Real-time Sentiment-Based Anomaly Detection in Twitter Data Streams

Khantil Patel, Orland Hoeber, and Howard J. Hamilton

Department of Computer Science
University of Regina, Canada
patel26k@uregina.ca, orland.hoeber@uregina.ca,
howard.hamilton@uregina.ca

Abstract. We propose an approach for real-time sentiment-based anomaly detection (RSAD) in Twitter data streams. Sentiment classification is used to split the data into independent streams (positive, neutral, and negative), which are then analyzed for anomalous spikes in the number of tweets. Four approaches for evaluating the data streams are studied, along with the parameters that adjust their sensitivity. Results from an evaluation show the effectiveness of a probabilistic exponentially weighted moving average (PEWMA) coupled with a sliding window that uses median absolute deviation (MAD).

1 Introduction

Time-series data streams have become a popular way of characterizing the data generated by real-time applications with a temporal attribute. Since such data can introduce new patterns very quickly, *data stream mining* has drawn interest from many researchers, with a focus on developing anomaly detection techniques that are both computationally efficient and memory efficient [4, 5, 10]. Anomaly detection in time-series data streams is challenging in three aspects [11]: (1) the dynamic nature of the data streams may result in changes in the data distributions over time (called *concept drift*); (2) storing the data for further analysis is not feasible given the high-velocity and infinite nature the data; and (3) the analysis must happen sufficiently quickly to be able to operate in real-time.

Twitter has become a popular micro-blogging platform where millions of users express their opinions on a wide range of topics on a daily basis via *tweets*, producing large amounts of data every second that can be modelled as time-series data streams and analyzed for anomalies. Twitter allows real-time collection of streams of tweets related to any specified topic keywords, hash tags, or user names through their *public streams* service [13]. This easy access to the data has enabled researchers to study and propose a broad range of techniques, including visual analytics [6, 8], sentiment analysis [2, 12], and anomaly detection [5].

The work in this paper is motivated by the challenge of providing users with timely information about different opinions relevant to topics of interest without requiring continual observation. In order to derive public opinions, tweets can

be subjected to sentiment classification, resulting in a labelling of individual tweets as positive, neutral, or negative [2]. A visual analytics based approach has been used in our prior work to discover and analyze the temporally changing sentiment of tweets posted by fans in response to micro-events occurring during a multi-day sporting event [6]. However, with this approach, in order to discover emerging micro-events that are causing significant increases in positive, neutral, or negative tweets, one must analyze data continuously. The goal of the research described in this paper is to detect in real-time anomalies in Twitter sentiment data streams, providing alerts to the analysts of the change, and enabling them to conduct further analysis immediately.

Real-time sentiment-based anomaly detection (RSAD) starts by classifying the tweets and aggregating them in temporal bins of a fixed interval (e.g., 15 minutes). Candidate anomalies are detected based on their deviation from the distribution of recent data; these are then compared to other previously seen anomalies within a sliding window to identify legitimate anomalies. This approach is resilient to concept drift, makes use of an incrementally updatable model, and is efficient enough to handle high-velocity data streams.

2 Methodology

We consider a data point to be an *anomaly* if it deviates sufficiently from nearby data points or a specified group of data points in the past. We define a *candidate anomaly* to be a data point that deviates from the local or nearby data points. Moreover, if this candidate anomaly deviates from the group of other previously detected candidate anomalies in some limited timeframe, we consider it a *legitimate anomaly*. In the remainder of this section, we explain the approach used in RSAD to detect these types of anomalies.

2.1 Pre-Processing

A unique feature of RSAD is the detection of anomalies within pre-classified data streams. The rationale for this is to allow for the independent detection of anomalous increases in tweets that are *positive*, *neutral*, or *negative* in nature. From the perspective of anomaly detection, we can consider the classification process as a pre-processing step. We use an online sentiment analysis service called Sentiment140 [12], which was designed specifically to address the short and cryptic nature of English language tweets.

In order to turn the streams of tweets into time-series data, we aggregate them over a pre-determined interval of time (e.g., 15 minutes). The granularity of this binning will affect the sensitivity to small-scale vs. large-scale anomalies and can be set based on an expectation of the velocity patterns of the tweets for the given query. Since the goal is to analyze these data streams only based on the tweet frequency, once the classification and temporal binning are performed, the actual contents of the tweets are forgotten. All that remains is the number of positive, neutral, and negative tweets that were seen in each time period. These frequency counts serve as the data points (d_t) for anomaly detection stage.

2.2 Candidate Anomaly Detection

To detect the candidate anomalies from among the local context of data points, we consider a deviation-based approach using two possible methods for determining the average of the previously seen data points: exponentially weighted moving average (EWMA) [9] and probabilistic exponentially weighted moving average (PEWMA) [3]. While each of both these approaches have been used to detect outliers in streaming data in two separate contexts [3, 8], it is not clear which is most appropriate for the RDAS approach and Twitter data.

An anomaly score of a data point d_t is calculated to represent its deviation from the mean of the data points in its neighbourhood. The candidate anomaly score (CAS) is evaluated using following formula:

$$CAS(d_t) = \frac{|d_t - \mu_{c(t-1)}|}{\mu_{c(t-1)}} \tag{1}$$

where t is the time of current bin, and $\mu_{c(t-1)} = \sum_{i=(t-1)}^{N} d_i$ is the mean of recent data points. CAS was adapted from the A-ODDS technique [10], in which the neighbourhood density of each data point is determined using a probability density estimator and then the anomaly score of a data point is computed in terms of the relative distance between its neighbourhood density and the average neighbourhood density of recent data points. In the A-ODDS approach, the neighbourhood consists of a set of data points up to radius r on both sides of the data point. In the streaming context, the local neighbours of an newly arrived data point are the ones that recently arrived because the following data points have not yet been received. Thus, in our work, CAS is the relative distance of d_t to the mean of the recent data points.

If the CAS of the current data point is near zero; the point is close to the other data points. If the CAS of the current data point is a large value, then it is significantly larger or smaller than the other data points. In order to label the current data point as a candidate anomaly, the CAS should be larger than the standard deviation of the previously seen data points by some factor. The threshold condition for a data point d_t to be so labeled is given as:

$$CAS(d_t) > \tau_c * \sigma_{c(t-1)} \tag{2}$$

where $\sigma_{c(t-1)} = \sqrt{\frac{1}{N}\sum_{i=(t-1)}^{N}(d_i - \mu_{c(t-1)})^2}$ is the standard deviation of the of recent data points, and τ_c is a threshold factor for candidate anomalies. τ_c can be set by a domain expert according to the particular features of the data stream. A lower value of τ_c increases sensitively to clustered anomalies, whereas a higher value increases sensitivity to dispersed anomalies.

With each new data point, it is necessary to update $\mu_{c(t)}$ and $\sigma_{c(t)}$. A naïve approach is to maintain all the data points in the N previous steps, and use the standard formulation to evaluate $\mu_{c(t)}$ and $\sigma_{c(t)}$. However, it would be difficult to determine an appropriate value for N that would be feasible in the streaming context without losing accuracy. Another approach is to use an exponentially

weighted moving average (EWMA) [9] and incrementally update $\mu_{c(t+1)}$ and $\sigma_{c(t+1)}$ as given in the equations:

$$\mu_{c(t)} = \alpha_{EWMA} * \mu_{c(t-1)} + (1 - \alpha_{EWMA}) * d_t \tag{3}$$

$$\sigma_{c(t)} = \alpha_{EWMA} * \sigma_{c(t-1)} + (1 - \alpha_{EWMA}) * |d_t - \mu_{c(t-1)}| \tag{4}$$

Here $0 < \alpha_{EWMA} < 1$ is the decay weighting factor. The α_{EWMA} parameter controls the weight distribution between the new data point d_t and the old mean $\mu_{l(t-1)}$; a value of 0 implies no weight on the history, while a value of 1 implies all weight on the history. An inherent assumption with the EWMA approach is that the mean is changing gradually with respect to the exponential weighting parameter α_{EWMA}, as shown in equation 3. Thus, a significant change in d_t will result in a significant increase in $\mu_{c(t)}$ and an even greater increase in $\sigma_{c(t)}$.

To increase resiliency against such changes in d_t, the value of weighting parameter α_{EWMA} can be dynamically adjusted. More precisely, if d_t changes with respect to recent data points, then a higher weight (α_{EWMA} close to 1) should be given to the recent data points; otherwise more weight should be given to d_t. Probabilistic EWMA [3] adjusts the weighting parameter based on the probability of the occurrence of the value of the current data point. The probabilistic weighting parameter is given as $\alpha_{PEWMA} = \alpha_{EWMA}\,(1 - \beta P_t)$, where P_t is the probability of occurrence of d_t and β is the weight placed on P_t. The parameter α_{EWMA} is multiplied by $(1 - \beta P_t)$ to reduce the influence of abrupt change in d_t on the moving average.

The probability density estimator equation with the standard normal distribution for P_t is given as $P_t = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{Z_t^2}{2}\right)$. While evaluating P_t for the current data point d_t, it may happen that $P_t \to 0$, if $\sigma_{c(t-1)} \to \infty$. To avoid such situations, normalization is applied to the input data points to obtain a zero-mean and unit standard deviation random variable $Z_t = (d_t - \mu_l)\,/\sigma_l$. The factor $\frac{1}{\sqrt{2\pi}}$ is the constant height and it is selected to normalize P_t such that $0 < P_t < \frac{1}{\sqrt{2\pi}}$. The drawback of considering the standard normal distribution is that for larger value of d_t, $P_t \to 0$. However, our approach does not require that the deviation of d_t be large, as long as it is sufficiently deviated from the underlying data distribution. By adjusting equation 3, with the probabilistic weighting factor [3], we get:

$$\mu_{c(t)} = \alpha_{PEWMA} * \mu_{c(t-1)} + (1 - \alpha_{PEWMA}) * d_t \tag{5}$$

2.3 Legitimate Anomaly Detection

To detect whether a candidate anomaly should be considered a legitimate anomaly, we use a one-sided sliding window of length W_t (e.g., 6 days). In contrast to the conventional method of maintaining all past data points in the sliding window, we maintain only those data points that are identified as candidate anomalies. We consider a window-based deviation approach using two possible methods for determining the deviation of the data points in the window: standard deviation

(STD), based on the simple arithmetic mean, and median absolute deviation (MAD), based on the median. While each approach has been used to detect outliers in static time series data, it is not clear which is most appropriate for a sliding window.

To determine whether a candidate anomaly should be considered as legitimate, the legitimate anomaly score (LAS) is calculated. The LAS of a data point represents its deviation from the mean of the candidate anomalies in the window. For the current data point d_t, the equation for LAS is computed as:

$$LAS(d_t) = \frac{|d_t - \mu_{w(t-1)}|}{\mu_{w(t-1)}} \tag{6}$$

where $\mu_{w(t-1)} = \sum_{i=(t-1)}^{W_t} A_{c(i)}$ and W_t is the window length. LAS gives the relative distance of d_t with respect to the mean of the candidate anomalies in the window.

The significance of LAS is similar to that of CAS in equation 1. Thus, the value of LAS for data point d_t should be sufficiently large in order to label it as a legitimate anomaly. The cutoff condition is given as:

$$LAS(D_t) > \tau_l * \sigma_{w(t-1)} \tag{7}$$

where $\sigma_{w(t)} = \sqrt{\frac{1}{W_t} \sum_{i=(t-1)}^{W_t} (d_i - \mu_{w(t-1)})^2}$, the standard deviation (STD) estimated from the simple arithmetic mean of the recent candidate anomalies in the window. τ_l, is a threshold factor for legitimate anomalies.

At each step of the algorithm we update the mean $\mu_{w(t)}$ and standard deviation $\sigma_{w(t)}$ with respect to the sliding window. Since only candidate anomalies are maintained in the window, the number of data points is relatively small. In such small data cases, the standard deviation technique is strongly affected by presence of extreme values [7]. As a result, statistical techniques that are robust against extreme anomalies are recommended, such as median and median absolute deviation (MAD) [7]. The median of the sliding window of previously detected candidate anomalies is given as: $\hat{\mu_{w(t)}} = median(W_t)$. The median absolute deviation (MAD) is calculated as, $\hat{\sigma_{w(t)}} = median_i\left(|d_t - median_j\left(W_t\right)|\right)$.

3 Preliminary Experimental Evaluation

In order to evaluate and compare the different alternatives for identifying candidate and legitimate anomalies, we performed anomaly detection experiments using Twitter data streams collected during 2013 Le Tour de France cycling races [6]. This event was held from June 29 - July 21, 2013, and is the premier race in professional cycling. The data set contains 449,077 English tweets retrieved from the Twitter public stream that were posted using the official hash tag ("#tdf") during the event period. Since the event is no longer live, for the purposes of this evaluation, we simulated an artificial data stream using these tweets. Given the features of this data, the aggregation period was set to 15 minutes, and the sliding window length was set to 6 days.

The combination of the two models that can be applied to detect candidate anomalies (EWMA and PEWMA) and the two models that summarize the statistical properties of the sliding window (STD and MAD) result in four approaches to be evaluated. The threshold parameters for the candidate and legitimate anomaly detection steps (τ_c and τ_l, respectively) were independently manipulated in the range $[1, 5]$. The decay factor for both EWMA and PEWMA was fixed at $\alpha_{EWMA} = 0.97$ and $\alpha_{PEWMA} = 0.99$ respectively, which are optimal minimum mean square error parameters in many settings [3].

For the experiments, we leveraged an open source, real-time distributed stream processing framework, called *Apache Storm* [1]. The four approaches were implemented in the Storm framework independently and then evaluated with the input of the simulated data stream. In the absence of classification labels indicating known anomalies in the tweets data stream, we worked with domain experts to assess the false positives and false negatives identified in the data. For each experimental setting, precision and recall were calculated, along with the F-score. Furthermore, since our goal was to discover an approach that works well across all three sentiment classes, we averaged the F-score over the positive, neutral, and negative data streams for each experimental setting.

3.1 Results and Analysis

Given the combination of the two alternatives in the candidate anomaly step with the two alternatives in the legitimate anomaly step, we arrived at four approaches to evaluate: EWMA-STD, EWMA-MAD, PEWMA-STD, and PEWMA-MAD. The results of the experiments in the manipulation of the threshold parameters τ_c and τ_l are provided in Figure 1.

The first item of note from these experiments is that the STD approach (Figures 1a and 1c) is very sensitive to the value of τ_l. As this parameter is increased, the method for determining if a candidate anomaly is considered a legitimate anomaly will be more strict. While this resulted in high precision (those that met this criteria were clearly anomalies), the recall was adversely affected with many actual anomalies not being detected. This pattern held for both the EWMA and PEWMA approaches.

Considering the MAD approach, a similar pattern of the F-score decreasing as τ_l increases holds for the EWMA approach (Figure 1b). Furthermore, as τ_c increases, there is also a general pattern of the F-score decreasing. This is due to extreme anomalies having an significant impact on the mean value, making it difficult to discover additional anomalies in the local context when the threshold value is high. As a result, this produced a high precision and a low recall. For the PEWMA approach (Figure 1d), it is clear that this method was more resilient to the settings of the parameters. This was due to the more effective approach for calculating the mean value within the local context, which also made the candidate anomalies within the sliding window more representative of the true anomalies.

The highest F-score achieved across all 100 experimental settings was 0.80 (PEWMA-MAD, $\tau_c = 4$, $\tau_l = 4$). While the other methods approached this

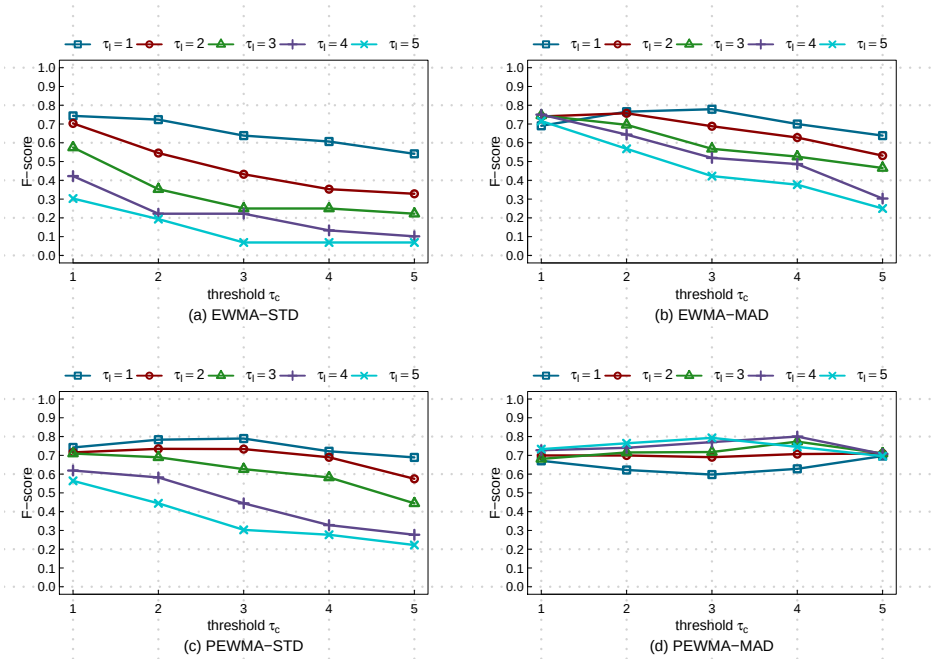

Fig. 1. Candidate anomalies identified as legitimate using STD and MAD, estimated wit mean and the median respectively. (Tweets aggregated at 15 minutes interval)

value for certain settings, given the resilience of PEWMA-MAD to the threshold parameters, we conclude that it is the superior approach for our purposes.

3.2 Real-Time Performance

In terms of computational complexity, the calculations used to determine the candidate anomalies are linear due to the incremental nature of calculating the EWMA and PEWMA. When determining whether a candidate anomaly is a legitimate anomaly, it is necessary to loop over all of the candidate anomalies within the current window. As such, this step has a complexity of $O(n)$, where n is the maximum number of potential candidate anomalies. Given a window size of 6 days and an aggregation interval of 15 minutes, the worst-case value for n is 576. Clearly, with these settings, the approach can be considered to run in real-time. Even with an extremely high velocity data stream, as long as the aggregation interval is kept in the minute-range, the approach will be able to keep up on a sufficiently fast computer system.

4 Conclusion and Future Work

In this paper we have highlighted the problem of real-time detection of changes in the sentiment in Twitter data steams. We showed that the proposed RSAD

approach can efficiently detect anomalies in presence of temporal drift when used with PEWMA-MAD technique. In the experimental evaluation of the candidate algorithms for detecting anomalies within the 2013 Le Tour de France data set, we found that the PEWMA-MAD approach was accurate and resilient to the settings of threshold parameters. Future work will focus on evaluating the RSAD approach over multiple datasets and compare it to other approaches from the literature. Furthermore, we wish to expand this approach to identify cyclical patterns in the data, in order to exclude these from being detected as anomalies.

References

1. Apache Storm. https://storm.apache.org/, (accessed January 1, 2015)
2. Bifet, A., Frank, E.: Sentiment knowledge discovery in twitter streaming data. In: Pfahringer, B., Holmes, G., Hoffmann, A. (eds.) Discovery Science, Lecture Notes in Computer Science, vol. 6332, pp. 1–15. Springer Berlin Heidelberg (2010)
3. Carter, K.M., Streilein, W.W.: Probabilistic reasoning for streaming anomaly detection. In: Proceedings of the Statistical Signal Processing Workshop (SSP). pp. 377–380. IEEE (2012)
4. Gupta, M., Gao, J., Aggarwal, C.C., Han, J.: Outlier detection for temporal data: A survey. Knowledge and Data Engineering, IEEE Transactions on 26(9), 2250–2267 (Sept 2014)
5. Guzman, J., Poblete, B.: On-line relevant anomaly detection in the twitter stream: an efficient bursty keyword detection model. In: Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description. pp. 31–39. ACM (2013)
6. Hoeber, O., Hoeber, L., Wood, L., Snelgrove, R., Hugel, I., Wagner, D.: Visual twitter analytics: Exploring fan and organizer sentiment during Le Tour de France. In: Proceedings of the VIS Workshop on Sports Data Visualization. pp. 1–7. IEEE (2013)
7. Leys, C., Ley, C., Klein, O., Bernard, P., Licata, L.: Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. Journal of Experimental Social Psychology 49(4), 764 – 766 (2013)
8. Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: Twitinfo: Aggregating and visualizing microblogs for event exploration. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 227–236. ACM (2011)
9. Münz, G., Carle, G.: Application of forecasting techniques and control charts for traffic anomaly detection. In: Proceedings of the ITC Specialist Seminar on Network Usage and Traffic. Logos Verlag (2008)
10. Sadik, S., Gruenwald, L.: Online outlier detection for data streams. In: Proceedings of the Symposium on International Database Engineering & Applications. pp. 88–96. ACM (2011)
11. Sadik, S., Gruenwald, L.: Research issues in outlier detection for data streams. SIGKDD Explor. Newsl. 15(1), 33–40 (Mar 2014)
12. Sentiment140. http://www.sentiment140.com/, (accessed December 10, 2014)
13. Twitter Public Streams. https://dev.twitter.com/streaming/public, (accessed December 10, 2014)