Knowl Inf Syst
DOI 10.1007/s10115-015-0899-3

REGULAR PAPER

CPB: a classification-based approach for burst time prediction in cascades

Senzhang Wang[1] · Zhao Yan[1] · Xia Hu[2] · Philip S. Yu[3,4] · Zhoujun Li[1,6] · Biao Wang[5]

Received: 26 November 2014 / Revised: 25 August 2015 / Accepted: 26 October 2015

Abstract Studying the bursty nature of cascades in social media is practically important in many real applications such as product sales prediction, disaster relief, and stock market prediction. Although both the cascade size prediction and the burst patterns of the cascades have been extensively studied, how to predict when a burst will come remains an open problem. It is challenging for traditional time-series-based models such as regression models to address this task directly. Firstly, times-series-based prediction models focus on predicting the future values based on previously observed ones. It is hard to apply them to predict the time of a bursts with the "quick rise-and-fall" pattern. Secondly, besides the cascade popularity, a lot of other side information like user profile and social relation are available in social media.

✉ Zhoujun Li
lizj@buaa.edu.cn

Senzhang Wang
szwang@buaa.edu.cn

Zhao Yan
yanzhao@buaa.edu.cn

Xia Hu
hu@cse.tamu.edu

Philip S. Yu
psyu@uic.edu

Biao Wang
wangbiao120@sina.com

1 State Key Laboratory of Software Development Environment, Beihang University, Beijing, China

2 Department of Computer Science and Engineering, Texas A&M University, College Station, TX, USA

3 Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA

4 Institute for Data Science, Tsinghua University, Beijing, China

5 Information and Technology Department, University of International Relations, Beijing, China

6 School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Published online: 09 December 2015

Although the potential utility of such information can be high, it is also hard for time-series-based models to capture and integrate these rich information with diverse formats seamlessly. This paper proposes a classification-based approach for burst time prediction by exploiting rich knowledge in information diffusion. Particularly, we first propose a time-window-based transformation to predict in which time window the burst will appear. By dividing the time spans of all the cascades into the same number of time windows K, the cascades with diverse time spans can thus be handled uniformly. To exploit the rich and heterogenous information in social media, we next propose a scale-independent feature extraction framework to model the heterogenous knowledge in a scale-independent manner. Systematical evaluations are conducted on the Sina Weibo reposting dataset and MemeTracker dataset. Besides the superior performance of the proposed approach, we also observe that: (1) surprisingly, social/structure knowledge is more indicative of the bursts than the cascade popularity information, especially for the bursts occurring in a farther future. (2) Larger cascades are harder to predict as the spreading process of the cascades with higher popularity is usually more diverse and fluctuant. (3) The proposed approach is robust in the sense that the result is not much sensitive to the popularity of the training cascades.

Keywords Information diffusion · Cascade prediction · Burst · Sina Weibo

1 Introduction

Burst, defined as "a brief period of intensive activity followed by long period of nothingness" [25], is a common phenomenon in human activities. The bursty nature of human behavior is observed and studied extensively in many domains, such as electronic communication [24], library visiting [25], stock trading [21], and web browsing [20]. With the growing popularity of social networks, a large body of research has focused on investigating users' reposting or resharing behavior in social media [19,26–28,33,43]. An important finding of these works, which is consistent with the human behavior discovered in many other domains, is that the spreading process of the cascades formed by users successively reposting contents in social media also presents the bursty property [26,31]. For instance, Myers and Leskovec have found that the dynamics of information diffusion in Twitter can be characterized by "steady rates of changes, interrupted by sudden bursts" [31].

With the bursty nature of the cascades and the challenge of information overload in social media, an interesting problem arises: *Can we predict the burst time of the cascade with the observed partial data in its early stage of spreading?* Predicting the burst time of cascades is of outstanding interest for many real applications in various domains, such as product sales prediction [9], disaster relief [15], and stock market prediction [22]. Yahoo! Finance reported that, Didier Sornette, a former physicist has developed a statistical model with the help of social media data to predict when a financial bubble will burst [22]. As said in the report: " the Sornette model is now predicting a stock market crash as early as next year."

Existing related works on cascades prediction mainly focus on predicting their future volume, ranging from the future popularity prediction [27,28] to the aggregate size prediction [30,33]. Recently, some efforts have also been devoted to modeling the burst patterns of the cascades [18,19,26]. These works mainly focused on studying the patterns of the bursts and using these patterns to cluster the cascades. Another related area is burst detection [5,16,21]. Burst detection focuses on detecting instead of predicting the burst. Although both cascades

and bursts have individually been studied from several different aspects, how to predict the burst time of the cascades with rather dynamic and stochastic properties, while rich social-related knowledge still remains an open problem.

Predicting the burst time of the cascades is a non-trivial task due to the following two major challenges. Firstly, existing times-series-based approaches cannot be directly applied to predict the time of burst due to its "quick rise-and-fall" pattern. For example, traditionally, regression is widely used for predicting and forecasting [7,17] by learning relationships among features based on historical data. With the "quick rise-and-fall" property of bursts [26], the correlation between historical data and future data becomes difficult to be captured by regression-based methods. Meanwhile, a remarkable difference between the social media data and the traditional time series data is that a lot of other side information are available. The rich and heterogeneous social information like user profile and social relation may be potentially helpful [33]. However, it is challenging for traditional time series models to capture and utilize these rich information with heterogeneous formats seamlessly. This motivates us to study: what knowledge are helpful for our task and how to model them in a unified way?

The second challenge lies in the extremely skew distribution of the cascade size and their significantly distinct life spans. A widespread property of cascades is that large cascades are rare [32,33]. The skew distribution of the cascade size suggests that rather than developing a model that can only accurately predict cascades of certain sizes, a more robust prediction model is needed. More specifically, can the prediction model accurately predict the burst time of the large cascades while the training cascades are mostly small? In addition, the vast difference in magnitude and time span of the cascades makes extracting comparable features difficult, and results in building predictive models challenging. For example, given two cascades with the time spans of 10 days and 10 h respectively, it is more meaningful to predict the formal one on the daily basis and the latter one on the hourly basis. Therefore, instead of simply modeling all the cascades in the original time scales without distinguishing their diverse time spans, a more general and time-independent model is necessary for the task we study.

In this paper, we take the first step toward understanding the burst in cascades from the time dimension. Specifically, we propose to formulate the burst time prediction task as a classification problem by time-window-based transformation. Time-window-based transformation first divides the observed time series of all the cascades into the same number of time windows K. Instead of predicting the exact occurring time of the burst, we predict in which time window the burst will appear. Since we conduct the prediction in the time window granularity, cascades with diverse time spans can be handled in a unified way. Motivated by previous studies on utilizing social theories to analyze and predict information diffusion in social media [1–4,44], we explore rich social knowledge available during cascades spreading such as knowledge on user profile and social relation to help this task. To utilize rich knowledge in a unified way and eliminate the difference of cascades in magnitude and time span, we model them in a scale-independent manner by deriving scale-independent features. To summarize, the advantages of the proposed approach are as follows:

- *Flexibility* Time-window-based transformation enables us to predict cascades of different time spans and popularities with different time granularities.
- *Inclusiveness* Classification-based framework can handle various features extracted from different types of knowledge in a scale-independent manner.
- *Robustness* The prediction result is not much sensitive to the popularity of the training cascades. For those popular but rare cascades that people may concern more, the prediction result is still desirable even if most training cascades are small.

We evaluate the proposed approach on three real datasets: the Sina Weibo reposting dataset that contains 300,000 posts, the MemeTracker phrase cluster dataset that contains around 60,000 phrase clusters spreading in the web, and the MemeTracker raw phrases datasets that contains more than 80,000 raw phrases. The results show the effectiveness of the proposed approach in accurately predicting the burst time of the posts and memos. Besides the strong performance, we also have some interesting observations which may direct us to have a deeper understanding of information diffusion in social media. (1) Traditional time series models are not very effective to this task. This means that simply considering the popularity of cascades in different time intervals as time series data is not enough to predict the burst time of cascades. (2) Social/Structure knowledge is a good indicator of the evolving cascades. Our experimental results demonstrate that both user profile and social relation knowledge perform better than the pure cascade popularity information. (3) Larger cascade is harder to predict. The prediction accuracy on the cascades with more than 2000 reposts is roughly 10 % lower than those with only 100 reposts on the Sina Weibo dataset. This finding implies the spreading of popular cascades is more complex and fluctuant.

Compared with our previous work on burst time perdition [45], the new contributions and observations of this paper are as follows. (1) We conduct data analysis on the Sina Weibo dataset (Sect. 2). By clustering the cascades into six types of diffusion patterns, we discover that each diffusion pattern of Sina Weibo reposting data shows a significant burst. In addition, the diversity analysis shows that smaller cascades are more similar to each other, while the diffusion processes of larger cascades are more diverse and complicated. It implies that larger cascades may be harder to predict. (2) Besides the Sina Weibo dataset, two new datasets are used to evaluate the proposed model (Sect. 6). The desirable results on these datasets demonstrate that the proposed model can be widely used to predict the time of bursts of various social media and web data. 3) More experiments and analysis are added in the experiment part. The performance of more classification algorithms on this problem is studied (Sect. 6.2). We further study whether some algorithms perform significantly better than others on our task by adding the classification statistical significance test. The study on the performance versus cascade size (Sect. 6.5) verifies that the prediction performance decreases with the increase in cascade size. The robustness analysis shows the robustness of the proposed CPB model: the prediction performance is promising on large cascades even if most training samples are small (Sect. 6.7).

The remainder of the paper is organized as follows: in Sect. 2, we formally define the burst time prediction problem. Section 3 describes and analyzes the dataset. Section 4 introduces how to formulate the burst prediction task as a classification problem and elaborates how to extract the rich scale-independent features. In Sect. 5, we evaluate the proposed approach and report the results. We discuss related work in Sect. 6. Section 7 concludes this research with directions for the future work.

2 Related work

We present the related work in three areas: cascade prediction, burst detection, and time series analysis.

2.1 Cascade prediction

Many efforts have been devoted to this area recently. Most work relevant to ours focus on predicting the future volume of the cascades [6,27,28,30,33,34]. The future volume

prediction can be further categorized into the future popularity prediction [6,27,28] and the aggregate size prediction [30,33]. The future popularity prediction mainly aims to predict whether a new cascade will become a trending topic in the future. Hong et al. formulated the messages popularity prediction task into a classification problem [27]. Cui et al. [6] proposed a data-driven approach to predict which cascades will become outbreaks in the future by selecting important nodes as sensors. The aggregate size prediction task focused on predicting the future size of the cascades. Cheng et al. [33] proposed a framework for addressing the cascade growth prediction problem. More specifically, given a cascade with the size k, they aim to predict whether the cascade will double its size and reach the size $2k$. However, most previous work focused on the volume prediction. Predicting the time of a particular event like a burst is still an open problem.

2.2 Burst detection

Burst detection is a well-studied problem. The pioneering work on this topic is conducted by Kleinberg [16]. Other represent work include the efficient elastic algorithm proposed by Zhu and Shasha [21] and the scalable near real-time algorithm proposed by Parikh et al. [5]. Recently, the burst phenomenon in social media has attracted a lot of interest [18,19,26,31]. Previous work focused on studying the burst shapes and clustering the "rise-and-fall" patterns of the cascades. Yang and Leskovec [19] found that there are six main temporal shapes of time series in Twitter. Matsubara et al. [26] studied the "rise-and-fall" patterns of the cascades and discovered the burst in real data show an exponential rise and power law fall pattern. Instead of detecting bursts or mining the burst patterns, this paper focuses on a different aspect: how to predict when a burst will occur based on the early stage data of a new cascade. Therefore, the above-mentioned approaches cannot applied to our task directly.

2.3 Time series analysis

Time series analysis is an old research topic and has been extensively studied. Time-series-based prediction approaches aim to predict future values based on previously observed ones, such as Auto-Regression (AR) [7], the moving average (MA) models [36], and their variants [17]. These models are all linear methods and depend linearly on previous data points [35, 42]. For the burst time prediction task, with the "quick rise-and-fall" property, it is hard to directly apply such linear models to predict burst by simply considering previous observed values. Besides linear models, some nonlinear models are also proposed for forecasting [37,38]. Nonlinear methods for forecasting are usually hard to interpret, and these methods are not specifically focused on predicting burst either. Meanwhile, a lot of other information such as user profile and social relation are available in social media. Pure time-series-based prediction models are traditionally hard to integrate various rich information in information diffusion.

3 Problem statement

In this section, we start with some definitions to help us state the studied problem and then formally define the burst time prediction problem.

First, we introduce how we define "burst time" of a cascade. It is hard to exactly define at what time a burst begins or in which time interval a burst exists. Alternatively, we consider the time of the *global spike* of the cascade defined as follows as its burst time we need to predict.

Definition 3.1 *Global spike* Suppose (1) the time span T^c of the cascade c can be equally divided into K time windows, that is $T^c = \{(n_1^c, 1), (n_2^c, 2), \ldots (n_K^c, K)\}$, where $n_j^c (j = 1, 2 \ldots, K)$ is the number of reposts in the jth time window, and (2) the number of reposts n_k^c is a function of the time window k: $n_k^c = f_c(k)$. The global spike of c is the $f_c(k_{\max})$ that satisfies $\forall 1 \leq k \leq K$, $f_c(k_{\max}) \geq f_c(k)$, and $k_{\max}$ is the time window of the global spike.

The time series of the cascades may also have some local peaks where the values are larger than those in the neighborhood time windows. We define such values as the *local spikes* of the cascade.

Definition 3.2 *Local spike* Given the time window related function $n_k^c = f_c(k)$ of cascade c and the divided time windows $T^c = \{(n_1^c, 1), (n_2^c, 2), \ldots (n_K^c, K)\}$ in Definition 1, $f_c(k_{l\max})$ is a local spike of cascade c if the following condition is satisfied: $\forall -s \leq i \leq +s$, $f_c(k_{l\max}) \geq f_c(k_{l\max} + i)$, where s is a predefined threshold.

Previous studies discovered that most cascades in social media usually present one notable spike with several local spikes that are mostly not that remarkable [18,19,26]. Therefore, it makes sense to use the time of global spike as the burst time.

Due to the fact that the time spans of various cascades may differ significantly, it is impractical to model all the cascades in the initial time scale. To address this challenge, we propose a time-window-based approach to eliminate the difference of various cascades in time span. Specifically, we first divide the time spans of all the cascades into K time windows and then try to predict the bursts of the cascades appearing in which future time window. Before formulating the problem, we first define the μth future time window of a cascade as follow.

Definition 3.3 *The μth future time window* Given constant K, μ and the cascade c with an observed spreading time interval $[t_0^c, t_{\text{current}}^c]$, where t_0^c is the starting time of the cascade c and t_{current}^c is the current time, the μth future time window of c is defined as such a time interval $[t_{\text{current}}^c + \frac{\mu-1}{K} \times (t_{\text{current}}^c - t_0^c), t_{\text{current}}^c + \frac{\mu}{K} \times (t_{\text{current}}^c - t_0^c)]$.

Figure 1 gives an illustration of the studied problem. The x-axis is time, and the y-axis is the number of reposts. The red solid curve is the current observed data, and the dashed curve is the future data. The observed data are equally divided into K time windows, and our task is to predict the future time window μ in which the burst occurs. Based on the above definitions and the illustration, we can address the prediction task by answering the following two questions.

I. Given a new cascade c with an early stage of observed diffusion process, how could we predict whether a burst will occur in its μth future time window?
II. How could we further predict in which future time window the burst will appear?

Question **I** can be considered as a binary classification problem and solved by a general classification method, such as SVM or decision tree. If we can accurately answer Question **I**, Question **II** can be solved based on the answer of Question **I**. A straightforward approach is to consider Question **II** as a multi-classification problem. However, our later experiment results will show that this attempt usually can not get desirable results due to the fact that the classification performance with different μ may be significantly distinct. This is because it is much harder to predict the burst occurring in a far future time window than that occurring in a near one. That is, the classification performance decreases with the increase in the parameter u (We will show that in our experiment part later). In the next section, we will introduce an effective approach to answer question **II** by recursively solving question **I**.

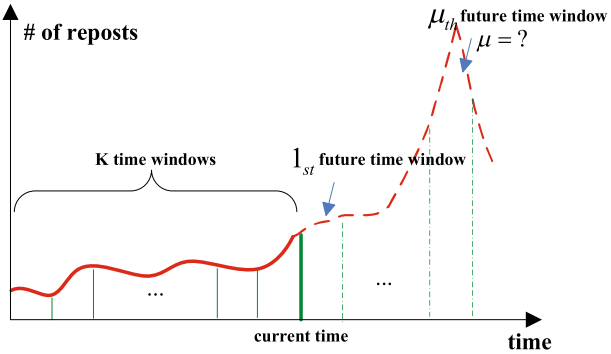


Fig. 1 An illustration of the burst time prediction problem in cascade. The x-axis is time, and the y-axis is the number of reposts. The observed part of the cascade is equally divided into K time windows. Assuming the burst of the cascade appears in the μth future time window, the problem is: *can we predict the value of* μ?

Based on the above analysis and definitions, we formally define the problem of burst time prediction in cascades as follow.

Definition 3.4 *Problem statement* Given a cascade c with the observed spreading process $\{(\text{user}_1^c, t_1^c), \ldots, (\text{user}_n^c, t_n^c)\}$ in the time interval $[t_0^c, t_{\text{current}}^c]$, where (user_1^c, t_1^c) means user_1 reposting c at the time t_1, a burst time prediction procedure attempts to learn an prediction function $f(c, \mu; K)$ such that

$$f(c, \mu; K) = \begin{cases} 1 & \text{If a burst occurs in the } \mu\text{th future window of } c; \\ 0 & \text{Otherwise.} \end{cases} \tag{1}$$

where K is a predefined parameter to determine the number of time windows the observed data are divided into.

4 Data analysis

In this section, we first conduct data analysis on the used dataset. Then we study the following two questions by analyzing the dataset: (1) *What are the patterns of the bursts in our dataset?* and (2) *How diverse the time series of the cascade popularity are?*

4.1 Dataset

Akin to Twitter, Sina Weibo is one of the most popular Chinese microblogging websites. In this paper, we study the public available dataset crawled from Sina Weibo[1] [14]. It contains 1,776,950 users, 308,489,739 following relationships, 300,000 popular microblog diffusion episodes with the original microblog and all its reposts. On average each microblog has been reposted for about 80 times.

For the purpose of this study, we first preprocess the dataset as follows. We remove some incomplete cascades. At the time of crawling data, some new posts have not shown the bursts yet. We identify and remove such incomplete cascades. To distinguish the complete and incomplete data, we first equally divide the observed time span of each cascade into

[1] http://arnetminer.org/Influencelocality#b2354.

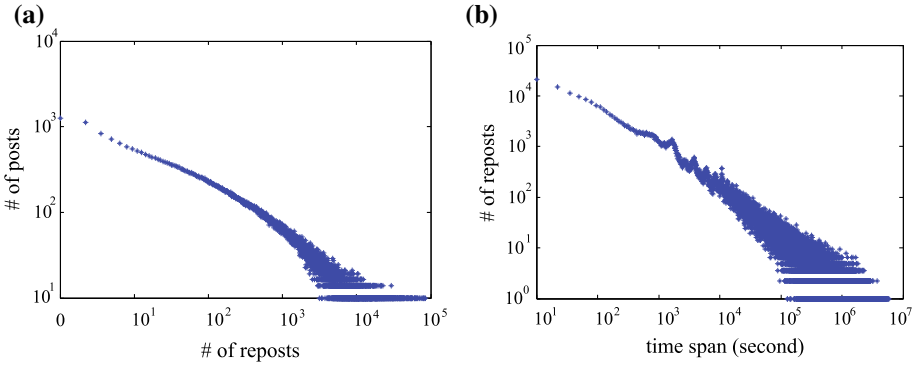


Fig. 2 Statistics of the Sina Weibo reposting dataset. **a** # of posts versus # of reposts, **b** # of posts versus time span of the posts

120 time windows [26]. Then the cascade c can be represented as such a time series: $c = \{n_1^c, n_2^c, \ldots n_{120}^c\}$, where n_i^c is the number of reposts in the ith time window. Based on the time series of the cascade, we find its global spike. The cascades with their global spikes in the observed time windows are kept, and all the other cascades are considered as incomplete data and removed from the dataset.

Figure 2 shows the statistics of the studied Sina Weibo reposting dataset. We plot the number of reposts for each post and the number of posts that shares the same number of reposts in Fig. 2a. Ones can see that it shows a power law distribution which is typical in social networks. It means only a smaller number of posts become highly popular eventually and get a large number of reposts, while most regular posts are relatively unattractive. We also plot the number of posts and the length of time span of the posts in Fig. 2b. One can also see a power law distribution that a smaller number of posts can survive for a rather long time and most ones only survive for several days.

4.2 Burst patterns analysis

To have an intuitive understanding of the burst patterns in Sina Weibo, we cluster the microblog diffusion episodes in the dataset using the K-Spectral Centroid algorithm proposed by Yang and Leskovec [19]. Figure 3 shows the clustering results. The horizontal axis is the time, and the vertical axis is the normalized volume of reposts. The results show that each temporal pattern of the Sina Weibo reposting data exhibits such a shape characteristic: a very rapid rise followed by a relatively slow decay, which can be considered as a burst. The temporal patterns on our dataset are consistent with previous studies on the hashtags adoption dataset in Twitter and the phrases propagation dataset on the Web [19,26].

One can also see that the shapes of the time series patterns are similar: a remarkable burst followed by a slow decay. The main different is that the burst time of the cascades for different patterns are different. This implies it may be very challenging to predict the bursts purely based on the time series of the cascade popularity. To accurately predict the time of the bursts, we need explore more knowledge for help.

4.3 Cascades diversity analysis

To study how diverse the spreading process of the cascades, we also investigate the diversity of the time series of the cascades. Here we use the Jensen–Shannon divergence (JS) [13] to

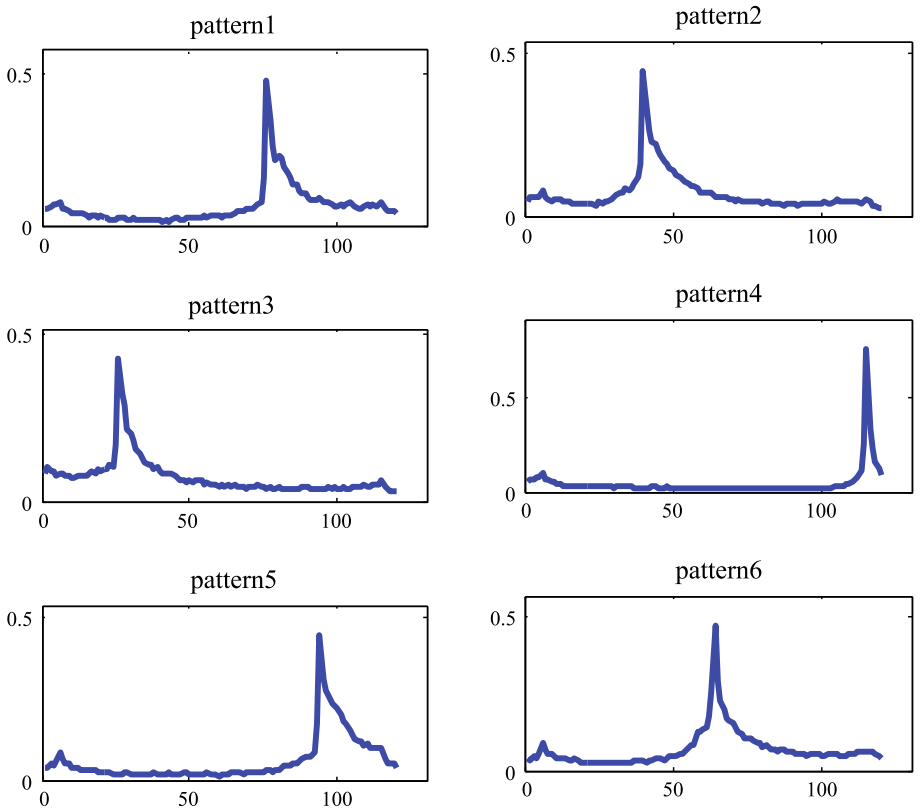


Fig. 3 Burst patterns of the cascades in Sina Weibo reposting dataset discovered by K-SC

quantitively measure the different between the time series distributions of each two cascades. JS divergence is a metric computed from Kullback–Liebler (KL) divergence to measure the distance between two distributions. Given two distributions Q and P, the Jensen–Shannon divergence between them can be computed by

$$\mathrm{JS}(P||Q) = \frac{1}{2}\left[\mathrm{KL}(P||M) + \mathrm{KL}(Q||M)\right] \tag{2}$$

where $M = \frac{1}{2}(P + Q)$, and $\mathrm{KL}(P||M)$ is the KL divergence between distributions P and M

$$\mathrm{KL}(P||Q) = \sum_{i=1}^{|P|} P_i \cdot \log\left(\frac{P_i}{Q_i}\right) \tag{3}$$

For each cascade c, we calculate its distribution of reposts in each time windows $P^c = (\frac{n_1^c}{N^c}, \frac{n_2^c}{N^c}, \ldots, \frac{n_{120}^c}{N^c})$, where n_i^c is the number of reposts in the ith time windows, and $N^c = \sum_{i=1}^{120} n_i^c$. To study how different the cascades are from those with the similar size and different sizes, we calculate the average JS divergence between two cascades with the similar size and different sizes, respectively. Figure 4 shows the results. Figure 4a shows the average JS divergence between two cascades with the same size, and Fig. 4b demonstrates the average JS divergence between two cascades with different sizes. From both Fig. 4a, b, one can see that

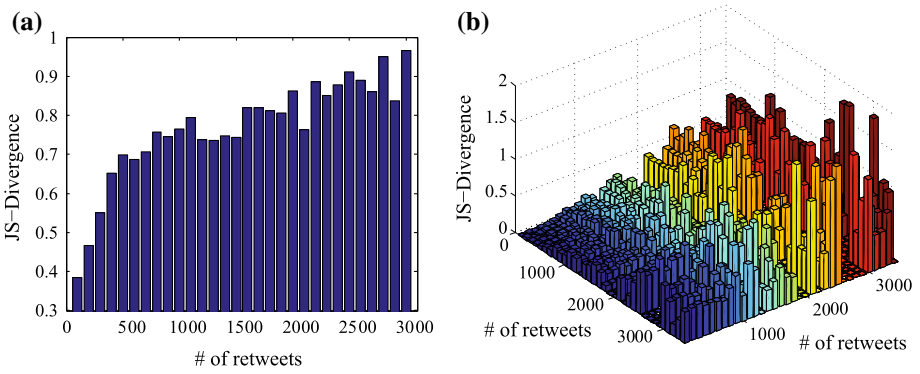


Fig. 4 The JS divergence between the time series distributions of two cascades. **a** Shows the average JS divergence between two cascades with the same size, and **b** shows the average JS divergence between two cascades with different sizes

with the increase in reposting size, the average JS divergence between two similar cascades increases with the increase in their reposing size. It means that the smaller cascades are more similar to each other than larger ones. One can further infer that the diffusion processes of larger cascades are more complicated and therefore, they may be harder to predict than smaller ones.

5 CPB: classification-based framework for burst time prediction

In this section, we first describe how we transform the time prediction task to a classification problem. Then we introduce what knowledge we exploit and how to model them in a scale-independent manner to help the classification task.

The intuition is that though the magnitudes and time spans of the cascades may be significantly different, the shapes of their time series curves may be similar (Fig. 3). Motivated by this, we propose the ***time-window-based transformation*** which equally divides the time spans of all the cascades into the same number of time windows. Then instead of predicting at which exact time point the burst will appear, we predict in which time window it occurs. This paves the way for transforming the time prediction task into a classification problem. Meanwhile, the time granularity of the prediction can be independent of the original time scale and only related to the number of time windows K.

The general methodology would be to represent a cascade with a set of features extracted from rich information diffusion related knowledge, and then we use **C**lassifiers to **P**redict the **B**urst will occur in which future time window (CPB). As an illustration, Fig. 5 shows how we construct the classifier to predict whether a burst will occur in the 1st future time window based on two cascades with significantly different time spans and popularities. The upper part illustrates how we extract the positive and negative samples from the raw times series data of the cascades. Horizontal axis is time and vertical axis is the reposting count. Given a cascade c with observed spreading process in the time interval $[t_0^c, t_{\text{current}}^c]$ (the green vertical line represents t_{current}^c), we first equally divide $[t_0^c, t_{\text{current}}^c]$ into K time windows. If a burst appears in the following future time window, the partial data between $[t_0^c, t_{\text{current}}^c]$ is considered as a positive sample; otherwise, it is a negative sample. We will introduce how to extract these training samples in details later (Sect. 4.1). Next we construct the classifier based on the extracted samples. Since the popularity information is insufficient, we will later

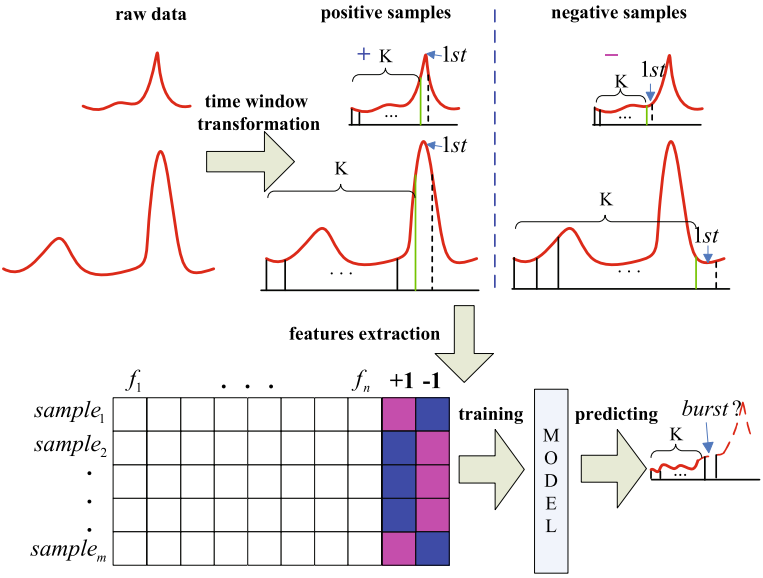


Fig. 5 Illustration of the classifier construction to predict whether a burst will occur in the 1st future time window. The *upper part* shows how we extract training samples from the raw time series data of the cascade popularity. The *lower part* shows we extract rich features from various information in each time window, and use them to train a classifier. When a new cascade comes, we predict whether the burst will appear in the coming time window

elaborate what knowledge we use and how to extract features from them (Sect. 4.2). One can see that the proposed framework enables us to handle cascades with various time spans and popularities uniformly. Finally, when a new cascade comes, we use the trained model to predict whether a burst will appear in its next time window.

5.1 Time-window-based transformation to construct classifiers

To answer question **I** in Sect. 2, we first introduce how to extract training samples based on the time-window-based transformation and how to use these samples to construct classifiers. We next present how to answer question **II** by recursively solving question **I**.

Classifier Construction for Question I For each cascade c, we construct classifiers for predicting whether the burst will appear in its 1st, 2nd,...μth future time windows, respectively, and extract corresponding training samples. For brevity, we only introduce how to construct the 1st future time window classifier as an example, and all the other classifiers can be constructed in the similar way.

To construct positive samples for the 1st future window classifier, we first identify the time of the global spike $t^c_{\max}$ for cascade c, and then we equally divide the time interval $[t^c_0, t^c_{\max}]$ into $K+1$ time windows $\{[t^c_0, t^c_1], (t^c_1, t^c_2] \ldots (t^c_K, t^c_{\max}]\}$. The time of the burst $t^c_{\max}$ can be considered to be in the last time window. If the current time is t^c_K and we can only observe the reposting data of c before t^c_K, the burst will occur in the next time window, namely the 1st future time window. Therefore, the reposting data of c in the time interval $[t^c_0, t^c_K]$ can be considered as a positive sample of the 1st future time window classifier. For the negative samples, similarly, we first randomly select a time point $\bar{t}^c_K$ such that the burst will not appear in the next time window. To do this, we equally divide the time interval $[t^c_0, t^c_{\max}]$ into $K+l+1$ time windows $\{[t^c_0, \bar{t}^c_1], \ldots (\bar{t}^c_{K-1}, \bar{t}^c_K] \ldots, (\bar{t}^c_{K+l}, t^c_{\max}]\}$, where l is

a random positive or negative integer. A positive l means the burst does not occur before the Kth window, while a negative l means the burst occurs before the Kth window. The reposting data of c in the time interval $[t_0^c, \bar{t}_K^c]$ can be considered as a negative sample of the 1st future time window classifier. For a testing sample, assuming the start time is t_0 and the current time is t_{current}, we also divide $[t_0, t_{\text{current}}]$ into K time windows.

Answering Question II Question **II** can not be directly answered by a single classifier mentioned above. Given a new cascade c_{new} and the μth future time window classifier, we can predict whether the burst will appear in its μth future time window. If the answer is *NO*, we still do not know when the burst will appear. That is, it is hard to answer question **II** by only one classifier. However, we can answer question **II** by recursively answering question **I** as follows. First we set the maximum future time window $\mu_{\max}$ and only predict whether bursts will appear in the first $\mu_{\max}$ time windows. We start with predicting whether the burst will appear in the 1st future time window using the 1st future time window classifier. If the answer is *YES*, the process stops and outputs the result; otherwise, we use the 2nd future time window classifier to predict whether it will appear in the 2nd future time window. The above process continues recursively until some classifier gives a positive prediction. If all the classifiers give the negative answer, we predict the burst appears in the last time window.

The reason why we conduct the prediction in this way is that, as shown in later experiment, the classification performance decreases with the increase in the parameter μ. Intuitively, bursts in near future time windows are easier to predict than those in farther future time windows. If two classifiers, for example the 1st and 2nd future time window classifiers both give positive predictions, we think the burst is more likely to appear in the 1st future time window because the former classifier is more accurate.

5.2 Model information diffusion related knowledge in a scale-independent manner

Besides the repost count, the cascades are also associated with a lot of other information, such as user profile and social relation. Oh et al. [2] study showed that there are a number of mechanisms by which social influence is transmitted such as networked structure and conformity. Cheng et al. [33] also found that the network structure information are helpful to predict cascades. Motivated by these works, we explore rich knowledge in information diffusion and categorize them into four types: general time-series-based knowledge, fluctuation knowledge, user profile knowledge, and social relation knowledge. For each type of knowledge, we extract scale-independent features that are derived from some initial features and independent from their absolute values.

5.2.1 General time-series-based knowledge

By simply considering the repost count in each time window as the time series data, we can extract some general time series features. Here we derive the following 6 features from the time series of repost count.

Average spreading speed (ASS) Suppose cascade c is represented as such a time series: $\{(n_1^c, 1), \ldots (n_K^c, K)\}$, where (n_i^c, i) denotes there are n_i^c reposts in the ith time window. The average spreading speed of c is

$$\text{ASS}^c = \frac{1}{K} \sum_{i=1}^{K} \frac{n_i^c}{t^c} \tag{4}$$

where t^c is the time window length of cascade c.

Average one-step increase rate (AIR_{+1}) Given the number of reposts n_k^c and n_{k+1}^c in the kth and $(k+1)$th time window, respectively, the one-step increase rate between the two successive time windows is defined as

$$\mathrm{AIR}_{+1}^c(k, k+1) = \frac{n_{k+1}^c - n_k^c}{n_k^c}. \tag{5}$$

Based on the one-step increase rate defined above, we can further calculate the average one-step increase rate by

$$\mathrm{AIR}_{+1}^c = \frac{1}{K-1} \sum_{i=1}^{K-1} \mathrm{AIR}_{+1}^c(k, k+1). \tag{6}$$

Average two-step increase rate (AIR_{+2}) Similarly, the two-step increase rate between every other time windows k and $k+2$ is defined as

$$\mathrm{AIR}_{+2}^c(k, k+2) = \frac{n_{k+2}^c - n_k^c}{n_k^c} \tag{7}$$

Similarly, we can calculate the average two-step increase rate by

$$\mathrm{AIR}_{+2}^c = \frac{1}{K-2} \sum_{k=1}^{K-2} \mathrm{AIR}_{+2}^c(k, k+2) \tag{8}$$

Recent data may be more important to help us predict the future trend of the cascade; hence, we also extract some features which are only related to the latest data.

Average spreading speed in the latest l time windows (ASS_l) The average spreading speed in the latest l windows can be defined as

$$\mathrm{ASS}_l^c = \frac{1}{l} \sum_{i=1}^{l} \frac{n_{K-i}^c}{t^c} \tag{9}$$

Average one-step increase rate in the latest l time windows (AIR_l_{+1}) The average one-step increase rate in the latest l time windows is defined as

$$\mathrm{AIR\_}l_{+1}^c = \frac{1}{l-1} \sum_{i=0}^{l-1} \mathrm{AIR}_{+1}^c\,(K-i-1, K-i) \tag{10}$$

Average two-step increase rate in the latest l time windows (AIR_l_{+2}) The average two-step increase rate in the latest l time windows is defined as

$$\mathrm{AIR\_}l_{+2}^c = \frac{1}{l-2} \sum_{i=0}^{l-2} \mathrm{AIR}_{+2}^c(K-i-2, K-i) \tag{11}$$

5.2.2 Fluctuation knowledge

The spreading process of cascades is rather dynamic and fluctuates over time [28,30,31]. An important reason causing the temporal dynamic is that users' behaviors are highly related to the time. To show this, in Fig. 6 we plot the number of posts and their burst time in hour and day. The left figure shows in each hour of day how many posts show their bursts, and the right figure shows the similar thing in each day of the week. The left figure shows that the bursts are much more likely to appear in the time interval from 8 am to 12 am and less likely

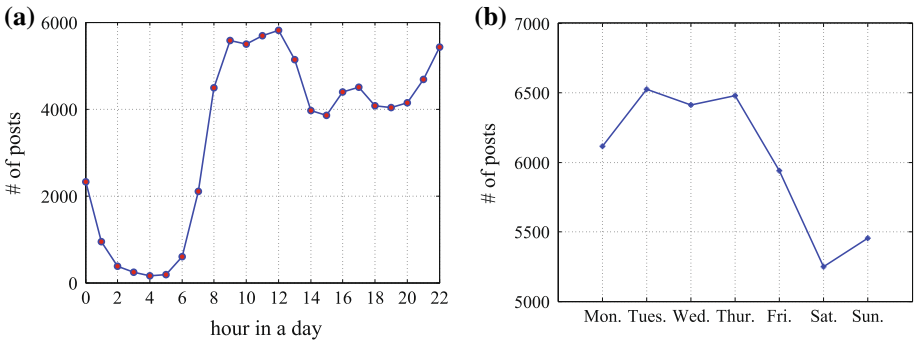


Fig. 6 The burst time distribution on hours of a day (**a**) and days of a week (**b**)

to appear in the time interval from 1 am to 6 am. From the daily basis, bursts are much less likely to appear at weekends. Based on the above observations, we use the hour and the day as our features.

Hour (H) We use the hour of current time as a time-related feature with 24 values from 0 to 23.
Day (D) The day of the week is selected as the second time related feature with 7 values from 0 to 6.

There usually exists several local spikes before the global spike comes during the cascades spreading [30,31]. The local spikes may also help us more accurately predict when the global spike will come. Therefore, we also extract some local spikes related fluctuation features.

Number of local spikes N_l Given a cascade c and the current time t_{current}, we identify all the local spikes before t_{current} and use the number of local spikes N_l as a feature.

Average normalized distance between two successive local spikes ($ADLL$) Assuming $t^c_{l\text{max\_}k}$ and $t^c_{l\text{max\_}k+1}$ are two successive local spikes of cascade c, the normalized distance between the two time points can be denoted as

$$d^c(k, k+1) = \frac{t^c_{l\text{max\_}k+1} - t^c_{l\text{max\_}k}}{t^c} \tag{12}$$

where t^c is the time window length. The average normalized distance between two successive spikes can be computed by

$$\text{ADLL}^c = \frac{1}{m}\sum_{i=1}^{m} d^c(i, i+1) \tag{13}$$

The normalized distance between the latest local spike and the current time (DLC) Assuming $k_{l\text{max\_}l}$ is the time window of the latest local spike and k_{current} is the current time window, the normalized distance between the latest local spike and current time can be defined as

$$\text{DLC}^c = k_{\text{current}} - k_{l\text{max\_}l} \tag{14}$$

One-step consistency (F^c_{+1}) Given the time series $\{(n^c_1, t^c_1), (n^c_2, t^c_2), \ldots (n^c_K, t^c_K)\}$ of cascade c, the one-step consistency between two successive time windows k and $k+1$ is defined as

$$f^c_{+1}(k, k+1) = \begin{cases} 0 & \text{if } n^c_k \geq n^c_{k+1} \\ 1 & \text{if } n^c_k < n^c_{k+1} \end{cases} \tag{15}$$

The one-step consistency of c is the sum of the one-step consistency between all the successive two time windows

$$F^c_{+1} = \sum_{k=1}^{K-1} f^c_{+1}(k, k+1) \tag{16}$$

Two-step consistency F^c_{+2} Similar to the one-step consistency, the two-step consistency between the time window k and $k+2$ is defined as

$$f^c_{+2}(k, k+2) = \begin{cases} 0 & \text{if } n^c_k \geq n^c_{k+2} \\ 1 & \text{if } n^c_k < n^c_{k+2} \end{cases} \tag{17}$$

The two-step consistency of c is the sum of all the two-step consistency between all the two time windows

$$F^c_{+2} = \sum_{k=1}^{K-2} f^c_{+2}(k, k+2) \tag{18}$$

5.2.3 User profile knowledge

Different from traditional time series data, the cascades are triggered and driven by users. The posts originating from different users may have significantly different impact on the spreading of the cascades [29]. For example, a tweet posted by an influential user in Twitter is more likely to obtain more retweets than that by a less influential user. Hence we also use the user profile knowledge and categorized them into two types: profile-based knowledge and authority-based knowledge. The profile-based knowledge includes **gender**, **location**, and **number of posts**. The authority-based knowledge includes **number of followers**, **number of followees**, whether the user is a **verified** user, **PageRank score**, and **HITS score** of the user. The PageRank [39] and HITS [40] scores are computed based on the following relationship graph of all the users. For each type of knowledge, we first obtain the corresponding data in each time window, and then derive scale-independent features based on the time series data in all the time windows. Due to space limitation, we only take the gender as an example to illustrate how we extract scale-independent features from it.

Gender In each time window (n^c_k, k), assuming the numbers of male and female users reposting post c are $n^c_{k\_m}$ and $n^c_{k\_f}$ respectively, we compute the ratio $g^c_k = \frac{n^c_{k\_m}}{n^c_{k\_f}}$. By calculating the ratios in all the time windows, we obtain such a gender related time series $G^c = \{g^c_1, g^c_2, \ldots, g^c_K\}$. Based on G^c, we can further extract the following derived features: *average gender ratio*, *average one-step increase rate of gender ratio*, *average two-step increase rate of gender ratio*, *the gender ratio in the last time window*, *the latest one-step gender ratio*, and *the latest two-step gender ratio*.

5.2.4 Social relation knowledge

In social networks, users are connected in the form of following and being followed by other users. The structure information of users and the spreading paths of the cascades may also potentially help us better predict the future trend of the cascades. Previous work studied whether the cascade is spreading primarily within a community or across many to predict the future number of users adopting the cascade [33]. Ma et al. [12] also studied that the total number of exposed users is an important feature in predicting Twitter hashtag popularity.

To study whether the structure information can be helpful, we extract some social relation related features.

Wiener index Recently, Goel et al. [8] have proposed the Wiener index as a measure of the structure vitality of a cascade. Cheng et al. [33] studied the importance of Wiener index as a structure feature to predict the future size of cascade. Hence we utilize it as the first structure feature. Wiener index is defined as follows,

$$v(T) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} \tag{19}$$

where d_{ij} denotes the length of the shortest path between nodes i and j. Intuitively, a cascade with low Wiener index value suggests that most nodes follow from a small number of hub nodes; while high Wiener index means that the cascade has many long paths.

Graph edit distance Graph edit distance is used to measure how similar two graphs are [11]. Given two graphs G and H, their graph edit distance is defined as

$$d(G, H) = |V_G| + |V_H| - 2|V_G \cap V_H| + |E_G| + |E_H| - 2|E_G \cap E_H| \tag{20}$$

where V_G, V_H are the nodes of graph G and H, and E_G, E_H represent the edges. In our case, we first extract the users following graphs G_i^c and G_{i+1}^c in two successive time windows i and $i+1$. Then we can compute the graph edit distance of the two graphs.

Vertex and edge overlap (VEO) Another metric to measure the similarity of two graphs is vertex and edge overlap [11]. The vertex and edge overlap of graph G and H is defined as

$$\text{Sim}_{\text{VEO}}(G, H) = \frac{|V_H \cap V_G| + |E_G \cap E_H|}{|V_G| + |V_H| + |E_G| + |E_H|} \tag{21}$$

Graph density As studied in [33], the density of the initial reposting graph of a cascade is helpful to predict the future trend of a cascade; hence, we also extract the graph density as a structure feature. The graph density of graph G can be computed by

$$\text{Density}_G = \frac{|E|}{|V| \times (|V| - 1)} \tag{22}$$

Entropy of degree distribution As an important property of a graph, degree distribution is discovered to be useful in predicting the bursting hashtags in Twitter [10]. Here we use the entropy of degree distribution of a graph as a feature. Entropy of degree distribution can be used to measure the heterogeneity of the network and can be computed by

$$\text{Entropy}_G = -\sum_{k=1}^{|V|} p(k) log(p(k)) \tag{23}$$

where $p(k)$ is the probability of a node with degree k in G.

We take the graph edit distance as an example to show how we model the social relation knowledge and extract features. All the other measures can be modeled in the similar way. For each time window k, we first extract a graph G_k^c based on the nodes involved in cascade c before the current time window and the following relationships among these nodes. Then we can obtain a set of graphs $\mathcal{G}^c = \{G_1^c, G_2^c \ldots, G_K^c\}$. Based on $\mathcal{G}^c$ we can compute the graph edit distance $d_{k,k+1}$ between two successive graphs G_k^c and G_{k+1}^c. Then we can further obtain the time series data of the graph edit distance $D = \{d_{1,2}, d_{2,3} \ldots, d_{K-1,K}\}$. Using the time series data of D, we derive some scale-independent features similar to the general time series features.

6 Experiments

We conduct extensive experiments to systematically evaluate our approach in this section. First, we verify whether the proposed approach can learn an accurate burst time prediction model by examining the classification performance with various learning algorithms. Then we conduct parameter analysis to examine how sensitive the CPB model is to the parameter K. Next we perform feature importance analysis and investigate how the four types of features impact learning performance. Finally, we quantitively study how accurate CPB can predict the burst time of the cascades compared with baselines. To study how robust the CPB model is, we also test the prediction performance on some large cascades with only small cascades as training samples.

6.1 Dataset

We use three datasets to evaluate the proposed burst time prediction model. The first dataset is the Sina Weibo reposing dataset we described in Sect. 3. As we have discussed and analyzed this dataset, we will omit the description on it here. The second and third datasets are both extracted from the MemeTracker dataset.[2] We briefly describe this dataset as follows.

The MemeTracker dataset contains more than 300 million blog posts and news articles collected from 1 million websites. Memes are short textual phrases or quotes that spread through the web. Each meme can be considered as a piece of information, and all the time-stamped webpages which contain the same meme forms a diffusion cascade. Each webpage may also contain some hyper-links pointing to other webpages on the webs. Similar to the following relationships in social networks, the hyper-links among these webpages can be also consider as the networking information. We extract the hyper-links among the webpages and construct a directed graph and use the constructed graph to extract *social relation knowledge* to structure based features. As there are no profile-based knowledge like location, gender, and number of posts as in Sina Weibo, we only extract the authority-based knowledge as the *user profile knowledge* to construct user/node features. The authority-based knowledge includes number of webpages pointing in and pointing out the webpage, the PageRank and HITS scores. As the general time serious features and fluctuation features only rely on the time serious data of cascade, the two types of features for the MemeTracker dataset can be extracted in the same manner as the Sina Weibo dataset. Thus we omit the details here.

We extract two datasets from the initial MemeTracker dataset. (1) MemeTracker phrase cluster dataset. As the whole MemeTracker dataset is larger and some memes are very similar to each other. The similar memes are merged as a meme cluster. We select the phrase clusters that are mentioned more than 20 times in different webpages in August 2008. In all we obtained around 68,316 cascades and the average size of the meme clusters is 90. (2) MemeTracker raw phrases dataset. To further evaluate the effectiveness of the proposed model, we also use the raw phrases which are mentioned more than 20 times as a dataset. In all, we extract 81,504 cascades in this dataset. On average, each phrase has been mentioned for 73 times in different webpages. One can see that the cascades of the raw phrase data are smaller than those of the phrase cluster data.

[2] http://www.memetracker.org/data.html.

Table 1 Classification performance for various learning algorithms on the three datasets

Algorithm	Sina Weibo			MT phrase cluster			MT raw phrases		
	F1	AUC	Acc (%)	F1	AUC	Acc (%)	F1	AUC	Acc (%)
Naive Bayes	0.739	0.798	74.1	0.752	0.814	75.8	0.743	0.812	76.4
BayesNet	0.840	0.917	83.4	0.824	0.854	83.5	0.833	0.858	84.3
KNN	0.827	0.832	83.2	0.832	0.845	83.7	0.818	0.847	81.7
Logistic Regression	0.826	0.895	82.8	0.818	0.834	82.6	0.824	0.843	81.6
Multilayer Perceptron	0.899	0.915	90.2	0.900	0.921	91.1	0.910	0.924	90.6
Adaboost	0.854	0.929	85.4	0.867	0.890	88.7	0.837	0.856	82.9
Bagging	0.912	**0.937**	91.4	0.898	**0.927**	90.6	0.904	0.931	91.6
Random Forest	0.892	0.904	89.2	**0.925**	0.914	**92.2**	0.894	0.910	90.2
J48 Decision Tree	**0.928**	0.922	92.2	0.913	0.923	91.7	**0.922**	**0.928**	91.4
Random Tree	0.904	0.934	**92.4**	0.897	0.914	90.8	0.914	0.923	**92.0**
LibSVM (linear)	0.824	0.843	81.6	0.815	0.843	82.6	0.824	0.847	81.8
LibSVM (polynomial)	0.832	0.847	82.5	0.822	0.854	83.7	0.842	0.861	84.5
LibSVM (RBF)	0.824	0.828	82.8	0.845	0.864	85.7	0.841	0.852	85.1
LibSVM (sigmoid)	0.852	0.873	86.2	0.841	0.866	84.5	0.824	0.846	83.1
LibLinear	0.829	0.834	83.4	0.834	0.845	83.7	0.837	0.852	84.2

For short, we use MT to represent MemeTracker. The results on three metrics: F1-measure, AUC, and accuracy are reported
Bold values indicate the best results of all the classification algorithms

6.2 Performance analysis with various learning algorithms

We first exam the classification performance of various learning algorithms. We use 10-fold cross validation to evaluate on three metrics F1-measure, Area Under ROC Curve (AUC) and classification accuracy. In this experiment, we divide the time spans of all the cascades into 10 time windows and predict whether the burst occurs in the first future time window. That is we set $K = 10$ and $\mu = 1$, and the result is given in Table 1. As shown in the table, the classification accuracy of most classification techniques, except for Naive Bayes, are over 80 %, which means the results are rather good. The performance of Multilayer Perceptron, Random Forest, Bagging, and J48 Decision Tree are not significantly different: the accuracy is around 90 %. It implies that when sufficient features are available, this prediction task is not much sensitive to the choice of the learning algorithms. For the Sina Weibo dataset, Random Tree is shown to be the most accurate algorithm with a classification accuracy of 92.4 % and AUC value of 0.934. For the MemeTracker phrase cluster dataset, the Random Forest algorithm achieves the highest accuracy of 92.2 %. The performance of Random Tree is also desirable: AUC is 0.914 and accuracy is 90.8 %. One can see that Random Tree also achieves desirable performance for the MemeTracker raw phrase dataset. Thus in the following experiments, we use Random Tree algorithm as our classification method.

6.2.1 Statistical significance test for the classification performance

To further study whether some classification algorithms perform significantly better than others, we conduct t test to compare the classification performance of these algorithms. Following the method proposed in Thomas [44], we conduct 30 trials. For each trial, we

first randomly split the dataset into the training set and testing set. Specifically, two-thirds of the entire dataset are selected as training data and the remaining are testing data. Then we train various learning algorithms on the training data and test the performance on the test data. Given two classifiers A and B, let p_A^i (respectively, p_B^i) be the observed proportion of test examples misclassified by algorithm A (respectively B) during trial i. We establish the null and alternative hypothesis as follows: null hypothesis $H_0 : p_A = p_B$, alternate hypothesis $H_1 : p_A > p_B$. If we assume that the 30 differences $p^i = p_A^i - p_B^i$ were drawn independently from a normal distribution, then we can apply Student's t test, by computing the statistic

$$t = \frac{\bar{p} \cdot \sqrt{n}}{\sqrt{\frac{\sum_{i=1}^{n} (p^i - \bar{p})^2}{n-1}}} \tag{24}$$

where $\bar{p} = \frac{1}{n} \sum_{i=1}^{n} p^i$. If we set the p value = 0.05, the null hypothesis can be rejected if $|t| > t_{29,0.975} = 2.04523$ for the 30 trials.

We conduct t test on the three datasets to further study the classification performance of the algorithms. In our last experiment, the Random Tree, Randoms Forest, and J48 Decision Tree algorithms perform best on the Sina Weibo dataset and MemeTracker dataset, respectively. We choose Random Tree, Random Forest, and J48 Decision Tree as classifier A for the Sina Weibo dataset, MemeTracker phrase cluster dataset, and MemeTracker raw phrases dataset, respectively. For each group of t test, we select four algorithms that achieve the similar classification accuracy with the classifier A as classifier B. For the Sina Weibo dataset, we choose Bagging, Adaboost, LibLinear, and J48 Decision Tree as classifier B. For the MemeTracker phrase cluster dataset, the classifier B are selected as Bagging, Multilayer Perceptron, LibSVM, and J48 Decision Tree. For the MemeTracker raw phrases data, we select Random Tree, Multilayer Perceptron, Random Forest, and Bagging as classifier B. Table 2 gives the results of the t test. One can see that for the Sina Weibo dataset, the hypotheses that accuracy of Random Tree is similar to Bagging, Adaboost, LibLinear should be rejected as the t values are all larger than $t_{29,0.975} = 2.04523$. However, the hypothesis should be accepted for the J48 Decision Tree, which means that there is no significant difference between the classification performances of Random Tree and J48 Decision Tree. For the MemeTracker phrase cluster dataset, all the hypotheses should be rejected which means that Random Forest is significantly better than the other methods. For the MemeTracker raw phrases dataset, Random Tree and Multilayer Perceptron can achieve similar performance as J48 Decision Tree, while Bagging and Random Forest are both significantly inferior to J48 Decision Tree.

6.3 Effect of parameter K

To study the effect of parameters K and μ on the classification performance, we conduct experiments with various K over different μ. Here K is the number of divided time windows and μ is the future time window in which we predict whether the burst occurs. Note that given a cascade c, the burst time window μ may change if K changes. The results are given in Fig. 7. The x-axis is the μth future time window with μ from 1 to 5, and the y-axis shows the classification accuracy. We first set K to a set of relatively small numbers: $K = 6$, $K = 8$, and $K = 10$. We can observe a monotonically decrease trend in classification accuracy with the increase in the parameter μ. It means the burst occurring in a farther away future time window is harder to predict than that occurring in a time window closer to the current time. One can also see the classification accuracy does not show significant difference with relatively small K values.

Table 2 t test for the classification accuracy on the three datasets

t test on the Sina Weibo dataset				
Random Tree	Bagging	Adaboost	LibLinear	J48 Decision Tree
versus t value	2.8742	3.6754	4.5674	1.6543
t test on the MemeTracker phrase clusters				
Random Forest	Bagging	Multilayer Perceptron	LibSVM (sigmoid)	J48 Decision Tree
versus t value	3.8740	2.9785	4.8675	2.1456
t test on the MemeTracker raw phrases				
J48 Decision Tree versus t value	Bagging	Multilayer Perceptron	Random Forest	Random Tree
	2.1754	1.8975	2.7582	1.4568

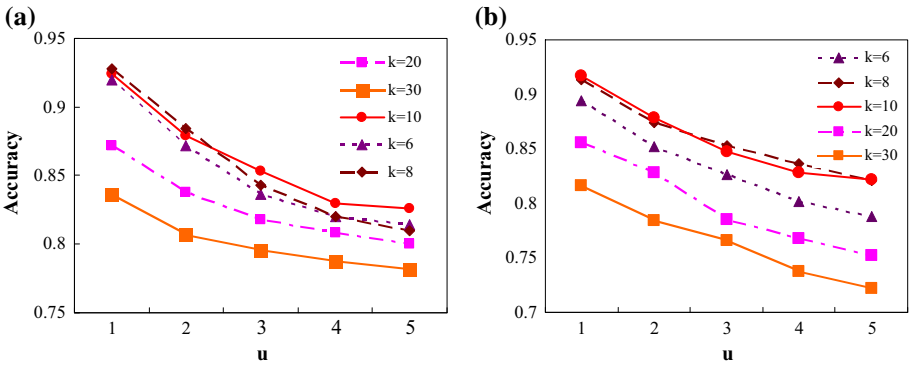


Fig. 7 Classification accuracy on different time windows K and the future time windows μ on the two datasets. The x-axis is the μth future time window, and the y-axis is the classification accuracy. We report the results with $K = 6$, 8, 10, 20, and 30. **a** Sina Weibo dataset, **b** MemeTracker phrase cluster dataset

To further verify whether larger K can significantly impact the classification performance, we set K to two larger values: $K = 20$ and $K = 30$. The results are shown in the same figure. One can see that the classification performance decreases significantly if K is set to a relatively large value. The result is not surprising, because larger K means smaller time window and more fine-grained prediction. Smaller time window makes the difference between the data in two successive time windows smaller and harder to distinguish. It is intuitively harder to predict whether the burst of a cascade will occur in some hour than in some day in the future. The result also shows that $K = 10$ seems to be a reasonable choice for the classification task. In the following experiments we use $K = 10$ as the parameter of choice.

6.4 Feature importance analysis

Feature importance analysis studies how important the different features are in learning the prediction task. In other words, we want to investigate which features contribute most to the classification task. From a macroscopic view, we first study the importance of the features derived from different types of knowledge. Figure 8 shows the classification accuracy from

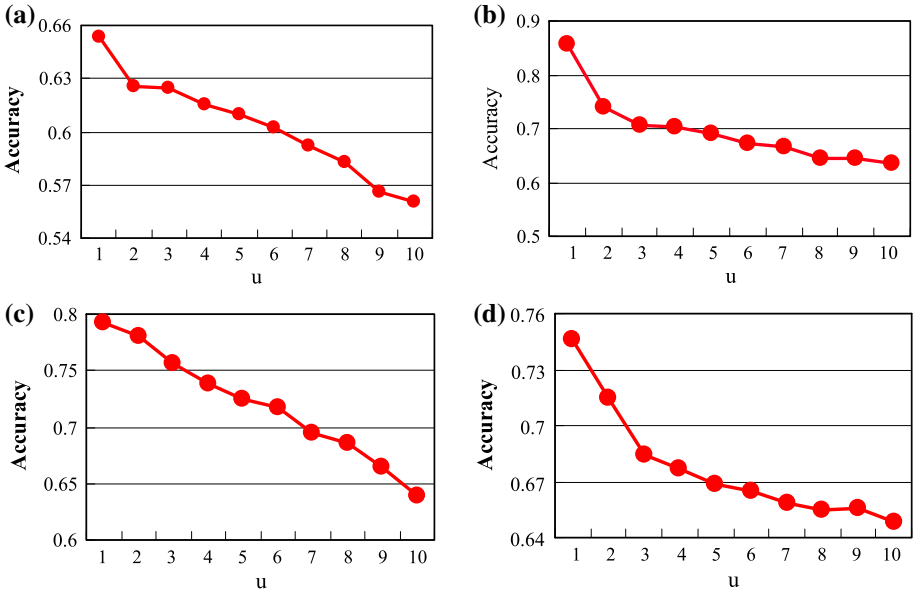


Fig. 8 The classification accuracy of the classifiers trained on each group of features separately on the Sina Weibo dataset. The x-axis is the μth future time window, and the y-axis is the classification accuracy. **a** General time series features, **b** fluctuation features, **c** user features, **d** structure features

$\mu = 1$ to $\mu = 10$ achieved by the classification functions trained on each group of features separately. We summarize the results by the following observations:

- *General time series features are surprisingly not very useful* Figure 8a shows that the classification accuracy is less that 66 % if we use the general time series features only. It implies that it is hard to predict the time of bursts simply using the time series of the cascade popularity.
- *The most important features are fluctuation features* Only using the fluctuation features, the classification accuracy for the 1st future time window is near 90 %, which is the highest in the four groups of features.
- *Both the user profile features and social relation features are helpful* Surprisingly, both user profile and social relation features perform much better than general time series features. User profile features seem slightly better than social relation features, and both types of features perform rather well. Only using the user profile features, the accuracy is around 80 % for the 1st future time window prediction task, and the figure is 75 % for the social relation features only.

Next we study the importance of features from a microcosmic point of view. We utilize the Information Gain (IG) method to rank all the features, and the top 10 important features are given in Table 3.

One can see that the top-10 important features include four fluctuation knowledge related features, three user profile related features and three social relation related features. Top-4 important features are all fluctuation features, which also demonstrate their importance in our prediction task. In the top-4 features, three of them is related to the local spikes, and the other one is the *hour* of the current time. It shows that local spike information is a good

Table 3 Top-10 important features

Top-10 features
Average normalized distance between two successive local spikes
Normalized distance between the latest local spikes and the current time
Number of local spikes
Hour
Average one-step increase rate of the number of followees
The latest one-step increase rate of PageRank score
Average one-step increase rate of Wiener index
The latest one-step increase rate of Wiener index
The latest two-step increase rate of number of followers
The latest one-step increase rate of graph density

indicator of the occurrence of the global spike. None of general time series features are in the top-10 features, which implies they are less helpful to predict bursts.

6.5 Classification performance versus cascade size

As we studied in Sect. 4.3, the time series of the reposing size between two smaller cascades is more similar than that between two larger ones, which implies the spreading of larger cascades may be more complex than smaller ones. Thus it is nature to ask whether larger cascades are harder to predict their bursts than smaller ones. Here we study whether the classification accuracy increases or decreases with the increase in cascade size.

Figure 9a plots the classification accuracy curve for the posts from the size 100 increasing to 2,000 over the Sina Weibo dataset. Figure 9b shows the classification accuracy curve for the phrase clusters from the size 100 to 3000 over the MemeTracker dataset. The curve in Fig. 9a shows a decrease trend in classification accuracy with the increase in the cascade size. There is a sharp drop for the accuracy from size 100 to about 600. From 600 to 2000, the decrease trend becomes gently. It implies that larger cascades are harder to predict that smaller ones. This is mainly because larger cascades are usually more diverse and complex. Larger cascades may experience several peaks and are harder to fade out quickly. The spreading process of relatively smaller cascades, on the contrary, is simper and more similar to each other: experiences a remarkable peak and fades out quickly. One can also see that the curve in Fig. 9b shows the similar trend. This results on both dataset verify our observation in Sect. 4.3 that larger cascades are harder to predict as their diffusion processes are more complicated.

6.6 Quantitive comparison with baselines

In previous sections, we evaluate the classification performance of our approach on predicting whether the burst will occur in a particular future time window; in this section, we aim to quantitively evaluate whether our approach can accurately predict the burst occurs in which time window. In order to demonstrate the effectiveness of CPB, we implemented the following four methods as baselines.

- *Random (RD)* We randomly select a future time window as the time window in which the burst occurs.
- *Auto-regressive moving average (ARMA)* ARMA is a popular statistical analysis model of time series. Given a time series data X_t, the ARMA model aims to predict future values in

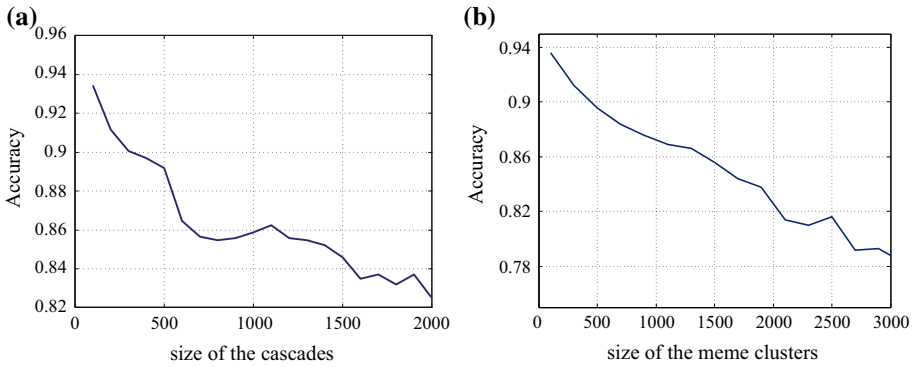


Fig. 9 Classification performance versus cascade size on the two datasets. The x-axis is the size of the cascades, and the y-axis is the classification accuracy. One can see that the accuracy decreases with the increase in cascade size, which shows larger cascades are harder to predict. **a** Sina Weibo dataset, **b** MemeTracker phrase cluster dataset

this series. As ARMA is a value prediction model rather than the time prediction model, to make it comparable, we first use ARMA to predict the values in several future time windows. The time window with the maximum prediction value is considered as the burst time window.

- *Multi-classification (Multi_C)* We consider the problem of predicting the time window in which the burst occurs as a multi-classification problem and use a multi-class classifier to predict the burst occurs in which time window.
- *SPIKEM* SPIKEM [26] is designed to capture the diffusion patterns of cascades. To make it comparable, we first use SPIKEM to forecast the future volume of a cascade based on its early data and then identify the burst time window based on the predicted future volume in each future time window.
- *CPB only using time series features of the cascades popularity (CPB_CP)* To study whether the rich knowledge can improve the prediction performance, we also use CPB with only the time series features of the cascades popularity as a baseline.

We use the mean absolute error (MAE) computed by $MAE = E(\frac{|I^c - \hat{I}^c|}{l})$ as the evaluation metric. Here I^c denotes the true future time window in which the burst occurs, $\hat{I}^c$ is the predicted time window, and l is the number of future time windows. The results with various μ are given in Table 4.

For ease of comparison, we first fix the number of future time windows l. Then we select the testing samples whose bursts occur in one of the l future time windows. Table 4 gives the results with l from 2 to 8. The figures in bold show the best results. Ones can see that on both datasets the proposed CPB performs significantly better than the five baselines in terms of MAE in all the cases. The MAE increases with the increase in l, which implies bursts in a farther away time window is harder to predict. One can also see that although multi-classification approach is significantly better than random method, it is less effective than CPB. The performance of SPIKEM and ARMA is not desirable: even inferior to multi-classification method. This is mainly because the two methods only utilizes the time series data of the cascade popularity, but cannot capture and handle various other knowledge in information diffusion. Compared to CPB_CP, the MAE value achieved by CPB decreases by an average of about 30 %. It shows that the rich knowledge in information diffusion does help our task.

Table 4 Quantitive comparison against various baselines

μ	RD	ARMA	Multi_C	CPB_CP	SPIKEM	CPB
Sina Weibo dataset						
2	0.250	0.084	0.060	0.056	0.108	0.042
3	0.296	0.227	0.132	0.125	0.224	0.065
4	0.312	0.233	0.135	0.158	0.227	0.102
5	0.320	0.244	0.172	0.186	0.246	0.142
6	0.325	0.246	0.178	0.182	0.222	0.144
7	0.330	0.245	0.176	0.184	0.254	0.146
8	0.336	0.258	0.188	0.192	0.267	0.152
MemeTracker phrase cluster dataset						
2	0.252	0.093	0.062	0.066	0.118	0.044
3	0.293	0.242	0.166	0.132	0.234	0.074
4	0.315	0.274	0.153	0.163	0.233	0.112
5	0.322	0.255	0.182	0.191	0.252	0.146
6	0.328	0.266	0.188	0.190	0.263	0.148
7	0.331	0.254	0.189	0.187	0.286	0.152
8	0.342	0.261	0.191	0.194	0.287	0.155
MemeTracker raw phrases dataset						
2	0.246	0.087	0.060	0.056	0.128	0.042
3	0.289	0.256	0.145	0.136	0.224	0.069
4	0.322	0.271	0.157	0.158	0.252	0.123
5	0.314	0.258	0.178	0.186	0.273	0.152
6	0.343	0.274	0.192	0.178	0.263	0.147
7	0.352	0.271	0.194	0.182	0.266	0.158
8	0.347	0.265	0.187	0.184	0.274	0.162

6.7 Robustness analysis of CPB

In practice, the size distribution of cascades is extremely skew, which means most posts only have a small number of reposts and only a small number of posts are highly popular. In the Sina Weibo dataset, only less than 1 % posts are reposted more than 1000 times. Can the CPB model give desirable prediction results for the large cascades that people may concern more in reality?

To study this problem, we test the robustness of the CPB model by such an experiment. For the Sina Weibo dataset, we first choose all the cascades that are reposted for more than 1000 times; and for the MemeTracker dataset we choose all the meme phrases that are mentioned in more than 1500 different webpages. We then only choose some small cascades as training samples to get a prediction model. With such a prediction model, we test its effectiveness in predicting the selected large cascades. Figure 10 shows the experiment results by choosing cascades with various popularities of posts and meme phrases. For a fair comparison, we select the same number of training samples for all the classifiers. One can see that the accuracy increases with the increase in cascade popularity on both datasets. However, the increase trend is not that remarkable. On the Sina Weibo dataset, even if we use the classifier trained on the cascades with only around 100 reposts, the accuracy is about 5 % lower that the prediction model trained on the selected large cascades themselves. If

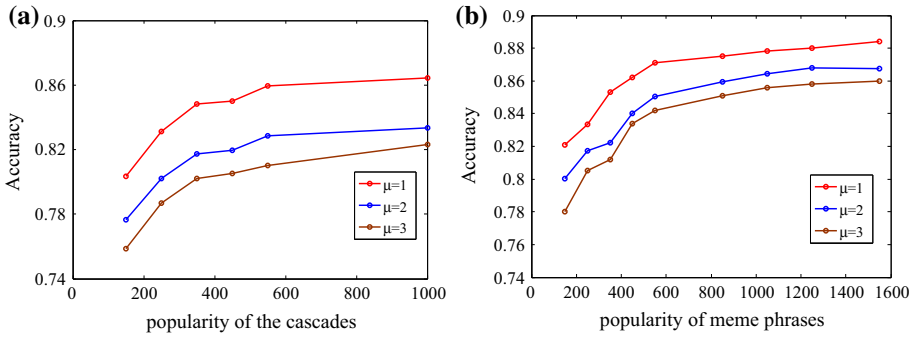


Fig. 10 The robust experiment on the two datasets. The results with parameter $\mu = 1$, $\mu = 2$, and $\mu = 3$ are reported. **a** Sina Weibo dataset, **b** MemeTracker phrase cluster dataset

we use larger number of smaller cascades, the difference can be even smaller. Similarly, on the MemeTracker dataset, the classification performance of the classifier trained on the meme phrases with a popularity of 600 is almost the same as that trained on the samples with a popularity of 1500. It demonstrates that the CPB model is rather robust and not much sensitive to the popularity of the training cascades.

7 Conclusion and future work

In this paper, we studied the problem of burst time prediction in cascades. Although the cascade volume prediction and the bursty nature of the cascades are well studied, predicting when a burst will occur is less touched. We proposed a novel classification-based approach CPB to predict the burst time of the cascades by extracting rich scale-independent features. Our solution allows us to predict the cascades with diverse magnitudes and time spans in a unified manner, since we conduct the prediction in the time window granularity by a novel time-window-based transformation. Extensive evaluations on a real social network dataset demonstrate the effectiveness of CPB. Meanwhile, we also give some interesting observations about burst in information diffusion, which may direct us to have a deeper understanding of information diffusion in social media.

Potential avenues of future work include a deeper study on the underlying spreading mechanisms of the bursts. A more comprehensive analysis on the reasons causing the bursts of cascades may lead us to new insights on better understanding on human behaviors in information diffusion. It would also be interesting to further study some other interesting properties of the bursts in cascades such as the duration and the size of the bursts.

Acknowledgments This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 61170189, 61370126, 61202239), National High Technology Research and Development Program of China under Grant (No. 2015AA016004), Major Projects of the National Social Science Fund of China under Grant (No. 14&ZH0036), Science and Technology Innovation Ability Promotion Project of Beijing (PXM2015-014203-000059), the Fund of the State Key Laboratory of Software Development Environment (No. SKLSDE-2015ZX-16), Microsoft Research Asia Fund (No. FY14-RES-OPP-105), the Innovation Foundation of BUAA for PhD Graduates (No. YWF-14-YJSY-021), US NSF through Grants III-1526499, CNS-1115234, and OISE-1129076.

References

1. Hu X, Tang L, Tang JL, Liu H (2013) Exploiting social relations for sentiment analysis in microblogging. In: Proceedings of the sixth ACM international conference on web search and data mining, pp 537–546
2. Oh J, Susarla A, Tan Y (2008) Examining the diffusion of user-generated content in online social networks. Soc Sci Res Netw. doi:10.2139/ssrn.1182631. http://ssrn.com/abstract=1182631
3. Wang SZ, Hu X, Yu PS, Li ZJ (2014) MMRate: inferring multi-aspect diffusion networks with multi-pattern cascades. In: Proceedings of the 20th ACM SIGKDD conference on knowledge discovery and data mining, pp 1246–1255
4. Wang SZ, Zhang HH, Zhang JW, Zhang XM, Yu PS, Li ZJ (2015) Inferring diffusion networks with sparse cascades by structure transfer. In: Proceedings of the 20th international conference on database systems for advanced applications, pp 405–421
5. Parikh N, Sundaresan N (2008) Scalable and near real-time burst detection from e-commerce queries. In: Proceedings of the 14th ACM SIGKDD conference on knowledge discovery and data mining, pp 972–980
6. Cui P, Jin SF, Yu LY, Wang F, Zhu WW, Yang SQ (2013) Cascading outbreak prediction in networks: a data-driven approach. In: Proceedings of the 19th ACM conference on knowledge discovery and data mining, pp 901–909
7. Mill TC (1990) Time series techniques for economists. Cambridge University Press, Cambridge
8. Goel S, Anderson A, Hofman J, Watts D (2013) The structure virality of online diffusion (preprint)
9. Gruhl D, Guha R, Kumar R, Novak J, Tomkins A (2005) The predictive power of online chatter. In: Proceedings of the 11th ACM SIGKDD conference on knowledge discovery and data mining, pp 78–87
10. Kong SB, Mei QZ, Feng L, Zhao Z, Ye F (2014) On the Real-time prediction problem of bursting hashtags in twitter. CoRR abs/1401.2018
11. Papadimitriou P, Dasdan A, Garcia-Molina H (2008) Web Graph Similarity for Anomaly Detection. iN: Proceedings of the 17th International World Wide Web Conference, pp 1167–1168
12. Ma ZY, Sun AX, Cong G (2013) On predicting the popularity of newly emerging hashtags in twitter. J Am Soc Inf Sci Technol 7(64):1399–1410
13. Lin JH (1991) Divergence measures based on the Shannon entropy. IEEE Trans Inf Theory 1(37):145–151
14. Zhang J, Liu B, Tang J, Chen T, Li JZ (2013) Social influence locality for modeling retweeting behaviors. In: Proceedings of the 23rd international joint conference on artificial intelligence, pp 2761–2767
15. Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international World Wide Web conference, pp 851–860
16. Kleinberg J (2002) Bursty and hierarchical structure in streams. In: Proceedings of the 8th ACM SIGKDD conference on knowledge discovery and data mining, pp 91–101
17. Li L, Liang CJM, Liu J, Nath S, Terzis A, Faloutsos C (2011) Thermocast: a cyber-physical forecasting model for data centers. In: Proceedings of the 17th ACM SIGKDD conference on knowledge discovery and data mining, pp 1370–1378
18. Crane R, Sornette D (2008) Robust dynamic classes revealed by measuring the response function of a social system. Proc Natl Acad Sci USA 41(105):15649–15663
19. Yang J, Leskovec J (2011) Patterns of temporal variation in online media. In: Proceedings of the fourth ACM international conference on web search and data mining, pp 177–186
20. Kleinberg J (2005) Temporal dynamics of on-line information streams. In: Data Stream Managemnt: Processing High-speed Data. Springer
21. Zhu YY, Shasha D (2003) Efficient elastic burst detection in data streams. In: Proceedings of the 9th ACM SIGKDD conference on knowledge discovery and data mining, pp 336–345
22. Pinsen D (2012) Predicting the bursting of a market bubble. http://finance.yahoo.com/news/predicting-bursting-market-bubble-171432469.html
23. Barabási A (2011) BURSTS: the hidden pattern behind everything we Do, from Your E-mail to Bloody Crusades. Penguin, New York
24. Barabási A (2005) The origin of bursts and heavy tails in human dynamics. Nature 435:207–211
25. Vazquez A, Oliveira JG, Dezso Z, Goh K, Kondor I, Barabási A (2006) Modeling bursts and heavy tails in human dynamics. Phys Rev E 73, 036126:1-19
26. Matsubara Y, Sakurai Y, Prakash BA, Li L, Faloutsos C (2012) Rise and fall patterns of information diffusion: model and implications. In: Proceedings of the 18th ACM SIGKDD conference on knowledge discovery and data mining, pp 6–14
27. Hong LJ, Dan O, Davison BD (2011) Predicting popular messages in twitter. In: Proceedings of the 20th international World Wide Web conference, pp 57–58
28. Szabo G, Huberman BA (2010) Predicting the popularity of online content. Commun ACM 53(8):81–88

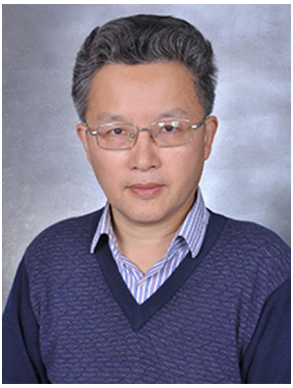
29. Kupavskii A, Umnov A, Gusev G, Serdyukov P (2013) Predicting the audience size of a Tweet. In: Proceedings of the seventh international AAAI conference on weblogs and social media, pp 693–696
30. Petrovic S, Osborne M, Lavrenko V (2011) RT to Win! Predicting message propagation in twitter. In: Proceedings of the fifth international AAAI conference on weblogs and social media
31. Myers S, Leskovec J (2014) The bursty dynamics of the twitter information network. In: Proceedings of the 23th international World Wide Web conference, pp 913–924
32. Goel S, Watts DJ, Goldstein DG (2012) The structure of online diffusion networks. In: Proceedings of conceptual modeling—31st international conference, pp 623–638
33. Cheng J, Adamic LA, Dow PA, Kleinberg J, Leskovec J (2014) Can cascades be predicted? In: Proceedings of the 23rd international World Wide Web conference, pp 925–936
34. Kupavskii A, Ostroumova L, Umnov A, Usachev S, Serdyukov P, Gusev G, Kustarev A (2012) Prediction of retweet cascade size over time. In: Proceedings of the 21st ACM international conference on information and knowledge management, pp 2335–2338
35. Gershenfeld N (1999) The nature of mathematical modeling. Cambridge University Press, Cambridge, pp 205–208
36. Said SE, Dickey DA (1984) Testing for unit roots in autoregressive-moving average models of unknown order. Biometrika 71(3):599–607
37. Motulsky H, Christopoulos A (2004) Fitting models to biological data using linear and nonlinear regression: a practical guide to curve fitting. England Oxford University Press, Oxford
38. Chakrabarti D, Faloutsos C (2002) Large-scale automated forecasting using fractals. In: Proceedings of the eleventh international conference on information and knowledge management
39. Page L, Brin S, Motwani R, Winograd T (1999) The PageRank citation ranking: bringing order to the Web. Technical Report Stanford InfoLab
40. Kleinberg JM (1999) Hubs, authorities, and communities. ACM Comput Surv 31(4):5
41. Gomez-Rodriguez M, Leskovec J, Scholkopf B (2013) Modeling information propagation with survival theory. The 30th international conference on machine learning
42. Wang SZ, Xie SH, Zhang XM, Li ZJ, Yu PS, and Shu XY (2014) Future influence ranking of scientific literature. In: 2014 SIAM international conference on data mining
43. Cui P, Wang F, Liu SW, Ou MD, Yang SQ (2011) Who should share what? Item-level social influence prediction for users and posts ranking. In: The 34th international ACM SIGIR conference on research and development in information retrieval
44. Dietterich TG (1998) Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput 10:1895–1923 (1998)
45. Wang SZ, Yan Z, Hu X, Yu PS, Li ZJ (2015) Burst time prediction in cascades. In: The twenty-ninth AAAI conference on artificial intelligence

Senzhang Wang received the M.S. degree in Southeast University, Nanjing, China in 2009. He currently is a fifth-year Ph.D. student in the School of Computer Science and Engineering at Beihang University, Beijing, China, and also a visit student in the Department of Computer Science at University of Illinois at Chicago, USA. His main research focus is on data mining, social computing, and urban computing.

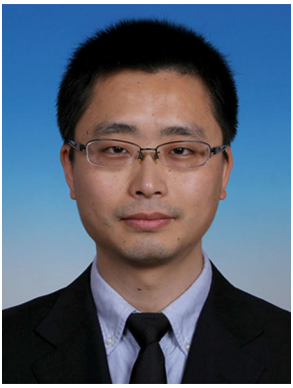
Zhao Yan received the B.S. degree in the School of Computer Science and Engineering from Beihang University in 2011. Currently, he is a Ph.D. candidate in Beihang University. His research interests include data mining and natural language processing.

Xia Hu is currently an assistant professor at the Department of Computer Science and Engineering, Texas A&M University. He obtained his Ph.D. in Computer Science and Engineering from Arizona State University, and M.S. and B.S. in Computer Science from Beihang University, China. His research interests are in data mining, social network analysis, machine learning, etc. As a result of his research work, he has published nearly 40 papers in several major academic venues, including WWW, SIGIR, KDD, WSDM, IJCAI, AAAI, CIKM, SDM, etc. One of his papers was selected in the Best Paper Shortlist in WSDM'13. He is the recipient of the 2014 ASUs Presidents Award for Innovation, and Faculty Emeriti Fellowship. He has served on program committees for several major conferences such as IJCAI, KDD and WWW, and reviewed for multiple journals, including IEEE TKDE, ACM TOIS and Neurocomputing.

Philip S. Yu is a professor in the Department of Computer Science at the University of Illinois at Chicago and also holds the Wexler Chair in information technology. He spent most of his career in the IBM Thomas J. Watson Research Center and was the manager of the Software Tools and Techniques Group. His research interests include data mining, Internet applications and technologies, database systems, multimedia systems, parallel and distributed processing, and performance modeling. He is a fellow of ACM and IEEE. He is the editor-in-chief of the ACM Transactions on Knowledge Discovery from Data.



Zhoujun Li received the B.S. degree in the School of Computer Science from Wuhan University in 1984, and the M.S. and Ph.D. degrees in the School of Computer Science from National University of Defense Technology. Currently, he is working as a professor of Beihang University. His research interests include data mining, information retrieval and information security. He is a member of the IEEE, ACM and AAAI.



Biao Wang received the B.S., M.S. and Ph.D. degree in the School of Mathematics from SiChuan University. Currently, he is working as an associate professor and the director of Network and Educational Center of University of International Relations. His research interests include information security, Cryptology and Smartphone Security. He is a senior member of the China Computer Federation (CCF).