Incremental Active Opinion Learning Over a Stream of Opinionated Documents

Max Zimmermann
Swedish Institute of Computer Science (SICS Swedish ICT)
E-164 29 Kista, Sweden
max.zimmermann@sics.se

Eirini Ntoutsi
Ludwig-Maximilians University
Munich 80538, Germany
ntoutsi@dbs.ifi.lmu.de

Myra Spiliopoulou
Otto-von-Guericke University
Magdeburg 39106, Germany
myra@iti.cs.uni-magdeburg.de

ABSTRACT

Applications that learn from opinionated documents, like tweets or product reviews, face two challenges. First, the opinionated documents constitute an evolving stream, where both the authors's attitude and the vocabulary itself may change. Second, labels of documents are scarce and labels of words are unreliable, because the sentiment of a word depends on the (unknown) context in the author's mind. Most of the research on mining over opinionated streams focuses on the first aspect of the problem, whereas for the second a continuous supply of labels from the stream is assumed. Such an assumption though is utopian as the stream is infinite and the labeling cost is prohibitive. To this end, we investigate the potential of active stream learning algorithms that ask for labels on demand. Our proposed ACOSTREAM[1] approach works with limited labels: it uses an initial seed of labeled documents, occasionally requests additional labels *for documents* from the human expert and incrementally adapts to the underlying stream while exploiting the available labeled documents. In its core, ACOSTREAM consists of a MNB classifier coupled with "sampling" strategies for requesting class labels for new unlabeled documents. In the experiments, we evaluate the classifier performance over time by varying: (a) the class distribution of the opinionated stream, while assuming that the set of the words in the vocabulary is fixed but their polarities may change with the class distribution; and (b) the number of unknown words arriving at each moment, while the class polarity may also change.[2] Our results show that active learning on a stream of opinionated documents, delivers good performance while requiring a small selection of labels.

Keywords

opinion mining, active learning, stream mining

[1]Source code is available in R at: `https://www.dropbox.com/s/y2ptl486f4rvohx/acostream_src.zip?dl=0`

[2]Datasets are available at: `https://www.dropbox.com/s/gcpcyazp7fqentb/streams_acostream.zip?dl=0`

This paper was presented at the Fourth International Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM 2015), held in conjunction with KDD'15 in Sydney on 10 August 2015.

1. INTRODUCTION

New communication media promote sharing social content conveniently, e.g. opinions, ideas, thoughts etc., with everyone connected to the *WWW*. Blogs, social networks and microblogging are the common services to pose experiences [4]. Peoples contributions to such services ordered by time of their publication constitute a *stream of opinions*.

An opinion is represented by a document that conveys sentiment; some of its words have a polarity, but these word polarities do not necessarily determine the polarity of the document. On the other hand, a word appears in many opinionated documents, and the polarity of these documents gives an indication on whether this word is used to describe positive or negative sentiment. Moreover, polarity learning on a stream of documents is driven by scarcity of labeled data, since up to date labeled reviews or tweets are not available – it is impractical to expect that a human expert inspects and labels arriving reviews or tweets on sentiment, especially in an infinite data stream scenario [16]. In this study, we investigate how the active acquisition of labels on *document* polarity can contribute to learning and adapting upon an ongoing stream of documents.

According to Mohri [20], the goal of active learning is to achieve a performance comparable to the standard supervised learning scenario, but with fewer labeled examples. Model inference and adaption over streams lends itself to active learning, since the acquisition of fresh labels for all documents of an ongoing fast stream is impracticable. However, learning polarity on streams is subject to two challenges. First, the vocabulary evolves, as new words show up, and as the positive/negative connotation of some words changes. Second, the document polarity model evolves, in the conventional sense of concept drift – the likelihood of one polarity class becomes higher than before. Most conventional polarity stream mining algorithms, including active learning variants address drift of the document polarity model but assume that the vocabulary is fixed and known in advance [15, 3]. In this work, we propose an active stream learning approach for evolving feature spaces. In the core of our approach, there is a Multinomial Naive Bayes (MNB) classifier, which allows for an easy maintenance of class and word-class statistics over time.

In our earlier work [31, 25], we proposed polarity stream learning algorithms that adapt to an evolving vocabulary in the stream. However, in [25] we assume that fresh docu-

ment labels are made available at each moment, while in [31] we assume solely an initial seed of labeled documents and then we adapt the model in a semi-supervised way. On the contrary, in this study, we propose an active stream learning algorithm which requests document labels on demands based on the need for adapting the classifier to the underlying stream.

This work is organized as follows. Related work is discussed in Section 2. The basic concepts of ACOSTREAM, the incrementally updating process and the sampling strategies for document label acquisition are presented in Section 3. Experimental results are shown in Section 4. Conclusions and open issues are discussed in Section 5.

2. RELATED WORK

Active learning is a prominent choice when dealing with problems where labeled data are expensive to obtain, e.g. polarity classification or computational biology applications. There exist various active learning approaches, provided in recent surveys such as [5, 21]. They differ in their heuristics to select instances for which the true label is requested. Garnett et al. [7] use the most likely or the most pessimistic posterior $P(c|d)$ made by a current model. In contrast Krempl et al. [12] and Ho et al. [10] weight the posteriors by their likelihood resp. use hypotheses testing to include the reliability of the posterior when selecting the next instance. All these approaches follow the same framework: they select the next instance and relearn the classifier with the new instance. Relearning is expensive in terms of runtime when dealing with large streams as we do. Our approach works incrementally, thus it does not require relearning rather is expands the current model with new instances.

In context-sensitive learning, it is assumed that the label of a word depends on the context it is used in. Methods that trace recurring concepts [6, 13] and those that monitor context change [17, 9] can trace the association of a word to a label, but only for a limited number of existing contexts, respectively recurring concepts, and for a fixed vocabulary. Therefore, we concentrate on learning with an evolving vocabulary without making assumptions about concept recurrence or context switching.

Zliobaite et al. [32] propose two sampling strategies which are flexible towards a growing collection as well as considering concept change. The latter is covered while allowing the learner to select also samples which are not close to the decision boundary, i.e. for which the classifier is very certain, so that the classifier will not miss concept change. Boy et al. [2] test uncertainty and relevance [3] sampling with different classifiers. It is used to acquire more examples from a class which is scarce. Their results expose that Multinomial Naive Bayes (MNB) classifier performs best for both sampling techniques on polarity classification. We also use MNB as classifier.

Yerva et al. [26] propose an active stream learning based classifier for classifying tweets into relevant or irrelevant for a given company. Their idea is to built a company profile of positive and negative words and test the tweet against the profile to decide on its class. The profile is maintained online over the stream; initially a small set of words is included but the seed set is expanded by also including words that co-occur often in the stream with words in the seed set. We also expand in a word-basis, however our approaches are broader rather than topic specific.

Recently Kranjc et al. [11] present an active learning framework for selecting the most suitable tweets w.r.t. an initial trained classification model. They use as a Support Vector Machine and re-build the model as soon as new suitable tweets are selected. They select suitable tweets based on uncertainty and random sampling. Similarly [22] contribute an active learning approach distinguishing opinionated (positive and negative) from non-opinionated (neutral) tweets in finance twitter data streams. Based on an SVM classifier, Smailovic et al. determine a query strategy for active learning, combining advantages from uncertainty and random sampling.

We skip a discussion on the most recent polarity classification algorithms such as Socher et al. [23] as the contribution of our work is towards active learning strategies for polarity classification rather than pure polarity classification.

3. ACTIVE OPINION STREAM LEARNING

We observe a growing collection $\mathcal{D}$ of documents that constitute a stream, which we monitor at distinct timepoints $t_0, t_1, \ldots, t_i, \ldots$. Documents arrive at each t_i. A document $d \in \mathcal{D}$ is represented by the *bag-of-words* model, i.e. $d = w_1, w_2, \cdots, w_n$. We further assume an *initial labeled seed set* $\mathcal{S}$ of documents: for each $d \in \mathcal{S}$, an expert has assigned a polarity label $c \in C$ (C is the set of possible labels, e.g., positive, negative). We borrow the notation of *initial seed set* from our previous work proposed in [29]. As the stream progresses the concept of words might change, i.e. a word which is used to express positive polarity might change its contextual relation so that it is used to expressing negative thoughts. Moreover, new words - previously unknown words - might appear as peoples' vocabulary to express their positive or negative opinion evolves over time. The mining goal is to assess the polarity label of incoming documents while considering concept change and new words in the stream.

3.1 ACOSTREAM Overview

An overview of our approach is depicted in Algorithm 1. Briefly, it works as follows: The seed set $\mathcal{S}$ is used to initially train a classifier $\Delta(\mathcal{S})$ upon the true labels of $\mathcal{S}$; the document labels are propagated to their component words; this way the vocabulary V (line 2) is derived. The vocabulary consists of the words observed in $\mathcal{S}$ and their distribution in the positive, negative class. Note that these counts are adequate to approximate the class-conditional word probabilities and the class probabilities in MNB. We employ the classifier to predict the label for each arriving new document d from the stream (line 4). Depending on the active learning sampling strategy (cf. Section 3.3), we might request the true label c for d by an expert (line 7). If this is the case, we update the related word-class counts and class counts in the model, for all words appearing in d and the true label c of d (lines 8-10). If we encounter some new word, i.e., not in the current vocabulary, we expand the vocabulary accord-

[3]Relevance sampling regards the labeling of those examples which are most likely to be class members [14].

ingly and start monitoring their occurrences in the different classes (lines 10-12). Moreover we update the documents-class counts and the seed set while adding documents to $\mathcal{S}$ (lines 13-14).

Note that the classifier's predictions are always made on the current (updated) seed set $\mathcal{S}$. That is, the classifier is a *lazy learner*. Moreover, the seed set consists always of true-labeled documents, i.e., labeling was done by an expert. This implies, that the classifier is always trained upon true labeled (and therefore, reliable) instances.

Algorithm 1: ACOSTREAM

Input: initial seed $\mathcal{S}$, stream $\mathcal{D}$

1 $\Delta \leftarrow$ train initial classifier on seed $\mathcal{S}$; predictedLabels $\leftarrow \emptyset$
2 $V \leftarrow$ extract all words from $\mathcal{S}$
3 **while** $\mathcal{D}$ **do**
4 $d \leftarrow$ next document from stream
5 $p \leftarrow$ predict label for d by $\Delta(\mathcal{S})$
6 **if** *d is sampled w.r.t. p* **then**
7 c $\leftarrow$ request true label for d
  `// incrementally update word-class counts`
8 **for** *i=1* **to** $|d|$ **do**
  `// for existing words`
9 **if** $w_i \in V$ **then** $N_{ic} = N_{ic} + 1$
  `// for new words`
10 **else**
11 $N_{ic} = 1$
12 $V \leftarrow V \cup w_i$ `// expand vocabulary`
13 $N_c = N_c + 1$ `// update class counts`
14 $\mathcal{S} \leftarrow \mathcal{S} \cup d$ `// update seed set`

We provide more details in the next subsections.

3.2 Building and Maintaining a Polarity Classifier Over Time

Based on the initial seed set $\mathcal{S}$, we propagate the class labels of the documents to their component words $w_i \in V$, where V is the set of words derived from the documents $d \in \mathcal{S}$. We obtain for each word the word-class counts N_{ic} stating the number of times w_i has occurred in documents with class label c, i.e.

$$N_{ic} = |\{w_i : \exists d \in \mathcal{S}, w_i \in d \wedge class(d) = c\}|$$

We further derive the document class counts N_c expressing the number of documents with label c, i.e.

$$N_c = |\{d : class(d) = c\}|$$

Upon the class and word-class counts we compute the empirical class distributions $\hat{P}(c)$ w.r.t. class c and the empirical word-class distributions $\hat{P}(w_i|c)$ for each word $w_i \in V$, as described in the following section. We use a "hat" as in $\hat{P}$ to denote empirical estimates hereafter.

Framing the empirical distributions we build a Multinomial Naive Bayes classifier Δ. It is very fast for induction, robust to irrelevant attributes, while providing good prediction performance [18]. We assess the polarity label of an arriving document d while employing Δ on d:

$$class(d) = \underset{c \in \{+,-\}}{\operatorname{argmax}} \hat{P}(c|d) \propto \hat{P}(c) \prod_{i=1}^{|d|} \hat{P}(w_i|c)$$

That is, the class label of a new document d is the one maximizing the posterior probabilities $\hat{P}(c|d), c \in C$, which depends on the class conditional probabilities of the words in the document and makes the assumption that these words are independent given the class.

The class prior equals to the ratio of documents in $\mathcal{S}$ labeled as c and the total number of labeled documents $|\mathcal{S}|$, i.e.

$$\hat{P}(c) = |N_c|/|\mathcal{S}| \tag{1}$$

Analogously, the conditional probability of a word w_i given a class c equals to the ratio of documents in $\mathcal{S}$ which are labeled as c and contain the word w_i.

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

We apply the Laplace corrector, $1/|V|$, to alleviate the zero frequency problem for words that have not been observed under a given class.

3.3 Actively Selecting Documents to Acquire New Labels

As the stream of documents underlies changes w.r.t. the empirical word-class distributions $\hat{P}(w_i, c)$, the empirical class distributions $\hat{P}(c)$ and new appearing words, the initial classifier $\Delta(\mathcal{S})$ trained upon the initial seed set $\mathcal{S}$ might become outdated over time. The solution is to update the classifier in order to respond to these changes. To this end, we incorporate new documents into the seed set $\mathcal{S}$ and accommodate new words to the vocabulary. We further incrementally update word-class counts N_{ic} and document-class counts N_c. However, we only extend $\mathcal{S}$ by documents which are actively sampled, i.e. for which we requested a true label by an expert. There are different techniques for actively sampling labels for new documents; we instantiate our approach with two alternative strategies, one based on information gain and another based on uncertainty, discussed in the following sections. Our approach, though, can be coupled with different sampling approaches for labeled document acquisition.

3.3.1 Sampling by Information Gain

We select a new document d for the extension of $\mathcal{S}$ that shows a gain in information with respect to the thus far observed word-class distribution of words $w_i \in d$ and the distribution after considering the predicted label for d. The usage of the information gain is motivated by the attribute selection measures used in decision trees [19] and our previous work [30]. It is defined as follows:

Definition 1. [Information Gain] Let d be a new document containing words $w_i \in d$, for which the current classifier Δ predicts, for instance, the positive polarity label $+$. The *Information Gain* of d w.r.t. the predicted label relies

upon the difference in entropy before and after the addition of the new label +.

$$IG(d) = \sum_{w_i \in d \wedge \in V} H(N_{i+}, N_{i-}) - H(N_{i+} + 1, N_{i-}) \quad (3)$$

Here, $H(N_{i+}, N_{i-})$ is the entropy of w_i regarding the two polarity classes + and −, which expresses the purity of the class distribution based solely on w_i. The second term, $H(N_{i+} + 1, N_{i-})$, is the entropy of w_i when considering d and its predicted label, + in this example, as part of the seed set.

The entropy of two positive values $a, b \in \mathbb{N}$ is defined as:

$$H(a, b) = -\left[\frac{a}{a+b} * log_2(\frac{a}{a+b})\right]$$

Documents that increase the information reflect the current classifier very well and also enhance the classifiers performance while following the thus far observed word-class distributions, i.e. the distributions become more pure and thus the predictions are less random. A document that shows a gain in the information w.r.t. a predicted label c is sampled, i.e. the true label provided by an expert is requested and then utilized to update the classifier.

We update the classifier based on the received true label. Considering the predicted and the true label for d, there are two possible scenarios: (i) the predicted label matches the true label, i.e. the classifier is enhanced in its decision when being updated with the true label, and (ii) the predicted label is different from the true label. The latter case occurs if the classifier does not reflect the current concept underlying the stream, i.e. it makes a wrong prediction. The current concept is assumed to be reflected by the true label of the document. Therefore, Δ must be updated with the true label so that the concept of the related word-class distributions can be changed according to the underlying population of the stream. Hence, we do not miss concept change since we update with the true label.

In case of changes in the word-class distribution, the information gain relies on frequent and old words, i.e. words which have appeared in many documents over time, rather than on words that just newly appeared. This is to be preferred as frequent and old words carry more evidence regarding the class. A toy example shall help to depict this: assuming a word w that occurred thus far in 30 positive documents, further a new document d appears bearing w and the classifier predicts the negative label for d, so there is a change in w. The entropy difference for w would be $(1/31) * log_2(1/31)$, this is a small value so that it is likely that there is still a gain in information if the class distribution of the other words in d are promoted by the negative label; and thus d is selected to update the classifier. It is easy to see that the entropy difference regarding w is higher if w has appeared less than 30 times before the change occurs. Hence, we trust more frequent and old words when a change occurs.

It is noted that when computing the information gain we consider only the entropy difference over words $w_i \in d$, instead of all words $w \in \mathcal{S}$, i.e. we do not iterate over all the words.

3.3.2 Sampling by Uncertainty

As a second sampling strategy for acquiring true document labels, we utilize *uncertainty*. The idea of uncertainty sampling is to ask the expert for labeling an instance for which the current classifier is less certain, i.e. for which the certainty is below some fixed threshold α [21]. Since uncertain examples are close to the classifier's decision border, accommodating them makes the predictions of a classifier more distinctive. According to our MNB classifier we use the posterior probability estimates $\hat{P}(+|d)$ and $\hat{P}(-|d)$ computed by MNB as measure for certainty. A low posterior probability means that the classifier is less certain. The uncertainty is then defined as:

Definition 2. [Uncertainty] Let d be a new document and $\Delta(\mathcal{S})$ be the current classifier that computes the posterior probabilities of the two classes (+, -). The predictions of $\Delta(\mathcal{S})$ are considered as uncertain if:

$$\underset{c \in \{+,-\}}{\text{argmax}}\, \hat{P}(c|d) \leq \alpha$$

where α is a value in (0,1).

The parameter α is selected manually: small values ensure only few documents to be sampled and thus to update the classifier with documents very close to the decision boundary. That is, if the threshold is selected too small then the classifier will miss changes. In contrast, bigger values assume more examples to be sampled. They also allow sampling of documents that are far from the border and which might bear concept change. This also implies, more label requests though.

4. EXPERIMENTS

To evaluate ACOSTREAM, we experiment with two real world datasets of opinionated documents (product reviews and tweets). The original streams were modified in order to test the performance of ACOSTREAM in extreme and less extreme cases. A detailed description of the datasets is given in Section 4.1). We compared our ACOSTREAM against several baselines presented in Section 4.2. The results of our experiments are presented in Section 4.3.

4.1 Datasets

Stream StreamJi comes from a dataset first introduced by Yu et al. [28] which contains data crawled from cnet.com, viewpoints. com, reevoo.com and gsmarena.com. The true labels of the reviews were derived by the authors from star-ratings. The reviews cover mostly products and their properties such as “phone”, “firmware” and “price”. We use only reviews describing single product features, after removing very short reviews containing less than 2 adjectives. More details on the dataset and our preprocessing are also provided in our previous work [30]. The StreamJi dataset contains 11.374 product reviews and a vocabulary of 3.048 different words.

Stream TwitterSentiment, first introduced in [8], was collected by querying the (non-streaming) Twitter API for mes-

sages between April 2009 and June 25, 2009. The stream is very heterogeneous regarding the content. The true labels (ground truth) of the tweets were acquired through the Maximum Entropy classifier using emoticons as class labels. The stream also depicts a very strong concept shift towards its end, as only one of the two classes, the negative ones, is observed at the end of the stream. The original stream contains 1.600.000 tweets; we focus on the last part of the stream, tweets 1.235.000 - 1.485.000, reflecting concept drift. The selected dataset consists of 250.000 tweets with a vocabulary of 169.853 different words.

In StreamJi we focused only on adjectives and adverbs for sentiment analysis since, according to [24, 27], these words bear the actual opinion of the author; similar observation were shown in [30]. Stream TwitterSentiment comes with nouns and verbs as stated in [8].

4.1.1 *The effect of new appearing words*

In our experiments we show how ACOSTREAM performs in a continuously expanding vocabulary V,i.e., when new words arrive over time from the stream. To this end, we re-order the original streams so that the number of appearances of words from the initial seed set $\mathcal{S}$ decreases over time whereas the number of new words increases. That is, vocabulary-wise the initial seed set becomes "outdated" w.r.t. the evolving stream. The ordering was done as in our previous work [30].

Based on the ordering procedure, we obtain for each original stream a re-ordered counterpart which begins with documents that contain only words from the initial vocabulary V extracted from $\mathcal{S}$; as the stream progresses, the number of new words increases while documents arrive that also contain words $w \notin V$. In Figure 1 we draw the percentage of known and new words per document over time for the re-ordered versions of the streams averaged over batches of size 42 resp. 5000. In the very beginning all words are known, over time though, the ratio of known words decreases with unknown words dominating the stream.

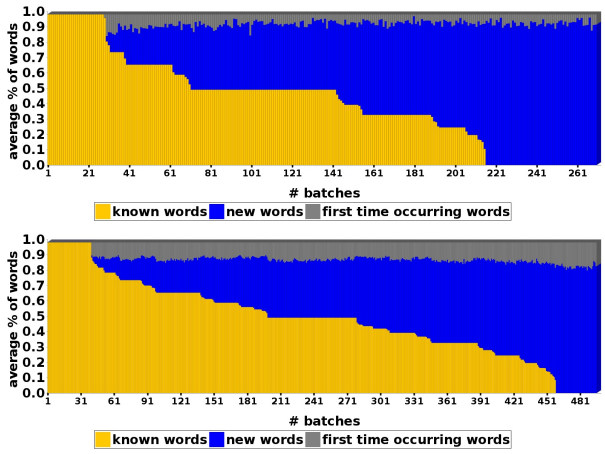

Figure 1: Percentage of known, new and first appearing words over time (avg. per batch) for the re-ordered version of stream StreamJi (top) $|\mathcal{S}|$=140 resp.TwitterSentiment (bottom) $|\mathcal{S}|$=5.000

We distinguish the unknown w.r.t. to the initial seed set words into i) first-time observed new words (in gray) and ii) already monitored new words (in blue). In the *re-ordered* versions in Figure 1, the number of new words is increasing over time and after some point the stream bears merely new words; whereas the number of first-time observed words is rather static over time showing a continuously increasing variety of words. The reason for re-ordering is to show how the classifier deals with an expanding vocabulary.

The class distributions of the streams is depicted in Figure 2: StreamJi is slightly skewed towards the negative class over the whole stream while TwitterSentiment is uniformly distributed at the beginning whereas, as the stream progresses, the distribution moves more towards the positive class.

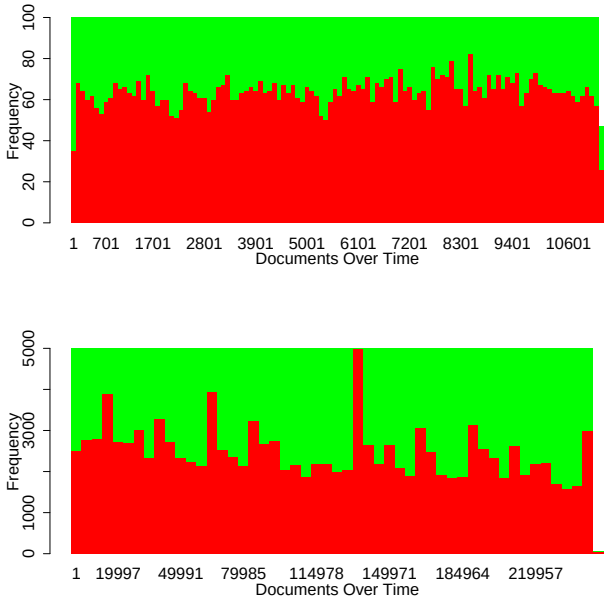

Figure 2: Class distribution on StreamJi (top) and TwitterSentiment (bottom) accumulated over batches of size 100 resp. 5.000

The obtained re-orderings of the original streams bear also changes in the polarity of words, i.e. the word-class distributions changes over time. Figure 3 depicts the word distribution of the words "best" on StreamJi and "tomorrow" on TwitterSentiment as accumulated ratio of documents with positive(green) resp. negative label(red): the distribution of both words change over time, e.g. for word "tomorrow", the ratio of negative documents alternates heavily as for instance, at document 13.800 only negative documents are shown followed by a majority of positive documents.

4.1.2 *Fixed Vocabulary*

The scenario where new words appear over time is an extreme one; though, it is a rather realistic one in polarity learning over streams. To apply our approach on a less extreme scenario, we run experiments on streams showing up *NO* new words over time, i.e. the seed contains all words of the stream.

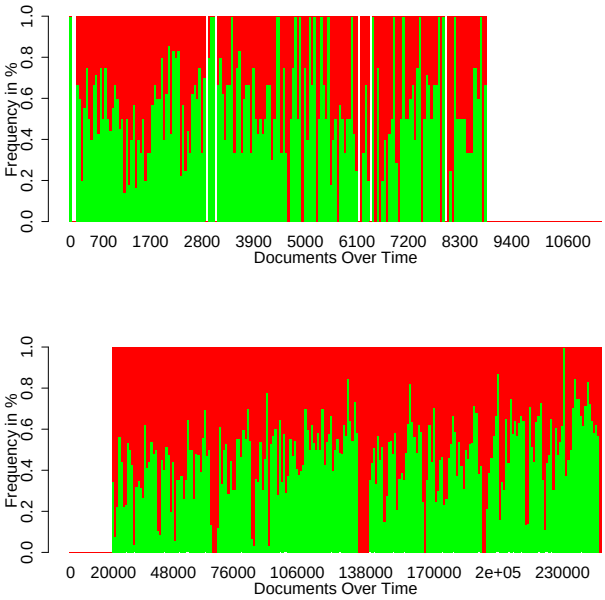

Figure 3: Word-class distribution of the words "best" on StreamJi (top) and "tomorrow" on TwitterSentiment (bottom) accumulated over batches of size 50 resp. 1.000 and depicted as frequency in percentage

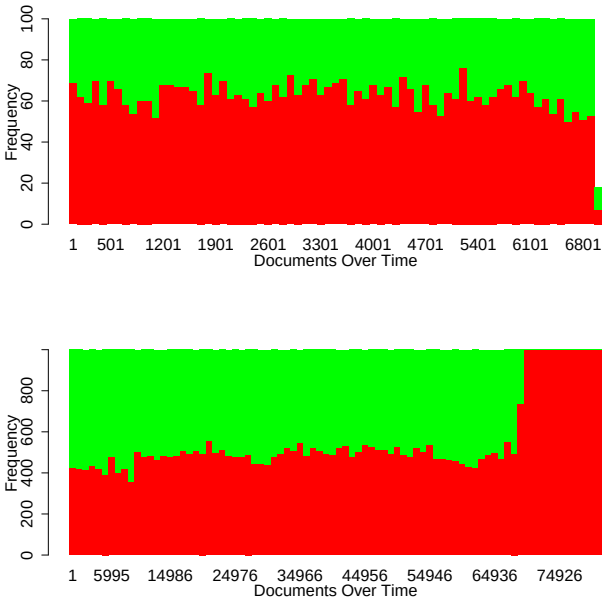

Figure 4: Class distribution of streams StreamJi and TwitterSentiment showing no new words over time w.r.t. seeds with sizes of 1.000 resp. 10.000 documents

Therefore, we reduced the original streams, keeping the *original order* though, to documents that contain only words which are part of the initial vocabulary V extracted from the seed. We acquired the shortened stream while selecting a relatively large seed $\mathcal{S}$ (1.000 for StreamJi and 10.000 for TwitterSentiment). Based on $\mathcal{S}$, we extracted the vocabulary V, and as the stream progresses we considered the documents d that contain only words $w \in V$.

The class distribution of the constituted streams is depicted in Figure 4. We aggregated the number of positive and negative documents over batches of size 100 and 1000 for StreamJi resp. TwitterSentiment. The resulting versions of the stream are smaller than the original version: StreamJi contains 7.018 documents and 759 words while TwitterSentiment covers 81.480 tweets and 14.785 words.

Similar to the re-ordered versions of the streams, described in cf. Section 4.1.1, the shortened stream bears concept change of the words. We skip detailed figures on specific words though as they mostly conform with the word distributions depicted in Figure 3.

4.2 Learning methods and quality measures

Below we outline the approaches we used to compare against ACOSTREAM. They all use Naive Bayes as classifier but differ on which documents they use for adaptation.

- **IncrementalMNB:** The classifier is updated gradually with each incoming instance based on the true labels of the instances. It assumes 100% availability of true labels. This approach serves as an upper baseline.
- **StaticMNB:** The classifier is not updated over time, rather is is trained once upon the initial seed set and remains static over the whole stream. This approach serves as a lower baseline.
- **Random:** The random sampling strategy labels the incoming instances at random instead of deciding actively on the relevance of the label. For every incoming instance the true label is requested with a probability B, where B is the budget [32]. We switch the budget in our experiments among 0,3 and 0,6, e.g., 30% of the documents from the stream are asked for the true label.

To evaluate the quality of our classifiers, we use the *kappa statistic*, which normalizes the classifier's accuracy by the accuracy of a chance classifier: $k = \frac{p_0 - p_c}{1 - p_c}$ [1].

p_0 is the accuracy of a classifier and p_c is the probability of making a correct prediction by a chance classifier that assigns the same number of examples to each class as the classifier under consideration. The kappa varies among -1 and 1: a value ≤ 0 indicates that the classifier's predictions coincide with, or are worse, than the predictions of the chance classifier. A value > 0 implies that the classifier's predictions overcome these of a chance classifier. The higher the value, the more often the predictions match with the true labels. Kappa is preferred to accuracy for data streams as it can handle imbalanced class distributions.

4.3 Performance evaluation

In this section, we compare ACOSTREAM using *information gain* and *uncertainty* sampling strategies against the *IncrementalMNB*, the *StaticMNB* as well as the *random* sampling based on the performance of kappa over time. As we deal with an evolving stream of documents a fixed budget of true labels cannot be utilized which is normally applied when comparing across different sampling strategies [32]. This would, however, lead to an unfair comparison as the budget would be spent differently among the strategies. Rather we used different values for the uncertainty threshold α and for random sampling across our experiments yielding to different number of requested labels over the stream. We depict the number of requested labels over the stream in percentage of the stream length in Table 1: *IncrementalMNB* always asks for 100% of the labels, while *StaticMNB* uses only the true labels of the training set $\mathcal{S}$. We implemented two experiments: i) we kept the vocabulary fixed over the stream while considering documents that contain only words $w \in V$, cf. Section 4.1.1, and ii) we allow the set of words V to evolve as including new appearing words.

In the following we examine the performance of ACOSTREAM on the two experiments comparing against the baselines described in Section 4.2.

4.3.1 Results on the Fixed Vocabulary Stream

We report on the results carried out from our experiments upon streams with a fixed set of vocabulary and evolving word-class counts, cf. Section 4.1.2, and while using kappa as evaluation measure. Figure 5 depicts the kappa over time for ACOSTREAM using information gain (*Acostream_ig*) and uncertainty (*Acostream_u*) sampling, *IncrementalMNB*, *staticMNB* and *Random* sampling on the shortened streams StreamJi (upper picture) and TwitterSentiment lower picture.

ACOSTREAM shows a good performance on both streams when applying information gain sampling. The results expose, upon stream StreamJi, a kappa that is rather close to the kappa of the upper baseline while utilizing only 44% of the true labels (cf. Table 1); on TwitterSentiment it overcomes the *IncrementalMNB* in most times of the stream using only 40% of the labels. Hence, information gain sampling performs very well requesting only 40% resp. 44% of the labels to achieve a comparable or higher kappa than *IncrementalMNB* which samples 100% of the labels. In contrast, uncertainty sampling, which uses 40% resp. 47% of the labels on StreamJi resp. TwitterSentiment, shows a lower kappa similarly to the results obtained by random sampling. The reason why kappa drops to 0 at the end of TwitterSentiment is because there only negative documents arrive, consequently one cannot be better than a chance classifier.

4.3.2 Results on continuously expanding vocabulary

We examine how the performance of ACOSTREAM is affected by a continuously expanding vocabulary and evolving word-class distributions. On stream StreamJi information gain sampling performs very well showing the highest and most robust kappa over time among all approaches to which we compare using only 60% of the labels, depicted by the picture on top of Figure 6, Uncertainty sampling does not perform well on stream StreamJi showing a lower

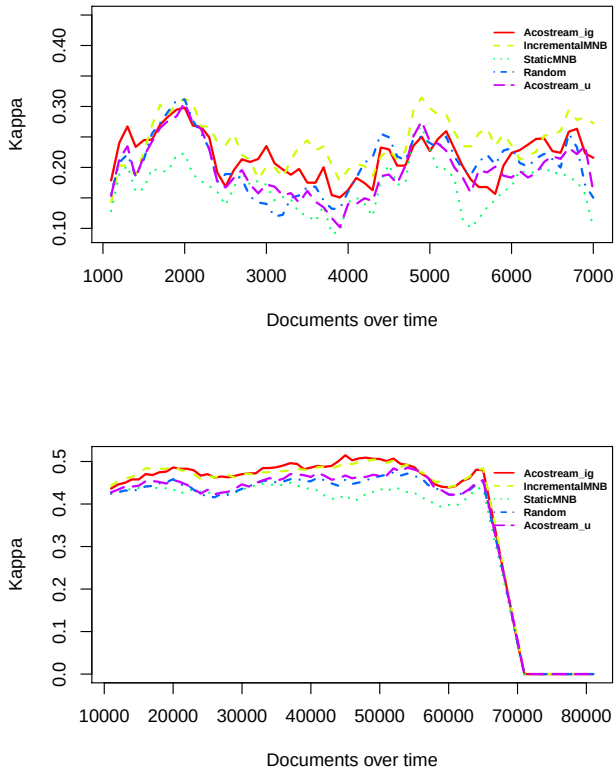

Figure 5: Kappa for the three methods to which we compare and ACOSTREAM on stream StreamJi (top) and TwitterSentiment (bottom) with a fixed vocabulary

Experiment + Dataset	ACOSTREAM (IG)	ACOSTREAM (U)	IncrementalMNB	StaticMNB	Random
fixed V: StreamJi	44	40	100	1	40
fixed V: TwitterSentiment	40	47	100	1	42
evolving V: StreamJi	60	59	100	1	60
evolving V: TwitterSentiment	52	88	100	2	31

Table 1: Requested labels per method and experiment: numbers in percentage regarding the length of the stream including the documents to train the classifier, i.e. the size of the seed. (IG=Information Gain), (U=Uncertainty)

kappa in comparison to random sampling. On TwitterSentiment it performs well but requiring 88% of the labels to be competitive with ACOSTREAM when using information gain sampling that acquires only 52% of the labels. The results on stream TwitterSentiment, depicted by the bottom picture of Figure 6, reveal that *IncrementalMNB* performs best on large streams with many words (169.853). ACOSTREAM (both sampling strategies) follows while showing a similar pattern of the kappa curve but with slightly lower values.

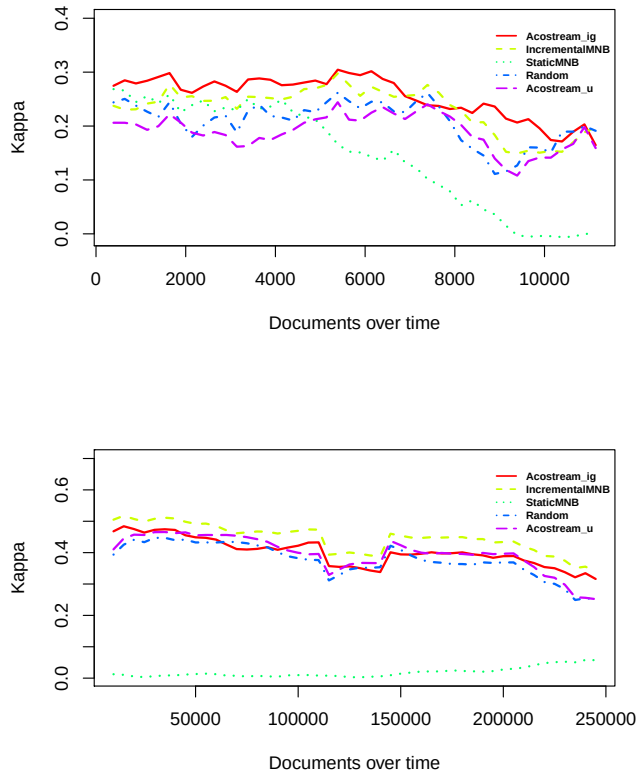

Figure 6: Kappa for the three methods to which we compare and ACOSTREAM on stream StreamJi (top) and TwitterSentiment (bottom) under an evolving vocabulary

ACOSTREAM is not negatively affected by new words and exposes a stable performance across both streams. Also, the curves show a pattern similar to the one obtained from the *IncrementalMNB* that adapts with all documents of the stream and thus considers all changes of the word distributions. That is, ACOSTREAM, in particular when information gain is used, adapts well to the underlying change in the population of the stream.

4.3.3 *Effect of the uncertainty threshold α*

To show the effect of the uncertainty threshold α, cf. Section 3.3.2, we varied values of α on stream StreamJi and TwitterSentiment when the vocabulary has a fixed size. Figure 7 depicts kappa over time on StreamJi (upper picture) and TwitterSentiment (lower picture) for different settings of α when uncertainty is used as sampling strategy upon ACOSTREAM. We varied among five values: e(-2), e(-10), e(-20), e(-30) and e(-40), where *e()* is the exponential function. Note that the posteriors become rather small as dealing with a sparse feature space. Thus we had to set small values for α in order to cause difference in the consumption of labels.

The results on both streams show and increasing performance while taking larger values for α into account. This is not surprising, as with increasing α also the number of considered samples grows which intuitively leads to a better performance. The gap in performance among values for α is huge on stream StreamJi where 100%, 87%, 33%, 19%, and 15% percent of labels are requested; while on TwitterSentiment 100%, 92%, 74%, 55% and 40% of the documents are sampled, leading to smaller gaps between the curves.

5. CONCLUSION

Polarity learning on an evolving stream is a challenging task as the stream is subject to concept changes; existing words might change sentiment over time due to e.g., different context, but also new words might occur to express opinions. Another challenge for a stream polarity learner is the scarcity of the class labels, assuming manual labeling of the (infinite) stream is unrealistic. Responding to these challenges requires adaptation of the model to the underlying stream population based on only a few labeled examples.

In this work, we proposed our active stream learning framework ACOSTREAM for incrementally updating a polarity learner based on actively acquired document labels. We instantiate our framework with two sampling strategies, *information gain* and *uncertainty*. We compare our method to a traditional active learning approach (random sampling), an incremental approach that requires all arriving document labels and a non-adaptive method. Our results show that actively asking for labels, pays off as the performance of the classifier is quite good while the label consumption remains low. Comparing the two sampling approach, *information gain*-based sampling shows good performance on all datasets

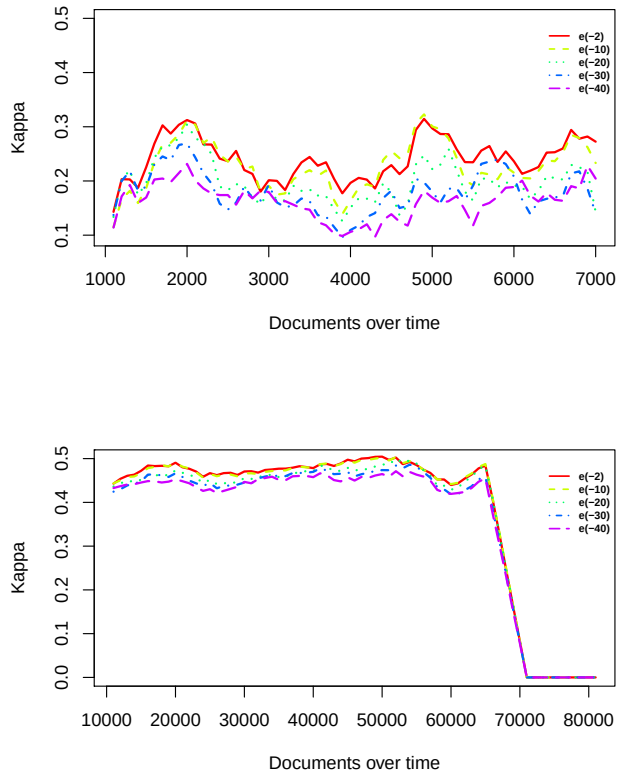

Figure 7: Kappa for different settings of α when using uncertainty as sampling strategy for ACOSTREAM on stream StreamJi (top) and TwitterSentiment (bottom)

w.r.t. the number of required labels, the accuracy of predictions and adaptation to concept change. The *uncertainty*-based sampling on the contrary shows a poor performance.

Our ongoing work involves more elaborated techniques on propagating document labels to words, considering that not all words contribute the same to the polarity of a document. Furthermore, we want to diminish independence between new documents when deciding to sample them. This will allow us to sample in a wider prospect detecting change early and address emerging scenarios comprehensively. Moreover, we plan to instantiate ACOSTREAM with different classifiers (except for the currently employed MNB) and different sampling strategies for active learning.

6. ACKNOWLEDGMENT

The work of Max Zimmermann was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme.

7. REFERENCES

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

[2] E. Boiy and M. francine Moens. A machine learning approach to sentiment analysis in multilingual web texts. *Information Retrieval*, pages 526–558, 2009.

[3] P. H. Calais Guerra, A. Veloso, W. Meira, Jr., and V. Almeida. From bias to opinion: A transfer-learning approach to real-time sentiment analysis. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 150–158, New York, NY, USA, 2011. ACM.

[4] E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2):15–21, Mar. 2013.

[5] Y. Fu, X. Zhu, and B. Li. A survey on instance selection for active learning. *Knowl. Inf. Syst.*, 35(2):249–283, 2013.

[6] J. Gama and P. Kosina. Recurrent concepts in data streams classification. *Knowl. Inf. Syst.*, 40(3):489–507, Sept. 2014.

[7] R. Garnett, Y. Krishnamurthy, X. Xiong, J. Schneider, and R. P. Mann. Bayesian optimal active search and surveying. In J. Langrod and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, pages 1239—1246, Madison, WI, USA, 2012. Omnipress.

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

[9] H. Gu, X. Xie, Q. Lv, Y. Ruan, and L. Shang. Etree: Effective and efficient event modeling for real-time online social media networks. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '11, pages 300–307, Washington, DC, USA, 2011. IEEE Computer Society.

[10] S.-S. Ho and H. Wechsler. Query by transduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(9):1557–1571, Sept. 2008.

[11] J. Kranjc, J. Smailović, V. Podpečan, M. Grčar,

M. Žnidaršič, and N. Lavrač. Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the clowdflows platform. *Information Processing & Management*, Apr. 2014.

[12] G. Krempl, D. Kottke, and M. Spiliopoulou. Probabilistic active learning: Towards combining versatility, optimality and efficiency. In *Discovery Science - 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014. Proceedings*, pages 168–179, 2014.

[13] M. Lazarescu. A multi-resolution learning approach to tracking concept drift and recurrent concepts. In H. Gamboa and A. L. N. Fred, editors, *PRIS*, page 52. INSTICC Press, 2005.

[14] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 3–12, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[15] R. Lourenco Jr., A. Veloso, A. Pereira, W. Meira Jr., R. Ferreira, and S. Parthasarathy. Economically-efficient sentiment stream analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research; Development in Information Retrieval*, SIGIR '14, pages 637–646, New York, NY, USA, 2014. ACM.

[16] M. M. Masud, C. Woolam, J. Gao, L. Khan, J. Han, K. W. Hamlen, and N. C. Oza. Facing the reality of data stream classification: coping with scarcity of labeled data. *Knowl. Inf. Syst.*, 33(1):213–244, 2011.

[17] M. Mathioudakis and N. Koudas. Twittermonitor: Trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 1155–1158, New York, NY, USA, 2010.

[18] 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.

[19] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[20] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[21] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[22] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič. Stream-based active learning for sentiment analysis in the financial domain. *Information Sciences*, Apr. 2014.

[23] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. P. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.

[24] P. D. Turney. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proc. of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. ACL.

[25] S. Wagner, M. Zimmermann, E. Ntoutsi, and M. Spiliopoulou, editors. *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 07-11, 2015. Proceedings*, Lecture Notes in Computer Science. Springer, 2015. to appear.

[26] S. R. Yerva, Z. Miklós, and K. Aberer. Entity-based classification of twitter messages. *IJCSA*, 9(1):88–115, 2012.

[27] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 129–136, Stroudsburg, PA, USA, 2003. ACL.

[28] J. Yu, Z.-J. Zha, M. Wang, K. Wang, and T.-S. Chua. Domain-assisted product aspect hierarchy generation: Towards hierarchical organization of unstructured consumer reviews. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 140–150, Stroudsburg, PA, USA, 2011.

[29] M. Zimmermann, E. Ntoutsi, and M. Spiliopoulou. Adaptive semi supervised opinion classifier with forgetting mechanism. In *Proc. of the 29th Annual ACM Symposium on Applied Computing*, SAC'14. ACM, 2014.

[30] M. Zimmermann, E. Ntoutsi, and M. Spiliopoulou. A semi-supervised self-adaptive classifier over opinionated streams. In *2014 IEEE International Conference on Data Mining Workshops, ICDM Workshops 2014, Shenzhen, China, December 14, 2014*, pages 425–432, 2014.

[31] M. Zimmermann, E. Ntoutsi, and M. Spiliopoulou. Discovering and monitoring product features and the opinions on them with OPINSTREAM. *Neurocomputing*, 150:318–330, 2015.

[32] 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.