Mining Streaming Tweets for Real-Time Event Credibility Prediction in Twitter

Jun Zou, Faramarz Fekri, and Steven W. McLaughlin
Georgia Institute of Technology
Atlanta, GA 30332, USA
Email: {junzou, fekri, swm}@ece.gatech.edu

***Abstract*—Social media like Twitter has been widely adopted for information dissemination due to its convenience and efficiency. However, false information and rumors on social media are undermining its utility as a valuable real-time information source. Existing works for information credibility analysis are based on offline batch analysis, often incurring a long lag since the event first occurs. In this paper, we develop a generative probabilistic model for real-time event credibility prediction in Twitter. We propose an online prediction algorithm based on streaming tweets, without storing or reprocessing the past tweets. We evaluate both the offline batch prediction and online streaming prediction performance of the proposed model on the Twitter dataset. The empirical results show that its batch prediction performance outperforms other algorithms based on aggregation analysis, and the online prediction performance quickly approaches that of the batch prediction with only a few hundred tweets.**

I. Introduction

Online social media services like Twitter are widely adopted by people to self-report activities and stories happening around them. Monitoring social media streams, e.g., tweets in Twitter, becomes an effective way to detect real-time events and monitor emergent situations [1], [2]. However, social media is also increasingly exploited to spread rumors and false information, e.g., fake images during Hurricane Sandy [3]. False rumors in social media can potentially reach millions of people in short amount of time. Counter measures are thus needed to curb false information from undermining the integrity and utility of social media.

Researchers have been exploring ways of automatically assessing information credibility [4], [5]. The existing works relied on offline aggregation analysis, where a complete set of tweets related to social events are required to extract aggregation features such as the depth of the propagation tree based on the retweets. However, because collecting a complete set of tweets often causes significant delay, this approach is not suitable when we need to detect false events as early as possible.

In this paper, we develop a probabilistic generative model for real-time event credibility prediction with streaming tweets. The model depicts the generative process of individual tweets. Tweets related to true and false events are drawn from different distributions. In addition, the model also captures the interaction between the social media community and tweets. The research in [6] on information propagation in Twitter showed that tweets related to false rumors are propagated differently from tweets of true news because rumors are more likely to be questioned and denied by the Twitter community.

Based on the generative model, we propose an online streaming prediction algorithm. In contrast to offline aggregation analysis that requires a complete set of tweets related to an event, the proposed algorithm only uses the currently observed streaming tweets. The algorithm updates prediction without the need to store or reprocess the past tweets. We conduct experiments on the dataset of tweets collected from Twitter. The results show that the online prediction performance quickly approaches that of the batch prediction with only a few hundred tweets.

II. Related Works

Automatic methods for assessing event credibility in social media services like Twitter were studied in [4], wherein the authors explored various aggregation features given a set of tweets related to an event, and employed machine learning tools, e.g., Decision Tree, for predicting event credibility. In [5], the authors also identified other prominent features including temporal, structural, and linguistic features. However, those features proposed in the above works require almost a complete set of tweets related to the event, which usually means continuously collecting tweets over a long period of time, causing significant delay in the prediction task since the event first takes place.

The work in [7] proposed methods for real-time credibility assessment of individual tweets, where a semi-supervised ranking model was used to assign credibility scores to tweets in a user's timeline based on features derived from single tweets. Since it only requires the data of each individual tweet without assuming the complete data of an event, it is able to generate results in real time. Another work in [8] studied the credibility of tweets during high impact events. But neither of the works addressed the real-time prediction of event credibility.

Streaming processing of tweets was studied in [9]–[11]. In [9], the authors proposed a streaming model of computation for detecting new events from a stream of Twitter posts. They achieved significant speedup in processing the large volume of data coming from Twitter. The work in [10] predicted latent user attributes in Twitter with stream-based classification, and it showed that the bag-of-words classification model can be decomposed into a series of streaming updates. The work in [11] applied Bayesian rule update to dynamical models for real-time streaming prediction of political preferences of Twitter users.

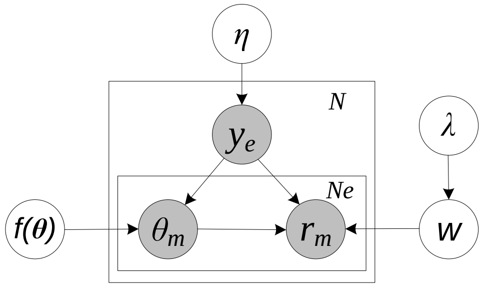

Fig. 1: Graphical representation of the generative model.

III. Generative Model

In Twitter, users post messages, also known as tweets, to describe happening events around them. Previous research in [4] has shown that the tweets related to true events and false events exhibit different characteristics, which can be exploited for automatic event credibility analysis. Meanwhile, the social media community interact with messages by retweeting and favoriting messages. According to the study in [6], the community interact differently with tweets related to true and false events, as messages related to false events are questioned and denied more than messages related to true events.

Our goal is to predict the credibility label of a new event with the relevant messages collected from the streaming tweets in Twitter. In particular, we hope to predict the label in real time, that is, to start prediction after observing only a few tweets in the early stage, rather than wait until collecting a complete set of tweets over a long period of time. In the following, we introduce a generative probabilistic model that describes the generative process of tweets related to true and false events as well as the community interactions, and specify the dependency relationships between various variables. Using the generative model, we develop a real-time online prediction algorithm with streaming tweets.

A. Generative Process

We denote an event in Twitter as e, and use y_e to denote its label, $y_e = 1$ if true and $y_e = 0$ if false. The label y_e is modelled by $y_e \sim \text{Bernoulli}(\eta)$, where η can be used to specify prior knowledge about labels. In this work, we always set $\eta = 0.5$, i.e., no prior information is known. We define a set of messages related to event e as $M_e = \{m_1(e), m_2(e), \ldots, m_{N_e}(e)\}$, where N_e is the number of observed messages. Each message m can be characterized by a set of features represented as $\boldsymbol{\theta}_m = [\theta_1, \ldots, \theta_L]$, where L is the number of features. The features can be directly extracted from the individual messages. To model feature vector $\boldsymbol{\theta}_m$ related to true and false events, we use a mixture distribution $f(\boldsymbol{\theta})$, where

$$f(\boldsymbol{\theta}|y_e) = y_e f_1(\boldsymbol{\theta}) + (1 - y_e) f_0(\boldsymbol{\theta}). \tag{1}$$

Given the event is true, $y_e = 1$, $\boldsymbol{\theta}_m$ is drawn from the distribution $f_1(\boldsymbol{\theta})$, otherwise $\boldsymbol{\theta}_m$ is drawn from the distribution $f_0(\boldsymbol{\theta})$. We use non-parametric density estimation methods to learn $f(\boldsymbol{\theta})$ from training data.

The community interaction on message m is denoted by r_m, $r_m = 1$ for positive feedbacks and $r_m = 0$ for negative feedbacks. We consider a message receives positive feedback if the total number of "Retweets" and "Favorites" is greater than a preset threshold, and negative feedback vice versa. The community feedback r_m is modelled as $r_m \sim \text{Bernoulli}(p_m)$, where p_m depends on both the massage feature $\boldsymbol{\theta}_m$ and event label y_e, $p_m = y_e S(\mathbf{w}_1, \boldsymbol{\theta}_m) + (1 - y_e) S(\mathbf{w}_0, \boldsymbol{\theta}_m)$. $S(\mathbf{w}_i, \boldsymbol{\theta})$, $\forall i = 0, 1$, represents a sigmoid function, $S(\mathbf{w}_i, \boldsymbol{\theta}) = \frac{1}{1+\exp\{-(w_{i0}+w_{i1}\theta_1+\ldots+w_{iL}\theta_L)\}}$. The coefficients $\mathbf{w}_i = [w_{i0}, w_{i1}, \ldots, w_{iL}]$ capture how the community interact with messages related to true events and false events. Let $W = \{\mathbf{w}_i, \forall i = 1, 0\}$. We can write the probability distribution of r_m conditioned on $\boldsymbol{\theta}_m$ and y_e as

TABLE I: The notations of the generative model.

N	number of events
N_e	number of relevant messages for event e
y_e	label of event e
$\boldsymbol{\theta}_m$	feature vector of message m
r_m	community feedback on message m
p_m	parameter of the Bernoulli distribution for r_m
$\mathbf{w}_i$	coefficient vector for sigmoid function S
λ, η	prior parameters for $\mathbf{w}_i$ and y_e, respectively

$$P(r_m|\boldsymbol{\theta}_m, y_e, W) = \Big[(S(\mathbf{w}_1, \boldsymbol{\theta}_m))^{y_e} (S(\mathbf{w}_0, \boldsymbol{\theta}_m))^{1-y_e}\Big]^{r_m} \times \Big[(1 - S(\mathbf{w}_1, \boldsymbol{\theta}_m))^{y_e} (1 - S(\mathbf{w}_0, \boldsymbol{\theta}_m))^{1-y_e}\Big]^{1-r_m}. \tag{2}$$

The prior distributions for coefficients $\mathbf{w}_i$ are given by $\mathbf{w}_i \sim \text{Gaussian}(0, \lambda^{-1}\mathbf{I})$ for regularization.

The probabilistic graphical representation of the generative model is provided in Fig. 1. The notations are summarized in Table I. We describe the generative process as follows:

1) Draw $\mathbf{w}_i \sim \text{Gaussian}(0, \lambda^{-1}\mathbf{I})$, $\forall i \in \{0, 1\}$.
2) For each event e:
   a) Draw its label $y_e \sim \text{Bernoulli}(\eta)$;
   b) For each message m related to event e:
      - Draw its feature $\boldsymbol{\theta}_m \sim f(\boldsymbol{\theta})$;
      - Draw its feedback $r_m \sim \text{Bernoulli}(p_m)$.

B. Model Learning

Let $E = \{e_1, e_2, \ldots, e_N\}$ represent the set of events. We denote $Y = \{y_e | e \in E\}$ as the set of event labels, $M = \cup_{e \in E} M_e$ as the set of relevant messages of all events, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_m | m \in M\}$ as the set of features of all messages, $R = \{r_m | m \in M\}$ as the set of community feedbacks on all messages, and $\Gamma = \{\eta, \lambda\}$ as the set of hyper parameters. The joint distribution of the observed data is given by

$$\begin{aligned} P(\boldsymbol{\Theta}, R, Y; W, \Gamma) &= P(R|\boldsymbol{\Theta}, W, Y) P(\boldsymbol{\Theta}|Y) P(Y|\Gamma) \\ &= \prod_{e \in E} \prod_{m \in M_e} P(r_m|\boldsymbol{\theta}_m, y_e, W) f_1(\boldsymbol{\theta}_m)^{y_e} f_0(\boldsymbol{\theta}_m)^{1-y_e} \\ &\quad \times \eta^{y_e} (1 - \eta)^{1-y_e}. \end{aligned} \tag{3}$$

Since $f(\boldsymbol{\theta})$ is independent of other variables given y_e and $\boldsymbol{\theta}_m$, we can estimate $f(\boldsymbol{\theta})$ from the observed data of y_e and $\boldsymbol{\theta}_m$. We apply the non-parametric kernel density estimation method. There are various kernel functions can be used, e.g., Tophat and Gaussian kernels.

We next estimate the parameter W by maximizing the posterior probability given the observed data

$$\begin{aligned} L(W) &= \log P(W|\boldsymbol{\Theta}, R, Y, \Gamma) \\ &= \sum_{i\in\{0,1\}} \sum_{e\in E_i} \sum_{m\in M_e} r_m \log S(\mathbf{w}_i, \boldsymbol{\theta}_m) + \\ &(1-r_m)\log(1-S(\mathbf{w}_i, \boldsymbol{\theta}_m)) - \frac{\lambda}{2}\mathbf{w}_i^\top \mathbf{w}_i + c_0 \qquad (4) \end{aligned}$$

where c_0 is some constant. We employ the numerical Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [12] to solve for the unique optimal solution.

C. Online Prediction with Streaming Tweets

Let $M_e^{(h)}$ denote the observed tweets related to event e during a short period of T_h. The posterior probability of the event label is given by

$$P(y_e|M_e^{(h)}) \propto \prod_{m\in M_e^{(h)}} P(r_m|\boldsymbol{\theta}_m, y_e) f(\boldsymbol{\theta}_m|y_e) P(y_e|\eta). \qquad (5)$$

At the beginning, we can initialize the parameter η for the prior distribution of y_e with any prior information. We then use the posterior to update the prior for y_e,

$$P(y_e|\eta^{(h)}) = P(y_e|M_e^{(h)}). \qquad (6)$$

As we continue to observe new streaming tweets $M_e^{(h+1)}$ for period T_{h+1}, we compute the new posterior for y_e as

$$P(y_e|M_e^{(h+1)}) \propto \prod_{m\in M_e^{(h+1)}} P(r_m|\boldsymbol{\theta}_m, y_e) f(\boldsymbol{\theta}_m|y_e) P(y_e|\eta^{(h)}). \qquad (7)$$

Therefore, the online streaming prediction algorithm does not need to store or reprocess the tweets observed in the past. After each period T_h, we can update the prediction of event label as $\arg\max P(y_e|M_e^{(h)})$.

IV. Experiments

We run experiments on a dataset of tweets collected from Twitter.com, and evaluate the event credibility prediction performance of the proposed model.

A. Datasets

1) Collection: A set of events and trending topics were collected from facebook.com, twitter.com, and snopes.com on a daily basis. Note that snopes.com also provides labels for the listed events. For other unlabelled events, we asked 5 annotators to label the events as true or false. They consulted external resources including major news websites such as cnn.com and nytimes.com to determine the credibility of events. To reduce uncertainty in event labelling, we only included the events whose labels were agreed upon by at least 4 annotators.

For each event, we gathered tweets relevant to the event from twitter.com via Twitter Search API[1]. We constructed a query for each event with a group of keywords, and sent the search request to twitter.com to retrieve tweets contain all keywords in the query. Each returned tweet contains the tweet text, the author information, and the number of "Retweets" and "Favorites". In the experiments, we define a tweet receives positive feedback if the total count of "Reweets" and "Favorites" is greater than 1 and negative feedback vice versa. The Twitter Search API only returns tweets from the past week. In addition, to avoid collecting duplicated tweets, we removed all retweets.

TABLE II: Aggregation features for event classification.

Author	Average registration age Average number of posted tweets Average number of followers Average number of friends
Content	Fraction of messages contain URLs Average sentiment score Fraction of messages with positive sentiment Fraction of messages with negative sentiment Fraction of messages contain user mentions Fraction of messages contain question marks Fraction of messages contain first pronouns Fraction of messages with positive feedback

We limited the maximum number of collected tweets to 500 per event, and also discarded events with less than 30 tweets. We continued this process for a period of three months from November 2014 to January 2015 until collected 104 events consisting of 52 true events and 52 false events. The total number of collected relevant tweets is 29,345.

2) Features: For each collected tweet, we extract a set of features similar to those discovered in [4], including the author-based features: registration age, number of posted tweets, number of followers, number of friends, and friend-follower ratio; the content-based features: sentiment score, subjectivity score, contains URLs, contains user mentions, contains question marks, and contains first pronouns. Note that the friend-follower ratio is computed as $(\text{number of friends} + 100)/(\text{number of followers} + 100)$, so as to smooth the ratio when the number of followers is very small.

The raw values of author features vary over a very wide range. We apply the transformation $F(f_a) = \frac{f_a}{f_a + d_f}$ to author feature f_a, where d_f is some constant for each feature. We set d_f as the median of observed f_a values. The transformed feature values are now in the range of $[0, 1]$.

B. Batch Prediction Performance

We first evaluate the offline batch prediction performance of the proposed algorithm, where all collected relevant tweets are used for prediction. We set the prior parameter for regularization of W as $\lambda = 1$, and use Tophat kernel with a bandwidth of 0.4 for the nonparametric kernel estimator in $f(\boldsymbol{\theta})$.

1) Baselines: The baseline algorithms are the classification algorithms using the aggregation features of events as proposed in [4]. We apply the Decision Tree and linear Support Vector Machine (SVM) methods for this purpose. For those algorithms, we use the aggregation features presented in

[1] https://dev.twitter.com/rest/public/search

TABLE III: Batch prediction performance for the true event and false event classes.

Algorithm	All	True Event Class			False Event Class		
	Accuracy	Precision	Recall	F_1 score	Precision	Recall	F_1 score
Proposed	0.823 ± 0.083	0.829 ± 0.096	0.826 ± 0.105	0.822 ± 0.085	0.838 ± 0.098	0.820 ± 0.105	0.821 ± 0.083
SVM	0.771 ± 0.126	0.807 ± 0.125	0.760 ± 0.154	0.763 ± 0.127	0.766 ± 0.149	0.783 ± 0.171	0.765 ± 0.151
Decision Tree	0.725 ± 0.091	0.735 ± 0.100	0.740 ± 0.165	0.715 ± 0.106	0.764 ± 0.135	0.710 ± 0.121	0.720 ± 0.093

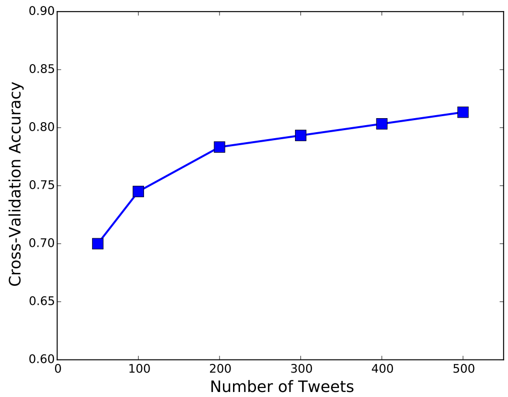

Fig. 2: Online prediction performance with streaming tweets.

Table II including the author-based features and the content-based features, which were shown to have good discrimination power. We omit some propagation-based aggregation features, e.g., the maximum size of a level in the propagation tree, because they require the complete set of tweets relevant to an event in order to reconstruct the tweet propagation tree, but our dataset does not guarantee such completeness. Also, generating those features often causes much longer delay since the propagation of tweets takes time.

2) Performance Comparison: We run 10-fold cross validation and report the average results with 95% confidence intervals. Table III shows the overall classification accuracy as well as the precision and recall performance of both the true event and false event classes. The proposed model achieves an overall prediction accuracy of 82.3%, and it outperforms the baseline methods across all metrics. Therefore, modelling events at the individual message level is a more effective approach to event credibility prediction when only a subset of tweets related to events are available.

C. Online Streaming Prediction

We investigate the online prediction performance of the proposed model with streaming tweets, where the probability of the event label is updated online through (7). In Fig. 2, we show the online prediction accuracy versus the number of observed tweets. The performance first improves significantly as the number of tweets increases. The algorithm achieves a quite good accuracy of 78.3% even with only 200 tweets. After the number of tweets reaches around 200 to 300, the performance improvement slows down. This is because after the number of past tweets becomes large, a relatively small number of new streaming tweets can only slightly influence the prediction. Also, as more and more tweets are observed, the accuracy converges to the limit of the prediction performance of the model. The results confirm that the online prediction algorithm quickly approaches the prediction accuracy of the batch prediction with only a few hundred tweets.

V. Conclusions

In this paper, we developed a probabilistic generative model for real-time event credibility prediction in Twitter. In contrast to aggregation analysis techniques that require a complete set of tweets related to events, our model only needs signals extracted from individual tweets. Using the generative model, we proposed an online streaming prediction algorithm without storing or reprocessing the past tweets. Through experiments on the Twitter dataset, we showed that the batch prediction performance of our model outperforms other algorithms based on aggregation analysis, and the online streaming prediction performance quickly approaches that of the batch prediction with only a few hundred tweets.

VI. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1115199.

References

[1] T. Sakaki, M. Okazaki, and Y. Matsuo, “Earthquake shakes twitter users: real-time event detection by social sensors,” in *WWW’10*.

[2] X. Wang, L. Tokarchuk, and S. Poslad, “Identifying relevant event content for real-time event detection,” in *ASONAM’14*, 2014.

[3] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, “Faking sandy: Characterizing and identifying fake images on twitter during hurricane sandy,” in *WWW’13 Companion*, 2013, pp. 729–736.

[4] C. Castillo, M. Mendoza, and B. Poblete, “Information credibility on twitter,” in *WWW’11*.

[5] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wnag, “Prominent features of rumor propagation in online social media,” in *ICDM’13*.

[6] M. Mendoza, B. Poblete, and C. Castillo, “Twitter under crisis: Can we trust what we rt?” in *SOMA’10*, 2010, pp. 71–79.

[7] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, “Tweetcred: Real-time credibility assessment of content on twitter,” in *SocInfo’14*, 2014, pp. 228–243.

[8] A. Gupta and P. Kumaraguru, “Credibility ranking of tweets during high impact events,” in *Proc. 1st Workshop on Privacy and Security in Online Social Media (PSOSM)*, 2012.

[9] S. Petrovi, M. Osborne, and V. Lavrenko, “Streaming first story detection with application to twitter,” in *Proc. NAACL HLT*, 2010, pp. 181–189.

[10] B. V. Durme, “Streaming analysis of discourse participants,” in *Proc. EMNLP-CoNLL*, 2012, pp. 48–58.

[11] S. Volkova, G. Coppersmith, and B. V. Durme, “Inferring user political preferences from streaming communications,” in *ACL’14*, 2014, pp. 186–196.

[12] J. Nocedal and S. J. Wright, *Numerical Optimization*, 1999.