Detecting Sentiment Change in Twitter Streaming Data

Albert Bifet ABIFET@CS.WAIKATO.AC.NZ
Geoff Holmes GEOFF@CS.WAIKATO.AC.NZ
Bernhard Pfahringer BERNHARD@CS.WAIKATO.AC.NZ
Department of Computer Science
University of Waikato, Hamilton, New Zealand

Ricard Gavaldà GAVALDA@LSI.UPC.EDU
LARCA Research Group
UPC-Barcelona Tech, Catalonia

Editor: Tom Diethe, José L. Balcázar, John Shawe-Taylor, and Cristina Tîrnăucă

Abstract

MOA-TweetReader is a real-time system to read tweets in real time, to detect changes, and to find the terms whose frequency changed. Twitter is a micro-blogging service built to discover what is happening at any moment in time, anywhere in the world. Twitter messages are short, and generated constantly, and well suited for knowledge discovery using data stream mining. MOA-TweetReader is a software extension to the MOA framework. **M**assive **O**nline **A**nalysis (MOA) is a software environment for implementing algorithms and running experiments for online learning from evolving data streams.

1. Introduction

Traditional web search engines are useful because they capture people's intent, what they are looking for, what they desire, and what they want to learn about. Instead, Twitter data streams help to capture what people are doing and what they are thinking about. Twitter popularity is growing, and the most interesting aspect from the data analysis point of view, is that a large quantity of data is publically available, as most of the people prefer to publish their posts openly, in contrast to other social networks like Facebook or LinkedIn, where the information is only accesible to people that are friends or connections.

Twitter has its own conventions that renders it distinct from other textual data. Consider the following Twitter example message or Tweet: `RT @toni has a cool #job`. It shows that users may reply to other users by indicating user names using the character @, as in, for example, `@toni`. Hashtags (#) are used to denote subjects or categories, as in, for example `#job`. `RT` is used at the beginning of the tweet to indicate that the message is a so-called "retweet", a repetition or reposting of a previous tweet.

Twitter is still growing. On the Twitter Blog in March 2011, the company presented some statistics about its site, in a blog post titled "#numbers." (Penner, 2011). In 2011 users send a billion tweets each week. The average number of tweets people sent per day in 2010 was 50 million. One year later, this number grew to 140 million. For example, the number of tweets sent on March 11, 2011 was 177 million. The number of new accounts created on March 12, 2011 was 572,000, and the average number of new accounts per day over February 2011 was 460,000. From 2010 to 2011 the number of mobile users grew 182%.

One of the main characteristics of Twitter is that tweets are arriving in real time following the data stream model. In this model, data arrive at high speed, and algorithms that process them must do so under very strict constraints of space and time. Data streams pose several challenges for data mining algorithm design. First, they must make use of limited resources (time and memory). Second, they must deal with data whose nature or distribution changes over time.

The main Twitter data stream that provides all messages from every user in real-time is called Firehose and was made available to developers in 2010. This streaming data opens new challenging knowledge discovery issues. To deal with this large amount of data, streaming techniques are needed (Bifet and Frank, 2010).

In this paper we present a methodology to analyse and mine the Twitter data in real time using data stream mining methods.

2. Real-Time Twitter Analysis Framework

We design a new general framework to mine tweets in real time adapting to changes in the stream. We are interested in

- classifying tweets in real time
- detecting changes
- showing what are the changes in the most used terms

Our main goal is to build a system able to train and test from the Twitter streaming API continuously. The input items are the tweets obtained from the Twitter stream. These tweets are preprocessed and converted by `MOA-TweetReader` to vectors of attributes or machine learning instances. The second component of the system is a learner trained with several instances, and that is able to predict the class label of incoming unlabeled instances. Finally, a change detector monitors the predictions, and outputs an alarm signal when change is detected.

2.1. The Twitter Streaming API

The Twitter Application Programming Interface (API) currently provides a Streaming API and two discrete REST APIs. The Streaming API (Kalucki, 2010) provides real-time access to Tweets in sampled and filtered form. The API is HTTP based, and GET, POST, and DELETE requests can be used to access the data.

In Twitter terminology, individual messages describe the "status" of a user. The streaming API allows near real-time access to subsets of public status descriptions, including replies and mentions created by public accounts. Status descriptions created by protected accounts and all direct messages are not available. An interesting property of the streaming API is that it can filter status descriptions using quality metrics, which are influenced by frequent and repetitious status updates, etc.

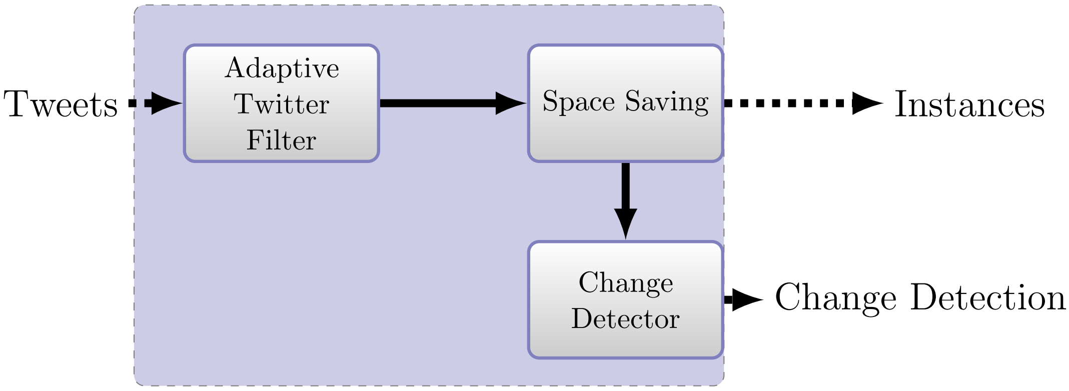

Figure 1: The `MOA-TweetReader`

The API uses basic HTTP authentication and requires a valid Twitter account. Data can be retrieved as XML and the more succinct JSON format. Parsing JSON data from the streaming API is simple: every object is returned on its own line, and ends with a carriage return.

The main Twitter stream, which provides all status updates from everyone in real-time, is called Firehose. Two subsamples of this stream are defined as the so-called "Spritzer" role and "Gardenhose" role respectively. The sampling rate is 5% for the Spritzer role and 15% for Gardenhose.

3. `MOA-TweetReader`

We present `MOA-TweetReader`, a new method to read tweets in real time adaptively using the Twitter streaming API.

Figure 1 shows the architecture of `MOA-TweetReader`. The input items are the tweets obtained from the Twitter stream. These tweets are preprocessed and converted by a tf-idf filter to vectors of attributes or machine learning instances. The second component of the system is a frequent item miner that stores the frequency of the most frequent terms. Finally, a change detector monitors changes in the frequencies of the items.

3.1. `MOA-TweetReader` Feature Generation Filter

`MOA-TweetReader` is able to use standard streaming machine learning methods. Tweets are list of words, and the adaptive Twitter filter will transform them to vectors of features, obtaining the most relevant ones.

We use an incremental *tf-idf* weighting scheme similar to the one used by Salton (Salton and Buckley, 1988) :

$$f_{i,j} = \frac{\text{freq}_{i,j}}{\sum_\ell \text{freq}_{\ell,j}} \tag{1}$$

$$idf_i = \log \frac{N}{n_i} \tag{2}$$

where

- $f_{i,j}$ is the frequency of term i in document j, which is the number of occurences of term i in document j divided by the the sum of number of occurrences of all terms in document j, that is, the size of the document.
- idf_i is the inverse document frequency of term i.
- N is the number of documents
- n_i is the number of documents where the term i appears

The weight of each query term is given by:

$$w_{i,q} = f_{i,j} \cdot idf_i \tag{3}$$

3.2. Adaptive Frequent Item Miner for Data Streams

The most important part of this reader is the adaptive mechanism of feature generation. It is based on the Space Saving Algorithm.

We base `MOA-TweetReader` on Space Saving since it has the best performance results compared with other frequent miners as reported in (Liu et al., 2011; Cormode and Hadjieleftheriou, 2008). This method proposed by Metwally et al. (Metwally et al., 2005) is very simple and it has interesting and simple theoretical guarantees. The algorithm maintains in memory k pairs of (item,count) elements, initialised by the first k distinct elements and their counts. Every time a new item arrives, if it was monitored before, its count is incremented by one. If not, it replaces the item with the lowest count, and it increments its count by one. This is done using space $O(k)$, and the error of estimating item frequencies is at most n/k, where n is the number of elements in the stream.

3.3. Change Detection

We use `ADWIN` (Bifet and Gavaldà, 2007) as a change detector. `ADWIN` (`AD`aptive sliding `WIN`dow) also solves in a well-specified way the problem of tracking the average of real-valued numbers. `ADWIN` keeps a variable-length window of recently seen items, with the property that the window has the maximal length statistically consistent with the hypothesis “there has been no change in the average value inside the window”.

More precisely, an older fragment of the window is dropped if and only if there is enough evidence that its average value differs from that of the rest of the window. This has two consequences: one, that change is reliably detected whenever the window shrinks; and two, that at any time the average over the existing window can be reliably taken as an estimation of the current average in the stream (barring a very small or very recent change that is still not statistically visible).

4. Twitter Sentiment Analysis

Sentiment analysis can be cast as a classification problem where the task is to classify messages into two categories depending on whether they convey positive or negative feelings; see (Pang and Lee, 2008) for a survey of sentiment analysis, and (Liu, 2006) for opinion mining techniques.

To build classifiers for sentiment analysis, we need to collect training data so that we can apply appropriate learning algorithms. Labeling tweets manually as positive or negative is a laborious and expensive, if not impossible, task. However, a significant advantage of Twitter data is that many tweets have author-provided sentiment indicators: changing sentiment is implicit in the use of various types of emoticons. Hence we may use these to label our training data.

4.1. The 2010 Toyota Crisis

As an example of Twitter sentiment analysis, and the need to mine sentiments in real-time, we would like to show the case of the crisis of Toyota, the world's largest car manufacturer, during 2009 and 2010. During those days, it seems that Toyota had problems with accelerator pedals, and had to recall millions of cars to check that they were working properly.

In the book "Toyota under Fire" (Liker and Ogden, 2011), Akio Toyoda, president of Toyota, identifies the gap in understanding of local conditions and urgency between regions and headquarters as a major contributor to the evolution of the crisis:

> *There was a gap between the time that our U.S. colleagues realised that this was an urgent situation and the time that we realised here in Japan that there was an urgent situation going on in the U.S. It took* ***three months*** *for us to recognise that this had turned into a crisis. In Japan, unfortunately, until the middle of January we did not think that this was really a crisis.*

We think that looking at Twitter data in real time can help people to understand what is happening, what people are thinking about brands, organisations and products, and more importantly, how they feel about them.

Using the Edinburgh corpus (Petrovic et al., 2010) collected between November 11th 2009 and February 1st 2010, we apply our new methods to get some insights to the Toyota crisis. The number of tweets referring to Toyota is 4381. Applying our new `MOA-TweetReader` is possible to detect the following changes in terms of frequencies, see Table 1.

To see the evolution of positive and negative tweets, we train and test a Hoeffding tree learner (Figure 2). The most well-known tree decision tree learner for data streams is the *Hoeffding tree* algorithm. It employs a pre-pruning strategy based on the Hoeffding bound to incrementally grow a decision tree. A node is expanded by splitting as soon as there is sufficient statistical evidence, based on the data seen so far, to support the split and this decision is based on the distribution-independent Hoeffding bound.

What we see in Figure 2 is that there is a clear correlation between the sentiments in the tweets available in the Twitter streaming API, and the timeline of the crisis of Toyota (Liker and Ogden, 2011). It is very clear that around Christmas 2009 positive sentiment in tweets

Term	Before	After	Diff
gas	0.122	0.484	0.363
pedals	0.129	0.438	0.309
wonder	0.017	0.214	0.198
problem	0.163	0.357	0.194
good	0.016	0.205	0.190
recalling	0.012	0.106	0.095
gm	0.011	0.089	0.077
#heard_on_the_street	0.040	0.113	0.072
social	0.031	0.099	0.068
sticking	0.070	0.125	0.055
fix	0.026	0.076	0.050
popularity	0.016	0.037	0.021
love	0.017	0.024	0.008

Table 1: Frequency changes detected

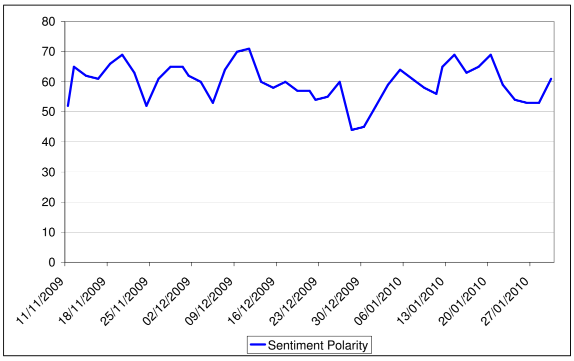

Figure 2: Positive sentiment detection on Toyota tweets retrieved from Twitter.

towards Toyota plunged below 50%. Looking at the changes in the frequencies of words, we can understand why these changes are happening. Almost every second tweet mentioning Toyota suddenly includes “gas” and “pedals”. A tool like `MOA-TweetReader` would have helped Toyota to understand the crisis sooner and to respond more appropriately.

5. Website, Tutorials, and Documentation

`MOA-TweetReader` is an extension of the MOA and it is going to be included in a next release of MOA. MOA is a classification and clustering system for massive data streams with the following characteristics:

- benchmark streaming data sets through stored, shared, and repeatable settings for the various data feeds and noise options, both synthetic and real
- set of implemented algorithms for comparison to approaches from the literature
- open source tool and framework for research and teaching similar to WEKA

MOA can be found at `http://moa.cs.waikato.ac.nz/`.

6. Conclusions

Twitter streaming data can potentially enable any user to discover what is happening in the world at any given moment in time. As the Twitter Streaming API delivers a large quantity of tweets in real time, we proposed `MOA-TweetReader`, a new system to perform twitter stream mining in real time using an adaptive frequent item miner for data streams.

References

Albert Bifet and Eibe Frank. Sentiment knowledge discovery in twitter streaming data. In *Discovery Science*, pages 1–15, 2010.

Albert Bifet and Ricard Gavaldà. Learning from time-changing data with adaptive windowing. In *SDM*, 2007.

Graham Cormode and Marios Hadjieleftheriou. Finding frequent items in data streams. *PVLDB*, 1(2):1530–1541, 2008.

John Kalucki. Twitter streaming API. `http://apiwiki.twitter.com/Streaming-API-Documentation`, 2010.

Jeffrey K. Liker and Timothy N. Ogden. *Toyota Under Fire: Lessons for Turning Crisis into Opportunity.* McGraw-Hill, 2011.

Bing Liu. *Web data mining; Exploring hyperlinks, contents, and usage data.* Springer, 2006.

Hongyan Liu, Yuan Lin, and Jiawei Han. Methods for mining frequent items in data streams: an overview. *Knowl. Inf. Syst.*, 26(1):1–30, 2011.

Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. Efficient computation of frequent and top-k elements in data streams. In *ICDT 2005*, pages 398–412, 2005.

Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

Carolyn Penner. #numbers. Twitter Blog Article, `http://blog.twitter.com/2011/03/numbers.html`, 2011.

Sasa Petrovic, Miles Osborne, and Victor Lavrenko. The Edinburgh twitter corpus. In *#SocialMedia Workshop*, pages 25–26, 2010.

Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.