Real Time Sentiment Change Detection of Twitter Data Streams

Sotiris K. Tasoulis, Aristidis G. Vrahatis, Spiros V. Georgakopoulos,Vassilis P. Plagianakos
Department of Computer Science and Biomedical Informatics,
University of Thessally,
Lamia, Greece.

***Abstract*—In the past few years, there has been a huge growth in Twitter sentiment analysis having already provided a fair amount of research on sentiment detection of public opinion among Twitter users. Given the fact that Twitter messages are generated constantly with dizzying rates, a huge volume of streaming data is created, thus there is an imperative need for accurate methods for knowledge discovery and mining of this information. Although there exists a plethora of twitter sentiment analysis methods in the recent literature, the researchers have shifted to real-time sentiment identification on twitter streaming data, as expected. A major challenge is to deal with the Big Data challenges arising in Twitter streaming applications concerning both Volume and Velocity. Under this perspective, in this paper, a methodological approach based on open source tools is provided for real-time detection of changes in sentiment that is ultra efficient with respect to both memory consumption and computational cost. This is achieved by iteratively collecting tweets in real time and discarding them immediately after their process. For this purpose, we employ the Lexicon approach for sentiment characterizations, while change detection is achieved through appropriate control charts that do not require historical information. We believe that the proposed methodology provides the trigger for a potential large-scale monitoring of threads in an attempt to discover fake news spread or propaganda efforts in their early stages. Our experimental real-time analysis based on a recent hashtag provides evidence that the proposed approach can detect meaningful sentiment changes across a hashtags lifetime.**

***Index Terms*—Twitter, Change Detection, Data Stream Mining**

I. Introduction

In the last decade, there has been a huge growth in the use of microblogging platforms such as Twitter [1] which is overwhelmed by amazing statistics. People send more than 500 million tweets per day (last update: 1/24/2017), 300 million are the total number of monthly active Twitter users (last update: 1/1/2018) and 100 million are the number of Twitter daily active users (last update: 1/24/2017). This wealth of information has attracted the interest of the research community, focusing on day-to-day emotion analysis that can be proven to be of great value in analyzing opinions about events, products, persons or political stances. It is well documented that people can feel and express emotions through Computer-Mediated Communication (CMC) even if it is asynchronous and text-based [2]. Sentiment analysis is a growing area of Natural Language Processing with research ranging from document level classification [3] to learning the polarity of words and phrases [4]. The sudden spurt of Twitter has enabled the emotion identification of several people at the same time for a specific subject, hence a plethora of studies have focused on Twitter Sentiment Analysis (TSA) research field. Nowadays, companies, media organizations and politicians strategy are affected by their Twitter popularity, since they can hear the common opinion on a daily basis [1], [5]. Twitter data streams are generated continuously at each trice offering the opportunity for a more realistic society's reflection on various issues. First steps towards this direction were made by Kalucki [6] providing a publicly available twitter streaming Application Programming Interface (API). Thenceforth, the "Twitter Streaming API" has been created by Twitter, allowing anyone to retrieve at most one percent sample of all the data by providing some parameters [1]. Twitter data streams pose several challenges for the data mining field, such as managing the limited resources (time and memory) and dealing with data shifts across time [7].

A typical Sentiment analysis software takes as input text and uses an algorithm to produce an estimate of its sentiment content. This estimate can be in several different forms: binary - either positive/negative or objective/subjective; trinary - positive/neutral/negative; scale - e.g., -5 (strongly negative) to 5 (strongly positive); dual scale - e.g., 1 (no positivity) to 5 (strong positivity) and -1 (no negativity) to -5 (strong negativity); and multiple - e.g., happiness $(0-100)$, sadness $(0-100)$, fear $(0-100)$. Sentiment analysis algorithms tend to use either machine learning or lexical approaches [8].

In this work, we focus on the lexical approaches in an attempt to build a completely on-line unsupervised lightweight methodology to identify the users' sentiment changes across the time. This is achieved by employing control charts to detect changes in the constructed times series of sentiment scores. To the best of our knowledge, no such methodology have been proposed before as usually an off-line phase, human interaction or historical information are required.

II. Related Work

Recently, studies for Twitter sentiment analysis have focused on design and implementation of scalable systems.

978-1-5386-5150-6/18/$31.00

[1] https://dev.twitter.com/docs/streaming-apis

The systems can be categorized in two broad categories: (i) real-time systems, and (ii) systems for batch processing [9]. Towards the direction of real-time systems is the work of Wang et al. [10] where they proposed a system for real-time sentiment analysis on Twitter streaming data towards presidential candidates (US 2012). However, their system is based on a crowd-sourcing approach to do sentiment annotation. In [9] a real-time architecture for scalable twitter sentiment analysis is presented, dealing with the dynamic content using a feedback mechanism in the sentiment analysis process, also incorporating supervised learning methods for sentiment analysis and additionally an off-line phase for feature extraction. Similarly, in [11] the authors introduced a transfer learning approach to performing real-time sentiment analysis. It is considered as the first study which measures the bias of social media users towards a topic providing evidence that user bias tends to be more consistent over time although the possible changes in the dynamic context (newcomer terms or old terms meaning change).

A crucial step in sentiment analysis is the challenge to identify the users' sentiment changes across the time. MOA-TweetReader, performs a stream mining from Twitter stream tweets, highlighting the sentiment changes [12]. It utilizes a feature generation filter to vectors of attributes or machine learning instances based on an incremental term frequency-inverse document frequency (tf-idf) weighting scheme. The system also applies the SPACE SAVING Algorithm [13] for mining and storing the frequency of the most frequent terms. More specifically, the system initialization is done by the first *k* distinct elements and their counts as they are stored in memory per *k* pairs elements (item and count). Then, it follows the rule which checks every time if the upcoming data has already been monitored. If the answer is yes, its count is incremented by one, if the answer is no, it replaces the least in count item and it increments its count by one. Lastly, MOA-TweetReader uses ADWIN [14] as a change detector, an Adaptive sliding window algorithm, thus memory requirements may grow significantly depending on the window size. In addition, although the authors manage to collect author-provided sentiment indicators to build classifiers for sentiment analysis there is still an off-line training stage.

III. Methodology

In this work, in an attempt to provide a lightweight solution that requires no training (no off-line phase) we employ the lexicon approach to characterize tweets. Then, we use change detection algorithms to detect opinion changes in a time series of sentiment scores taking advantage of the fact that such time series is bounded by nature (there is a limit in the number of words appearing in a post) and thus specifying appropriate parameters for control charts is relatively straightforward. In brief, the proposed methodology is constituted by two core parts:

- Collecting and characterizing tweets according to their sentiment.
- Detect significant changes in the time series of sentiment scores.

A. Constructing the Sentiment Time Series

Tweets are collected by the Twitter stream API continuously constituting a data stream. The API only requires a valid Twitter account for authentication, while our analysis is based on the open source tools provided by the R-project [15]. Using the "rtweet" package [16] we connect to the API to stream tweets filtered by keywords, for example, Twitter hashtags. Data are retrieved in JSON format and are easily parsed using the "rtweet" package. Once we retrieve text for each tweet we need to "clean" it removing hashtags, spaces, numbers, punctuations, URLs etc, employing functions from the "stringr" and "glue" packages respectively [17], [18]. Next, we process the resulting text by extracting tokens (tokenization) and retrieving only sentiment words using the "tidyverse" package [19]. Tokenization is a process of creating a bag-of-words from the text where the incoming string gets broken into comprising words using white space in separating individual words. Usually, tokenization of social-media data is considerably difficult but the aforementioned cleaning process has proven to be very successful. Each word from the bag-of-words is compared against the lexicon and if the word is found, we update the sentiment score of the post accordingly. Examples of the existing lexicons include: Opinion Lexicon [20] which categorizes words in a binary fashion into positive and negative categories, AFINN Lexicon [21] which assigns words with a score that runs between -5 and 5, with negative scores indicating negative sentiment and positive scores indicating positive sentiment and SentiWordNet which assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity [4].

B. Change Detection using CUSUM

The cumulative sum (CUSUM) algorithm was first proposed by Page in [22] for on-line and off-line change detection and it has been shown to be more efficient than Shewhart charts in detecting small shifts in the mean of a process. The CUSUM control chart have received a great deal of attention in modern industries while being an active research topic with various recent applications [23]–[25] and proposed variations or extensions [26]–[28].

In this work, we will utilize the online version of the CUSUM algorithm focusing on a technique connected to a simple integration of signals with adaptive threshold [29]. To describe the change detection algorithm, we consider a sequence of independent random variables y_k, where y_k is a sensor signal at the current time instant k (discrete time), with a probability density $p_\theta(y)$ depending only upon one scalar parameter θ. Before the unknown change time t_0, the parameter θ is equal to θ_0, and after the change it is equal to $\theta_1 \neq \theta_0$. Then, the problem is to detect and estimate this parameter change. In this work, our goal is to detect the change assuming that the parameters θ_0 and θ_1 are known, which is a quite unrealistic assumption for practical applications. Usually

parameters θ_0 and θ_1 can be experimentally estimated using test data, which we also not consider as available for the application at hand. However, for strongly bounded problems parameter values can be assumed relatively easy. For example, we may consider θ_0 the state of a neutral conversation with sentiment scores gathered around 0 and $\theta_1 = 1$ when positive comments dominate the stream. As such we may use our prior knowledge about the signal to correctly set *the change magnitude*.

Next, we introduce the basic idea used in quality control. Samples are iteratively taken and at the end of each arrival a decision rule is computed to test the two following hypotheses concerning parameter θ:

$$\begin{aligned} H_0 &: \theta = \theta_0, \\ H_1 &: \theta = \theta_1. \end{aligned} \tag{1}$$

As long as the decision is in favor of H_0, the sampling and test continue. Sampling is stopped after the first sample of observations for which the decision is in favor of H_1. This sample also determines the stopping time.

In our case, the samples (tweets) are arriving at each time instant and the decision rule is computed. We will use the following notation. Let

$$S_k = \sum_{i=1}^{k} s_i, \quad \text{where} \quad s_i = \ln \frac{p_{\theta_1}(y_i)}{p_{\theta_0}(y_i)}, \tag{2}$$

is the log-likelihood ratio for the observations from y_i to y_k and k be the current time instant. We refer to s_i as sufficient statistic. Let us now consider the particular case where the distribution is Gaussian with mean value μ and constant variance σ. In this case, the changing parameter θ is μ. The probability density is

$$p_\theta(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\theta)^2}{2\sigma^2}}, \tag{3}$$

and the sufficient statistic s_i is

$$s_i = \frac{\theta_1 - \theta_0}{\sigma^2}\left(y_i - \frac{\theta_0 + \theta_1}{2}\right). \tag{4}$$

The corresponding decision rule is then, at each time instant, to compare this difference to a threshold as follows:

$$g_k = S_k - m_k \geqslant h, \quad \text{where} \quad \mathrm{m_k} = \min_{1 \geqslant \mathrm{j} \geqslant \mathrm{k}} \mathrm{S_j}. \tag{5}$$

The stopping time is

$$t_a = \min\{k : g_k \geqslant h\}, \tag{6}$$

which can be rewritten as

$$t_a = \min\{k : S_k \geqslant m_k + h\}. \tag{7}$$

This decision rule is a comparison between the cumulative sum S_k and an adaptive threshold $m_k + h$. Because of m_k, this threshold not only is modified on-line but also keeps the complete memory of the entire information contained in the past observations. Moreover, in the case of a change in the mean of a Gaussian sequence, S_k is a standard integration of the observations.

The detection threshold h is a user-defined tuning parameter in which the appropriate form for its determination is based on the average run length function, which is defined as the expected number of samples before an action is taken [22]. More precisely one has to set the mean time between false alarms ARL_0 and the mean detection delay ARL_1. These two specific values of the ARL function depend on the detection threshold h, and can thus be used to set the performance of the CUSUM algorithm to the desired level for a particular application [30].

1) Two-sided Algorithm and Resets: The described algorithm, which is called one-sided CUSUM, focuses on change detections in one direction only. However, in our application, it is necessary to detect changes in each direction discovering both positive and negative sentiment change of twitter posts. For this purpose, two one-sided algorithms were used, one to detect an increase and the other to detect a decrease in the parameter θ. This leads to two different instantaneous log-likelihood ratios. As such, two cumulative sums and two decision functions are computed, while a change is detected by testing both decision functions simultaneously.

When a change is triggered for any of the two-sided functions, the CUSUM algorithm reset to zero and a re-initialization takes place. The algorithm restarts with a new value for θ_0 equal to the average of the last observations. As such we define a new control state representing the current sentiment at time t, while θ_1 is recalculated based on a fixed *change magnitude*. The rest of the parameters remain the same. Under this perspective, we deal with the major challenge in twitter streaming data, which is the sentiment trend detection under a dynamic estimation. This is crucial since a hashtag trend could change several times across its lifetime and thus changes should be estimated based on its current sentiment state rather than its initial state. Towards this direction, our methodological framework can imprint on-line, the change detections of a hashtag by updating its initial state. A flowchart diagram of the methodology is presented in Figure 1.

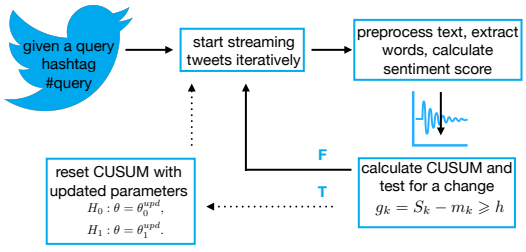

Fig. 1. Flowchart of the proposed methodology illustrating the core steps.

IV. Experimental Results

The described methodological framework was applied on a Twitter dataset streamed from 15-03-2018 to 24-03-2018 using the hashtag "theresamay". This term refers to Theresa Mary

May, Prime Minister of the United Kingdom and Leader of the Conservative Party since 2016. Our choice lies in the fact that politic-related Twitter hashtags offer a satisfying opportunity for testing sentiment changes since these are not one-sided (supporters are on both sides while critics are involved as well). In addition, this hashtag was selected considering that during this period the "Brexit" news topic attracted attention due to further discussions amongst high-level politicians regarding the relationships between the United Kingdom and European Union. 15491 posts are included in the streamed dataset after discarding non-English language posts. After sequentially applying the procedure described in Section III-A we retrieved the time series of sentiment score presented in Figure 2. To calculate the sentiment score for each tweet we used the "bing" lexicon [20] applied to the extracted words.

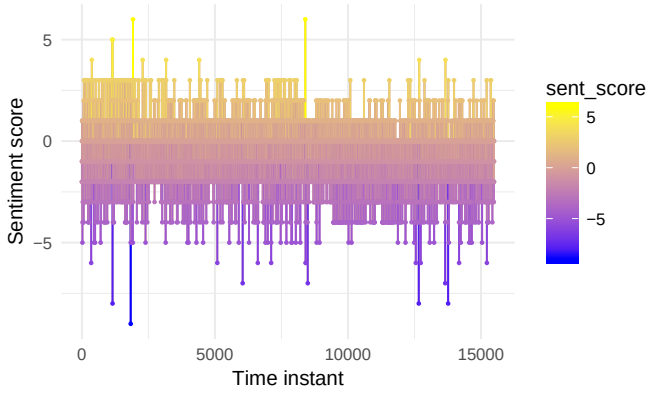

Fig. 2. The time series of sentiments score for the whole dataset across the specified period of time.

In an attempt to improve a visual inspection of the possible changes in sentiment, the corresponding moving average of sentiment scores was employed. Figure 3 illustrates the calculated moving average for a window of total size 200. As shown, we may visually discriminate the areas that could be characterized by changing distributions. We observe that the

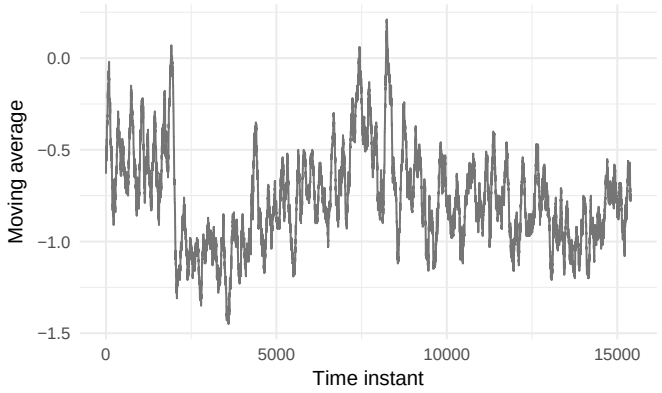

Fig. 3. The calculated moving average of window size 200.

whole thread is characterized by a slightly negative sentiment, which can be confirmed by the histogram of sentiments (see Figure 4). In what follows, the results of the change detection

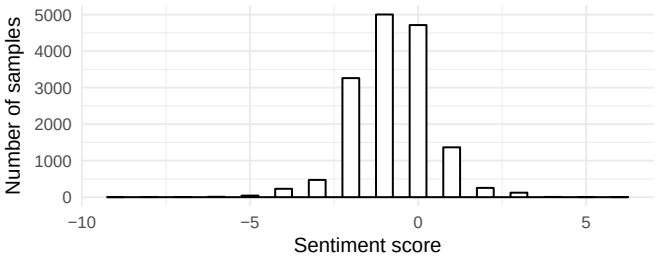

Fig. 4. Histogram of the sentiments score for the whole dataset.

algorithm are reported. For the initialization of parameters values we set $\theta_0 = -0.5$, while the *change magnitude* is set to 0.5; thus, we consider $\theta_1^{pos} = 0$ and $\theta_1^{neg} = -1$ for the two-sided CUSUM, respectively. In Figure 5 we show the CUSUM functions and the corresponding changes retrieved by the algorithm with vertical lines. We may recall that each time a change is detected the algorithm restarts with an updated initialization. Selecting the h parameter value is usually subject to question and depends on the user requirements. For this particular topic of study in our analysis we assume that a small number of reported changes per day are sufficient and thus we define $h = 20$. To visually investigate the reported

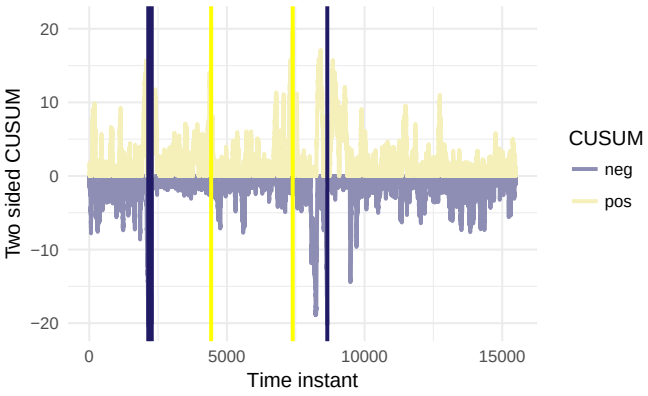

Fig. 5. The two sided CUSUM function along with the reported change points annotated by vertical lines. Yellow lines correspond to positive changes while blue lines correspond to negative changes.

change points we employed the calculated moving average presented in Figure 3. Vertical lines are depicted in the plot (see Figure 6) with the corresponding colors from Figure 5, where the blue lines correspond to negative changes, while the yellow lines correspond to positive changes. At this point, we can conclude that reported change points agree with a simple visual investigation of possible changes. In order to further verify this outcome, a very successful off-line methodology for change detection was employed. It is capable to discover multiple change points [31], [32] and estimate their number in the time series automatically if required. The penalty parameter of this algorithm used for this test was the default value $2 * \log(n)$, where n is the total number of samples. The

Q parameter specifying the maximum number of estimated change points is set equal to the total number of change points retrieved by the two sided CUSUM algorithm. The retrieved change points are also reported in Figure 6 with red vertical lines. We observe that this result highly agrees with the change points retrieved by CUSUM with only minor report of delays between the estimations.

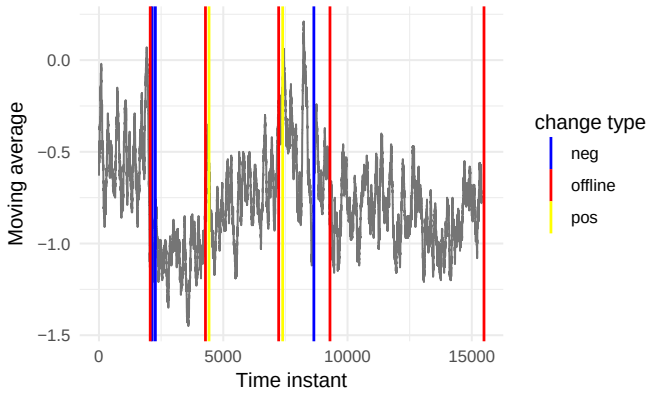

Fig. 6. The reported change points with respect to the calculated moving average. Yellow and blue lines correspond to positive and negative changes respectively while red lines correspond to the change points reported by the off-line algorithm.

The last step of the proposed methodology contains the content of two consecutive separate parts of the time series split by change points declaring the transition to a more negative sentiment and subsequently to a more positive sentiment. The first area ranges from the beginning of the time series until the first reported negative change point (sample count 2142 and corresponding date "2018-03-15 23:59:29 UTC") while the second part covers the area between the aforementioned point and the first positive change point (sample count 4427 and corresponding date "2018-03-16 17:42:08 UTC"). For each part, we report word-clouds plotting the frequency of words appearing in the corresponding collection of Twitter posts. The text is being preprocessed using the Term Frequency-Inverse Document Frequency (TF-IDF) methodology [33], where the frequency of words is rescaled by how often they appear in all tweets of the category, penalizing most frequent words (see Figure 7). It appears that there is discrimination between the word-clouds, while, we interestingly observe that in a hashtag about Teresa May the word-cloud that corresponds to the more positive part (before the negative change) is characterized by the appearance of words like "corbyn" and "labour".

V. Concluding Remarks

Twitter keeps attracting the interest of research community as in contrast to other social networks like Facebook or LinkedIn, its extensive data availability provides great potential for Machine Learning research. Sentiment analysis is a growing area of Natural Language Processing, which taking advantage of Twitter user activity has enabled the real-time emotion identification of large groups of people

Fig. 7. Word-clouds of two consecutive parts of the time series for which sentiment changes negatively from one to other. First part corresponds to the word-cloud at the left part of the Figure and the second part after the change corresponds to the word-cloud at the right part of the Figure.

for a specific subject. In this work, based on open source tools, we propose a complete methodology for continues real-time detection of sentiment changes in Twitter conversations. Focusing on simplicity, our methodology does not require training or an off-line phase, while memory and computational requirements are minimal. The experimental analysis provides enough justification of the usefulness of the methodology and exposes a great potential for further use. In our future research, we indent to examine approaches that enhance the robustness of the change detection algorithm, while further testing methodologies for sentiment characterization.

Acknowledgment

This research has been financially supported by the National Strategic Reference Framework (NSRF) Program with title: "Researcher Support with Emphasis on New Researches", co-financed by the European Union (European Social Fund ESF) and Greek national funds.

References

[1] E. Kouloumpis, T. Wilson, and J. Moore, *Twitter Sentiment Analysis: The Good the Bad and the OMG!* AAAI Press, 2011, pp. 538–541.

[2] J. Walther and M. R. Parks, "Cues filtered out, cues filtered in: Computer-mediated communication and relationships," 01 2002.

[3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, no. 1-2, pp. 1–135, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1561/1500000011

[4] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*, 2006, pp. 417–422.

[5] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle," in *Proceedings of the ACL 2012 System Demonstrations*, ser. ACL '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 115–120. [Online]. Available: http://dl.acm.org/citation.cfm?id=2390470.2390490

[6] J. Kalucki, "Twitter streaming api," 2010. [Online]. Available: http://apiwiki.twitter.com/Streaming-API-Documentation

[7] A. Bifet, G. Holmes, B. Pfahringer, and R. Gavalda, "Detecting sentiment change in twitter streaming data," in *Proceedings of the Second Workshop on Applications of Pattern Analysis*, ser. Proceedings of Machine Learning Research, T. Diethe, J. Balcazar, J. Shawe-Taylor, and C. Tirnauca, Eds., vol. 17. CIEM, Castro Urdiales, Spain: PMLR, 19–21 Oct 2011, pp. 5–11. [Online]. Available: http://proceedings.mlr.press/v17/bifet11a.html

[8] M. Thelwall, *The Heart and Soul of the Web? Sentiment Strength Detection in the Social Web with SentiStrength*. Cham: Springer International Publishing, 2017, pp. 119–134.

[9] M. Karanasou, A. Ampla, C. Doulkeridis, and M. Halkidi, "Scalable and real-time sentiment analysis of twitter data," in *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 944–951.

[10] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle," in *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, 2012, pp. 115–120.

[11] 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*. ACM, 2011, pp. 150–158.

[12] A. Bifet, G. Holmes, B. Pfahringer, and R. Gavalda, "Detecting sentiment change in twitter streaming data," 2011.

[13] A. Metwally, D. Agrawal, and A. El Abbadi, "Efficient computation of frequent and top-k elements in data streams," in *International Conference on Database Theory*. Springer, 2005, pp. 398–412.

[14] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 2007, pp. 443–448.

[15] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available: https://www.R-project.org/

[16] M. W. Kearney, *rtweet: Collecting Twitter Data*, 2017, r package version 0.6.0. [Online]. Available: https://cran.r-project.org/package=rtweet

[17] H. Wickham, *stringr: Simple, Consistent Wrappers for Common String Operations*, 2017, r package version 1.2.0. [Online]. Available: https://CRAN.R-project.org/package=stringr

[18] J. Hester, *glue: Interpreted String Literals*, 2017, r package version 1.2.0. [Online]. Available: https://CRAN.R-project.org/package=glue

[19] H. Wickham, *tidyverse: Easily Install and Load the 'Tidyverse'*, 2017, r package version 1.2.1. [Online]. Available: https://CRAN.R-project.org/package=tidyverse

[32] R. Killick, K. Haynes, and I. A. Eckley, *changepoint: An R package for changepoint analysis*, 2016, r package version 2.2.2. [Online]. Available: https://CRAN.R-project.org/package=changepoint

[20] M. Hu and B. Liu, *Opinion lexicon*, 2004, r package version 1.2.1. [Online]. Available: http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html

[21] F. Nielsen, "A new anew: Evaluation of a word list for sentiment analysis in microblogs," 03 2011.

[22] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[23] P. H. Tran and K. P. Tran, "The efficiency of cusum schemes for monitoring the coefficient of variation," *Applied Stochastic Models in Business and Industry*, vol. 32, no. 6, pp. 870–881. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/asmb.2213

[24] S. Tasoulis, C. Doukas, V. Plagianakos, and I. Maglogiannis, "Statistical data mining of streaming motion data for activity and fall recognition in assistive environments," *Neurocomputing*, vol. 107, pp. 87 – 96, 2013, timely Neural Networks Applications in Engineering. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231212007606

[25] S. V. Georgakopoulos, S. K. Tasoulis, and V. P. Plagianakos, "Efficient change detection for high dimensional data streams," in *2015 IEEE International Conference on Big Data (Big Data)*, Oct 2015, pp. 2219–2222.

[26] M. Abujiya, M. Riaz, and M. H. Lee, "Enhanced cumulative sum charts for monitoring process dispersion," vol. 10, p. e0124520, 04 2015.

[27] M. Perry and J. J. Pignatiello, "Estimating the time of step change with poisson cusum and ewma control charts," vol. 49, pp. 2857–2871, 05 2011.

[28] D. Wang, L. Zhang, and Q. Xiong, "A nonparametric cusum control chart based on the mann-whitney statistic," vol. 46, p. 2017, 02 2017.

[29] M. Basseville and I. V. Nikiforov, "Detection of abrupt changes: Theory and application," 1993.

[30] P. Granjon, "The cusum algorithm a small review," 2014.

[31] R. Killick, P. Fearnhead, and I. Eckley, "Optimal detection of changepoints with a linear computational cost," vol. 107, pp. 1590–1598, 12 2012.

[33] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*. Cambridge University Press, 2011.