Neurocomputing 149 (2015) 1469–1480

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Event detection and popularity prediction in microblogging

Xiaoming Zhang [a,*], Xiaoming Chen [a], Yan Chen [a], Senzhang Wang [a], Zhoujun Li [a], Jiali Xia [b]

[a] State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
[b] School of Software, Jiangxi University of Finance & Economics, Nanchang, China

ARTICLE INFO

Article history:
Received 1 September 2013
Received in revised form
6 August 2014
Accepted 22 August 2014
Available online 3 September 2014

Keywords:
Event detection
Popularity prediction
Burst words
Burst event

ABSTRACT

As one of the most influential social media platforms, microblogging is becoming increasingly popular in the last decades. Each day a large amount of events appear and spread in microblogging. The spreading of events and corresponding comments on them can greatly influence the public opinion. It is practical important to discover new emerging events in microblogging and predict their future popularity. Traditional event detection and information diffusion models cannot effectively handle our studied problem, because most existing methods focus only on event detection but ignore to predict their future trend. In this paper, we propose a new approach to detect burst novel events and predict their future popularity simultaneously. Specifically, we first detect events from online microblogging stream by utilizing multiple types of information, i.e., term frequency, and user's social relation. Meanwhile, the popularity of detected event is predicted through a proposed diffusion model which takes both the content and user information of the event into account. Extensive evaluations on two real-world datasets demonstrate the effectiveness of our approach on both event detection and their popularity prediction.

1. Introduction

Microblogging sites such as Twitter and SinaWeibo have become a popular way for users to share and disseminate information. Most microblogging services allow users to post short text message. For example, the length of content posted in SinaWeibo is limits to no more than 140 characters. As an important information sharing and consuming platform, microblogging sites usually hasa large volume of users, and generate huge number of contents every day. Take Twitter as an example, by July 2012 Twitter has over 500 million users, and the users collectively create over two billion tweets each week. With the huge volume of users and contents, microblogging provides us a desirable platform to study the spreading information from many perspectives. First, microblogging captures everything from mundanely daily routine of the masses to spectacular breaking events to other messages of significant historic impact. As a result, it is very difficult for users to capture new events which will be popular in the future. Thus, there is an urgent need to automatic detect event online. However, it is very challenging to effectively detect event in microblogging. The primary obstacle is that the language used in tweets is usually substantially different from that used in traditional text due to the length constraint of the tweets. Second, many researchers have shown that microblogging provides a probe into the sentiments and behaviors of entire popularity, enabling what has come to be known as computational social science. For example, content generated in social network sites can be used to predict the adoption of a new product or the result of an election [16]. As for the news events, it would be of interest to predict how popular they will be. Analyzing the spread of event would also be of interest in many domains. For example, online advertisers could use this analysis for efficiently targeted marketing campaigns. Public organizations can know how the society is influenced by the event and then determine how to reply to the public opinion.

In this paper, we study how to detect burst events from the stream of microblogging contents, such as “Hurricane Sandy hitting New York”, “Japan Earthquake in 2011” and “Beijing rainstorm in 2012”. Then, we further predict its popularity in the near future by modeling the spreading of the event. Burst event detection and information diffusion in social network are extensively studied For example, some works utilized a feature-pivot strategy to detect event in social media [11,19,43].These methods first detect burst words based on their frequency distributions on the time axis. Then, burst words are clustered, and each cluster represents a burst event. However, the definition of burst event in these works is only based on the novelty of event. The popularity of burst event in the near future is usually ignored. Moreover, as the length of micro-blog document is limited, the representation of micro-blog using terms vector faces the feature

* Corresponding author. Tel.: +86 10 82338247.
E-mail address: yolixs@buaa.edu.cn (X. Zhang).

http://dx.doi.org/10.1016/j.neucom.2014.08.045
0925-2312/

sparsity problem. This problem can be alleviated by exploiting other types of web resource. Usually, an event appears and spreads in multiple sources simultaneously, such as microblogging sites, blog sites, web forum and traditional news sites. Thus, these web resources can be used to enrich the information of micro-blog documents.

The information spread model has also been studied for many years. These works can be categorized into two groups. The first group focuses on the topology of social graph, investigating what topologies and what activation patterns facilitate efficient propagation of information. For example, the macro-level dynamics and characteristics of information diffusion are discussed [13,22], key factor that affect the adoption of behaviors is revealed [4,23] and contagion models are designed to simulate the diffusion process [36]. Some researches focus on the micro-level analysis of information spread in large scale social network, such as who got the gossip from whom or who infected whom [26]. The topology-based approaches usually make the following assumptions: (a) all the users have the same role in the spread process; (b) the spread path between users are known and the influence can only be transmitted over the edges of the underlying network; and (c) complete network data is available. However, in many scenarios, the underlying network is implicit or even unknown [40]. Furthermore, the topology-based approaches are not capable to analyze the content of the propagated information. They do not consider the difference among different users' reaction on the propagated information, either. For example, users who are interested in sports are more likely to be active in the spread of the event "Liu Xiang falls in Olympic game" than users who have little interest in sports. The other group of works mainly focuses on the analysis of the coarse-grained features of information content and network nodes [39], such as the average out-degree of node and the length of hashtag. They did not distinguish different activities and interest of users in event spread.

In this paper, we propose an approach to detect event from micro-blog stream. To tackle the challenge of data sparsity, we propose to enrich the microblogging information related to the event by exploiting multi-source content, such as blog sites, web forum and traditional news sites. We extend the burst event to incorporate temporal aspect of timeliness. In other words, we not only detect burst novel events, but also predict how popular they will be in the near future. This presents an additional challenge to model the temporal characteristics of event in real-time microblogging stream. We need to predict how popular an event will be, as soon as this event is detected. Detecting events and accurately predicting their future trends can contribute to better providing users and organizations potentially valuable information. For example, organizations may be interested in tracking the events related to them, and users would like to be informed of new burst events which are fast gathering momentum in microblogging. Considering the unique characteristics of microblogging, we use the feature-pivot method to detect micro-blogging event. Particularly, we first propose to combine the social relations of users and the frequency distribution of words to detect burst words. Then, the burst words are clustered into groups, and each group of words can be considered to be related to a specific event. The two-state model proposed by Kleinberg [19] is revised to detect burst words whose weights are real values. To cluster burst words, we construct a words graph based on their co-occurrence in micro-blogs and other web resources. Then, strongly connected burst words are clustered, and each of the clusters represents a burst event.

To predict the popularity of a detected event, we model the spread of an event by combing the posts related to the events and the social relations of related users. Different from most information diffusion models, our approach can handle graphs with incomplete structure information. The proposed approach utilizes a linear spread prediction function to predict the future popularity of the events. The linear function combines all kinds of information available, such as the influence power and interest of users, and the historical popularity information of the event. The motivation of this function is based on the following observations. First, a more active user usually contributes more to the spread of an event. To measure the activity of users in the social network, we introduce the influence power of users. Second, different users may have different interest on different topics. For example, some users prefer entertainment related topics, while some prefer politics related topics. Users are more likely to participate in events which are about the topics they are interested in and share them with their friends. To model the interest of users, a topic model is proposed. By extracting the profile of users and all the tweets of the events, the model automatically discovers the latent topics of users and the event and represents them as two topic vectors. The similarity between the two vectors is used to denote user's interest in the event. Third, if an event is very popular in the past, it is more likely to be popular in the future. To combine all above factors, the popularity of an event is assumed to be a linear function of the volumes produced by different users infected in the past and the volume introduced by its historical popularity. The main contributions of the paper can be summarized as follows:

1. We not only detect burst events that are novel, but also to predict how hot the events will be in the near future. To detect burst event by combining term's occurrence information and users' social relation information, the two-state model is improved to deal with real value.
2. We proposed a spread model based on the analysis of both event content and users' profile. The major advantage of our model is that it distinguishes users' contributions according to user's influence power and interest in the predicted event, which is different from other approaches that use the same parametric form for all events and users.
3. We further evaluate our approach in two real-life datasets, and experiment results indicate the efficiency of our approach.

The remainder of this paper is organized as follows. In the next section, we introduce related works. We formally formulate the problem in Section 3, and we propose our event detection algorithm in Section 4. Section 5 describes how to predict the popularity of an event, and the experiments are described in Section 6. Finally, the paper is concluded in Section 7.

2. Related works

The enormous amount of contents generated by social network users in the last decade is creating new challenges and new research interests for data mining, social network analysis and other related community. In this section, we present an overview of those works which are related with our work, i.e., event detection and information spread model.

The first issue when dealing with text stream is the aggregation of them. Many works try to aggregate text documents using the event detection approach [3,31,44]. Existing event detection approaches can be broadly classified into two categories: document-pivot approaches and feature-pivot approaches. The former ones detect events by clustering documents based on the similarity between documents [25,30,41], while the latter ones detect which words refer to event [19,11]. Since messages in microblogging sites are constrained to be short text (up to 140 characters), the sparse vector representation affect the similarity measure between two micro-blogs and hence affect the performance of document-pivot methods. Therefore, most of the works detect event in social media using the feature-pivot approach. We mainly introduce the works based on feature-pivot approach in the following part of this section.

Among the feature-pivot approaches, many works mainly depend on the analysis of term frequency. Kleinberg proposes to detect burst

event using an infinite-state automaton, in which burst words are modeled as state transitions [19]. Different from this work, individual word's appearance is modeled as binomial distribution in the other work [11], and then the burst of each word is identified by a threshold-based heuristic method. These works analyze word features in the time domain. Some other works analyze word features in the frequency domain. For example, Discrete Fourier Transform (DFT) is used to convert the word features from the time domain into the frequency domain [15]. However, this approach cannot locate the time periods of the bursts. Thus, the Gaussian Mixture model is used to remedy this by estimating such periods [15]. Unlike DFT, wavelet refers to a quickly varnishing oscillating function [28]. It has also been applied to detect events from folksonomies in some works [7,43,42]. A model called "keyword-based evolving graph sequences" (kEGS) is proposed to capture the characteristics of information propagation in social streams [20], which also identify events based on word frequency. However, most of these works cannot well exploit the social knowledge of micro-blog data.

Recently, there are great interests in harvesting social knowledge from microblogging. For example, new events discussed by users in Twitter are detected based on the novelty of propagated content [32]. However, these events may be trivial. A framework is proposed to detect localized events in real-time from a Twitter stream [1]. For this, spatiotemporal characteristics of keywords are continuosly extracted to identify meaningful candidates for event descriptions. Then, localized event information is extracted by clustering keywords according to their spatial similarity. In the other side, event detection is also formulated as a classification problem [37], in which new events cannot be detected. All of these works only focus on the novelty of event, and they have no analysis of event popularity in the future time units.

To predict the popularity of an event, we need to model the spread of this event when it is detected. Information spread has attracted a great attention with the developing of social media. In the early stage, a research on the spread of innovations [29] introduces a conceptual framework to study the emergence of information in network. Then, many works aim to describe the macro-level dynamics and characteristics of information spread [13,22], discover important factors which affect the adoption of behaviors [4], design a model to analysis the spread process [36,40], predict the popularity of new hashtags on Twitter by formulating the problem as a classification task [27], and predict the long time popularity of online content from early measurements of user's access [38]. In the micro-level, inference of spread stimulates great interest in indentifying the adoption of explicit behaviors [17,21,18], and machine learning approach based on the passive-aggressive algorithm is proposed to predict retweets of a tweet as well as humans [33]. Similar to these works, a rich set of efforts [10,14] have proposed viral marketing campaigns to exploit the word-of-mouth effect on information spread. To exploit the external influence, a topic spread and evolution model in social communities is also studied [35]. The space-time metadata in social media is also used to model the spread of events in space and time [12]. In particular, it illustrates the spread of one particular event – gas shortage in the aftermath of Hurricane Sandy. However, most of the works based on the micro-level analysis require knowledge of the complete network or some particular information.

The micro-blog content is also used in some works, such as profiling users [34] and modeling the inter-influence between users [9]. Some other works do refer to the topic of the propagated information [13,24], though topic is addressed in coarse granularity (e.g. "sports", "Apple", "Microsoft"). Another work examines user behavior relevant to information propagation through micro-blogging [2]. Specifically, it uses retweeting activities among Twitter users to model originating and promoting behavior. Moreover, the content feature is combined with some topology features (e.g., number of followers and retweets ratio, etc.) to minimize the error of popularity prediction [39]. However, most of these information spread models depend on a complete topology or information feature analysis, in which users' roles in the propagation is not well studied. In this study, we model the spread of event by combining the analysis of event content and user's interest, which is similar to some other macro-level works that do not need a complete topology of the network [36,40]. Moreover, we utilize event's history information and users' interest information to model the spread of event.

3. Problem formulation

In this section, we formally define the tasks of event detection and popularity prediction in micro-blog stream. Fig. 1 shows the flowchart of our approach, and Table 1 shows the notations used in this paper. We use the feature-pivot method to detect burst events. Each event is represented by a set of words that are used frequently when this event happen. These words are also known as burst words. When a burst event is detected, its popularity is

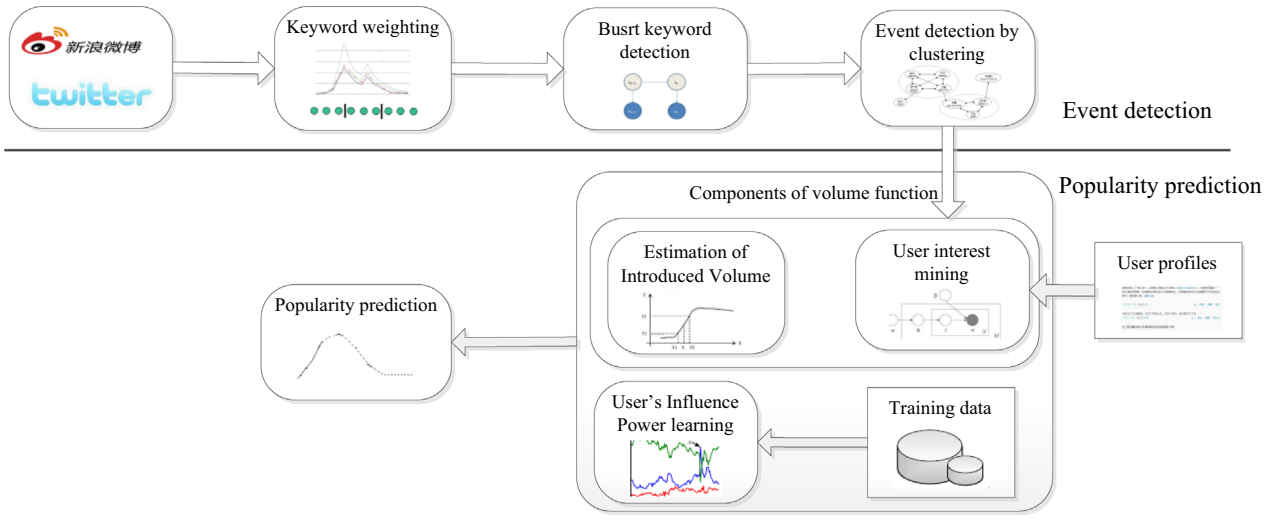


Fig. 1. Flowchart of our approach.

Table 1
Notations used in the paper.

Notation	Description
MC^t	Micro-blogs in time unit t
KW^t	The set of words within time unit t.
BW^t	Burst words BWt of the time unit t
bk_i^t	A burst key word in time unit t
E^t	Events detected in time unit t
$w_{j,v}$	Weight of the v^{th} word in micro-blog mc_j
$w_{.v}^t$	Local weight of the v^{th} word in time unit t: $w_{.v}^t = \sum_{j=1}^{\lvert MC^t\rvert} w_{j,v}$
w^t	The sum of all the words' local weights in time unit t: $w^t = \sum_v w_{.v}^t$
$rw_{i,j}^t$	The weight of relation between two burst words bk_i^t and
p_k	The probability to produce related documents in state k (0 or 1 corresponding to the "low" or "high" state)
s	The ratio between the numbers of relative documents produced in the "high" and "low" states
$C(q_{i-1},q_i)$	Cost of state transformation
$V_e(t)$	Volume of event m in time unit t
$F_u(d)$	User u's influence power d time units after he got infected
$Sim(u,e)$	User u's interest in event e
$H_e(t+1)$	The volume introduced in time unit $t+1$by its historical popularity

predicted using the spread model. We introduce the procedure of our approach based on the flowchart of Fig. 1 as following:

Given a continuous stream of incoming micro-blogs $I=<(c_1, \tau_1), (c_2, \tau_2), \ldots, (c_n, \tau_n)>$, where c_i denotes the micro-blog document and τ_i denotes the uploading time, we split it using a given time unit with length l, such as the document set in the tth time unit is: $<c_t, c_t+l>$. In each time unit t, we estimate the weight for each word and store it, and a revised state model is proposed to detect burst words BW^t based on the increase of word's weight. Then, a set of events E^t are detected by clustering those burst words BW^t into groups based on the relation graph of burst words. Each group of burst words represents an event, and the micro-blogs which contain these burst words are considered to be associated with the corresponding event.

When an event is detected, its popularity is predicted using the spread model. We define the popularity of an event $e \in E^t$ in time unit t as the volume, $V_e(t)$, which is the number of micro-blogs that discuss event e in this time unit. We model the volume $V_e(t+1)$ in the future time unit as a function of user's influence power, user's interest in the event, and the introduced volume of this event as shown in Fig. 1. We therefore learn a prediction function as following:

$$V_e(t+1) = f(U_{ie}, U_{if}, H_e, \sigma) \tag{1}$$

where U_{ie} denotes the infected users' interest in event e, U_{if} denotes the influence powers of the infected users, H_e is the volume introduced by event's historical popularity, and σ is the parameters related to the event.

Eq. (1) has several components. The first one is user' interest in event e. Usually, users who are more interested in an event would be more likely to adopt this event and more active in the spread of this event. The second one is user's influence power. User's influence power is a signature that, once this user adopts an event, how many other users will be infected by this user to adopt this event and then how many new micro-blogs will be produced. User's influence power can be learned from the training data of a number of events. Thus, if an event infects more number of influential and interested users, it will spread more widely and produce more number of micro-blogs. Besides the two components, some characteristics of event also affect its spread to some extent, e.g., its popularity in the past time units. For example, a more popular event is more likely to attract more attention in the next time units.

4. Event detection

In reality, when an event happens, many micro-blogs related to this event are produced in a short time period, and thus many words related to this event have a burst frequency in this period. We combine the normalized term frequency and user's social relation to weight words. Then, we revise the two-state model [19] to detect burst words. Finally, burst words are clustered based on the word relation graph.

4.1. Word weighting

For each micro-blog document, we extract its words and remove the stop-words. All the micro-blogs in the tth time unit t is denoted by MC^t, and KW^t is used to denote the set of words used in this time unit. The vector model is used to represent each micro-blog mc_j in MC^t as follows:

$$mc_j = (w_{j,1}, w_{j,2}, \ldots, w_{j,|KW^t|}) \tag{2}$$

where $w_{j,v}$ denotes the weight of the v^{th} vocabulary word in mc_j and is estimated using the augmented normalized term frequency as follows:

$$w_{j,v} = \left(0.5 + 0.5 \times \frac{tf_{j,v}}{tf_j^{\max}}\right) \tag{3}$$

where $tf_{j,v}$ denotes the term frequency of the v^{th} word in mc_j, and $tf_j^{\max}$ is the highest term frequency in mc_j. Because the frequencies of different words in a document may vary greatly, Eq. (3) is used to map the weight into a range of [0.5, 1].

Besides word frequency, the users who have used the word in their micro-blogs will also affect the burst of this word. User who has greater authority (e.g., has many followers) would make their words to be used by other users more frequently. This is because that many of other users will follow them and adopt these words. We use the PageRank-based method to estimate authority $au(u_i)$ of user u_i:

$$au(u_i) = \alpha + (1-\alpha) \times \sum_{u_j \in follower(u_i)} \frac{au(u_j)}{|follower(u_i)|} \tag{4}$$

where $\alpha \in (0, 1)$ is a dumping factor, and $follower(u_i)$ denotes the set of users who follow user u_i. Then the weight of the v^{th} word in micro-blog mc_j is replaced by the following equation:

$$w_{j,v} = \left(0.5 + 0.5 \times \frac{tf_{j,v}}{tf_j^{\max}}\right) \times au(user(mc_j)) \tag{5}$$

Let $w_{.v}^t = \sum_{j=1}^{|MC^t|} w_{j,v}$ be the local weight of the v^{th} word in time unit t. The greater increase of the value of $w_{.v}^t$ indicates that the word is more likely to be a burst word. Fig. 2 gives some examples of words' local weights allocated along the time line with the dataset collected from Sina Weibo site. Obviously, these words' local weights increase greatly from May 27 when the event "Shenzhen sports car accident caused 3 people death in May 2012" happens. In the next subsection, we will show how to detect burst words whose local weights increase greatly in some specific time units.

4.2. Burst word detection

As discussed above, word's local weight is a real value. We revised the discrete state model [19] to handle word's local weight. The state model uses HMM to represent the probabilistic automation of a word. The automation has two states corresponding to "low" and "high" respectively. It split the text-stream into blocks using a time unit. If a document contains a word w, it is considered

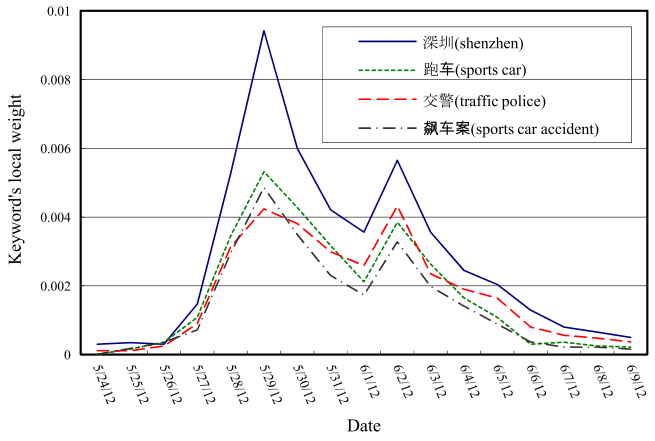


Fig. 2. Examples of word's local weight.

as a document related to w. Each block corresponds to a state, and the related documents are generated with a specific probability.

For example, all the text documents in a period are split into N sequential blocks (b_1, b_2,…, b_N) using a time unit. Given a word, its related document sets corresponding to these blocks are (r_1, r_2,…, r_N), and the corresponding state sequence is (q_1, q_2,…, q_N). In the initial state, the related documents are generated with a probability estimated as follows:

$$p_0 = \sum_{i=1}^{N} \frac{|r_i|}{|b_i|}/N \tag{6}$$

The probability that a block stays in a state follows an exponential distribution. The probability to generate related documents in state k (it is 0 or 1 in this paper) is $p_k = p_0 s^k$, where s is a parameter that denotes the ratio between the numbers of relative documents generated in the "high" state and "low" state respectively. In time unit t, the probability to generate the related document set rt with state k is estimated as follows:

$$prob(k, r_t, b_t) = \left(\frac{|b_t|}{|r_t|}\right) p_k^{|r_t|}(1-p_k)^{|b_t|-|r_t|} \tag{7}$$

We use $\delta(i, j)$ to denote the transformation probability from state i to state j. Therefore, the probability of a state sequence q= (q_1, q_2,…, q_N) can be estimated based on Bayes formula as follows:

$$P(q|r,b) = \frac{prob(q_1, r_1, b_1)\prod_{i=2}^{N}\delta(q_{i-1}, q_i) \times prob(q_i, r_i, b_i)}{Z}$$
$$Z = \sum_{r,d} prob(q_1, r_1, b_1)\prod_{i=2}^{N}\delta(q_{i-1}, q_i) \times prob(q_i, r_i, b_i) \tag{8}$$

where b denotes a set of sequential blocks, i.e., (b_1, b_2,…, b_N), and r denotes the sequence of related document sets corresponding to these blocks. The likelihood function with the logarithm form is represented as follows:

$$-\ln P(q|r,b) = -\sum_{i=1}^{N}\ln(prob(q_i, r_i, b_i)) - \sum_{i=2}^{N}\ln(\delta(q_{i-1}, q_i)) - \ln Z \tag{9}$$

We use the cost function $cost(q)$ to represent $-\ln P(q|r,b)$. Then, the maximization of $P(q|r,b)$ is equal to the minimization of $cost(q)$, which can be solved by the *Viterbi* algorithm.

In microblogging sites, a great volume of micro-blogs are generated in each minute. Thus, it needs an efficient method to detect burst words. However, the model discussed above is not competent in online processing of such a great volume of micro-blogs. For example, the calculation of initial probability p_0 needs the information of all the N blocks. Moreover, all of the words are treated equally in the search of burst states. To address these problems, we proposed an incremental strategy and use words' local weights to detect burst words.

In time unit t, we assume that the sum of all words' local weights is $w^t = \sum_v w^t_{,v}$. Since both $w^t_{,v}$ and w^t are real values and the binomial distribution of Eq. (7) is a discrete distribution, the generation probability of related document set cannot be estimated by Eq. (7) directly. However, when the number of samples n is great enough, the binomial distribution can be approximated by the normal distribution N(np, p(1−p)) according to the law of large numbers. In each time unit, the volume of generated micro-blogs is very great. Thus, the law of large numbers is satisfied. We use the normal distribution function instead of Eq. (7) to estimate the generation probability as follows:

$$prob(k, w^t_{,v}, w^t) = \frac{1}{\sqrt{2\pi\delta^2}} e^{-\left(\frac{x-\mu}{\delta}\right)^2} \tag{10}$$

where $\mu = p_k * w^t$, $\delta^2 = p_k(1\text{-}p_k)$, $x = w^t_{,v}/w^t$.

For the v^{th} word, the probability of generating its related document set in time unit t with the initial state is revised as follows:

$$p_0 = \begin{cases} \frac{1}{|KW^t|} & \text{it appear in the } t^{th} \text{ time unit first} \\ \frac{\sum_1^{t-1}\frac{w^t_{,v}}{w^t}}{t-1}\ (t \neq 1) & \text{else} \end{cases} \tag{11}$$

Then, by substituting Eqs. (10) and (11) into Eq. (9), the cost function can be revised as follows:

$$cost(q_{1,t}) = \min(cost(q_{1,t-1}) + C(q_{i-1}, q_i)) - \ln prob(k, w^t_{,v}, w^t) \tag{12}$$

where $q_{1,t}$is the state sequence ranging from q_0 to q_t, $C(q_{i\text{-}1},q_i)$= $-\ln\delta(q_{i\text{-}1},q_i)$ is the cost of state transformation. Now, the probability p_0 of the initial state is updated dynamically, and the estimation of $cost(q_t)$ is only depended on $cost(q_{t-1})$. Thus, it only need to store the cost value of the $(t-1)^{th}$ time unit to implement the incremental calculation. The complexity is only $O(n)$.

4.3. Burst word clustering

When the burst words are detected, events can be detected by clustering these burst words. Each group of burst words represents an event. To cluster burst words, we construct a word relation graph in which each vertex denotes a burst word and each edge denotes the relation between the two corresponding words. Then, a graph-based clustering method is used to split the word graph into sub-graphs each of which represents a burst event.

Given the burst words BK^t detected in time unit t. We estimate the weight of relation between two words by using the set of micro-blogs containing both of the words as positive evidence of the relation, and the set of micro-blogs containing only one of them as negative evidence against the relation [6,35]. Then the relation weight $rw^t_{i,j}$between two burst words bk^t_i and is estimated as follows:

$$rw^t_{i,j} = \log\frac{n_{i,j}}{n_i + n_j - n_{i,j}} \times \left|\frac{n_{i,j}}{n_i} - \frac{n_j - n_{i,j}}{n_{all} - n_i}\right| \tag{13}$$

where n_i, n_j, and $n_{i,j}$ denote the numbers of micro-blogs (in MC^t) containing bk^t_i, bk^t_j, and both of bk^t_i and bk^t_j respectively, and n_{all} denotes the total number of micro-blogs in time unit t. This equation is asymmetric, and it has two components. The first component estimates the strength of the relation based on their co-occurrence. The second one denotes which node is the lead of the relation. For example, if word bk^t_i appears in most of the micro-blogs and word bk^t_j appear in a few number of micro-blogs, the relation mainly depends on bk^t_j. This is because that word bk^t_i may represent a more abstract semantic concept (e.g., "Apple") and the

relation mainly depends on the co-occurrence with the more specific word bk_i^t(e.g., "Iphone 4 S").

Because of the feature vector sparsity problem of micro-blogs, the co-occurrence in micro-blog documents cannot efficiently measure the semantic relationship between two burst words. Thus, we use the burst words to search other types of web document from various web sites (e.g.,Yahoo and Sina). Then, Eq. (13) is also used to estimate the relation weight $rd_{i,j}^t$in other web documents, and the relation weight is revised as follows:

$$rw_{i,j}^{t'} = \alpha \cdot rw_{i,j}^t + (1-\alpha) \cdot rd_{i,j}^t \tag{14}$$

where $0 \le \alpha \le 1$ is a balance parameter. In the experiment, we download a set of Web pages from Sina and Yahoo to estimate the co-occurrence of burst words. These web pages are searched using the burst words as the query words during January 2012 and June 2012. In total, we crawled about 3.5 million web pages from these web sites. All the web pages are parsed to extract their text content. Then, the co-occurrence of words in this dataset is used to improve the measure of relation weight as shown in Eq. (14).

The burst word relation graph is a directed graph. A normalized method is used to remove these edges whose weights are relatively small. Since each event is defined as a set of semantically correlated words, we exploit the topological structure of the graph to detect event. In order to retrieve the burst events in time unit t, we search for the strongly connected sub-graphs from the graph constructed with the entire set of the burst words BK^t. Each sub-graph represents an event, and all the micro-blogs in MC^t containing the words in the sub-graph are considered to be related to the corresponding event. Particularly, a depth-first search algorithm is applied to find the strongly connected sub-graphs. For example, given a vertex a in a graph, we find the set of vertices A reachable from a through a path. Then, we repeat the process on the same graph with reversed edges in order to find the set of vertices B that can reach a through a path. The strongly connected sub-graph is formed by all the vertices within the intersection between A and B. Fig. 3 shows a part of the burst word relation graph built on the data collected from Sina Weibo. Obviously, it contains two sub-graphs that are strongly connected, and each of the sub-graphs represents an event discussed in Sina Weibo.

5. Prediction of event popularity

In this section, we formulate a linear spread model by assuming that the volume of an event depends on the volumes produced by the infected users and its historical popularity. The volume produced by each user depends on his influence power and his interest in the event.

5.1. Linear spread model of event

As discussed above, the volume of an event at time unit t depend on several factors, such as the influence powers of the infected users, infected users' interest in the event and event's historical popularity. By combining these factors, the volume $V_e(t+1)$of event e in the (t+1)th time unit can be formulated using a linear spread model as follows:

$$V_e(t+1) = \sum_{u \in In(t)} Sim(u,e) * F_u(t-t_u) + \chi_{t+1} H_e(t+1) \tag{15}$$

where $F_u(d)$ denotes user u's influence power d time units after he is infected, $Sim(u,e)$ denotes user u's interest in event e, $H_e(t+1)$ denotes the volume introduced by its historical popularity, and $In(t)$ denotes the set of already infected users and t_u is the time when user u is infected.

In Eq.(15), the volume of an event is a linear function of the weighted influence powers of the infected users and the volume introduced by the historical popularity, in which the weights indicate users' interest in the event. It means that the current volume of an event is a sum of the volumes expected to be produced by the infected users at the current time and the volume introduced by its historical popularity. Obviously, this model does not require the knowledge of the underlying network. Our model is similar to the model proposed in the previous work [40]. In the previous model, all of the users are considered to have the same contribution to event popularity. Moreover, the historical popularity information and event content are not well exploited in the previous model.

The key problem is how to estimate the weight value, i.e., user's interest $Sim(u,e)$, the influence power $F_u(d)$, and the introduced volume $H_e(t+1)$. We use a topic model to mine user's interest in the propagated event. To estimate the influence power $F_u(d)$, many works use a parametric approach to assume the functions of all users follow the same parametric form (e.g., exponential form $a_c e^{-\lambda_c d}$, and power law form $a_c d^{-\gamma_c}$). However, it may be difficult to capture the characteristics of different users in the environment of complex dynamics of event spread. In this study, we use a non-parametric approach to estimate the influence powers of different users, which is similar to the method used in the previous model [40]. The introduced volume $H_e(t+1)$ depends on the historical popularity information. By considering the history volumes as the points on the popularity curve of an event, the Newton interpolation method is used to estimate the introduced volumes in the next time units.

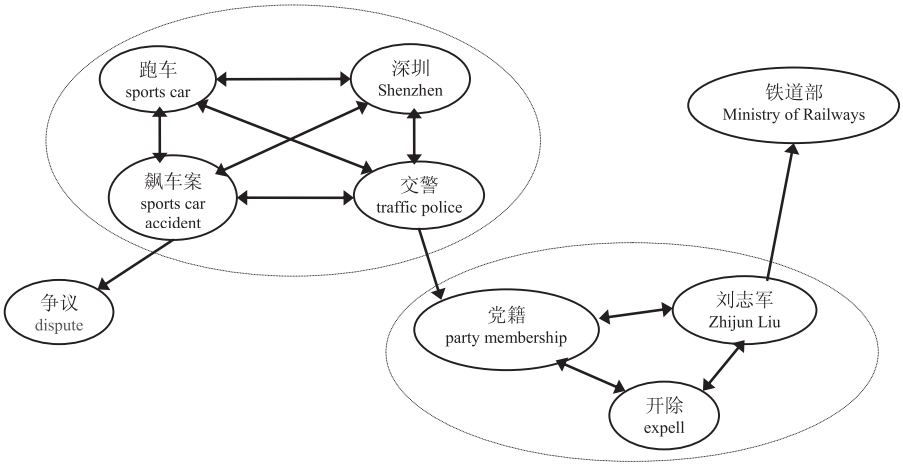


Fig. 3. Part of the burst word relation graph.

5.2. Mining users' interest

Usually, a user's interest is reflected by the micro-blogs that are uploaded by this user in the past. We use a topic model to indentify the topics that the user is interested in, i.e., the LDA model [5] is used to mine user' interest distribution over the latent topics. In the LDA model, each document is represented as a probability distribution over a set of given topics, while each topic is represented as a probability distribution over a set of words. This model has two parameters needed to be inferred, i.e., the document-topic distribution θ and the topic-word distribution ϕ.

However, our goal is to learn user's interest distribution over topics. We aggregate all of the micro-blogs generated by the same individual user into a profile that corresponds to a document in the LDA model. Then, user's interest distribution over topics is represented by the profile's probability distribution over topics. The intuition is that a profile's great probability on a topic indicates a great interest of the corresponding user in this topic. Therefore, each user's interest can be represented by a vector of probabilities distributed over the latent topics as follows:

$$\theta_u = < p_{u,1}, p_{u,2}, \ldots, p_{u,K} > \tag{16}$$

where $p_{u,k}$ represents the profile-specific probability of the k^{th} topic.

Usually, different users have different interest in a given event. The user whose profile's probability distribution over latent topics is more similar with the event's probability distribution over these latent topics should be more interested in this event. For example, if a large part of the micro-blogs produced by a user covers the topic of politics, this user should be more interested in the event related to politics. Thus, each event is also represented by a vector of probabilities distributed over the same latent topics. We aggregate some hot micro-blogs (i.e., the micro-blogs that have the most number of retweets and comments) selected from the event into an event profile. Then, the trained topic model is used to infer the probability distribution over topics of the event profile, such as Eq. (16). Finally, the cosine similarity between the probability distribution vectors of event profile and user profile is used to measure user's interest in the event, i.e., the $Sim(u,e)$ in Eq. (14).

5.3. Estimation of introduced volume

As discussed above, the introduced volume $H_e(t+1)$ is produced by the historical popularity information. We use a popularity curve to describe the development of an event. The x-axis denotes the time line and is equally spaced, and the y-axis denotes event volume. Then, the Newton interpolation is applied on the y values of the former x points to predict the y values in the next points. If the volume $V(i)$ of event e at the i^{th} ($i=0, 1, \ldots, t$) time unit is considered as the value y_i of the curve at point x_i, then the predicted value $V'(t+1)$ at point x_{i+1} is considered as the introduced volume $H_e(t+1)$ at the $(t+1)^{th}$ time unit as follows:

$$H_e(t+1) = V'(t+1) = \sum_{j=a}^{t} \left(\prod_{i=a, i\neq j}^{t} \frac{I_{t+1} - I_i}{I_j - I_i} \right) \cdot V(j) \tag{17}$$

where $0 \le a \le t$ is a index which indicates the starting point from which the volume values is used in the Newton interpolation equation. By setting the volume produced at each time unit as the y value of the curve, Eq. (17) can be used to predict the introduced volume in the next time units. Though the Newton interpolation rule can predict the next y values based on the previous x values, the predicted values may deviate from the real values greatly in the social network environment. For example, when the popularity of an event decreases greatly, the predicted volume is also decrease greatly. However, the event may become popular again in the next time unit if many influential users join in the discussion of the event. Thus, we combine the external factors and the volume introduced by event's historical popularity to predict event's total volume in the future.

5.4. Learning user's influence power

As discussed above, user's influence power is a signature that, once this user adopts an event, how many other users will be infected by this user to adopt the event and then how many new micro-blogs will be produced. To some degree, it is a measure of event volume that a user can produce for this event, and it is changed with time. We use the approach that is similar to the one used in the previous model [40] to infer user influence power $F_u(d)$ and the other parameter χ. Assuming there are U users and data about M different events spread in the network over time, each event can infect any number of users. We use a additional indicator function $R_{u,m}(t)$ to represent the infection of event e_m on user u. $R_{u,m}(t) > 0$ if user u is infected by event e_m at time unit t, and $R_{u,m}(t)=0$ otherwise. Then, the volume $V_m(t)$ of event e_m at time unit t is inferred as follows:

$$V_{e_m}(t+1) = \sum_{i=1}^{U} \sum_{d=0}^{D-1} R_{u_i,e_m}(t-d) \cdot Sim(u_i, e_m) \cdot F_{u_i}(d+1) + \chi_{t+1} H_{e_m}(t+1) \tag{18}$$

where D denotes the amount of time units, meaning that the influence power of a user can be neglected D time units after this user is infected. In the training process, we record T volume values of T sequential time units for each event. As there are M different events, the number of these volume equations denoted by Eq. (18) is $M \cdot T$. To infer these parameters, we can represent these equations in a matrix form:

$$V = \mathbf{R}F + H \tag{19}$$

where V is a volume vector of length $M \cdot T$, $\mathbf{R}$ is a matrix of size $M \cdot T \times U \cdot D$, F is a influence power vector of length $U \cdot D$, H is a vector of introduced volumes weighted by parameter χ. We compose V by taking the volume $V_m(.)$ of each event e_m as a vector of length T indexed by time unit order number t, and then merging the events for $m=1,\ldots, M$. The vectors of F and $H$are built in the same way. $\mathbf{R}$ can be considered as a super matrix which is composed by $M \times U$ matrixes each of which is a $T \times D$ matrix. Each row of the binary indictor matrix $\mathbf{R}$ is a vector of length $U \cdot D$, and it is composed by concatenating U vectors each of which is a vector with D elements as follows:

$$\begin{aligned}
&[R_{u_1,e_m}(t) \cdot Sim(u_1, e_m), \ldots, R_{u_1,e_m}(t-D+1) \cdot Sim(u_1, e_m),\\
&\qquad \vdots\\
&R_{u_i,e_m}(t) \cdot Sim(u_i, e_m), \ldots, R_{u_i,e_m}(t-D+1) \cdot Sim(u_i, e_m),\\
&\qquad \vdots\\
&R_{u_U,e_m}(t) \cdot Sim(u_U, e_m), \ldots, R_{u_U,e_m}(t-D+1) \cdot Sim(u_U, e_m)]
\end{aligned} \tag{20}$$

Now, the parameters can be inferred by solving a matrix equation as Eq. (19). Given the values of the volume vector V and the indicator matrix $\mathbf{R}$, the target is to resolve the values of the influence power vector F and the parameter vector χ. However, it is very difficult to get an exact solution because of the noise and over-determined problems. An approximate value of F can be inferred by minimizing the prediction error measured by the Euclidean distance between the true and the predicted volumes of the events:

$$\begin{aligned}
&\min \; ||V - (\mathbf{R}F + H)||_2^2\\
&subject\ to\ F \ge 0
\end{aligned} \tag{21}$$

This optimization problem is a non-negative least squares problem. It can be solved efficiently even for a large number of

nodes and events. In this paper, the Reflective Newton Method [8] is used to solve this problem.

6. Experiments

To validate the performance of our approach, we conduct a set of experiments on two datasets collected from Twitter and Sina Weibo respectively.

6.1. Dataset

The first dataset is collected from Twitter, which is a stream of about 31 million Twitter posts uploaded between July 2011 and June 2012. Considering the very large volume of users and the impressive amount of generated tweets in Twitter site, we monitor a stream of messages which are generated by a random sample of about 313 thousand users among all public messages. This access level provides a statistically significant input for data mining and research applications. The collected tweets are tokenized into words, and all of the stop-words are filtered. We also filter tweets with no more than three words. Then, about 16 million tweets and about 2.6 million unique words in total are remained after filtering and stemming.

The second dataset is collected from the Chinese microblogging site, i.e., Sina Weibo. In Sina Weibo, 10 hottest topics are recommended in its website every day. We collect a sample of users related to these events, and then download all the micro-blogs uploaded by these users between Jan 2012 and June 2012 using the API provided by Sina Weibo. Then, we collected about 6 million micro-blogs and 119 thousand users, and a set of 148 events that have the greatest volumes are selected for the experiments.

6.2. Evaluation criteria

Since the dataset collected from Sina Weibo is composed of the recommended events which are labeled by the site editors, we use these events as the ground-truth. Moreover, all these micro-blogs are manually labeled with the associated event. Thus, the amount of labeled micro-blogs of each event is considered as the volume of this event.

As for the dataset collected from Twitter, no ground truth is available for events and their volumes. We manually check the events detected by our approach one by one to evaluate the performance of event detection. Finally, a subset of 175 events that are detected by our approach (e.g., Hurricane " Elena " hit United States East Coast in Aug 2011, Iphone 4 s published in Oct 2011, etc.) and have the greatest volumes are selected to evaluate the performance of popularity prediction. The micro-blogs that contain the burst words are considered to associate with the corresponding event, and the amount of these micro-blogs is considered as the ground-true of the volume of this event in the popularity prediction experiments.

Some IR style criteria, i.e., Top-K precision *Precision@K* and macro-precision *MacroPrecision@K*, are used to evaluate the performance of event detection. Assuming we run the burst event detection process in N time units, the criteria are defined as follows:

$$Precision@K(t) = \frac{|TW_t \cap EW_{t,K}|}{|EW_{t,K}|} \tag{22}$$

$$MacroPrecision@K = \frac{\sum_{t=1}^{K} Precision@K(t)}{N} \tag{23}$$

where $EW_{i,K}$ denotes the set of top K events detected by the event detection approach in time unit t, and TW_t denotes the ground-truth of events in time unit t. In the event detection experiments, the length of time unit is set to be 12 h.

The volumes of an event can be viewed as a time series. In the experiments, when an event has been detected, its volume in the following time series is predicted previously. We evaluate the performance of popularity prediction on a time series prediction task, where we observe the volumes of event e up to the time unit t and aim to predict the volume $V_e(t+1)$ at the next time unit. To evaluate the prediction performance, we employ 10-fold cross validation. All the events in the two datasets are divided into 10 folds respectively, and then 9 folds are used to estimate the parameters and the remaining fold is used to evaluate. The metric of relative error is used to evaluate the performance of prediction as follows:

$$Err = \frac{\sqrt{\sum_{m,t}(V_{e_m}(t+1) - TV_{e_m}(t+1))^2}}{\sqrt{\sum_{m,t} TV_{e_m}(t+1)^2}} \tag{24}$$

where $TV_e(t+1)$ is the true volume of event e at the $(t+1)^{th}$ time unit.

6.3. Baseline methods

We compare our approach with other approaches in the tasks of event detection and popularity prediction respectively. As for the task of event detection, our approach is compared with other two event detection approaches ([43,42]. The first one named *EDCoW* [43] uses wavelet analysis on the frequency-based signal of words to filter away trivial words, and then the remaining words are clustered to form events. The second one named *MSBI* [42] detects burst indications on multiple sources (i.e., word frequency, and time-aware post coverage and user attractiveness), and then it combines multiple burst indications to detect burst events.

As for the task of popularity prediction, other two prediction approaches are compared with our approach [16,39]. We name them *HTall* [16] and *BassPre* [39] respectively. To implement these approaches, we build a set of features for each event (e.g., number of the burst words, average number of followers of the infected users, retweets ratio, volumes of the previous four time units, etc). Then the regression model [39] and Bass model [16] are used to predict event volume respectively.

6.4. Experiments of event detection

In the process of burst words detection, there are two parameters needed to be set, i.e., the ratio s between the numbers of relative documents generated with the "high" state and "low" state respectively, and the cost $C(q_{i-1},q_i)$ of state transformation. To facilitate further evaluation, we first analyze the effect of different parameter values in the performance of event detection. Then, the parameter values corresponding to the best performance of different datasets are selected for performance comparison. The macro precision values of top 10 events are shown in Figs. 4 and 5 with s varied from 3 to 15 and $C(q_{i-1},q_i)$ varied from 100 to 500 respectively. These figures show that the performance curves of different datasets are similar though their peaks are different. The performance in twitter dataset is the best when s is 9 and $C(q_{i-1}, q_i)$ is 400, while the performance is the best when s is 6 and $C(q_{i-1}, q_i)$ is 200 in Sina Weibo dataset.

Based on the aforementioned parameters evaluation, we compare the performance of burst event detection among different approaches. Figs. 6 and 7 show the macro-precision values of different approaches with K varied from 5 to 25, where *TSUB* is the name of our approach. As shown in these figures, our approach consistently outperforms both of *EDCoW* and *MSBI*. There are several reasons. First, our approach uses the directed relation

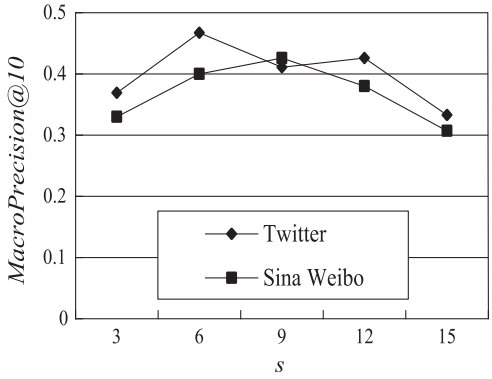


Fig. 4. Performances with different values of s.

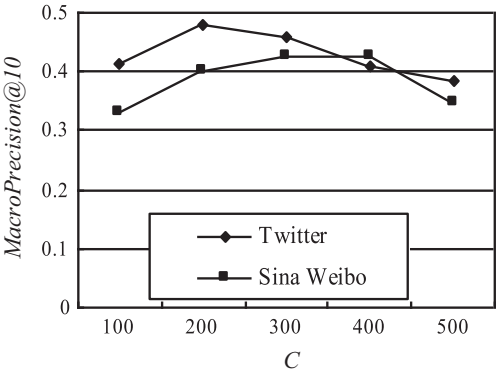


Fig. 5. Performances with different values of C.

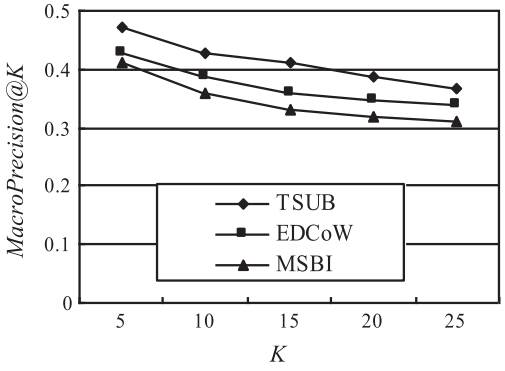


Fig.6. Comparision of event detection in Twitter dataset.

between words. It decreases the noisy effect of hot words which have high frequencies in many time units. Second, we combine users' influence in the detection of burst word, which can highlight the words that do not appear as burst words obviously but are very important for event development. Though MSBI also utilizes user's social relation, it considers user's social relation independently from the burst indication of word. Instead, it uses a mixture model to combine the burst detection results of multiple sources. All of the performances degrade with the increasing of K. This is because that many noisy messages exist in microblogging sites. Usually, these messages are related to mundanely daily routine of users. Among these messages, many of the hot words are detected as burst words, and thus the performance degrade when more noisy data is included.

Table 2 gives some examples of events each of which is represented by a set of burst words detected by our approach. The second column gives the examples of burst word clusters. In reality, the burst words of each event appear frequently when the

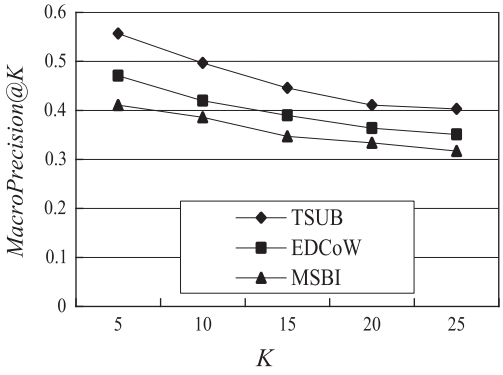


Fig. 7. Comparision of event detection in Sina Weibo dataset.

corresponding event happens, and they also have strong relation with each other. We add the description of each event in the third column. It shows that our approach can detect some event exactly in time.

6.5. Experiments of popularity prediction

Though many related works have been done for burst event detection, the definition of burst event in these works only focuses on the novelty of event. Our approach extends the definition of burst event to incorporate temporal aspect of timeliness. In other words, when an event is detected we also predict how hot the event will be in the future. Therefore, in this set of experiments, we evaluate the popularity prediction performance of our approach.

The complexity of the training of our prediction model is $O(\delta \cdot M \cdot T \cdot U \cdot D)$, where δ is the number of iterations in the process of inferring parameters. The complexity is mainly depended on the value of user amount U because it may be millions or greater. To reduce the complexity, we group the users who have the similar interests into a super node. In particularly, we use the k-means clustering method to cluster users based on the cosine similarity of users' vector of probability distribution over latent topics. Then each cluster of users is used as a super node to replace the original user nodes in the social network. Finally, we build a set of 500 super nodes for these datasets. Moreover, the value of $R_{c,m}(t)$ in Eq. (18) is set with the ratio of infected users to the total users in a super node.

In our prediction model, there are other parameters needed to be set, i.e., the length of time unit l and the tenure D of user's influence. We set different values to analysis their effect in the performance of popularity prediction. Since the lifetimes of many events are very short, the length of the volume time series is set to be no more than 10 days, i.e., $T=\mathbf{min}(20, (10\times 24)/|time\ unit|)$. For each event, we set the starting point ($t=0$) of $V_m(t)$ to be the first time unit that the event is detected.

First, we evaluate the effect of parameter D. Fig. 8 shows the relative error of our approach with the value of D varied from 1 to 7 time units. From this figure, it can be concluded that our approach achieves the best performance on the two datasets when the length of D is 4. The performance is improved with the length of D increasing from 1 to 4, while the performance falls down with the length of D increasing from 4 to 7. There are several reasons. First, when D is too short, each user's influence power is estimated based on a short period, which decreases his influence power and hence affects the prediction. Second, when D is too long, the out of date information may be noisy to the estimation of user's influence power.

Then, we analyze the effect of time unit length l and compare the performance of different approaches. Figs. 9 and 10 show the relative errors of different approaches used in Twitter and Sina

Table 2
Examples of events with the burst words

Source	Burst words of events	Event description
twitter	Hurricane, Aileen	Hurricane " Elena " hit United States East Coast in Aug 2011.
twitter	Iphone 4 S, Apple, Steve Jobs	Iphone 4 s was published in Oct 2011.
Sina Weibo	深圳(Shenzhen), 跑车(sports car), 交警(traffic police), 飙车案(sports car accident)	Shenzhen sports car accident caused 3 people death in May 2012
Sina Weibo	刘志军 (Zhijun Liu), 开除(expel), 党籍(party membership)	The former railways minister Zhijun Liu was expelled from the Communist Party of China in May 2012

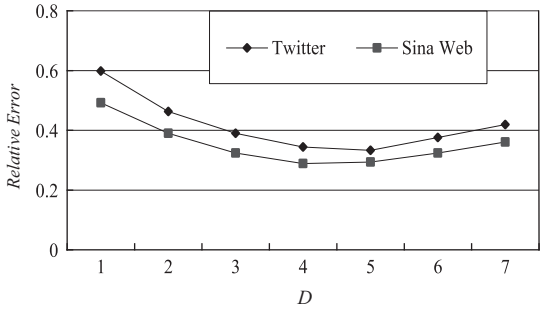


Fig. 8. Effect of parameter *D*.

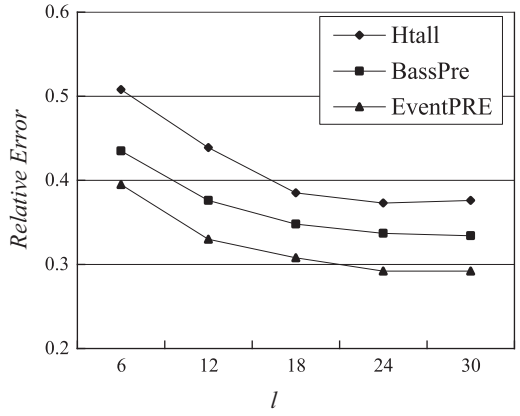


Fig. 9. Comparision of prediction on Twitter dataset.

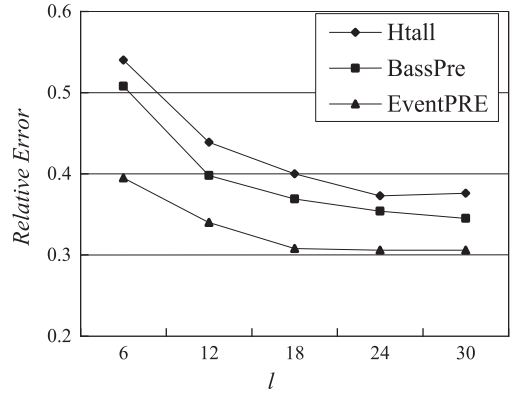


Fig. 10. Comparision of prediction on Sina Weibo dataset.

Weibo datasets respectively. Results are presented for five different lengths of time unit. It shows that our approach outperforms other two approaches consistently. This is because that our approach uses users' interest information to distinguish different users' contributions in the spread of event, while other approaches assume that all the users have the same contribution. Moreover, event's popularity information in previous time units is also used in our approach. It also shows that all the approaches achieve better prediction when the time unit is longer. This can be attributed to the deceased standard deviation of event volumes.

There are several interesting observations that can be found in the experiments. First, *HTall* is a linear regression model in which all of the users are considered to have the same influence power and interest in each of the events. *BassPre* also use a similar strategy. However, in reality, different users have different influence powers and interests in each event. Our approach receives a great benefit by distinguishing user's influence power and interest in the propagated event. Second, when the spread model is trained for different categories of events, we find that our approach has better results when predict the volumes of some categories of events, e.g., local disasters, accident events, etc. From the observation of the volume curves of these events, it can be found that most of the events of these categories follow the same evolution pattern. For example, some users uploaded a small number of micro-blogs discussing an event in the beginning. Then, many users follow these users and generate a great number of micro-blogs in a relatively short time. After it reaches the peak, the spread of this event begins to decrease relatively slowly. Thus, the parameters (i.e. $F_u(d)$ and χ in Eq. (18)) trained by these events give a great benefit to predict the popularities of events with the same evolution pattern. However, the prediction of some other categories of event has worse performance, e.g., entertainment event, politics event, etc. The reason may be that different events of these categories have different evolution patterns, and some events have more than one peak. Fig. 11 shows some examples of comparison between the true volumes and predicted volumes of accident event and entertainment event. In these figures, each point represents a separate prediction based on the previous *D* points.

7. Conclusion

With the fast developing of microblogging, a great volume of micro-blogs is generated in each minute. Thus, it is very difficult for users to know what are happing now and how popular the event will be. In this study, we present an approach to detect real-time events from microblogging text streams and then predict the popularity of the detected event in online communities. Specifically, our contributions are concluded as follows: First, we not only detect burst event that is novel, but also to predict how hot the event will be in the near future. We combine term's occurrence information and users' social relation information to detect burst words by revising the two-state model to handle real value. Second, we proposed a spread model based on the analysis of both event content and users' profile. The major advantage of our model is that it distinguishes users' contributions according to user's influence power and interest in the predicted event, which is different from other approaches that use the same parametric form for all events and users. Moreover, the historical popularity information of each event is also used to model its own effect on

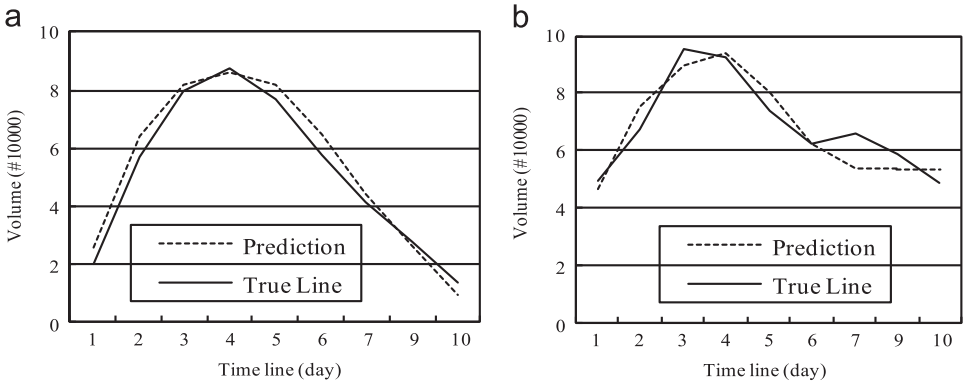


Fig. 11. Examples of prediction, (a) A accident event and (b) A entertainment event.

the future popularity. Third, we further evaluate our approach in two real-life datasets, and experiment results indicate the efficiency of our approach.

The tasks of Event detection and popularity prediction are very interesting from both commercial and sociological perspectives. Although the approach presented in this paper is independent of the language, but it might be affected by the culture and user behaviors of different regions. Thus, in the future work, we would like to further investigate the effect of users' behavior patterns with different cultures on the production and spread of event.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 60973105, 90718017, 61170189, and 61202239), the Fundamental Research Funds for the Central Universities (YWF-13-T-RSC-072, YWF-14-JSJXY-16), the Research Fund for the Doctoral Program of Higher Education (No. 20111102130003), and the Fund of the State Key Laboratory of Software Development Environment (No. SKLSDE-2013ZX-19).

References

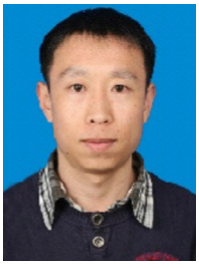
[1] H. Abdelhaq, C. Sengstock, M. Gertz EvenTweet: Online Localized Event Detection from Twitter. in: Proceedings of the VLDB Endowment, vol. 6, 12, 2013.
[2] P. Achananuparp, E. Lim, J. Jiang, T.-A. Hoang Who is retweeting the tweeters? Modeling, Originating, and Promoting Behaviors in the Twitter Network. ACM Transactions on Management Information Systems, vol. 3, 3, Article 13, October 2012.
[3] Agarwal P., Vaithiyanathan R., Sharma S., Gautam Shroff. Catching the long-tail: Extracting local news events from twitter. in: Proceedings of the 6th International Conference on Weblogs and social Media (ICWSM'12), 2012, pp. 379–382.
[4] L. Backstrom, D.P. Huttenlocher, J.M. Kleinberg, X. Lan Group formation in large social networks: membership, growth, and evolution. in: Proceeding of the 12th ACM Conference on Knowledge Discovery and Data Mining (SIGKDD'06), ACM, New York, 2006, pp. 44–54.
[5] D.M. Blei, A.Y. Ng, M. I Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[6] M. Cataldi, U. Torino, L.D. Caro, C. Schifanella Emerging topic detection on Twitter based on temporal and social terms evaluation, in: Proceeding of 10th International Workshop on Multimedia Data Mining (MDMKDD'10), 2010, pp. 1–10.
[7] L. Chen, A. Roy Event detection from flickr data through wavelet-based spatial analysis, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09), ACM, New York, 2009, pp. 523–532.
[8] T.F. Coleman, Y. Li, A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables, SIAM J. Optim. 6 (4) (1996) 1040–1058.
[9] C. Danescu-Niculescu-Mizil, M. Gamon, S. Dumais Mark my words!: linguistic style accommodation in social media, in: Proceeding of the 20th International Conference on World Wide Web (WWW'11), ACM, New York, 2011, pp. 745–754.
[10] P. Dodds, D. Watts, A generalized model of social and biological contagion J. Theor. Biol. 232 (4) (2005) 587–604.
[11] Fung G.P.C., Yu J.X., Yu P.S., Lu H.. Parameter free bursty events detection in text streams, in: Proceeding of the 31st International Conference on Very large data bases (VLDB'05), ACM, New York, NY, 2005, pp. 181–192.
[12] Ganti R., Srivatsa M., Liu H., and Abdelzaher. T.. Spatio-Temporal Spread of Events in Social Networks: A Gas Shortage Case Study. In Proceeding of 2013 IEEE Military Communications Conference.
[13] Gruhl D., Guha R.V., Liben-Nowell D., and Tomkins A.. Information diffusion through blogspace. In Proceeding of the 13th international conference on World Wide Web (WWW'04), ACM, New York, 43–52.
[14] Hartline J., Mirrokni V.S., Sundararajan M.. Optimal Marketing Strategies over Social Networks. In Proceeding of the 17th international conference on World Wide Web (WWW'08), ACM, New York, pp. 189–198.
[15] He Q., Chang K., and Lim E.-P. Analyzing feature trajectories for event detection, in: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'07), ACM, New York, 2007, pp. 207–214.
[16] X. Jin, A. Gallagher, L. Cao, J. Luo, J. Han The wisdom of social multimedia: using flickr for prediction and forecast, in: Proceeding of the 18th ACM Multimedia International Conference (MM'10), ACM, New York, NY, 2010, pp. 1235–1244.
[17] M. Kimura, K. Saito, R. Nakano, H. Motoda, Extracting influential nodes on a social network for information diffusion, Data Min. Knowl. Discov 20 (1) (2010) 70–97.
[18] Kimura M., Saito K., and Motoda H. Minimizing the spread of contamination by blocking links in a network, in: Proceeding of the 23th National Conference on Artificial Intelligence (AAAI'08), AAAI Press, 2008, pp. 1175–1180.
[19] Kleinberg J.. Bursty and hierarchical structure in streams, in: Proceeding of the 8th ACM Conference on Knowledge Discovery and Data Mining (SIGKDD'02), ACM, New York, NY, 2002, pp.91–101.
[20] E. Kwan, P.-L. Hsu, J.-He Liang, Y.-S. Chen Event identification for social streams using keyword-based evolving graph sequences, in: Proceeding of 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2013.
[21] Lee C., Kwak H., Park H., Moon S.B. Finding influentials based on the temporal order of information adoption in twitter, in: Proceeding of the 19th International Conference on World Wide Web (WWW'10), ACM, New York, pp. 1137–1138.
[22] J. Leskovec, L.A. Adamic, B.A. Huberman, The dynamics of viral marketing, ACM Trans. Web (TWEB) 1 (1) (2007) (Article 5).
[23] J. Leskovec, M. McGlohon, C. Faloutsos, N.S. Glance, M. Hurst Patterns of cascading behavior in large blog graphs, in: Proceeding of the 17th SIAM Conference of Data Mining (SDM'07), Minneapolis, Minnesota, 2007, pp. 551–556.
[24] Leskovec J., Backstrom L., Kleinberg J.. Meme-tracking and the dynamics of the news cycle, in: Proceeding of the 15th ACM Conference on Knowledge Discovery and Data Mining (SIGKDD'09), ACM, New York, 2009, pp. 497–506.
[25] Li C., Sun A., and Datta A. Twevent: segment-based event detection from tweets, in: Proceeding of the 21st ACM Conference on Information and Knowledge Management (CIKM'12), ACM, New York, 2012, pp. 155–164.
[26] C.X. Lin, Q. Mei, J. Han, Y. Jiang, M. Danilevsky The Joint Inference of Topic Diffusion and Evolution in Social Communities, in: Proceeding of the 2010 IEEE International Conference on Data Mining (ICDM'11), 2011, pp. 378–387.
[27] Z. Ma, A. Sun, G. Cong, On predicting the popularity of newly emerging hashtags in twitter, J. Am. Soc. Inf. Sci. Technol. 64 (7) (2013) 1399–1410.
[28] S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Trans. Pattern Anal. Mach. Intell. 11 (7) (1989) 674–693.
[29] N. Meade, T. Islam, Modelling and forecasting the diffusion of innovation – A 25-year review, Int. J. Forecast. 22 (3) (2006) 519–545.
[30] C.-C. Pan, P. Mitra Event detection with spatial latent Dirichlet allocation, in: Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries (JCDL'11), ACM New York, NY, 2011, pp. 349–358.

[31] N. Pathak, C. Delong, A. Banerjee, K. Erickson Social topics models for community extraction. in: Proceeding of the 2nd SNA-KDD Workshop (SNA-KDD'08), 2008.
[32] S. Petrović, M. Osborne, V. Lavrenko Streaming first story detection with application to twitter, in: Proceeding of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'10), Stroudsburg, PA, 2010, pp. 181–189.
[33] S. Petrovic, M. Osborne, V. Lavrenko RT to win! predicting message propagation in twitter, in: Proceeding of the 5th International AAAI Conference on Weblogs and Social Media ICWSM, 2011.
[34] D. Ramage, S. Dumais, D. Liebling Characterizing microblogs with topic models, in: Proceedings of AAAI on Weblogs and Social Media, 2010, pp. 130–137.
[35] I. Ruthven, M. Lalmas, A survey on the use of relevance feedback for information access systems, Knowl. Eng. Rev. 18 (2) (2003) 95–145.
[36] K. Saito, M. Kimura, K. Ohara, H. Motoda Selecting information diffusion models over social networks for behavioral analysis, in: Proceeding of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III (ECML-PKDD'10), 2010, pp. 180–195.
[37] T. Sakaki, M. Okazaki, Y. Matsuo Earthquake shakes twitter users: real-time event detection by social sensors, in: Proceeding of the 19th International Conference on World Wide Web (WWW'10), ACM, New York, 2010, pp. 851–860.
[38] G. Szabo, B.A. Huberman, Predicting the popularity of online content, Commun. ACM 53 (8) (2010) 80–88.
[39] Tsur O., and Rappoport A.. What's in a Hashtag? Content based Prediction of the Spread of Ideas in Microblogging Communities, in: Proceeding of the 5th ACM international conference on Web search and data mining (WSDM'12), ACM New York, NY, pp. 643–652.
[40] Yang J., Leskovec J.. Modeling information diffusion in implicit networks. in: Proceeding of IEEE Conference of Data Mining (ICDM), 2010, pp. 599–608.
[41] Y. Yang, T. Pierce, J. Carbonell A study of retrospective and on-line event detection, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98), ACM, New York, 1998, pp. 28–36.
[42] J. Yao, B. Cui, Y. Huang, X. Jin Temporal and Social Context based Burst Detection from Folksonomies, in: Proceeding of the 25th National Conference on Artificial Intelligence (AAAI'10), AAAI Press, 2010, pp. 1474–1479.
[43] J. Weng, B.-S. Lee Event detection in twitter, in: Proceeding of the 5th International AAAI Conference on Weblogs and Social Media, 2011, pp. 401–408.
[44] H. Zhang, C. Lee Giles, H.C. Foley, J. Yen Probabilistic community discovery using hierarchical latent gaussian mixture model, in: Proceeding of the 22nd National Conference on Artificial Intelligence (AAAI'07), AAAI Press, 2007, pp. 663–668.

Xiaoming Zhang was born in Hunan, China, on December 7, 1980. He received the B.Sc. degree, and the M.Sc. degrees in computer science and technology from the National University of Defence Technology, China, in 2003, 2007 respectively. He received his Ph.D degrees in computer science from Beihang University, in 2012. He is currently working at the school of computer, Beihang University, and he has been the lecturer since 2012. His major interests are text mining, image tagging, and TDT.

Xiaoming Chen was born in Hunan, China, on November 30, 1980. He received the M.Sc. degrees in computer science and technology from the National University of Defence Technology, China, in 2006. He is currently working his Ph.D degrees at the school of computer, Beihang University. His major interests are text mining, information retrieval, and CQA.

Yan Chen received her B.Sc. and M.Sc. degrees in computer science from the Central South University and Hohai University, China in 2007 and 2010. She is currently a Ph.D candidate in Beihang University. Her research interests include social media analysis and information retrieval.



Senzhang Wang was born in Yantai, China, on January 23, 1986. He received the M.Sc. degree in Southeast University, Nanjing, China in 2009. He currently is a third-year Ph.D student in the School of Computer Science and Engineering at Beihang University, Beijing, China. His main research focus is on data mining and social network analysis.

Zhoujun Li received his M.Sc. and Ph.D degrees in computer science from the National University of Defence Technology, China, in 1984 and 1999, respectively. He is currently working at the school of computer, Beihang University, and he has been the professor since 2001. His research interests include the data mining, information retrieval, and database.