Concept Drift Awareness in Twitter Streams

Joana Costa*†, Catarina Silva*†, Mário Antunes*‡, Bernardete Ribeiro†

*School of Technology and Management, Polytechnic Institute of Leiria, Portugal
{joana.costa,catarina,mario.antunes}@ipleiria.pt
†Center for Informatics and Systems, University of Coimbra, Portugal
{joanamc,catarina,bribeiro}@dei.uc.pt
‡Center for Research in Advanced Computing Systems, INESC-TEC, University of Porto, Portugal
mantunes@dcc.fc.up.pt

***Abstract*—Learning in non-stationary environments is not an easy task and requires a distinctive approach. The learning model must not only have the ability to continuously learn, but also the ability to acquired new concepts and forget the old ones. Additionally, given the significant importance that social networks gained as information networks, there is an ever-growing interest in the extraction of complex information used for trend detection, promoting services or market sensing. This dynamic nature tends to limit the performance of traditional static learning models and dynamic learning strategies must be put forward.**

In this paper we present a learning strategy to learn with drift in the occurrence of concepts in Twitter. We propose three different models: a time-window model, an ensemble-based model and an incremental model. Since little is known about the types of drift that can occur in Twitter, we simulate different types of drift by artificially timestamping real Twitter messages in order to evaluate and validate our strategy. Results are so far encouraging regarding learning in the presence of drift, along with classifying messages in Twitter streams.

I. Introduction

Social networks have become popular in recent years with millions of daily users sharing their everyday activities with friends and family. Users link themselves by defining others to follow, and consequently have their own followers based not only on social relations but also related with topics of interest.

Twitter is one of the most well-known social media platforms, being characterized by providing a microblogging service where users are able to post text-based messages of up to 140 characters, mimicking the SMS (Short Message Service) messages, and known as *tweets*. According to the Twitter website (`http://www.twitter.com`) the broad coverage of this social network is confirmed by having 255 million monthly active users that post 500 million *tweets* per day. Another interesting characteristic of Twitter is the presence of *hashtags*, single words started with the symbol `"#"`, used to classify each message content. Recently, the use of *hashtags* became popular, being adopted by other social networks like Facebook or Instagram, as more roles were identified to the use of *hashtags*, like bringing a wider audience into discussion [1], spreading an idea [2], get affiliated with a community [3], or bringing together other Internet resources [4].

Although mostly considered as an entertainment tool, *tweets* may contain information of broad interest [5] and are being widely studied as they have a wide range of applications and uses, like event detection [6]–[9], academic tool [10]–[12], news media [6], [13] or mining political opinion [14], [15].

Additionally, it is relevant to note how challenging can be to learn in social network environments be, especially in Twitter. Because of its nature of a *small document* social network, users often post on a daily basis and use mostly their mobile devices, which means they can post easily and everywhere, creating a deluge of data in real time. Twitter can be seen as a particular form of a temporal data stream with a considerable amount of noise, not only because it is easy to post, but also because no rules are applied to the posted material. Besides that, concepts appear and disappear often, as users post quickly as an event occur, like an earthquake, but they tend to naturally dissapear after a few days, reoccurring or not some time later. The main focus of our work will be to identify which learning model is better suited to learn in this dynamic environment, where the frequency of concepts drift over time.

To deal with the concept drift in the Twitter stream we propose a threefold approach: a time-window model, an ensemble based model and an incremental model. We propose the simulation of different types of drift by artificially timestamping real Twitter messages in a sequential way, as a way to guarantee that we have a ground truth goal and hence can evaluate and validate our strategy. By studying different types of drift we aim to identify the learning characteristics best tailored to learn in such environments, where each drift might occur.

Our time-window model is characterized by just taking into account recent information, given a time-window, disregarding previously seen examples from time to time. The ensemble model is based on the idea that the use of a committee of classifiers can provide better results than the best of the single classifiers, when correctly combined. To characterize an ensemble-based system two choices must be taken into account: the choice of the classifiers and the choice of the combination function, i.e., the voting algorithm used to combine the output of multiple classifiers into a single decision. Finally, the incremental model is characterized by retaining in one single classifier all the information gathered over time.

Our contribution regarding the ensemble model, is to evaluate how to combine the ensemble members to dynamically update the classifiers' weights, so that the ensemble can learn incrementally and does not need to store previously seen data. The rest of the paper is organized as follows. We start in

978-1-4799-7415-3/14 $31.00 © 2014 IEEE
DOI 10.1109/ICMLA.2014.53

Section II by describing the related work regarding social networks and concept drift. We then proceed in Section III by detailing the proposed approach and in Section IV explaining the experimental setup, including the dataset description, the preprocessing methods and learning and evaluation approaches. In Section V we present and discuss the obtained results. Finally, in Section VI we present the most relevant conclusions and delineate some directions for future work.

II. Related Work

A. Social Networks

Social networks have gained significant importance and are being widely studied in many fields in the last years. Modern challenges in social networks involve not only computer science matters but also social, political, business, and economical sciences. In computer science, and considering our focus on Twitter, recent works comprise event detection [7], [8], information spreading [16], community mining [17], crowdsourcing [18] and sentiment analysis [15].

In [19] we have proposed the use of meta-classes to boost the performance of Twitter messages classification. This preliminary study shed light on the possibility of evaluating message content in order to predict *hashtags*. Regarding Twitter *hashtags*, and particularly *hashtag* recommendation, we have also identified the recent study presented in [20], where an approach for *hashtag* recommendation is introduced. This approach computes a similarity measure between *tweets* and uses a ranking system to recommend *hashtags* to new *tweets*. In [21] the use of *hashtags* to classify Twitter messages is done by clustering similar *tweets* in a graph based collective classification strategy. Although the presented results seem promising, we have identified the lack of adaptiveness in this strategy.A different approach is proposed in [22], where an event detection method is described to cluster Twitter *hashtags* based on semantic similarities between the *hashtags*. This work is in line with our previous work except for the fact that the semantic similarities are computed based on the message content similarities rather than being based on semantic *hashtag* similarities.

B. Concept Drift

In the presence of concept drift, the learning task is not easy and requires a special approach, different from those commonly used, as the arriving instances can not be treated as equally important contributors to the final concept [23]. In non-stationary environments like the Twitter stream, effective learning requires a learning algorithm with the ability to detect context changes without being explicitly informed about them, quickly recover from the context change and adjust its hypothesis to the new context. It should also make use of previous experienced situations when old contexts and corresponding concepts reappear [24].

According to [25], there are 3 approaches to handle concept drift: (1) instance selection, (2) instance weighting and (3) ensemble learning. A review of concept drift applied to intrusion detection is presented in [26].

In [27] the algorithm Learn++.NSE is proposed as an algorithm to deal with drift. To deal with scenarios of imbalanced data, the authors in [28] proposed Learn++.CDS, a combination of the Learn++.NSE algorithm with the SMOTE algorithm proposed by [29]. A different ensemble method called DWM-WIN was recently proposed in [30], to overcome the known limits of [31] namely not considering the time classifiers were define nor the past correct classifications.

The related work presented so far sheds light on the importance of dealing with concept drift specially in dynamic scenarios like social networks, and particularly in Twitter, where important information can be mined. Multiple applications like spam email filtering, intrusion detection, recommendation systems, event detection or improve search capabilities are just pointed examples.

III. Proposed Approach

A. Twitter classification problem

Twitter classification is a multi-class problem that can be cast as a time series of *tweets*. It consists of a continuous sequence of instances, in this case, Twitter messages, represented as $\mathcal{X} = \{x_1, \ldots, x_t\}$, where x_1 is the first occurring instance and x_t the latest. Each instance occurs at a time, not necessarily in equally spaced time intervals, and is characterized by a set of features, usually words, $\mathcal{W} = \{w_1, w_2, \ldots, w_{|\mathcal{W}|}\}$. Consequently, instance x_i is denoted as the feature vector $\{w_{i1}, w_{i2}, \ldots, w_{i|W|}\}$.

When x_i is a labelled instance it is represented as the pair (x_i, y_i), being $y_i \in \mathcal{Y} = \{y_1, y_2, \ldots, y_{|\mathcal{Y}|}\}$ the class label for instance x_i.

We have used a classification strategy previously introduced in [19], where the Twitter message *hashtag* is used to label the content of the message, which means that y_i represents the *hashtag* that labels the Twitter message x_i.

The purpose of this classification problem is to define the unknown predict function $h^t : \mathcal{X} \rightarrow \mathcal{Y}$, that predicts the class label y_i, the *hashtag*, according to x_i, the Twitter message. In a time line perspective, h^t uses the historical data $\{x_1, \ldots, x_t\}$ to predict x_{t+1}. The function h^t is then the Twitter message classifier used to predict the *hashtag* of the set of *tweets* presented in the subsequent time-windows.

Notwithstanding the Twitter message classification is a multi-class problem in its essence, it can be decomposed in multiple binary tasks in a one-against-all binary classification strategy. In this case, a classifier h^t is composed by $|Y|$ binary classifiers.

B. Learning models

For classifying time series like the Twitter stream we have to devise proper learning models. In particular if we want to forecast which topics will become trends on Twitter, we have to carefully choose our guiding hypothesis for this setting. We present three models to tackle the Twitter classification problem: a time-window model, an ensemble model and an incremental model. The time-window model is a batch learning model unable to retain all the previously seen examples. The incremental model learns batches of *tweets* and has a memory mechanism that allows awareness of previously seen examples. Unlike the previous ones, the ensemble model has a modular structure which enables temporal adaptation to new incoming *tweets* on the basis of the data sampling real distribution over time. In a way, the built-in memory mechanism is inherited from the (recent) past.

Algorithm 1 defines the basic steps of the time-window model. For each collection of documents $\mathcal{T}$ in a time-window t, $\mathcal{T}^t = \{x_1, \ldots, x_{|T^t|}\}$ with labels $\{y_1, \ldots, y_{|T^t|}\} \rightarrow \{-1, 1\}$, the dataset $\mathcal{D}^t$ is updated with the newly seen documents. No previously seen documents are stored in $\mathcal{D}^t$ and thus $\mathcal{C}^t$ classifier is always trained with the examples of the most recent time-window.

Algorithm 1: Time-Window Model

Input:
For each collection of documents $\mathcal{T}$ in a time-window t,
$\mathcal{T}^t = \{x_1, \ldots, x_{|T^t|}\}$ with labels
$\{y_1, \ldots, y_{|T^t|}\} \rightarrow \{-1, 1\} \quad t = 1, 2, \ldots$ T

for *t=1,2, ...* T **do**
    $\mathcal{D}^t \leftarrow \mathcal{T}^t$
end
BaseClassifier $\mathcal{C}^t$: Learn ($\mathcal{D}^t$), obtain: h^t: $\mathcal{X} \rightarrow \mathcal{Y}$
Time-Window Classifier $\mathcal{C}^t$: Classify ($\mathcal{T}^{t+1}$), using: h^t: $\mathcal{X} \rightarrow \mathcal{Y}$

Unlike the time-window model, the incremental model uses all the previously seen examples, as can be illustrated in Algorithm 2, by updating the documents collection $\mathcal{D}^t$ in an incremental manner. Even though this model retains all the information gathered over time, one can argue that continuously increasing $\mathcal{D}^t$ would lead to storing problems. This drawback leads to find a new way to circumvent this ever-growing problem.

Algorithm 2: Incremental Model

Input:
For each collection of documents $\mathcal{T}$ in a time-window t,
$\mathcal{T}^t = \{x_1, \ldots, x_{|T^t|}\}$ with labels
$\{y_1, \ldots, y_{|T^t|}\} \rightarrow \{-1, 1\} \quad t = 1, 2, \ldots$

for *t=1,2,...* T **do**
    $\mathcal{D}^t \leftarrow \mathcal{D}^t \cup \mathcal{T}^t$
end
BaseClassifier $\mathcal{C}^t$: Learn ($\mathcal{D}^t$), obtain: h^t: $\mathcal{X} \rightarrow \mathcal{Y}$
Incremental Classifier $\mathcal{C}^t$: Classify ($\mathcal{T}^{t+1}$), using: h^t: $\mathcal{X} \rightarrow \mathcal{Y}$

Therefore, the ensemble model, presented in Algorithm 3, proposes to store all the information gathered in a different way. For each collection of documents $\mathcal{T}$, that contain both positive and negative examples and occur in a time-window t, a classifier $\mathcal{C}^t$ is trained and stored. When a new collection of documents in the subsequent time-window occur, all the previously trained classifiers are loaded, and will classify the newly seen examples. The prediction function of the ensemble, composed by the set of classifiers already created, is a combined function of the outputs of all the considered classifiers. Several strategies can be used herein. We will use a majority voting strategy where each model participates equally. If the sum of all votes is a null value, which means a tie, the classification of the most recent classifier is used to untie. The documents of the previously seen time-windows are not stored in this approach even though the possible learning information is stored along in the classifier trained immediately after it.

Algorithm 3: Ensemble Model

Input:
For each collection of documents $\mathcal{T}$ in a time-window t,
$\mathcal{T}^t = \{x_1, \ldots, x_{|T^t|}\}$ with labels
$\{y_1, \ldots, y_{|T^t|}\} \rightarrow \{-1, 1\} \quad t = 1, 2, \ldots$ T

for *t=1, 2, ...* T **do**
    $\mathcal{D}^t \leftarrow \mathcal{T}^t$
    BaseClassifier $\mathcal{C}^t$: Learn ($\mathcal{D}^t$), obtain: h^t: $\mathcal{X} \rightarrow \mathcal{Y}$
end
for *k=1, ... , t* **do**
    ModuleClassifier $\mathcal{C}^k$: Classify ($\mathcal{T}^{t+1}$), using: h^k: $\mathcal{X} \rightarrow \mathcal{Y}$
end
Ensemble $\mathcal{E}^t$: Classify ($\mathcal{T}^{t+1}$), using:

$$e^t = \begin{cases} \frac{\sum_t h^t(\mathcal{T}^{t+1})}{|\sum_t h^t(\mathcal{T}^{t+1})|} & \text{if } \sum_t h^t(\mathcal{T}^{t+1}) \neq 0 \\ h^t(\mathcal{T}^{t+1}) & \text{if } \sum_t h^t(\mathcal{T}^{t+1}) = 0 \end{cases}$$

IV. Experimental Setup

A. Dataset

There are few works regarding the learning process in the occurrence of concept drift in the particularly field of social networks and little is known about the types of drift that can occur. Hence, we propose to generate a dataset that simulates different types of drift in Twitter by artificially timestamping real *tweets*, in order to evaluate and validate our strategy. By inducing different types of drift with controlled features, we intend to identify the learning characteristics needed to deal with them and thus define best tailored learning features to this specific problem. One should always consider that given the amount of data produced in Twitter, storage can be of major importance.

The drifts we intend to represent are based on the four different major types proposed in [32], namely (i) *sudden*, (ii) *gradual*, (iii) *incremental*, and (iv) *reoccurring*. Thus, we have made use of these four types of drift and have defined ten different drift patterns which cover different behaviours corresponding to classes (hashtags).

Table I shows the different types of drifts sorted by the artificially generated timestamps and their corresponding time-windows. Time is represented as 24 continuous time-windows, in which the frequency of each *hashtag* is changed in order to represent the defined drifts. Each *tweet* is then timestamped so it can belong to one of the time-windows we have defined.

Timewindow	Sudden#1	Sudden#2	Gradual#1	Gradual#2	Incremental#1	Incremental#2	Reoccuring	Regular#1	Regular#2	Regular#3
1	0	0	0	60	0	60	0	20	50	20
2	0	0	10	60	0	60	0	20	50	50
3	0	0	10	0	0	60	0	20	50	20
4	0	0	0	0	0	60	0	20	50	50
5	0	50	0	50	0	60	0	20	50	20
6	0	50	20	50	4	56	50	20	50	50
7	0	50	20	0	8	52	50	20	50	20
8	0	50	0	0	12	48	50	20	50	50
9	0	50	0	40	16	44	0	20	50	20
10	0	50	30	40	20	40	0	20	50	50
11	0	0	30	0	24	36	0	20	50	20
12	0	0	0	0	28	32	0	20	50	50
13	0	0	0	30	32	28	0	20	50	20
14	0	0	40	30	36	24	50	20	50	50
15	50	0	40	0	40	20	50	20	50	20
16	50	0	0	0	44	16	50	20	50	50
17	50	0	0	20	48	12	0	20	50	20
18	0	0	50	20	52	8	0	20	50	50
19	0	0	50	0	56	4	0	20	50	20
20	0	0	0	0	60	0	0	20	50	50
21	0	0	0	10	60	0	0	20	50	20
22	0	0	60	10	60	0	50	20	50	50
23	0	0	60	0	60	0	50	20	50	20
24	0	0	0	0	60	0	50	20	50	50
Total:	**150**	**300**	**420**	**420**	**720**	**720**	**450**	**480**	**1200**	**840**

Table I
FREQUENCY OF HASHTAGS OVER TIME.

Table I also represents the frequency we have defined for each *hashtag* over time.

Analysing Table I, one can also notice that there are 2 instances of sudden, gradual and incremental drifts, 1 instance of a reoccurring drift and 3 with *regular* feeds. For instance, in the reoccurring drift we introduce 50 *tweets* in time-windows 6, 7, and 8 and later in time-windows 14, 15, and 16. Regularity is represented here to show *tweets* that occur in a continuous frequency, i.e. without drift.

Hence, the main idea is to represent drift in the Twitter message classification. Since a Twitter labelled dataset is missing so far, we use the *hashtags* enclosed in the message as the message classification, as previously introduced by the authors in [19].

In order to accomplish this purpose, we start by defining 10 different *hashtags*, one for each defined drift, representing different concepts and hence different classes, such as *realmadrid* and *literature*. By trying to use mutually exclusive concepts we intend to avoid misleading the classifier, since two different *tweets* may represent the same concept. Table II shows the chosen *hashtags* and the corresponding drift.

Drift	Hashtag
Sudden #1	#bradpitt
Sudden #2	#realmadrid
Gradual #1	#ryanair
Gradual #2	#literature
Incremental #1	#twitter
Incremental #2	#ferrari
Reoccuring	#syria
Regular #1	#jobs
Regular #2	#sex
Regular #3	#nowplaying

Table II
CORRESPONDENCE BETWEEN TYPE OF DRIFT AND HASHTAG.

The Twitter API (`https://dev.Twitter.com`) was then used to request public *tweets* that contain the defined *hashtags*. The requests have been made between 19 October 2013 and 31 October 2013 and *tweets* were only considered if the user language was defined as English. We have requested more than 10.000 *tweets*, even though some of them were discarded, like for instance those *tweets* containing no message content besides the *hashtag*.

B. Representation and Pre-processing

A *tweet* is represented as a vector space model, also known as *Bag of Words*. The collection of features is built as the dictionary of unique terms present in the documents collections. Each tweet of the document collection is indexed with the *bag* of the terms occurring in it, i.e., a vector with one element for each term occurring in the whole collection. The *hashtag* was removed from the message content in order to be exclusively used as the document label.

High dimensional space can cause computational problems in text-classification problems where a vector with one element for each occurring term in the whole connection is used to represent a document. Also, over-fitting can easily occur which can prevent the classifier to generalize and thus the prediction ability becomes poor. Pre-processing methods were applied in order to reduce feature space. These techniques, besides reducing the document size, prevent the mislead classification as some words, such as articles, prepositions and conjunctions, are non-informative words. These words, called *stopwords*, occur more frequently than informative ones. *Stopword removal* was then applied, preventing those non informative words from misleading the classification.

Stemming method was also applied. This method consists in removing case and inflection information of each word, reducing it to the word stem. Stemming does not alter significantly the information included, but it does avoid feature expansion.

C. Learning and Evaluation

The evaluation of our approach was done by the previously described dataset and using the Support Vector Machine (SVM). SVM constitute currently the best of breed kernel-

	Time-window	Ensemble	Incremental
Sudden #1	55.93%	58.42%	54.85%
Sudden #2	60.22%	80.12%	88.84%
Gradual #1	49.88%	40.45%	65.21%
Gradual #2	45.08%	74.53%	82.41%
Incremental #1	41.41%	30.69%	60.35%
Incremental #2	52.01%	61.72%	79.45%
Reoccurring	73.59%	82.92%	89.74%
Regular #1	55.78%	55.53%	84.44%
Regular #2	57.69%	88.05%	93.23%
Regular #3	23.71%	30.65%	65.35%
Average:	**51.53%**	**60.31%**	**76.39%**

Table III
COMPARATIVE RESULTS: F_1 MEASURE

based technique, exhibiting state-of-the-art performance in text classification problems [33], [34] and was used in our experiments to construct the proposed models.

In order to evaluate the binary decision task of the proposed models we defined well-known measures based on the possible outcomes of the classification, such as, error rate ($\frac{FP+FN}{TP+FP+TN+FN}$), recall ($R = \frac{TP}{TP+FN}$), and precision ($P = \frac{TP}{TP+FP}$), as well as combined measures, such as, the van Rijsbergen F_β measure, which combines recall and precision in a single score: $F_\beta = \frac{(\beta^2+1)P\times R}{\beta^2 P+R}$.

F_β is mostly used in text classification problems with $\beta = 1$, i.e. F_1, an harmonic average between precision and recall.

V. RESULTS AND DISCUSSION

In this section we evaluate the performance obtained on the Twitter data set using the three approaches described in Section III. Table III summarizes the performance results obtained by classifying the dataset, considering the F_1 measure.

Analysing the table we can observe that the use of the incremental approach outperforms the overall classification of the time-window model and the ensemble model, except in the *Sudden #1* drift. In this particular drift, the ensemble model outperform the incremental model with a F_1 score of 58.42% against the F_1 score of 54.85%. Nevertheless, this is the only drift in which this occurs, which seems to be explained by being a fast occurring sudden drift, as it appears and disappears rapidly, differently from the *Sudden #2* that occurs during a longer period. The incremental model, by having a much broad view of the whole time collection, fails to identify smaller and faster occurring drifts. In this particular case, a model with less memory can be appropriate as there is no gain in retaining the information about this drift.

When considering the average of the F_1 score, the time-window model scores 51.53%, it is outperformed by the ensemble model with 60.31% and finally by the incremental model with a F_1 score of 76.39%. The obtained results were expected as the incremental model decides with the knowledge of the whole collection previously seen, differently from the time-window model and even the ensemble model. One can argue that the ensemble model, by using the time-window models created in each time-window, could still have the information of the previously seen examples, however, as the decision is combined, errors can also arise as all the models previously created contribute to the decision and can induce errors in the final decision.

It is also important to note that the ensemble model performs better than the time-window in the majority of drifts, nevertheless, in the drift *Gradual #1* and in the drift *Incremental #1*, the ensemble scores 40.45% against 49.88% and 30.69% against 41.41%, respectively, which are significant results. These drifts have the particularity of being the only ones that increase their frequency over time, which seem to denote that there is a relation between their nature and the performance obtained by the ensemble model. The explanation for this phenomenon is that in the first occurring time-windows, considering the time line, the time-window models that are created to compose the ensemble tend to fail, as they have not seen enough positive examples. In the last time-windows they contribute equally to the output of the ensemble and influence in a negative way the classification provided by the ensemble. This does not occur with a decreasing frequency drifts because when the models that have seen less positive examples start to participate in the ensemble decision, the examples they have to identify are less (as the frequency is decreasing) and thus the ensemble fails in a smaller proportion. This also seems to denote that the ensemble model tends to take more time to adapt to a changing environment.

Besides the mentioned drifts, in *Regular #1* the ensemble model is also outperformed by the time-window model, but in this case with less significant results, 55.53% against 55.78%. We believe that this is related to the tie mechanism, as the examples miss classified are just a few (when comparing with the time-window model) and are those in which there was a tie and the last model, that is called to untie, fails the decision. Finally it seemed strange in a first glance that *Regular #3* had such a bad performance, specially when compared with a pronounced drift. The results might be explained by the *hashtag* we choose to represent it, *#nowplaying*. This *hashtag* is commonly used to refer the songs that users are playing in their computers or mobile devices, usually just posting the song name and the corresponding artist. Considering the spectrum of musics and artists we suspect that the diversity of those *tweets*, along with its short encoding, compromises the performance of the classifier. One can also notice that *Regular #1* with the *hashtag #jobs* could suffer from the same characteristics, still, in this particular case, *tweets* have linking words like hiring, recruiting or opportunity that might be informative of the tweet content.

VI. CONCLUSIONS

In this paper we have presented a threefold approach to learn in the presence of concept drift in Twitter streams. Three different models were proposed: a time-window model, an ensemble based model and an incremental model. We have also simulated different types of drift by artificially timestamping real *tweets* to evaluate and validate our strategy.

The results obtained revealed the usefulness is using different strategies in the awareness of different kinds of drift. More precisely, we have identified that the same learning model performs equally with drifts of the same nature, and that for instance in the case of a decreasing frequency drift, which means a concept that tends to disappear, it is better to use a time-window model instead of an ensemble model.

Furthermore, the results have also shown that memory, or the ability to keep the information already gathered, is important in the adaptability to drift in the learning process, as the incremental model tends to perform better than the other two models. Nevertheless, as storing can be a constraint in the Twitter stream data, it is important in future approaches to identify what is an outdated example, and a more profound study is imperative to detail for how long it is useful to store examples. This can be done by analyzing different time-window sizes, so we can reach an equilibrium between the computational burden of storing and processing huge amounts of data and the usefulness of storing those examples.

Another strategy is to use selective pruning, which means to understand which examples are considered most significant, the ones with interest to keep, and those which might not be relevant to identify future drifts of the same nature. Those could be discarded to reduce the computational effort.

Future work will include not only the identification of examples that can be discarded, as outdated or not significant, but also further study on the characteristics of each drift. Further steps also include the study of different kinds of drift, where the same concept is used with different meanings in two different moments.

A more intensive study on the drift patterns is also foreseen; especially by identifying those that occur in the Twitter scenario. It is also relevant to extend the learning models to include different weighting mechanisms in the ensemble model, as the models that compose the ensemble may contribute differently to the final decision when different drift patterns are present.

ACKNOWLEDGMENT

We gratefully acknowledge iCIS project (CENTRO-07-ST24-FEDER-002003).

REFERENCES

[1] S. Johnson, "How twitter will change the way we live," *Time Magazine*, vol. 173, pp. 23–32, 2009.

[2] O. Tsur and A. Rappoport, "What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities," in *Proc. 5th ACM International Conference on Web Search and Data Mining*, 2012, pp. 643–652.

[3] L. Yang, T. Sun, M. Zhang, and Q. Mei, "We know what @you #tag: does the dual role affect hashtag adoption?" in *Proc. 21st International Conference on World Wide Web*, 2012, pp. 261–270.

[4] H.-C. Chang, "A new perspective on twitter hashtag use: diffusion of innovation theory," in *Proc. 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem*, 2010, pp. 85:1–85:4.

[5] M. Efron, "Hashtag retrieval in a microblogging environment," in *Proc. 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '10, 2010, pp. 787–788.

[6] F. Abel, Q. Gao, G.-J. Houben, and K. Tao, "Semantic enrichment of twitter posts for user profile construction on the social web," in *Proc. 8th Extended Semantic Web Conference on The Semanic Web: Research and Applications - Volume Part II*, ser. ESWC'11, 2011, pp. 375–389.

[7] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proc. 19th International Conference on World Wide Web*, 2010, pp. 851–860.

[8] A.-M. Popescu and M. Pennacchiotti, "Detecting controversial events from twitter," in *Proc. 19th ACM international conference on Information and knowledge management*, ser. CIKM '10, 2010, pp. 1873–1876.

[9] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on twitter," in *Proc. Fifth International Conference on Weblogs and Social Media*, 2011.

[10] S. Ovadia, "Exploring the Potential of Twitter as a Research Tool," *Behavioral & Social Sciences Librarian*, vol. 28, pp. 202–205, 2009.

[11] M. Rowe and M. Stankovic, "Mapping tweets to conference talks: A goldmine for semantics," *Workshop on Social Data on the Web*, 2010.

[12] K. Weller and C. Puschmann, *Twitter for Scientific Communication: How Can Citations/References be Identified and Measured?*, 2011, pp. 1–4.

[13] F. Abel, Q. Gao, G.-J. Houben, and K. Tao, "Analyzing user modeling on twitter for personalized news recommendations," in *Proc. 19th international conference on User modeling, adaption, and personalization*, 2011, pp. 1–12.

[14] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Proc. Fourth International AAAI Conference on Weblogs and Social Media*, 2010, pp. 178–185.

[15] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series," in *Proc. International AAAI Conference on Weblogs and Social Media*, 2010.

[16] B. Doerr, M. Fouz, and T. Friedrich, "Why rumors spread so quickly in social networks," *Commun. ACM*, vol. 55, no. 6, pp. 70–75, 2012.

[17] C. Tantipathananandh and T. Y. Berger-Wolf, "Finding Communities in Dynamic Social Networks," in *2011 IEEE 11th International Conference on Data Mining*, 2011, pp. 1236–1241.

[18] M. Treiber, D. Schall, S. Dustdar, and C. Scherling, "Tweetflows: flexible workflows with twitter," in *Proc. 3rd International Workshop on Principles of Engineering Service-Oriented Systems*, 2011, pp. 1–7.

[19] J. Costa, C. Silva, M. Antunes, and B. Ribeiro, "Defining semantic meta-hashtags for twitter classification," in *Adaptive and Natural Computing Algorithms*, ser. Lecture Notes in Computer Science, vol. 7824. Springer Berlin Heidelberg, 2013, pp. 226–235.

[20] E. Zangerle, W. Gassler, and G. Specht, "Recommending #-tags in twitter," in *Proc. Workshop on Semantic Adaptive Social Web 2011 in connection with the 19th International Conference on User Modeling, Adaptation and Personalization, UMAP 2011*, 2011, pp. 67–78.

[21] Y. Duan, F. Wei, M. Zhou, and H.-Y. Shum, "Graph-based collective classification for tweets," in *Proc. 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 2323–2326.

[22] H. O. O Ozdikis, P Senkul, "Semantic Expansion of Hashtags for Enhanced Event Detection in Twitter," in *The First International Workshop on Online Social Systems (WOSS)*, 2012.

[23] A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen, "Dynamic integration of classifiers for handling concept drift," *Information Fusion*, vol. 9, no. 1, pp. 56–68, 2008.

[24] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Mach. Learn.*, vol. 23, no. 1, pp. 69–101, 1996.

[25] A. Tsymbal, "The Problem of Concept Drift: Definitions and Related Work," Department of Computer Science, Trinity College Dublin, Tech. Rep., 2004.

[26] J. Kim, P. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, and J. Twycross, "Immune system approaches to intrusion detection - a review," *Natural Computing*, vol. 6, no. 4, pp. 413–466, 2007.

[27] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *Trans. Neur. Netw.*, pp. 1517–1531, 2011.

[28] G. Ditzler and R. Polikar, "Incremental learning of concept drift from streaming imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2283–2301, 2013.

[29] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[30] D. Mejri, R. Khanchel, and M. Limam, "An ensemble method for concept drift in nonstationary environment," *Journal of Statistical Computation and Simulation*, vol. 83, no. 6, pp. 1115–1128, 2013.

[31] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: A new ensemble method for tracking concept drift," in *Proc. Third International IEEE Conference on Data Mining*, 2003, pp. 123–130.

[32] I. Zliobaite, "Learning under concept drift: an overview," Vilnius University, Faculty of Mathematics and Informatic, Tech. Rep., 2010.

[33] T. Joachims, *Learning Text Classifiers with Support Vector Machines*. Kluwer Academic Publishers, Dordrecht, NL, 2002.

[34] J. Costa, C. Silva, M. Antunes, and B. Ribeiro, "On using crowdsourcing and active learning to improve classification performance," in *Proc. 11th International Conference on Intelligent Systems Design and Applications (ISDA)*. IEEE, 2011, pp. 469–474.