Learning with Drift in Twitter

Joana Costa[12]
joana.costa@ipleria.pt,joanamc@dei.uc.pt
Catarina Silva[12]
catarina@ipleiria.pt,catarina@dei.uc.pt
Mário Antunes[13]
mario.antunes@ipleiria.pt,mantunes@dcc.fc.up.pt
Bernardete Ribeiro[2]
bribeiro@dei.uc.pt

[1] School of Technology and Management
Polytechnic Institute of Leiria, Portugal
[2] Center for Informatics and Systems
University of Coimbra, Portugal
[3] Center for Research in Advanced Computing Systems
INESC-TEC, University of Porto, Portugal

Abstract

Social networks have become part of our daily routine as Internet users. Reading news, looking for a service, asking for help, or simply sharing emotions or thoughts with family and friends turned social networks into a huge repository of information as users share daily valuable information. Learning in such a dynamic environment requires specific approaches, not only because of the diversity of data but because time plays an important role, drifting concepts over time. In this paper we propose a learning strategy to learn in the presence of concept drift in Twitter, one of the most well known social networks. Two learning models are proposed: a time-window model and an ensemble based model. We also present the QtSim framework, designed to simulate different types of drift by artificially timestamping real Twitter messages, that allows us 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.

1 Introduction

Over the last few years, with the burst of social networks, people became easily connected and can communicate, share and join together. This can obviously endorse noteworthy changes in information spreading, as information is being shared publicly among users. One of the most well-known social media platforms is Twitter, a microblogging service where users post text-based messages, *tweets*, of up to 140 characters. Another interesting characteristic of Twitter is the presence of *hashtags*, single words started with the symbol "#", used to classify each message content. Along with the deluge of data created, time is an important constraint, as the flow of information is continuous and changes over time: one might be refering to an important event that might be occurring today, and in a few days those *tweets* might have disappeared and new content arises. Learning in the presence of concept drift is not an easy task and requires a specific approach. The learning model must have not only the ability to continuously learn, but also the ability to change concepts already acquired. To deal with concept drift in the Twitter stream we propose a two-fold approach: a time-window model and an ensemble based model. We also propose a framework to simulate different types of drift by artificially timestamping real Twitter messages in a sequential way in order to 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.

2 Related Work

In [1] an approach for *hashtag* recommendation in Twitter is introduced. This approach computes a similarity measure between *tweets* and uses a ranking system to recommend *hashtags* to new *tweets*. In [2] 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 [3], 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.

3 Proposed Approach

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, occurring each instance at a time, not necessarily in equally spaced time intervals, and is characterized by a set of features, usually words. A labelled instance is represented as a pair between the feature vector of that instance along with the associated class label.

We have used a classification strategy previously introduced in [4], where the Twitter message *hashtag* is used to label the message content. 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, which means one classifier for each class.

For classifying time series like the Twitter stream we propose a two-fold approach: a time-window model and an ensemble model. The time-window model is a batch learning model unable to retain all the previously seen examples. Differently, 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. The main purpose is to design a memory mechanism that allows newly seen examples to be identified based on past experiences.

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

1 **for** *t=1,2, ... T* **do**
2 $\quad \mathcal{D}^t \leftarrow \mathcal{T}^t$
3 **end**
4 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: $\mathcal{X} \rightarrow \mathcal{Y}$

The ensemble model, presented in Algorithm 2, proposes to store all the information gathered with the previously seen examples. 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 is presented to the ensemble model, all the previously trained classifiers are loaded, and each one 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 propose a majority voting strategy where each classifier participates equally. When there is a tie, i.e. the votes account to zero, the classification of the most recent classifier is used to untie.

4 The QtSim Framework

In this work we have developed the QtSim framework that dynamically creates datasets by artificially timestamping real *tweets*. The major goal of this framework is to create labelled datasets that can be used to simulate different drift patterns that will evaluate and validate our previously introduced strategy. The framework receives a document set for each document class, typically *tweets* that contain the same *hashtag*, along with the frequency of that class during previously defined time-windows. The main idea is to use the frequency to reproduce artificial drifts. For instance, a sudden drift might be represented by *tweets* from a given *hashtag* that in a given temporal moment start to appear with a significant

Algorithm 2: 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:

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

frequency. Besides artificially timestamping real tweets, our framework represents each *tweet* as a vector space model, also known as *Bag of Words*. In this representation the collection of features is built as the dictionary of unique terms present in the documents collections and each tweet is indexed with the *bag* of the terms occurring in it. We have also integrated in our framework the INDRI API from the Lemur Project (`http://www.lemurproject.org/`) to add more features like indexing, parsing and querying. As our main intent is to create datasets for text classification approaches our framework can also apply pre-processing methods like *stopword removal* and *stemming*. The framework creates datasets in the ARFF format and in SVMLight format.

5 Dataset

We have created a dataset using our QtSim framework in order to evaluate and validade our strategy. As previously stated, we used a classification strategy introduced in [4], where the Twitter message *hashtag* is used to label the message content. We have simulated 10 different drift patterns and are based on those proposed in [5] , namely (i) *sudden*, (ii) *gradual*, (iii) *incremental*, and (iv) *reoccurring*. We have represented 2 instances of sudden, gradual and incremental drifts, to represent both increasing (refered as #1) or decreasing (refered as #2) frequencies. Regularity is represented here to show *tweets* that occur in a continuous frequency, i.e. without drift. We chosen 10 different *hashtags*, one for each defined drift, representing mutually exclusive concepts and hence different classes, such as *realmadrid* and *literature*. Table 1 shows the chosen *hashtags* and the corresponding drift.

The Twitter API (`https://dev.Twitter.com`) was then used in October 2013 to request public *tweets* that contain the defined *hashtags*. Besides having requested more than 10.000 *tweets*, those containing no message content besides the *hashtag*, along with all in non-English languages were discarded. Finally, we used 5700 *tweets* that were split in 24 timewindows according to the drift patterns previouly defined. In each timewindow the number of *tweets* is variable, as for simulating the drift patterns each class frequency varies along with time it is not compensated by any other.

6 Results and analysis

Table 1 summarizes the results obtained considering the F_1 measure.

Analysing the table we can observe the time-window model scores 51.53% of F_1 and it is outperformed by the ensemble model with 60.31%. Besides performing 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 their nature and the performance obtained are related. The explanation is that in the first occuring time-windows, the time-window models used in 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 in the drifts with a decreasing frequency because, as the frequency is decreasing, the newly created models have seen less positive examples, but when they start to influence the ensemble decision, that in the beginning is mainly composed by models that have seen much positive examples, the examples they have to identify are less (as the frequency is decreasing) and thus the ensemble fails in a smaller proportion.

	Hashtag	Time-window	Ensemble
Sudden #1	#bradpitt	55.93%	58.42%
Sudden #2	#realmadrid	60.22%	80.12%
Gradual #1	#ryanair	49.88%	40.45%
Gradual #2	#literature	45.08%	74.53%
Incremental #1	#twitter	41.41%	30.69%
Incremental #2	#ferrari	52.01%	61.72%
Reoccurring	#syria	73.59%	82.92%
Regular #1	#jobs	55.78%	55.53%
Regular #2	#sex	57.69%	88.05%
Regular #3	#nowplaying	23.71%	30.65%
	Average:	**51.53%**	**60.31%**

Table 1: Comparative results: F_1 measure

Moreover, in *Regular #1* the ensemble model is also outperformed, 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 misclassified are just a few 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 pronunciated drift. The results might be explained by the *hashtag* we choose to represent it, *#nowplaying*. This *hashtag* is commonly used to refer songs that users are playing in their devices, and considering the spectrum of musics and artists we suspect that the diversity of those *tweets* compromises the performance of the classifier.

7 Conclusions

We have presented two models to learn in the presence of concept drift in Twitter streams: a time-window model and an ensemble based model. We have also presented the QtSim frameworks, used to simulate different types of drift by artificially timestamping real *tweets* to evaluate and validate our strategy.

The results obtained revealed the usefulness of keeping information already gathered and 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 in the case of a decreasing frequency drift it is better to use a time-window model instead of an ensemble model. Another solution is to combine the ensemble so that models with less positive examples participate with less score than those better suited to identify positive examples. Though, as storing can be a constraint in the Twitter stream data, it is important in future approaches to identify an outdated example, and for how long it is useful to store examples. This can be done by analyzing different time-window sizes, so we can reach a balance between the computational burden of storing and processing and the usefulness of storing.

Future work will include a more intensive study of the drift patterns in Twitter in order 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 in the presence of different drift patterns. Furthermore, another study is to identify if there are *tweets* more informative than others, so pruning strategies can be used to relief the computational burden.

Acknowledgment

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

References

[1] E. Zangerle, W. Gassler, and G. Specht. Recommending #-tags in twitter. In *Proceedings of the Workshop on Semantic Adaptive Social Web 2011 in connection with the 19th International Conference on User Modeling, Adaptation and Personalization, UMAP 2011*, pages 67-78, 2011.

[2] Y. Duan, F. Wei, M. Zhou, and H.-Y. Shum. Graph-based collective classification for tweets. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2323-2326, 2012.

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

[4] Joana Costa, Catarina Silva, Mário Antunes, and Bernardete Ribeiro. Defining Semantic Meta-hashtags for Twitter Classification. In *Adaptive and Natural Computing Algorithms, Lecture Notes in Computer Science, vol. 7824. Springer Berlin Heidelberg*, pages 226-235, 2013.

[5] Indre Zliobaite. Learning under Concept Drift: an Overview. Tech. Report, Vilnius University, Faculty of Mathematics and Informatic, 2010.