prior probability of class $c$, $P(w_i|c)$ is the conditional probability that word $w_i$ belongs to $c$ and $f_i^d$ is the number of occurrences of $w_i$ in document $d$.

The estimates of these probabilities (estimates indicated by a "hat" as in $\hat{P}$) are re-computed on the contents of the seed set $\mathcal{S}$ every time the model is updated, keeping in mind that $\mathcal{S}$ also changes are new documents with derived labels are added (forward adaptation, Section 3.2) and very old ones are forgotten (backward adaptation, Section 3.3). The formula for the conditional propability estimate of a word $w_i$ in documents of class label $c$ is:

$$\hat{P}(w_i|c) = \frac{N_{ic}+1}{\sum_{j=1}^{|V|} N_{jc} + |V|} \quad (2)$$

where $N_{ic}$ is the number of occurrences of word $w_i$ in documents with label $c$, $V$ is the vocabulary over $\mathcal{S}$ and $P(d)$ is the prior of $d$ (assumed the same for all documents). We apply Laplacian correction (initializing to 1 instead of 0) to avoid the zero-frequency problem. In Section 3.3 we extend Eq. 2 to take the age of documents into account.

## 3.2 Forward Adaptation – Incorporating New Documents into the Model

As the stream of documents evolves, we update the initial classifier $\Delta(\mathcal{S})$ by incorporating new documents into the seed set $\mathcal{S}$ after deriving their labels with $\Delta(\mathcal{S})$. We use the extended training set $\mathcal{S}'$ to adapt the model into $\Delta(\mathcal{S}')$. To select new documents for the extension of $\mathcal{S}$, we introduce the concept of *usefuleness*, which is based on entropy.

*Definition 1.* [Usefulness] Let $d$ be a new document, to which $\Delta(\mathcal{S})$ assigns the label $c$. The *usefuleness* of $d$ is

$$Usefuleness(d) = \sum_{w_i \in d} H(\mathcal{S}, w_i) - H(\mathcal{S} \cup d, w_i) \quad (3)$$

and $d$ is *useful* for learning if $Usefuleness(d)$ is greater than a threshold $\alpha \in (-1, 0]$: Here, $H(\mathcal{S}, w_i)$ is the entropy of $\mathcal{S}$ w.r.t. $w_i$, which expresses how pure $\mathcal{S}$ is w.r.t class when considering only $w_i$; $H(\mathcal{S} \cup d, w_i)$ is the entropy w.r.t. $w_i$ when considering $d$ as part of the seed set, i.e. over the set $\mathcal{S} \cup d$. The entropy $H(\mathcal{S}, w_i)$ is defined as:

$$H(\mathcal{S}, w_i) = -\sum_{c \in C} p_{ic} * log_2 p_{ic}$$

where the probability that word $w_i$ belongs to class $c$ according to the seed set $\mathcal{S}$ is:

$$p_{ic} = \frac{\{|d \in \mathcal{S} : w_i \in d \wedge class(d) = c\}|}{|d \in \mathcal{S} : w_i \in d\}|}$$

Informally, a document that decreases the entropy difference in Eq. 3 is useful because it "boosts" the performance of the old classifier by adding to $\mathcal{S}$ documents that are very likely to have indeed the label assigned to them. On the other hand, a document that increases the entropy difference is also useful, since it forces the classifier to adapt to documents that are different from those seen thus far. We regulate the usefuleness of documents with the threshold $\alpha \in (-1, 0)$: values close to 0 promote *smooth adaptation*, because they require that the newly added documents in the model agree with the old classifier, while values close to $-1$ promote diversity.

It is noted that in the usefulness definition we use the entropy difference over all words $w_i \in d$, instead of over all words in $\mathcal{S}$ and $\mathcal{S} \cup d$, respectively. The reason is that $d$ is the only difference between the two sets, so there is no need to iterate over all the words. If $d$ is useful w.r.t. the usefulness threshold $\alpha$, the seed set $\mathcal{S}$ is expanded by $d$, so the new seed set is $\mathcal{S} \cup d$. Also, the parameters of the MNB classifier are updated accordingly based on $d$. This is an efficient update, as we need to update only the counts $N_{ic}$ for all words $w_i \in d$ and class label $class(d) \in C$.

## 3.3 Backward Adaptation – WeightingDocuments By Age

Next to forward adaptation, we weight documents by their age, so that older documents have gradually less effect on the classifier and very old ones get discarded from $\mathcal{S}$ completely. For the weighting scheme we use the exponential ageing function, which has been widely used in temporal applications and data streams, see e.g. [11]. More formally:

*Definition 2.* [Document Age] Let $d$ be a document that arrived at $t_d$. The age of $d$ at the current timepoint $t$ is $age(d,t) = e^{-\lambda \cdot (t - t_d)}$ where $\lambda > 0$ is the decay rate. The higher the value of $\lambda$, the lower the impact of old documents.

We incorporate ageing into Algorithm 1 (cf. lines 8-9, 13-15) by replacing $N_{ic}$ in Eq. 2 with $N_{ic}^{aged}$, defined as:

$$N_{ic}^{aged} = \sum_{d=1}^{|\mathcal{S}|} f_{ic}^d * age(d,t) \quad (4)$$

where the number of occurrences of word $w_i$ in $d$ with label $c$ is weighted by the age of $d$. Hence, the conditional probability of $w_i$ given class $c$ (Eq. 2) is replaced by:

$$\hat{P}(w_i|c)_{aged} = \frac{N_{ic}^{aged} + \mu}{\sum_{j=1}^{|V|} N_{jc}^{aged} + \sum\limits_{d \in \mathcal{S}} age(d,t)} \quad (5)$$

The parameters $\mu$ and $\sum_{d \in \mathcal{S}} age(d,t)$ serve as Laplace correction; $\mu$ is the smallest weight – referring to a document that appeared at timepoint 0 (beginning of the stream).

In principle, we should update the word counts after the arrival of each document. However, when a document $d$ arrives, only the counts of words in $d$ must be updated, and only for the class of $d$. So, we rather update once per timepoint: we modify the weights of all documents that arrived between the previous and current timepoint and then adjust the counts of each affected word and class.

# 4. EXPERIMENTS

We study the performance of our semi-supervised stream classification method and other variations on two streams

of tweets that differ in size and in the presence of drift (cf. Subsection 4.1). In particular, we consider the following classification methods for the comparison:

- *AdaptFull*: our stream classification algorithm with both forward & backward adaptation (cf. Section 3)
- *AdaptFWD*: our stream classification algorithm with forward adaptation only (cf. Subsection 3.2)
- *AdaptBWD*: the stream classification algorithm with backward adaptation only (cf. Subsection 3.3)
- *MNB_Baseline*: the basic Multinomial Naive Bayes (cf. Subsection 3.1) when using it as a fully supervised approach, i.e. we incrementally adapt by the *true labels* of the tweets. No forgetting is considered.
- *SelfLearner*: the opinion classification algorithm of [18]. This algorithm learns from the training data in the initial seed set and from all *predicted labels* derived using this seed, without adaptation nor filtering on usefulness or age of the data.

We employ two different evaluation techniques, prequential and holdout evaluation which are two basic evaluation procedures in the case of data streams [2]. Additionaly we compute accuracy and kappa statistic as evaluation measures (cf. Subsection 4.2).

We study the effect of the different parameters in the performance, namely, the size of the initial seed set `seedSize` $|\mathcal{S}|$, the usefulness threshold $\alpha \in (-1, 0]$ and the decay rate $\lambda > 0$. Recall that the usefulness threshold $\alpha$ introduced in forward adaptation (cf. Subsection 3.2) "controls" which documents should be considered for seed expansion. Lower values of $\alpha$ allow for considering documents that are very different from the seed. The decay factor $\lambda$ introduced in backward adaptation (cf. Subsection 3.3) determines how fast an old document is forgotten; lower values of $\lambda$ imply that old documents are remembered longer.

## 4.1 Datasets

We used two Twitter streams with different characteristics to test our methods.

**TwitterSentiment (TS) dataset** is a balanced dataset containing 1.600.000 tweets, half positive and half negative. It was collected by querying the (non-streaming) Twitter API for messages between April 2009 and June 25, 2009. It is very heterogeneous towards content by capturing a mixture of different topics. To label the tweets, the Maximum Entropy classifier has been used [9]. [2]

**ICA dataset** is a dataset created by Charlotte Priess and Alina Sinelnikova as part of their bachelor theses [16, 12] [3]. This dataset is described hereafter.

The initial ICA dataset is the result of a crawl (run etween 20.11.2011 and 16.12.2012) that collected 7.730 tweets, using the Twitter search API and the topsy otter API. The crawler collected tweets belonging to 3 different topics and having16 different hashtags (cf. Table 4.1). Each day, 25 new tweets for each topic were added, resulting in 400 new tweets per day. Retweets and non-English tweets were removed.

[2]Available at: http://help.sentiment140.com/for-students

[3]Available at: www.dbs.ifi.lmu.de/cms/Publications/TwitterSentimentDataset, acknowledging Charlotte Priess and Alina Sinelnikova.

| Topic | Hashtags |
|---|---|
| fast food | #pizzahut, #mcdonalds, #burgerking, #hooters, #kfc |
| mobile | #motorola, #nokia, #HTC, #iphone, #blackberry |
| TV | #dexter, #himym, #xfactor |

**Table 2: Hashtags of ICA dataset**

The crawled tweets were labeled manually, to achieve results of higher quality despite the presence of challenging tweets (e.g. sarcastic ones and tweets in colloquial language). The labeling was performed by independent web users according to following workflow: users registered (to ensure that no user assigns labels more than once) and then the tweets were shown to them; for each tweet, they had to select among the rating options "netative", "positive", "neutral", "not english" and "skip". The first three options were intended for tweets with clear context and sentiment; the "skip"-option was intended for tweets with unclear context or intention. The "not english"-option has been necessary, because some non-english tweets contained english words and were thus misjudged as english by our crawler; these tweets had to be skipped.

In total, 50 users registered for labeling these tweets. Each tweet was labeled by at least 3 users. Almost 49% of the tweets were marked as "neutral", 28 % as "positive" and 13 % as "negative". The "skip" was used for 10 % of the tweets, the "not english" for 0.8% of the tweets. We skipped tweets with the last two labels. Among the remaining ones, only those having enough positive ('pos') resp. negative ('neg') ratings (absolutely as well as relatively) and an unambiguous polarity were added to the dataset. Since many tweets were marked as "neutral" and many of the opinionated ones had both positive and negative labels, only 35% of the tweets were retained. The final dataset, denoted as **ICA dataset** hereafter, contains 2.949 positive and 1.248 negative ones.

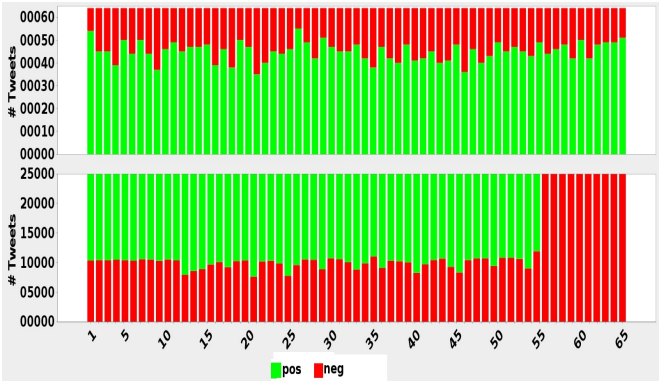


**Figure 1: Class distribution over time for the ICA dataset (top) and the TS dataset (bottom)**

The class distribution over time is shown in Figure 1. For the ICA dataset, the class distribution is almost the same over time, with the positive class (green color) dominating the negative one (red color). The TwitterSentiment dataset is balanced in the larger part of the stream, with positive and negative classes sharing almost the same number of in-

stances over time. Towards the end of the stream though, the dataset becomes highly imbalanced containing only instances of the negative class (red color). The preprocessing of the datasets was done according to [17]. More details can be found in [16].

## 4.2 Evaluation methods and quality measures

To evaluate the performance of the classifier, we employ two different evaluation methods: holdout evaluation and prequential evaluation [2].

- *Holdout evaluation*: The performance is measured by using a standard amount of instances from each batch for training and the rest for testing. In our case, we chose 2/3 of the batch instances for training and the rest for testing.

- *Prequential* (or *interleaved test-then-train*) *evaluation*: Each instance of the stream is first used for testing the performance of the classifier and then for training. [4] This method is more appropriate for data streams since its highly adaptive and most robust to overfitting.

As evaluation measures, we used *accuracy* and *kappa statistic* within a sliding window. The kappa statistic [2] normalizes accuracy by that of a chance classifier:

$$k = \frac{p_0 - p_c}{1 - p_c}$$

In the above equation, $p_0$ is the accuracy of the classifier and $p_c$ is the probability that a chance classifier, that assigns the same number of examples to each class as the classifier under consideration, makes a correct prediction. The kappa value can vary between 0 and 1, with 0 indicating that the classifier's predictions coincide with the predictions of the chance classifier. Kappa is more appropriate for data streams since the class distribution might change over time.

## 4.3 Results on the ICA dataset

We first compare our full adaptive classifier (*AdaptFull*) to its variations namely, the only forward adaptive classifier (*AdaptFWD*), the only backward adaptive classifier (*AdaptBWD*), and to the fully supervised MNB (*MNB_Baseline*) and the no adaptation at all classifier (*SelfLearner*). Recall that MNB_Baseline uses the true class labels for adaptation, whereas the rest of the methods rely upon the predicted labels derived from the seed set.

The results are displayed in Figure 2. We clearly outperform the SelfLearner and the AdaptBWD for both holdout and prequential evaluation under kappa. The forward adaptation (AdaptFWD) contributes to stability for both the evaluation methods and both the evaluation measures while the backward adaptation (AdaptBWD) performs well for accuracy under prequential adaptation. Since the ICA stream is rather homogeneous, containing 16 topics, and also quite unbalanced (cf. section 4.1) we do not overcome the fully supervised approach (MNB_Baseline). However our approach contributes to higher kappa and slightly higher accuracy values under holdout and prequential evaluation in comparison to the SelfLearner, while using only the true labels from the initial seed.

Next, we discuss the effect of the different parameters, namely decay rate $\lambda$, usefulness threshold $\alpha$ and initial seed size $|\mathcal{S}|$ on our method. The size of the batch is set to 65 across all experiments on the ICS dataset.

In Figure 3, we present the effect of the *decay rate* $\lambda$ in the performance of the proposed method while keeping constant the seed size $|\mathcal{S}| = 260$ and the usefulness threshold $\alpha = -0.4$. The kappa under holdout evaluation for $\lambda = 0.1, \lambda = 0.3, \lambda = 0.5$ is greater than for $\lambda = 0.7, \lambda = 0.9$. Thus, slow forgetting of old data (large $\lambda$) affects the performance under holdout because there are too few data to learn. When considering more data, i.e. under prequential evaluation, the kappa for $\lambda = 0.1$ achieves a slightly higher value while the other values remain stable among the $\lambda$'s. Values for the accuracy show similar trend as kappa but they are located around 0.6.

In Figure 5, we present the impact of the *usefulness threshold* $\alpha$ on the accuracy under holdout and prequential evaluation. The value $\alpha = 0$ reveals the worst value among the setting for $\alpha$. So, extending the training set with documents that have unexpected labels (small value for $\alpha$) has a positive influence on the performance of the accuracy for holdout and prequential evaluation.

In Figure 7, we present how the *size of the initial seed set* $|\mathcal{S}|$ affects the accuracy under both holdout and prequential evaluation. As was expected, small increases of the seed size also increase the accuracy because more true labels are considered. However, the accuracy levels out at around 0.7 when increasing the size of the seed. This might be due to the data, which capture a fix size of 16 topics that were collected everytime across the crawling. Consequently, to increase the size of the seed does not influence accuracy when the seed size was set to 390.

## 4.4 Results on the TwitterSentiment dataset

Similarly to the ICA dataset, we evaluate the performance of the different approaches and we study the effect of the different parameters on the quality of the proposed method while setting the size of the batch to 25000.

In Figure 10, a comparison of the different methods in terms of accuracy and kappa and for both holdout and prequential evaluation methods is shown. Our full adaptive method (AdaptFull), trained on a seed size of 100000 tweets, outperforms the fully supervised basic Multinomial Naive Bayes (MNB_Baseline) for both holdout and prequential evaluation in both kappa and accuracy. In particular the full adaptive approach is more stable than the baseline, i.e. the accuracy and kappa does not oscillate much while the baseline shows much oscillation.

The level of oscillation is pictured by the results of kappa for holdout evaluation (cf. Figure 9). It shows a huge oscillation of the baseline while drawing a rather stable line for the full adaptive approach as well as the approach when using only the forward adaptation. Accordingly, the forward adaptation contributes much to the stability of the performance when employing the usefulness threshold $\alpha$, i.e. documents that reveal very unexpected labels are not used to adapt the classifier. Considering the huge heterogeneity of the TS corpus, then the impact of adapting the classifier with documents that show expected or slightly unexpected labels is rather high. Note: the value for kappa becomes zero as soon

[4]For the proposed method and its variations, under prequential evaluation, all documents are considered for learning, but some may be discarded as non-useful based on the usefulness threshold $\alpha$.

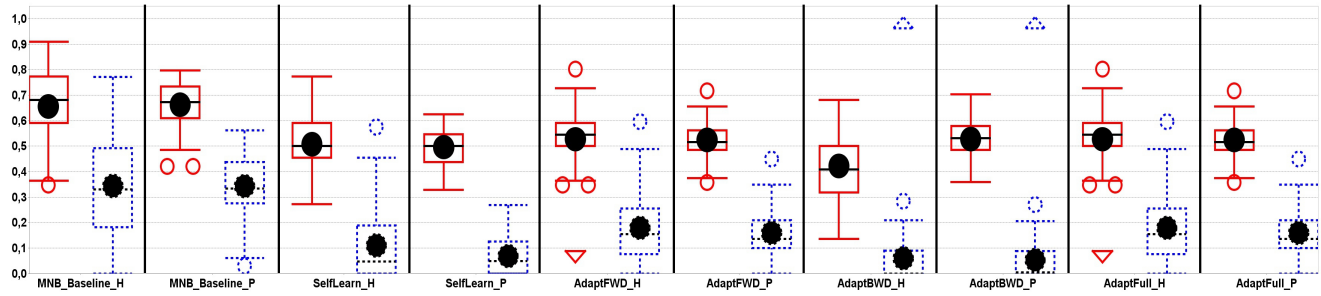


**Figure 2: ICA dataset: Kappa (dashed) and accuracy (drawn) over time for the different classifiers and for both holdout (_H) and prequential (_P) evaluation ($\alpha$=0.0, $\lambda$=0.5, $|\mathcal{S}|$=260).**

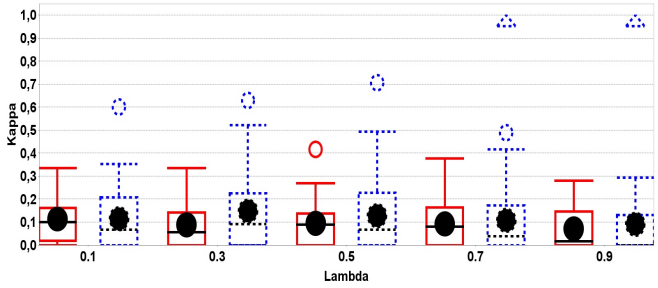


**Figure 3: ICA dataset: AdaptFull method. Holdout (dashed) and prequential (drawn) kappa over time for different $\lambda$ ($|\mathcal{S}|$=260, $\alpha$=-0.4).**

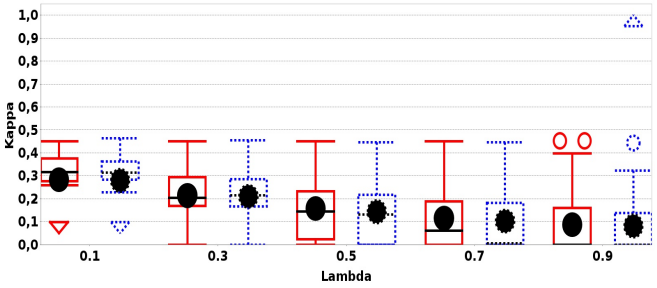


**Figure 4: TwitterSentiment dataset: AdaptFull method. Holdout (dashed) and prequential (drawn) kappa over time for different $\lambda$ ($|\mathcal{S}|$=10000, $\alpha$=-0.0).**

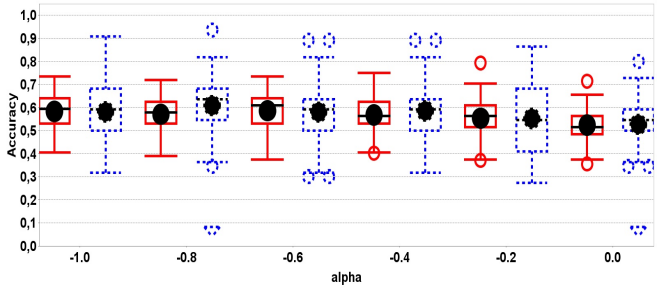


**Figure 5: ICA dataset: AdaptFull method. Holdout (dashed) and prequential (drawn) accuracy over time for different $\alpha$ ($|\mathcal{S}|$=260, $\lambda$=0.5).**

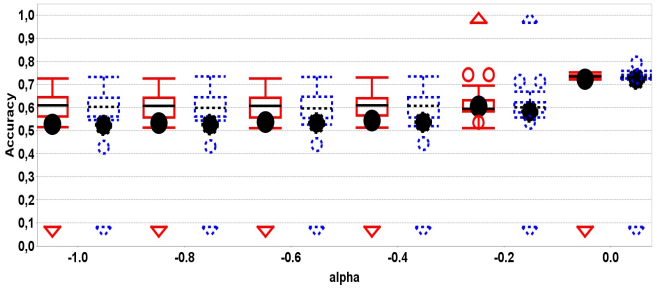


**Figure 6: TwitterSentiment dataset: AdaptFull method. Holdout (dashed) and prequential (drawn) accuracy over time for different values of the usefulness threshold $\alpha$ ($|\mathcal{S}|$=100000, $\lambda$=0.5).**

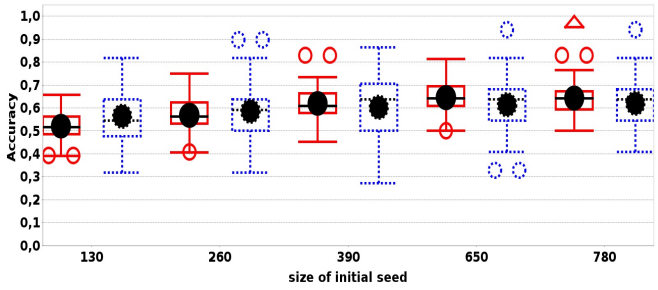


**Figure 7: ICA dataset: AdaptFull method. Holdout (dashed) and prequential (drawn) accuracy over time for different initial seed sizes $|\mathcal{S}|$ ($\alpha$=-0.4, $\lambda$=0.5).**

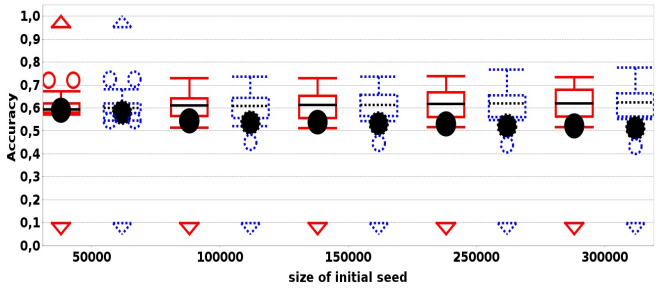


**Figure 8: TwitterSentiment dataset: AdaptFull method. Holdout (dashed) and prequential (drawn) accuracy over time for different initial seed sizes $|\mathcal{S}|$ ($\alpha$=-0.4, $\lambda$=0.5).**

as only one label occurs among the stream allowing a chance classifier to be as good as any other classifier.

Beside the contribution of the forward adaptation, figure 9 also shows the influence of the backward adaptation when comparing the full adaptive approach (includes backward adaptation) and the approach when using only forward adaptation. The stability and the kappa of the full adaptive approach is higher than for the forward adaptation only. However the backward adaption only does not perform well as the model are never added new documents that might reflect the underlying population. The backward adaptation improves the performance and the stability while weighting tweets according to their age only when applying it together with the forward adaptation. So, the coupling of forward and backward adaptation (full adaptive) allows to reflect the underlying population very well.

We varied the *usefulness threshold* $\alpha$ in the $(-1, 0)$ range and we show the effect on the accuracy of our method in Figure 6. As we can see, extending the seed set with documents that have expected labels (high value for $\alpha$) influences the performance of the accuracy positively for both holdout and prequential evaluation. In fact, it seems that a $\alpha$ smaller that -0.2 has no influence of the accuracy. Considering the results of the ICA dataset also, we can say that the effect of the $\alpha$ depends on the homogeneity of the dataset. The TwitterSentiment dataset contains a mixture of topics whereas ICA contains only 16 topics; moreover the experiments on the ICA corpus reveal a better accuracy for a high value of $\alpha$. So, when dealing with a rather heterogeneous dataset (like TwitterSentiment) then a small $\alpha$ should be chosen as noisy data are more likely. A homogeneous corpus (like ICA) though, requires a larger value for $\alpha$ allowing to adapt more documents that are not noise but have unexpected labels. The experiments on kappa show the same trend while having a value around 0.4.

The experiments on the *decay rate* $\lambda$ and the *size of the initial seed set* reveal the same trend as for the ICA corpus with slightly different values, i.e. slow forgetting (small value for $\lambda$) and a high number of documents in the seed has a positive influence on the performance (cf. figure 4 and 8).

## 5. CONCLUSION

We study the problem of opinion stream classification with only a small initial seed of labeled documents. Unlike semi-supervised learning on static data, stream classification with only an initial training set must cope with concept drift.

We propose an opinion stream classifier that uses two mechanisms to adapt to drift: forward and backward adaptation. Forward adaptation involves selecting only some of the arriving unlabeled documents for incorporation into the training set; these are documents, on whose derived label the classifier is confident, but also documents that are different from those seen thus far. We quantify the notion of usefulness for these documents, using entropy as basis. Backward adaptation involves ageing of old documents, gradually eliminating them from the model. Albeit window-based stream classification is a widespread strategy, the elimination of old documents from a learned model has not been used in conventional stream classification before (it has been used in relational stream classification [13]).

Our experiments on streams of opinionated tweets show that our method is competitive in comparison to a fully supervised method. We further show that the interplay of forward and backward adaptation ensures classifier stability in the presence of drift: backward adaption ensures that the influence of old documents is removed from the model after some time, so that the model is more oriented towards new documents.

Future steps include the study of more elaborate mechanisms for the selection of informative new documents in the forward adaption phase. We further want to investigate how our semi-supervised opinion stream classification method can be used to study how the attitude of people towards products or product features changes over time.

## 6. REFERENCES

[1] M. W. Bartosz Krawczyk. Incremental learning and forgetting in one-class classifiers for data streams. In *Proc. of the 8th Int. Conf. on Computer Recognition Systems CORES 2013, Milkow, Poland*, 2013.

[2] A. Bifet and E. Frank. Sentiment knowledge discovery in twitter streaming data. In *Discovery Science*, 2010.

[3] A. Bifet, G. Holmes, and B. Pfahringer. Moa-tweetreader: real-time analysis in twitter streaming data. In *Proc. of the 14th Int'l. Conf. on Discovery science*, DS'11, pages 46–60, 2011.

[4] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà. New ensemble methods for evolving data streams. In *KDD 09*, pages 139–148. ACM, 2009.

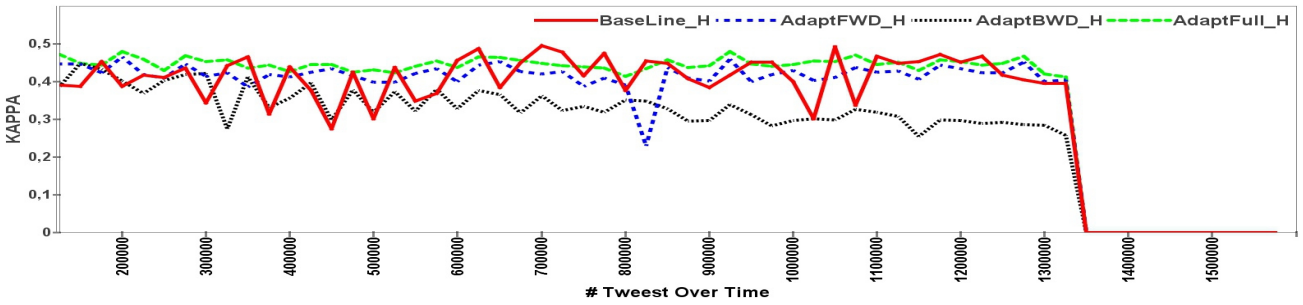


**Figure 9: TwitterSentiment dataset: Holdout Kappa over time for AdaptFull, AdaptBWD, AdaptFWD, and MNB_Baseline ($\alpha$=0.0, $\lambda$=0.1, $|\mathcal{S}|$=100000).**

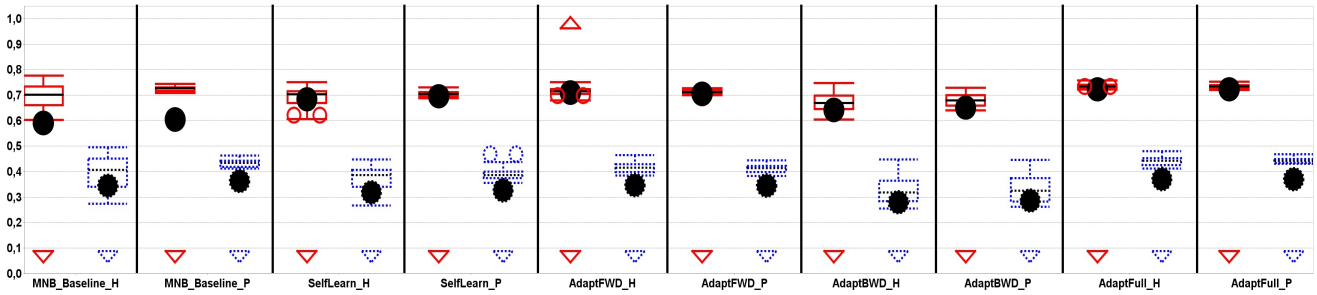


**Figure 10: TwitterSentiment dataset: Kappa (dashed) and accuracy (drawn) over time for the different classifiers and for both holdout (_H) and prequential (_P) evaluation ($\alpha$=0.0, $\lambda$=0.1, $|\mathcal{S}|$=100000).**


[5] B. Drury, L. Torgo, and J. J. Almeida. Classifying news stories with a constrained learning strategy to estimate the direction of a market index. *IJCSA*, 9(1):1–22, 2012.

[6] K. B. Dyer and R. Polikar. Semi-supervised learning in initially labeled non-stationary environments with gradual drift. In *The 2012 Int. Joint Conf. on Neural Networks (IJCNN)*, 2012.

[7] S. Fralick. Learning to recognize patterns without a teacher. *IEEE Trans. Inf. Theor.*, 13(1):57–64, Jan. 1967.

[8] J. Gama, P. Medas, G. Castillo, and P. Rodrigues. Learning with drift detection. In *Advances in Artificial Intelligence–SBIA 2004*, pages 286–295. Springer Berlin Heidelberg, 2004.

[9] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6, 2009.

[10] 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.

[11] E. Ntoutsi, A. Zimek, T. Palpanas, P. Kroger, and H.-P. Kriegel. Density-based projected clustering over high dimensional data streams. In *Proc. of the 12th SIAM Int. Conf. on Data Mining (SDM), Anaheim, CA*, 2012.

[12] C. Priess. Sentiment analysis in Twitter. Bachelor thesis, LMU, Munich, 2012.

[13] Z. F. Siddiqui and M. Spiliopoulou. Tree induction over a stream of perennial objects. In *Proc. of 22nd Int. Conf. on Scientific and Statistical Database Management, SSDBM '10*, volume 6187 of *LNCS*, pages 640–657. Springer, 2010.

[14] Z. F. Siddiqui and M. Spiliopoulou. Classification rule mining for a stream of perennial objects. In *Proc. of 5th Int. Symposium on Rules: Research Based, Industry Focused (RuleML-2011), collocated with IJCAI 2011)*, Barcelona, Spain, July 2011.

[15] I. S. Silva, J. Gomide, A. Veloso, W. Meira, Jr., and R. Ferreira. Effective sentiment stream analysis with self-augmenting training and demand-driven projection. In *Proc. of the 34th Int'l. ACM SIGIR Conf. on Research and development in Information Retrieval*, pages 475–484. ACM, 2011.

[16] A. Sinelnikova. Sentiment analysis in the Twitter stream. Bachelor thesis, LMU, Munich, 2012.

[17] A. Sinelnikova, E. Ntoutsi, and H.-P. Kriegel. Sentiment analysis in the twitter stream. In *36th Annual Conf. of the German Classification Society (GfKl 2012)*, Hildesheim, Germany, 2012.

[18] M. Zimmermann, E. Ntoutsi, and M. Spiliopoulou. Extracting opinionated (sub)features from a stream of product reviews. In *Proc. of the 16th Int. Conf. on Discovery Science (DS'2013)*, volume 8140 of *LNCS*, pages 340–355, Singapore, Oct. 2013. Springer.

[19] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with evolving streaming data. In *Proc. of ECML PKDD 2011*, volume 6913 of *LNCS*. Springer-Verlag, 2011.