Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Online detection of bursty events and their evolution in news streams*

Wei CHEN†, Chun CHEN, Li-jun ZHANG, Can WANG†‡, Jia-jun BU

(*Zhejiang Laboratory of Service Robot, Zhejiang University, Hangzhou 310027, China*)

†E-mail: {chenw, wcan}@zju.edu.cn

Received Apr. 29, 2009; Revision accepted Sept. 1, 2009; Crosschecked Apr. 9, 2010

Abstract: Online monitoring of temporally-sequenced news streams for interesting patterns and trends has gained popularity in the last decade. In this paper, we study a particular news stream monitoring task: timely detection of bursty events which have happened recently and discovery of their evolutionary patterns along the timeline. Here, a news stream is represented as feature streams of tens of thousands of features (i.e., keyword. Each news story consists of a set of keywords.). A bursty event therefore is composed of a group of bursty features, which show bursty rises in frequency as the related event emerges. In this paper, we give a formal definition to the above problem and present a solution with the following steps: (1) applying an online multi-resolution burst detection method to identify bursty features with different bursty durations within a recent time period; (2) clustering bursty features to form bursty events and associating each event with a power value which reflects its bursty level; (3) applying an information retrieval method based on cosine similarity to discover the event's evolution (i.e., highly related bursty events in history) along the timeline. We extensively evaluate the proposed methods on the Reuters Corpus Volume 1. Experimental results show that our methods can detect bursty events in a timely way and effectively discover their evolution. The power values used in our model not only measure event's bursty level or relative importance well at a certain time point but also show relative strengths of events along the same evolution.

Key words: Online event detection, Event's evolution, News stream, Affinity propagation
doi:10.1631/jzus.C0910245 **Document code:** A **CLC number:** TP391

1 Introduction

In recent decades the increasing number of electronically available news reports threatens to be overwhelming. This has led to a surge of research work in analyzing and utilizing news streams. Amongst these, the Topic Detection and Tracking (TDT) Community has been studying for a decade to give practical solutions for effectively monitoring news streams for important events (TDT Project, 2007). Detecting news events and summarizing their evolutions along the timeline will provide a conceptual structure for news stories in the news stream and greatly facilitate users navigation in news spaces. Furthermore, indexing and organizing news stories by events can help cluster and deliver personalized news stories, bringing better accessibility to the overwhelming amount of electronically available news.

In our previous work, we developed a personalized and phonic Web news recommendation system named EagleRadio, designed for blind and visually-impaired people. EagleRadio helps them access their daily amount of news on the Web with various types of smart terminals (Chen *et al*., 2008). Effectively detecting important news events and providing related news reports are extremely useful in such personalized news recommendation systems or other news retrieval systems. To better address users' needs in finding the new updates for these events and their related reports in history, both real-time detection of the events and discovery of their evolutions should be explored to more effectively present news stories by

‡ Corresponding author

* Project (No. 2008BAH26B00) supported by the National Key Technology R & D Program of China

events. Consider the example in Fig. 1, which shows the news events about Xiang Liu's injury in the 2008 Olympic Games and his recovery over time. After reading a report about Liu's return to China on '09/3/18', users probably want to retrieve related news reports in history. In such a case, a timeline-based event's evolution as shown in Fig. 1 would be a very informative summary of the events, which is also useful to have structured guidelines for users' navigation in the news space.

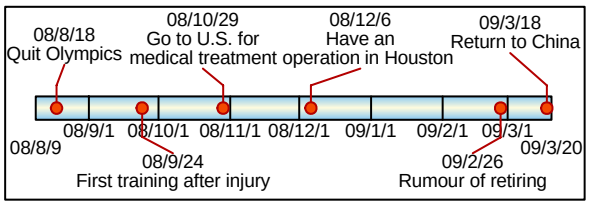

Fig. 1 News events about Xiang Liu's recovery after his quitting the Olympic Games

In addition to finding out real-time events and their evolution, measuring the events' bursty level or importance is also crucial, as there are currently often tens of thousands of news reports published every day. By analyzing the strength of events and ordering them accordingly, we can obtain the latest important news events (e.g., headlines) automatically. In addition, by evaluating the strengths of event's evolution along the timeline, we can easily discover the relative bursty level of related events which reveals the evolutionary trend of a specific event within a certain time period.

From the above discussion, we see an increasing need for online detection of important events and discovery of their evolution. Yet despite its importance, very few related works exist to address this issue. Most of the related studies are on clustering of documents corresponding to some real events (document sets manually identified) by decreasing the false alarms and the amount of missing documents (TDT Project, 2007), without much consideration to the temporal structures of the features. Our work in this paper is inspired by He *et al*. (2007) and Fung *et al*. (2005)'s work in bursty event detection, in which they detected a set of bursty features and grouped them together to reconstruct a bursty event. However, their work is about offline detection and the events' evolution is not studied. Section 2 will give the detailed discussion of related work.

In this paper, we study the online bursty events detection problem in a news stream. In addition, we address the issue discovering the evolutionary patterns of the detected bursty events along the timeline. We give a formal definition to the problem and present a solution with the following steps: (1) instantly identifying bursty features with different bursty periods in the current time window, (2) grouping online detected bursty features to form bursty events and associating each event with a power value representing its bursty level, and (3) discovering the events evolution and analyzing their bursty level along the timeline. We evaluate the proposed methods on Reuters Corpus Volume 1 (RCV 1)—a collection of 365-day (96/8/20–97/8/19) news reports from Reuters (Lewis *et al*., 2004). The results show that our method can effectively detect bursty events in real time and precisely rank these events with power values at a certain time point in time. The detected events and associated power values are consistent with the events that happened and their corresponding levels of importance in real life. The proposed evolution discovery approach is also effective. The powers of events in the same evolution also well model these events relative strengths over time.

The rest of the paper is organized as follows. Section 2 is the related work. In Section 3, we formally define the general problem of online bursty events detection and their evolution discovery. Section 4 presents our approach to identifying bursty features in real time. In Section 5, we give the method of online feature correlation analysis and further present an affinity propagation based method to perform online bursty event detection. An events evolution discovery method is introduced in Section 6. We discuss our experiments and results in Section 7. Finally, Section 8 concludes our work.

2 Related work

A news event is something that happens at a certain time in a certain place, which may be reported consecutively by many news reports over a period of time (TDT Project, 2007). In the TDT community, there are mainly two lines of research related to news event detection, i.e., retrospective news event detection (RED) (Yang *et al*., 1998; Li *et al*., 2005) and

new event detection (NED) (Allan *et al*., 1998; Yang *et al*., 1998; 2001; Lam *et al*., 2001; Kumaran and Allan, 2004; Zhang *et al*., 2007; 2008). RED detects previously unidentified events in a news corpus (Yang *et al*., 1998). NED is more related to our work, which detects news stories about previously unseen events in a stream of news stories.

The most popular approach of NED is based on the initial work of Yang *et al*. (1998) and Allan *et al*. (1998). In their work, each newly arrived report was compared to all the previously received ones. It became the first story of a new event if none of their similarities exceeded a threshold. There are various modifications to this approach. Some NED methods try to use name entities (such as person, organization, location, date, time, etc.) to improve accuracy (Lam *et al*., 2001; Yang *et al*., 2001; Kumaran and Allan, 2004). Events belonging to the same topic often share a set of keywords. These features are informative for discriminating on- and off-topic documents. Yang *et al*. (2001) and Kumaran and Allan (2004) classified documents into different categories firstly. And then specific stop words with respect to each category were removed. Their work showed significant improvements. In addition, to determine whether two reports of different classes belong to the same topic, different types of features (e.g., name entities and non-named terms) have different effects. Reweighting of terms is also widely used (Yang *et al*., 2001; Kumaran and Allan, 2004; Zhang *et al*., 2007; 2008), and contributes significantly to improvement in NED accuracy. For example, Yang *et al*. (2001) and Kumaran and Allan (2004) reweighted both named entities and non-named terms of each category. Zhang *et al*. (2007; 2008) introduced several reweighting approaches, such as adjusting term weights based on term distributions between the whole corpus and a cluster story set.

Recently, there has been significant interest in bursty event detection, which models an event in text streams as a bursty activity, with certain features rising sharply in frequency (i.e., bursty features) as the event emerges (Kleinberg, 2002; Fung *et al*., 2005; He *et al*., 2007). Different from traditional RED and NED methods, these methods group bursty features with identical trends to form events. There are two main issues related to bursty event detection, including bursty features identification and grouping bursty features into bursty events. Kleinberg (2002) modeled the stream and extracted bursty features using infinite-state automation. He *et al*. (2007) applied spectral analysis to categorize features for different event characteristics, important and less-reported, periodic and aperiodic. They modeled aperiodic features with Gaussian density and periodic features with Gaussian mixture densities. An unsupervised greedy event detection algorithm was used to detect both aperiodic and periodic events. Fung *et al*. (2005) also grouped bursty features to find bursty events. They identified a bursty feature by its distribution. Nevertheless, most of the bursty event detection methods are offline and belong to RED.

The online bursty detection in streams is well studied in data stream mining (Zhu and Shasha, 2002; Vlachos *et al*., 2004; Bulut and Singh, 2005; Yuan *et al*., 2007). Based on discrete Fourier transforms and a three-level time interval hierarchy, Zhu and Shasha (2002) monitored tens of thousands of time series data streams in an online fashion. Bursts may occur at variable temporal durations (maybe hours, days, or even weeks). Bulut and Singh (2005) proposed a multi-resolution indexing scheme to online discover meaningful behavior and monitor this over variable window sizes. Based on ratio aggregation pyramid (RAP) and slope pyramid (SP) data structure, Yuan *et al*. (2007)'s algorithm can also detect bursts in multiple window sizes. Vlachos *et al*. (2004) presented effective ways for identifying periodicities and bursts in the query logs of the MSN search engine.

In a way different from previous bursty events detection methods, bursty features are identified in real time with the help of data stream mining methods in our work. Accordingly, bursty events are detected in time by grouping bursty features and ranked by their bursty levels. Motivated by Mei and Zhai (2005) in summarizing the evolutionary patterns of themes in a text stream, we also discover events evolution over time.

3 Problem formulation

The general problem of online detection of bursty events and their evolution from a news stream is formulated in this section.

Given a stream of news stories $S=\{d_1, d_2, \ldots, d_i, \ldots\}$, where d_i is a document at time point t_i and

each document consists of a set of features in a vocabulary $F=\{f_1, f_2, \ldots, f_i, \ldots\}$, we treat the news stream S as tens of thousands of time series streams of features in F. The time series representation of a feature is defined as follows:

Definition 1 (Feature trail) The trail of feature f_i can be written as a discrete time series $f_i[1, 2, \ldots, t, \ldots]$, where each element $f_i[t]$ denotes the value of feature f_i at time point t (the unit of time point may be one hour, half a day or one day, etc.; one day in the following discussion), defined as the DFIDF score (He *et al*., 2007):

$$f_i[t] = \mathrm{DF}_i[t] \times \mathrm{IDF}_i = \frac{\mathrm{DN}_i[t]}{N[t]} \times \ln\left(\frac{N}{\mathrm{DN}_i}\right), \quad (1)$$

where $N[t]$ is the amount of documents at time point t while N is the amount of documents over the stream, $\mathrm{DN}_i[t]$ is the amount of documents including f_i at time point t, and DN_i is the amount of documents including f_i over the stream. We denote the subsequence of entries at time positions t_1 through t_2 as $f_i[t_1{:}t_2]$ in the following. Also t was used to denote the latest time point (i.e., now).

We can use a set of representative features to describe an event. Similarly, we have

Definition 2 (Bursty event) A bursty event is a minimal set of bursty features that occur together in a certain time window with a strong support of documents in the text stream (Fung *et al*., 2005). E.g., the event 'APEC forum' on '96/11/25' can be described by a few bursty features, such as 'APEC', 'forum', 'subic' and 'Philippines'. These features of the same bursty event share a similar bursty pattern and intersect highly in documents when the event happens.

Definition 3 (Bursty feature) A feature is identified as bursty within a certain window when the aggregate (here the sum) value of its feature trail within the window is much larger than the aggregate values in most other windows of the same size.

Therefore the key of online bursty events detection is to automatically indentify minimal sets of bursty features in the current time window. After identifying bursty events $E=\{e_1, e_2, \ldots, e_i, \ldots\}$ in the current window, we also want to find the closely related events (i.e., event's evolution) in history. We now define a particularly interesting concept called the "bursty event's evolutionary trail".

Definition 4 (Bursty event's evolutionary trail) The evolutionary trail of a bursty event e_i can also be written as discrete time series $e_i[1, 2, \ldots, t, \ldots]$. Each entry of the trail represents the relative strength of the event in the corresponding time point. The definition of 'entry' will be given in Section 6.

In the following three sections, we propose approaches to bursty features/events identification and event evolution discovery.

4 Online bursty feature identification

Formally, given an aggregate function G (here the sum), sliding windows of size w, and their corresponding thresholds $\gamma(w)$, the online bursty feature identification problem is to discover all these features such that the aggregate function G applied to feature trail $f[t-w+1{:}t]$ exceeds threshold $\gamma(w)$, i.e., check if

$$G(f[t-w+1{:}t]) \geq \gamma(w), \quad (2)$$

where t is the current time point. We use the historical data to obtain the threshold $\gamma(w)$. That is, we compute the aggregates within a sliding window of size w on the historical data (i.e., $G(f[t_i-w+1{:}t_i])$, $t_i<t$). Then, the threshold is set as $\gamma(w)=\mathrm{mean}(G(f[t_i-w+1{:}t_i]))+\varepsilon\times\mathrm{std}(G(f[t_i-w+1{:}t_i]))$, where mean(·) and std(·) are the average and standard deviation functions respectively (Vlachos *et al*., 2004). The complexity of obtaining $\gamma(w)$ is $O(L)$ for each feature, where L is the length of the feature trail to calculate the threshold. We set ε to 3 in our experiments.

If we know the bursty duration of a feature already, we can maintain the aggregate over the known window size and indicate whether it satisfies the above equation. However, we cannot predict the length of the feature's bursty duration. Some bursty features may span more than a dozen of days while others may last only a few days. Fig. 2 shows the feature trail of 'APEC' in RCV 1. The first bursty period of 'APEC' lasts more than ten days around '96/11/25', while the second bursty period around '97/4/5' lasts only five days.

Suppose we want to find all the bursty features with different bursty durations in the last W days. We need to detect bursts across multiple window sizes of each feature trail $f[t-W+1{:}t]$ (t is the current time point, and W is the maximum sliding window

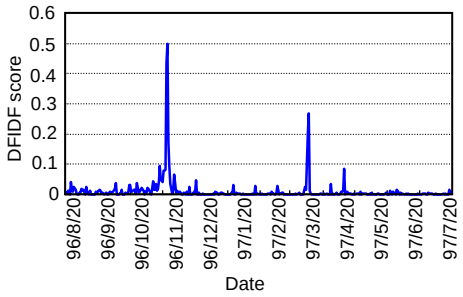

Fig. 2 Feature trail of 'APEC'

size by which we want to detect bursts). The naïve algorithm has to examine all the starting positions and window sizes in $f[t-W+1:t]$. It requires $O(kW)$ time (k is the number of windows) for each feature. It is very slow to detect bursts in streams with thousands of features.

We adopt a multi-resolution approach to detecting bursts in a feature trail. Multiple sliding windows with different resolutions are maintained to monitor the feature trail. The aggregate value of a specific resolution is maintained within a fixed size sliding window w. The size of the sliding window doubles as it goes up a resolution. Take Fig. 3 as an example. There are four resolutions with w=2, 4, 8, and 16. We maintain aggregate values for each resolution and update them when a new stream value arrives. The aggregate value of resolution i ($i\geq 1$, and the smallest sliding window is denoted as resolution 1) is represented as AGG[i] in our following discussion. We use the data structure shown in Fig. 3 to detect bursts in a feature trail.

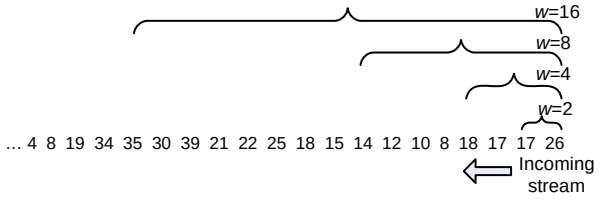

Fig. 3 Multi-resolution sliding windows over a data stream

4.1 Online multi-resolution burst detection

To find all the burst in $f[t-W+1:t]$, we need to maintain aggregate values of resolution 1 to $\lceil \log_2 W \rceil$, i.e., AGG[1, 2, ..., $\lceil \log_2 W \rceil$]. In addition, all the entries in the feature trail within a window equal to the largest sliding window in size, i.e., $2^{\lceil \log_2 W \rceil}$, should be maintained. Let $w_{\max} = 2^{\lceil \log_2 W \rceil}$ and the maintained feature trail be $f[t-w_{\max}+1:t]$. The online multi-resolution burst detection (denoted as OMRBD) algorithm will return the time series tag[$t-w_{\max}+1:t$], which denotes the bursty point of $f[t-w_{\max}+1:t]$. The tag entry is 1 for burst; otherwise it is 0. A feature is identified as bursty if any tag entry is set. The pseudo-code of the OMRBD algorithm is given in Fig. 4.

Input: $f[t-w_{\max}+1:t]$, W, AGG[1, 2, ..., $\lceil \log_2 W \rceil$] and threshold γ for each sliding window.
Output: tag[$t-w_{\max}+1:t$].
1 FOR $w=W$ TO 1
2 IF AGG[$2^{\lceil \log_2 w \rceil}$] $<\gamma(w)$
3 $w=w-1$;
4 ELSE
5 IF sum($f[t-w+1:t]$) $\geq\gamma(w)$
6 tag[$t-w+1:t$]=1;
7 BREAK;
8 END IF
9 END IF
10 END FOR

Fig. 4 Online multi-resolution burst detection algorithm

Because the aggregate sum of a non-negative feature trail is monotonically increasing, the sum of the feature trail within a sliding window of size w is bounded by the sum of its inclusive resolution with size $2^{\lceil \log_2 w \rceil}$. Thus, those subsequences whose sums are far below their thresholds can be eliminated (line 2 in Fig. 4).

The job of the OMRBD algorithm is to discover whether the old sequence plus the newly incoming entry (i.e., the entry at time point t) is burst in multiple windows. This is done by checking the sum of $f[t-w+1:t]$ (line 5). If the sequence in the sliding window w is identified as burst, we can stop OMRBD immediately. Because the feature trail $f[t-w+1:t]$ is denoted as burst and all further sliding windows that need to be checked is inside the current window w.

After a new data point of a feature from the stream becomes available, we need to update the feature trail, tag, and aggregate the value of each resolution, i.e., AGG[1, 2, ..., $\lceil \log_2 W \rceil$]. All the

values of the feature trail $f[t-w_{\max}+1:t]$ and tag$[t-w_{\max}+1:t]$ are shifted left by one point and the oldest value is eliminated. The value of $f[t]$ is replaced by newly incoming data while tag is updated using the OMRBD algorithm. AGG[1, 2, ..., $\lceil \log_2{}^{W} \rceil$] should be recomputed by adding the new data point while removing the oldest data in a corresponding resolution. The time complexity of the OMRBD algorithm is $O(W)$.

5 Online bursty events detection

After the OMRBD algorithm is applied to all the feature trails in the current window of size W, all bursty features and their bursty period can be identified. The next task is to analyze the correlation between bursty features, which corresponds to the probability of two features forming an event. Then by grouping highly correlated bursty features together, the bursty events on the current window can be detected.

5.1 Online feature correlation analysis

If features f_i and f_j belong to the same bursty event at present, they must meet the necessary conditions in the current window as follows (He *et al*., 2007):

1. Feature trails $f_i[t-W+1:t]$ and $f_j[t-W+1:t]$ have very similar bursty patterns.

2. Features f_i and f_j have a high document intersection in the current window of size W.

5.1.1 Measuring the similarity of bursty features

There are many ways to measure the similarity of two time sequences, such as the cosine similarity, Kullback-Leibler (KL) divergence, the Euclidean distance and its extensions to support various transformations (Chu and Wong, 1999; Kahveci and Singh, 2001). In our work, we use the cosine similarity to measure the similarity of two bursty feature trails. The feature trail $f[t-W+1:t]$ is treated as a vector. Each entry of the feature trail is a weight, corresponding to the DFIDF score. That is, we obtain the bursty similarity of feature trails $f_i[t-W+1:t]$ and $f_j[t-W+1:t]$ by the cosine of the angle between them. However, different features have different background distributions. For example, the feature 'April' has a much higher probability of occurrence than the rare feature 'zywnosciowej'. We define it as the average of the DFIDF score in the historical feature trail, denoted as mean(f[:]). The background information should be subtracted from the original feature trail before the cosine similarities are calculated.

In addition, we should give more weight on the more recently time point of the feature trail for online bursty event detection. In our work, we introduce an exponential decay vector $h[1:W]$ from the current to the earlier time point. It is defined as

$$h[i]=\mathrm{e}^{-(i-1)/\tau},\quad 1\le i\le W,\quad \tau=t_{1/2}/\ln 2, \tag{3}$$

where τ is the mean lifetime of news reports. $t_{1/2}$ is the half-life time. Dezso *et al*. (2006) reported that the half-life time of online news stories are typically 36 h (i.e., $t_{1/2}$=1.5). In our work, we set τ=2.

Let $f'[i]=f[t-W+i]-\mathrm{mean}(f[:])$, $1\le i\le W$. Then, we obtain the following definition:

Definition 5 (Updated feature trail)

$$f^*[t-W+i]=\begin{cases} f'[i]\times h[W-i+1], & f'[i]>0,\\ 0, & f'[i]<0. \end{cases} \tag{4}$$

Finally, we define the similarity of two bursty feature trails as

$$\mathrm{SIM}(f_i,f_j)=\frac{\sum_{k=t-W+1}^{t} f_i^*[k]\times f_j^*[k]}{\sqrt{\left(\sum_{k=t-W+1}^{t}(f_i^*[k])^2\right)\left(\sum_{k=t-W+1}^{t}(f_j^*[k])^2\right)}}. \tag{5}$$

To obtain the similarity of all bursty features, the time complexity is $O(W\times N^2)$, where N is the number of bursty features at time point t.

5.1.2 Measuring document intersection

Given two features f_i and f_j, we define D_i and D_j as the sets of all documents containing f_i and f_j respectively at the time point when both of them are bursty (at this time point both tag_i and tag_j are set) in the current window. Then $D_i\cap D_j$ is the intersecting document set including features f_i and f_j. The document intersection of two bursty features can be defined as (Vlachos *et al*., 2004)

$$\mathrm{INT}(f_i,f_j)=\frac{1}{2}\left(\frac{|D_i\cap D_j|}{|D_i|}+\frac{|D_i\cap D_j|}{|D_j|}\right). \tag{6}$$

Obviously, the larger the INT(f_i, f_j), the more likely that features f_i and f_j are highly correlated. Obtaining the document intersection is time consuming; it requires $O(|D_i|\times|D_j|)$ time for features f_i and f_j. Thus, the time complexity of document intersection is $O(N^2\times|D|^2)$, where N is the number of bursty features and $|D|$ is the average document length. Then the correlation of bursty features in the current sliding window W can be calculated as

$$\mathrm{COR}(f_i, f_j) = \mathrm{SIM}(f_i, f_j) \times \mathrm{INT}(f_i, f_j). \quad (7)$$

5.2 Bursty event detection: an affinity propagation approach

Intuitively, bursty events can be detected from a list of highly correlated features using unsupervised clustering methods. We find that there are usually several thousand of bursty features in each window with size w of RCV 1. However, most bursty features are not correlated; i.e., the correlation value is zero. Besides, we do not know how many bursty events happened previously at each time point. Therefore classical techniques for clustering such as k-means and spectral clustering (Luxburg, 2007) which need to specify the number of clusters are not very suitable for this application.

Recently a new clustering algorithm called affinity propagation has been proposed in *Science*, which efficiently clusters sparsely related data by passing messages between data points (Frey and Dueck, 2007; Xia *et al*., 2008). Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges (Frey and Dueck, 2007). Affinity propagation takes input as a collection of real-valued correlations between data points, corresponding to correlations between bursty features in our work. Rather than pre-specifying the number of clusters, affinity propagation requires a preference value for each data point. The point with a larger preference is more likely to be chosen as exemplars. The amount of identified exemplars (the same as the cluster number) is influenced by the input of preferences, but also emerges from the message-passing procedure (Frey and Dueck, 2007). In our work, we applied an affinity propagation clustering method to group events from bursty features. The time complexity of affinity propagation is $O(N^2\log_2 N)$, where N is the number of data points, corresponding to the number of bursty features here. Before going on, we give the following definitions.

Given a bursty feature f, its updated feature trail $f^*[t-W+1{:}t]$, and bursty tag$[t-W+1{:}t]$ obtained using the OMRBD algorithm, we have

Definition 6 (Feature power) The power of a feature is the sum of bursty entries in the updated trail of that feature $f^*[t-W+1{:}t]$:

$$\mathrm{POW}(f) = \mathrm{sum}(f^*[t-W+1{:}t] \cdot \mathrm{tag}[t-W+1{:}t]), \quad (8)$$

where '$\cdot$' is the dot product operation. The power of the feature represents its bursty level and importance in the current window W. A feature with a higher power is more likely to be the representative and descriptive word of a bursty event, i.e., exemplar of a bursty event. Thus, in our experiments we set the preferences of affinity propagation as the power of the features.

Given a bursty event e which is composed of a group of bursty features C, obtained by affinity propagation, we have

Definition 7 (Event power) The power of the bursty event e is

$$\mathrm{POW}(e) = \underset{f_i \in C}{\mathrm{mean}}(\mathrm{POW}(f_i) \times \mathrm{COR}(f_i, f_{\mathrm{exemplar}})), \quad (9)$$

where f_{exemplar} represents the exemplar of the bursty features set C, and $\mathrm{COR}(f_{\mathrm{exemplar}}, f_{\mathrm{exemplar}})=1$.

The power of an event shows its bursty level in its corresponding window. Thus, we can use the power to rank the events in a specific window. The top ranked events represent important bursty events happening at a corresponding time.

6 Event evolution discovery

Supposing we have detected the bursty events set $E=\{e_1, e_2, \ldots, e_i, \ldots\}$ at the current window, our goal here is to discover the evolution of interesting events in history.

In our work, we use an information retrieval method based on the cosine similarity to discover events evolution (Baeza-Yates and Ribeiro-Neto, 2004; Croft *et al*., 2009). Bursty events detected at a historical window are considered as documents, while

events at the current window are considered as queries. The corresponding bursty features of events are regarded as index terms of the documents/queries. A vector space model is used to represent documents and queries. That is, we assume documents and queries to be part of a z-dimensional vector space, where z is the number of index terms. We define a query as $q=\{w_{1,q}, w_{2,q}, \ldots, w_{z,q}\}$ and a document j as $d_j=\{w_{1,j}, w_{2,j}, \ldots, w_{z,j}\}$. Both weights of the queries and documents are defined as the power of features at the corresponding window. The cosine similarity is used to quantify the similarity of queries and documents, which is equivalent to the probability of queries and documents belonging to the same evolution. That is,

$$P(d_j, q) = \frac{\sum_{i=1}^{z} w_{i,j} \times w_{i,q}}{\sqrt{\left(\sum_{i=1}^{z} w_{i,j}^2\right)\left(\sum_{i=1}^{z} w_{i,q}^2\right)}}. \quad (10)$$

Suppose we want to find the events evolution of a specific event e_i (i.e., q in Eq. (10)) at present. We check every detected bursty event (i.e., d_j in Eq. (10)) in the historical window. Only those events d_j that satisfy $P(d_j, q)>\eta$ (η is an empirical tunable value) are returned and each window can return only at most one event with the largest $P(d_j, q)$. Then, we can find these events closely related to event e_i in history.

Finally, we conclude Definition 4 of bursty events evolutionary trail $e_i[1, 2, \ldots, t]$. Each entry of the trail is

$$e_i[k] = \mathrm{POW}(e_{i,k}), \quad 1 \le k \le t, \quad (11)$$

where $e_{i,k}$ represents the event at time point k in the evolution of event e_i.

7 Experimental studies

In this section, we report experimental studies based on a real news corpus. We introduce experimental setup and dataset first, and then evaluate the online multi-resolution burst detection algorithm. The bursty events detection method is studied next. Finally, we give examples of the events evolution discovery.

7.1 Data preparation

RCV 1 (Lewis *et al*., 2004) was used to evaluate the proposed methods. RCV 1 is an archive of 806 791 manually categorized newswire stories made available by Reuters Ltd. for research purposes. These news stories are from '96/8/20' to '97/8/19' (totally 365 days) using a daily resolution. The SimpleAnalyzer of open source full text indexing and searching toolkit Lucene 2.4.0 (Lucene Project, http://lucene.apache.org) was used to tokenize the news corpus, without removing a stopword or stemming. Only the text in <title> and <text> fields was processed. Finally, we obtained 378 454 features. We implemented all experiments in Matlab and Java. These experiments were carried out on a 3.16 GHz Intel Core 2 Duo PC running Windows Vista with 4 GB of memory.

7.2 Bursty feature detection

The OMRBD algorithm was used to detect bursty features with the maximum sliding window of size W=16, 12, 8, and 4, respectively. In our previous discussion, we need historical information to set parameters, such as the threshold value $\gamma(w)$ and the IDF value in the DFIDF score. Since there were very few historical data in the beginning month, all these parameters in the first month were set based on the whole corpus.

Fig. 5 shows the number of bursty features at each sliding window from '96/9/4' to '97/8/19' (totally 350 time points) by the OMRBD algorithm with the size of the maximum sliding window W=16, 12, 8, and 4, respectively. The numbers of detected bursty features at different time points differ. Take the case W=16. For example, the maximum number is 8736 on '97/1/5' while the minimum is only 4438 on '97/3/14'. In fact, there were many significant bursty events happening around the beginning of 1997, such as the famous 'Japanese embassy hostage crisis at Peru'. The number of detected bursty features is approximately proportional to the size of the sliding window. For example, the number in the case W=16 is nearly double that in the case W=8. We can further find that there are exactly 50 distinguishing peaks/troughs in the figure, which exhibits strong weekly periodicities. Possibly due to the periodicity of news reports numbers, typically there are around 3000 per day on weekdays while it is 500 at weekends.

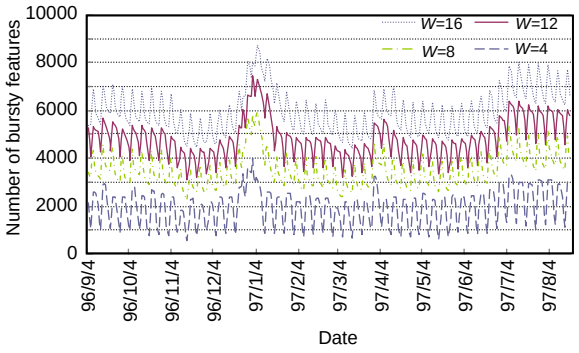

Fig. 5 Number of bursty features at each sliding window detected by OMRBD with *W*=16, 12, 8, and 4 respectively from '96/9/4' to '97/8/19'

Fig. 6 shows the power summation of the top 5/10/20 powerful bursty features detected using the OMRBD algorithm with *W*=16 from '96/9/4' to '97/8/19', which also exhibit strong weekly periodicities. The bursty features with the highest power usually reflect the most important bursty events or topics at the corresponding time. For example, the topmost bursty feature is 'iraq' from '96/9/4' to '96/9/10', 'thanksgiving' from '96/11/27' to '96/11/30', 'Christmas' from '96/12/23' to '96/12/30', 'deng' from '97/2/20' to '97/2/24' (paramount leader Deng Xiaoping of China died on '97/2/19'), etc. We find that there are many monthly related features (i.e., April, May, August, etc.) at the top of the list. Since an event usually identifies something happening at a certain time and monthly related features appear in these reports, the top powerful bursty features detected using the OMRBD algorithm with different sliding window sizes show similar results.

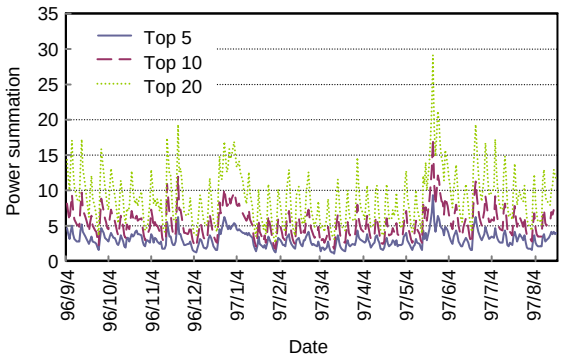

Fig. 6 The power summation of top 5/10/20 powerful bursty features detected using OMRBD with *W*=16 at each sliding window

The power distribution of all 6552 bursty features detected with *W*=16 on Christmas day is shown in Fig. 7. The distribution is much skewed and it is similar in other sliding windows. Only 32 top powerful bursy features have a power larger than 0.5. The top 400 powerful bursty features take up 50% of the total power of all bursty features in Fig. 7, which are the most descriptive and representative features at that time.

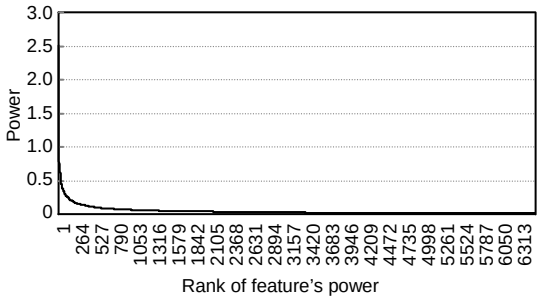

Fig. 7 The power distribution of 6552 bursty features detected using OMRBD with *W*=16 on '96/12/25'

7.3 Bursty event detection

After obtaining the bursty features using the OMRBD algorithm at the current window, we analyze the similarity of bursty patterns and document intersection between bursty features. As discussed in a previous section, bursty features belonging to the same bursty event have very similar bursty distributions. From Fig. 8 we can find that 'boeing', 'douglas', and 'merger' exhibit similar behavior from '96/11/30' to '96/12/15'. In addition, they have a high document intersection, INT('boeing', 'merger')=0.81, INT('douglas', 'merger')=0.84, INT('boeing', 'douglas')=0.43. In fact, three bursty features concern the 'Merger of Boeing and McDonnell Douglas', which created the world's largest aerospace company.

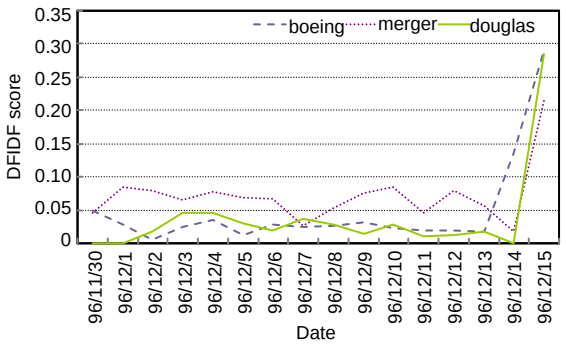

Fig. 8 The feature trail of 'boeing', 'merger', and 'douglas' from '96/11/30' to '96/12/15'

To evaluate how a different sliding window size affects the bursty events detected using our method, we firstly identified bursty features using the OMRBD algorithm with maximum sliding window size W=16, 12, 8, and 4, respectively. Then bursty events were formed by the affinity propagation approach. Table 1 shows three randomly selected bursty events with the bursty features obtained with W=16, 12, 8, and 4 respectively on '97/7/1'. The affinity propagation clustering method produced an exemplar for each event, which is listed first in bold in the table. The other bursty features in the table of the same event following exemplar are ordered by their correlation to the exemplar. We manually assign the description of detected events in the table.

Event 1 is about "Hong Kong's handover to China". We can see that the number of bursty features of Event 1 increased as the sliding windows size W decreased. In fact, the number of news reports about this event increased gradually before '97/7/1'. Correspondingly, the number of bursty features about the event "Hong Kong's handover" also increased. In the meanwhile, a few noise words, such as 'portrait', 'glimpse', 'yi', and 'licensee', were grouped into the feature lists of the detected bursty events with smaller sliding window sizes. Event 2 is about "Mike Tyson's disqualification after three rounds for twice biting Evander Holyfield's ears in the match for the World Boxing Association heavyweight title" on '97/6/28'. The OMRBD algorithm with the maximum sliding window size of 4 was enough for discovering all the bursty features related to the event. The detected bursty features about Event 2 of different sliding windows were nearly the same. Event 3 lasted a few months before '97/7/1', which is about 'general election Albania'. Most of the bursty features were the same in different window sizes.

In most cases, a sliding window of W=4 is adequate as most bursty events last only a few days. However, the measure of the feature trail's bursty similarity with a larger sliding window is more stable. In fact, we found that the detected bursty features of a specific event in larger windows were more general and closely related to the corresponding events in Table 1. Thus, we show only the results of W=16 in the following discussion.

The top 5 powerful bursty events on selected days (i.e., '96/11/24'–'96/11/26', '96/12/15'–'96/12/16', and '96/12/18'–'96/12/19') obtained using the proposed methods are listed in Table 2. With limited space, we do not list all bursty features of each event. Most exemplars of events in Table 2 are very representative and informative, such as 'airborne', 'lukashenko', 'apec', 'erm', 'merger', and 'boeing'. From the exemplar, we can discover approximately what the event concerns approximately. The detailed description of each event is manually assigned and listed in Table 3.

Table 1 Three random bursty events formed by bursty features obtained with W=16, 12, 8, 4 respectively on '97/07/01'

Event	Events formed by bursty features				Event description
	W=16	W=12	W=8	W=4	
1	**kong**, hong, handover, midnight, colony, sovereignty, pattern, reverts, mainland, garrison, fireworks, reversion	**kong**, hong, handover, colony, sovereignty, pattern, tiananmen, reverts, mainland, freedoms, farewell, garrison, fireworks, humiliation, reversion	**kong**, hong, handover, midnight, territory, colony, sovereignty, tiananmen, reverts, freedoms, mainland, farewell, ceremonies, motherland, garrison, fireworks, humiliation, reversion, portrait, yi	**kong**, hong, handover, china, midnight, colonial, beijing, territory, colony, sovereignty, tiananmen, autonomy, freedoms, reverts, mainland, peng, capitalist, sino, ceremonies, reunification, motherland, prosperity, opium, garrison, shenzhen, fireworks, barren, portrait, glimpse, yi, licensee	Hong Kong's handover to China
2	**tyson**, evander, boxing, biting, disqualified, ears, boxer	**tyson**, evander, boxing, biting, disqualified, ears, ear, boxer	**tyson**, evander, boxing, biting, disqualified, ears, ear, boxer	**tyson**, evander, boxing, biting, disqualified, ears, heavyweight, ear, boxer	Mike Tyson bit Evander Holyfield's ear in a heavyweight match
3	**albania**, berisha, sali, tirana, observers, fatos, albanians	**sali**, berisha, democratic, albania, fatos, albanians, nano, tirana, celebratory	**sali**, berisha, democratic, albania, fatos, albanians, nano, tirana, ballot, celebratory	**sali**, berisha, democratic, albania, fatos, albanians, nano, tirana, ballot, monarchy	Albania's general election

W is the maximum window size

Table 2 Top five powerful events on '96/11/24'–'96/11/26', '96/12/15'–'96/12/ 16', and '96/12/18'–'96/12/19' extracted from RCV1

Date & ID	Event 1	Event 2	Event 3	Event 4	Event 5
96/11/24 (I)	**airborne**, crew, bodies, ocean, transport, problems, crash, traffic, rescue, land, accident, evening, rescued, portland, seattle, electrical, controllers, …(8)	**mechanism**, erm, lira, ministers, Italian, committee, eu, join	**lukashenko,** referendum, belarus, Alexander, minsk, belarussian, extending	**apec** , subic, philippines, brunei	**maskhadov**, aslan, chernomyrdin, viktor, accord, Moscow, brigades, breakaway, describing, pipelines, importantly, guarantees, mulling
96/11/25 (II)	**parent**, specified, div, billions, eps, unless, yen, nippon	**apec**, forum, subic, philippines, ramos, computers	**mechanism**, erm, lira, entry, eu	**lukashenko**, referendum, belarus, powers, alexander, minsk, extending, belarussian	**airborne**, plane, crew, bodies, crash, ocean, traffic, contact, accident, problems, rescue, land, evening, portland, rescued, electrical, seattle, controllers, …(8)
96/11/26 (III)	**div**, eps, specified, parent, billions, unless, yen, ord, interim, kogyo	**erm**, lira, mechanism, entry, parity	**apec**, forum, subic, philippines	**clarke**, kenneth, chancellor, exchequer	**airborne**, traffic, crashed, contact, bodies, accident, crash, rescue, problems, portland, land, evening, rescued, electrical, controllers, seattle, sgt, …(8)
96/12/15 (IV)	**merger**, boeing, mcdonnell, douglas, aerospace, aircraft, phil, airbus, lockheed, tran, ual, commercial	**financially**, sectors, niche, barrier, marine, cancer, tougher, blow, insurers, assurance, thrilled, uninsured, stages, insurer, …(7)	**nationale**, partners, aerospatiale, consortium, europeans, arguments, employing, marietta, daimler, …(6)	**slobodan**, milosevic, zajedno, serbia, serbian, carnival, gestures, inviting, waved, pressure, accuse	**untrue**, diverting, grant, settlers, territories, anticipating, inaccurate, elements, ignite, pursued, destroy, troubling, settle, martin, son
96/12/16 (V)	**boeing**, merger, mcdonnell, douglas, daimler, cnbc	**financially**, huge, victims, matter, sectors, cancer, marine, niche, barrier, insurers, assurance, insurer, blow, …(8)	**sncf,** contractor, europeans, louis, casa, weston, nationale, subcontractors, employing, employers, irony	**slobodan**, milosevic, serbia, zajedno, serbian, inviting, provoking, pressure, pelted	**chaotic**, competition, reform, leon, father, priest, delays, trickle, bongo, multiparty, libreville, legislative, antoine, disciplined, founding
96/12/18 (VI)	**ambassador**, hostages, peru, lima, residence, peruvian, amaru, tupac, hostage, alberto, mrta, reception, inspired, fujimori, comrades, emperor, captives	**mcdonnell**, boeing, douglas, merger, daimler, cnbc	**financially**, huge, victims, sectors, cancer, marine, accident, niche, foothold, barrier, insurers, assurance, insurer, blow, …(9)	**christmas**, holidays, festive	**warsaw**, zlotys, polish, preliminary
96/12/19 (VII)	**japanese**, ambassador, hostages, lima, residence, peru, amaru, tupac, mrta, jailed, peruvian, hostage, comrades, stormed, reception, emperor, captives, captive, traps	**christmas**, holidays, festive	**mcdonnell**, boeing, douglas, daimler, dasa, cnbc	**financially**, huge, victims, sectors, cancer, marine, accident, niche, barrier, relieved, insurers, assurance, insurer, blow, thrilled, decade, uninsured, …(6)	**polish**, zlotys, warsaw, preliminary

The number of unlisted features is given in the brackets after '…'

Table 3 Description of events and their power corresponding to Table 2

ID	Description of events	Power
I	1. U.S. authorities stepped up search for survivors (a plane crashed on '11/22' night after the crew reported engine/electrical problems)	1.49
	2. Italian lira re-entered the exchange rate mechanism (ERM) on '11/24'	1.40
	3. Belarus entered a referendum vote on '11/24' (President is Alexander Lukashenko)	1.36
	4. Annual summit of Asia-Pacific Economic Cooperation (APEC) forum will be held on '11/25'	1.33
	5. Chechen rebel leader (Maskhadov Aslan) signed a weekend peace deal with Moscow on '11/24'	1.02
II	1. 96/97 or 6-month parent results/parent forecast of dozens of Nippon companies.	1.76
	2. Fourth annual summit of Asia-Pacific Economic Cooperation (APEC) forum was held at Subic, Philippines on '11/25'	1.50
	3. Follow-up reports of I-2	1.22
	4. Follow-up reports of I-3	1.18
	5. Follow-up reports of I-1	1.14
III	1. The same to II-1	1.59
	2. Follow-up reports of I-2	1.12
	3. Follow-up reports of I-4	0.98
	4. UK Chancellor of the Exchequer Kenneth Clarke presented his 97/98 budget on '11/26'	0.89
	5. The same as I-1	0.76
IV	1. Merger of Boeing and McDonnell Douglas	1.58
	2. U.S.-Japan insurance pact	1.48
	3. Airbus felt heat from Boeing-MD merger	1.01
	4. 250 000 people marched through Belgrade protesting against election fraud (Slobodan Milosevic VS Zajedno)	0.93
	5. Netanyahu sends envoy to Arafat	0.91
V	1. Follow-up reports of IV-1	1.29
	2. Follow-up reports of IV-2	1.23
	3. Follow-up reports of IV-3	0.83
	4. Follow-up reports of IV-4	0.69
	5. Gabon experienced fresh electoral confusion	0.66
VI	1. Japanese embassy hostage crisis in Lima, Peru	0.96
	2. Follow-up reports of IV-1	0.73
	3. Follow-up reports of IV-2	0.63
	4. News about Christmas day	0.60
	5. Publishing of economic indicator of hundreds of polish companies	0.54
VII	1. Japanese embassy hostage crisis	1.55
	2. News about Christmas day	0.55
	3. Follow-up reports of IV-1	0.43
	4. Follow-up reports of IV-2	0.43
	5. Follow-up reports of VI-5	0.42

The proposed method in our paper can effectively detect bursty events in time. All the bursty events listed in Table 2 were discovered in a timely fashion after they occurred. For example, our method detected the explosive news of 'Japanese embassy hostage crisis' on '96/12/18' just after 14 MRTA (Tupac Amaru Revolutionary Movement) members occupied the Japanese Ambassador's residence in Lima in the evening of '96/12/17'. This is mainly due to the use of the updated feature trail instead of the original feature trail to obtain the similarity of bursty pattern. The exponential decay vector introduced in the updated feature trail gives much more weight to more recent entries of the feature trail.

Most news in Table 2 is about politics. However, there are also exceptions. We found that the topmost powerful news on '96/11/25' and '96/11/26' was about '96/97' or 6-month parent results/parent forecast of Nippon companies. As a matter of fact, in that week and before, thousands of parent results/parent forecast reports were announced. Since Reuters did not publish these reports at the weekend, we did not find any news about it on '96/11/24' (Sunday).

The power information can well model the relative strengths of events. Each event's power is listed in Table 3. It is not difficult to find that the power of events at a specific day effectively exhibited the relative importance of the events on that day. In addition, they also took effect along the timeline. For example, the power of 'APEC forum' was 1.33 (event I-4), 1.50 (event II-2), and 0.98 (event III-3) from '96/11/24' to '96/11/26'. In the corpus, there were many more news reports about 'APEC' on '96/11/25' than on the other two days. In fact, the fourth annual summit of the Asia-Pacific Economic Cooperation (APEC) forum was held at Subic, Philippines on '96/11/25'. Take 'Japanese embassy hostage crisis' as another example. It broke out on '96/12/17' and there was no news about it before that day. The event was so explosive that it became the headline with the highest power on '96/12/18' just after it happened. Actually, it appeared in the headlines for more than ten days.

We further compared our proposed online bursty events detection approach (abbreviated as OBED) with He *et al*. (2007)'s work (abbreviated as HE), which were also evaluated on RCV 1. HE analyzed the feature trail in both time and frequency domains. They used spectral analysis to categorize features for different characteristics, important and less-reported,

periodic and aperiodic, so their methods can detect important/less reported periodic/aperiodic events. The events detected by OBED are usually bursty and important. Thus, we list the bursty events detected by OBED corresponding to the 17 important aperiodic events in He *et al*. (2007) in Table 4. In the column of 'HE', we list the events and their periods from He *et al*. (2007). The first bursty event in the timeline ('96/9/4'–'97/8/19') detected using our methods including the same or nearly the same features of HE is listed in column 'OBED', followed by the date on which our method detected it. The event description was also manually assigned.

In He *et al*. (2007), spectral analysis was applied to the whole line (i.e., one year RCV1) of the feature trail to identify the feature's characteristics (i.e., important and less-reported, periodic and aperiodic). These characteristics reflect the behavior of features in a year span, and have very coarse granularity. Our work differs from their work in that we analyze the features in a sliding window of size W (i.e., 16, 12, 8, and 4 days), which can reflect in a timely way the features' behavior in the timeline. In the experiments, we found that OBED detected the events in a more timely way and the detection time was more accurate than HE's. The detection time of 9 events in column OBED was much earlier than the start time of the corresponding period in column HE in Table 4. We believe that HE may have made some mistakes in events 3, 4, 10, and 16. There were no news reports about these events at the start of their period. Because our experiments were carried out from '96/9/4' to '97/8/19', the detection time of events 7 and 8 in column OBED was '96/9/4'. The bursty features of detected events of OBED were richer and had finer granularity than HE's in most events. We believe that our method is better for online bursty events detection while HE is fit for analyzing a whole corpus to discover the event list.

OBED first identifies bursty features, and then measures the similarity of the bursty features trail and document intersection between bursty features. Finally it groups bursty features to form bursty events. As we indicated in a previous section, the time complexity of achieving document intersection is the highest, which is $O(N^2\times|D|^2)$. Fortunately, the number of bursty features, N, in each time point is not very large, typically several thousand (Fig. 5). In our experiments, it took only 16.3 min on average to obtain document intersection on each day. Other steps were very fast. The average time to detect bursty events for each day was 20.2 min. OBED was highly efficient compared to other event detection methods. For example, HE took 742 min to obtain less-reported aperiodic events from low power and high periodicity features. In addition, they did not report how much time was taken to categorize these features' characteristics (i.e., important and less-reported, periodic and aperiodic) (He *et al*., 2007). In fact, their method is not incremental. Because step spectral analysis is based on the whole corpus, their method should repeat all the steps (firstly applying spectral analysis to identify features' characteristics) to detect events for any incoming news to the corpus.

7.4 Event evolution discovery

In this subsection, we illustrate the event evolution using the 'Japanese embassy hostage crisis' example. After finding an event about 'hostage crisis' (including features hostages, raid, ambassador, amaru, tupac, mrta, peruvian, hostage, revolutionary, movement, japanese, peru, lima, fujimori, alberto) on '97/4/25', the proposed event evolution discovery method can effectively discover related events in history. We list only the discovered major events related to crisis for space limit in Table 5. The crisis started on '96/12/17' and ended when all hostages were freed on '97/4/22'. In this period, there were thousands of related news items about the 'MRTA', 'Peru/Japan government', "MRTA's demands", 'Red Cross', 'Negotiations', 'military solution', etc.

The power values associated with events in the same evolution can well reflect events' strengths along the timeline. In Fig. 9, we plot the power of event 'Japanese embassy hostage crisis' over time. It received much attention gradually after it occurred. The power reached the first peak on '96/12/22' after President Fujimori made his first public announcement about the hostage crisis. On '96/12/26', a loud explosion was heard from ambassador's residence, resulting in the second peak. The armed forces abruptly stormed the Japanese ambassador's residence and rescued the remaining hostages at 4 p.m. '97/4/22', which led to another power peak on '97/4/23'. In other words, the proposed method provided topic visualization from a historical perspective. We can easily discover the media attention of a specific event's evolution or topic along the timeline.

Table 4 Comparison between our proposed online bursty events detection (OBED) approach and He *et al*. (2007)'s work

ID	HE (event and period)	OBED (event and detection date)	Event description	Remark
1	sali, berisha, albania, albanian, march 97/2/2–97/5/29	tirana, schemes, pyramid, sali, berisha, albanians, balkan, hall, heirs, thieves, fury, depositors, escalated 97/1/26	Protesters enraged at losing their life savings in pyramid investment schemes, and clashed with police in Tirana as protests across Albania grew on '97/1/26'	There are a lot of news reports about the unstable political situation in Albania from late January 1997 in RCV1
2	seko, mobutu, sese, kabila 97/3/22–97/6/9	seko, mobutu, sese, zairean, kinshasa, river, mineral, laurent, soldiers, ragged 97/3/15	Rebels seized the heavily defended airport at Zaire's third city of Kisangani on '97/3/15'	Zaire is in unstable political situation in 1997
3	marxist, peruvian 96/11/19–97/3/5	residence, ambassador, hostages, lima, peruvian, marxist, peru, hostage, ambassadors, reception, shining, siege, captive 96/12/20	Japanese embassy hostage crisis in Lima, Peru	There aren't any news reports including features 'Marxist' and 'Peruvian' on '96/11/19' and '96/12/19'. The number is 7 on '96/12/20'
4	movement, tupac, amaru, lima, hostage, hostages 96/11/16–97/3/20	ambassador, hostages, peru, lima, residence, peruvian, amaru, tupac, hostage, alberto, mrta, reception, inspired, fujimori, comrades, emperor, captives 96/12/18	Japanese embassy hostage crisis in Lima, Peru	The crisis broke out in the evening of '96/12/17'
5	kinshasa, kabila, laurent, congo 97/3/26–97/6/15	zaireans, truce, sahnoun, mohamed, transitional, ailing, seizing, congo, kabila, laurent 97/3/22	Zaire rebel leader Laurent Kabila outlined plans for a transitional government on '97/3/22', and told U.N. Mohamed Sahnoun there would be no truce before negotiations	Zaire is in unstable political situation in 1997
6	jospin, lionel, june 97/5/10–97/7/9	jospin, lionel, parliamentary, udf, socialist 97/4/21	French opposition Socialist leader Lionel Jospin accused President Jacques Chirac of planning a parliamentary election	Our approach detected totally 42 events about 'Lionel Jospin' after '97/4/21'
7	iraq, missile 96/8/31–96/9/13	iraq, missile, kurds, contradicts, sovereignty, skikda, hindering 96/9/4	U.S. missile attack on Iraq	'96/9/4' is the first day our experiment carried out
8	kurdish, baghdad, iraqi 96/8/29–96/9/9	iraqi, saddam, missiles, baghdad, hussein, kurdish, hostilities, kuwait, escalation, jerk, havens, simex, spiked, soar, unjustified, actions, flagrant, skies, chevron, onstream 96/9/4	Iraqi troop fought with Kurdish	'96/9/4' is the first day our experiment carried out
9	may, blair 97/3/24–97/7/4	tony, blair, revolution, labour 97/3/18	No revolution, pacesetter Tony Blair promises Britain	Our approach detected totally 70 events about 'blair' around the year
10	slalom, skiing 96/12/5–97/3/21	skiing, alpine 96/12/17	Austria's Thomas Sykora clinched men's Alpine skiing Slalom win, ahead of Italian Alberto Tomba	There are not any reports including 'slalom' and 'skiing' on '96/12/5'. The number is under 2 until '96/12/17'. Our approach detected more than 80 events about 'Slalom Game of Alpine Skiing' after '96/12/17'
11	interim, months 96/9/24–96/12/31	unless, specified, billions, yen, interim, div 96/09/04	Japan released 6-month forecast of 5 companies on '96/09/04'	Japan released forecast or interim results for thousands of companies after '96/9'
12	dole, bob 96/9/9–96/11/24	dole, bob, nominee, midwestern, refinanced, halving, affluent, hinges, vintage, hoover 96/9/4	Republican presidential candidate Bob Dole unveiled a series of plans	Our approach detected totally 32 events about 'Bob Dole' after '96/9/4'
13	july, sen 97/6/25–97/6/25	ranariddh, norodom, hun, sen, premiers, prince 97/6/25	Cambodia's Prince Norodom Ranariddh Ranariddh announced that Second Prime Minister Hun Sen agreed the Cambodia's next general election	Our approach detected totally 32 events about 'Hun Sen' after '97/6/25'
14	hebron 96/10/15–97/2/14	dore, hebron, ross erekat, saeb, rabin, expedite, pullback, modifications, yitzhak, renegotiate, jews, cool, spurred 96/10/06	Palestinian negotiator Saeb Erekat told reporters that there will be not any renegotiate or modifications of agreements signed	Our approach detected totally 57 events about 'hebron' around the year
15	april, easter 97/2/23–97/5/4	easter 97/3/20	News about Easter day in March and April in 1997	Our approach detected totally 31 events about Easter day around the year
16	diluted, group 97/4/27–97/7/20	specified, billions, eps, ord, unless, parent, div, diluted, vs, commem, group, prft, wholesaler, kogyo 97/5/15	Japan released 96/97 group results of 48 companies on '97/5/15'	Japan released 96/97 group results of thousands of companies from '97/5/15' and lasted 36 days except weekends
17	december, christmas 96/11/17–97/1/26	christmas 96/12/12	News about Christmas day in late '96/12'	Our approach detected totally 33 events about 'Christmas' around the year

Table 5 Key events related to 'Japanese embassy hostage crisis'

Date	Description of key events
96/12/17	Fourteen MRTA members occupied the Japanese Ambassador's residence in Lima, and made a series of demands
96/12/21	The leader of MRTA Néstor Cerpa announced that hostages who were not connected to the Peruvian government would be released gradually
96/12/22	Peru President Fujimori made his first public announcement about the hostage crisis, rejected the MRTA's demands. During the months that followed, the rebels released all female hostages and all but 72 of the men
96/12/26	A loud explosion was heard from the Japanese ambassador's residence in Lima
97/4/22	Under orders from President Fujimori, armed forces stormed the residence and rescued the remaining hostages and killed all 14 MRTA militants

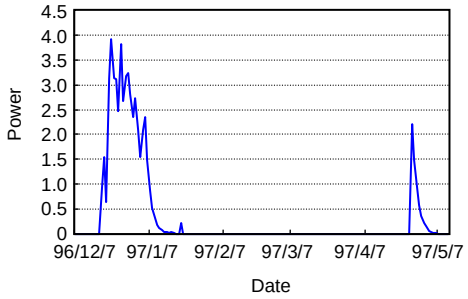

Fig. 9 The power of 'Japanese embassy hostage crisis' over the timeline

8 Conclusion

In this paper, we study the problem of the online detection of bursty events and the discovery of their evolution in a sequence of a chronologically ordered news stream. In our work, a bursty event is represented as a group of highly correlated bursty features. We applied a new algorithm, online multi-resolution burst detection (OMRBD), to detect bursty features of multiple bursty durations. To cluster bursty features into corresponding events, we applied an affinity propagation algorithm by taking correlation between bursty features and their preference as inputs. Correlations between bursty features are measured by considering both the similarity between the corresponding bursty patterns and the intersection between the corresponding documents. The preference of a bursty feature is set to its power value, which is a measurement we introduced to indicate a feature's and event's bursty level or importance. By calculating the cosine similarity of bursty event's feature vectors, similar events along the timeline are grouped into tracks of evolution. The power value events will also help us discover the headline events at a certain time point and provide topic visualization from a historical perspective.

We evaluated our method using Reuters Corpus Volume 1, which consists of 806 791 news reports over one year. Experimental results show that the proposed method can mine meaningful information hidden in the news stream. There are still many interesting aspects to extend our work further. For example, we can develop a news recommendation or retrieval system applying the proposed methods to help users navigate in the news information space. Better methods should be proposed to assign labels automatically for detected events and make our work more practical. In addition, there exists some interesting knowledge hidden in news streams that can be addressed in future work, such as events' association and events' periodicity. Finally, name entities and reweighting strategies may be incorporated into our method to improve detection accuracy.

References

Allan, J., Papka, R., Lavrenko, V., 1998. Online New Event Detection and Tracking. Proc. SIGIR Conf. on Research and Development in Information Retrieval, p.37-45. [doi:10.1145/290941.290954]

Baeza-Yates, R., Ribeiro-Neto, B., 2004. Modern Information Retrieval. China Machine Press, Beijing, China (in Chinese).

Bulut, A., Singh, A.K., 2005. A Unified Framework for Monitoring Data Streams in Real Time. Proc. 21st Int. Conf. on Data Engineering, p.44-55. [doi:10.1109/ICDE.2005.13]

Chen, W., Zhang, L.J., Wang, C., Chen, C., Bu, J.J., 2008. Pervasive Web News Recommendation for Visually-Impaired People. IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology, **3**:119-122. [doi:10.1109/WIIAT.2008.43]

Chu, K.K.W., Wong, M.H., 1999. Fast Time-Series Searching with Scaling and Shifting. Proc. 8th ACM SIGMOD Symp. on Principles of Database Systems, p.237-248. [doi:10.1145/303976.304000]

Croft, W.B., Metzler, D., Strohman, T., 2009. Search Engines: Information Retrieval in Practice. Addison Wesley, Boston.

Dezso, Z., Almass, E., Lukacs, A., Racz, B., Szakadat, I., Barabasi, A.L., 2006. Dynamic of information access on the Web. *Phys. Rev. E*, **73**(6):066132. [doi:10.1103/PhysRevE.73.066132]

Frey, B.J., Dueck, D., 2007. Clustering by passing messages between data points. *Science*, **315**(5814):972-976. [doi:10.1126/science.1136800]

Fung, G.P.C., Yu, J.X., Yu, P.S., Lu, H.J., 2005. Parameter Free Bursty Events Detection in Text Streams. Proc. 31st Int. Conf. on Very Large Data Bases, p.181-192.

He, Q., Chang, K., Lim, E., 2007. Analyzing Feature Trajectories for Event Detection. Proc. 30th Annual Int. ACM SIGIR Conf., p.207-214. [doi:10.1145/1277741.1277779]

Kahveci, T., Singh, A., 2001. Variable Length Queries for Time Series Data. Proc. 17th Int. Conf. on Data Engineering, p.273-282. [doi:10.1109/ICDE.2001.914838]

Kleinberg, J., 2002. Bursty and Hierarchical Structure in Streams. Proc. 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, p.91-101. [doi:10.1023/A:1024940629314]

Kumaran, G., Allan, J., 2004. Text Classification and Named Entities for New Event Detection. Proc. 27th Annual Int. ACM SIGIR Conf., p.297-304. [doi:10.1145/1008992.1009044]

Lam, W., Meng, H., Wong, K., Yen, J., 2001. Using contextual analysis for news event detection. *Int. J. Intell. Syst.*, **16**(4):525-546. [doi:10.1002/int.1022]

Lewis, D.D., Yang, Y.M., Rose, T.G., Li, F., 2004. RCV1: a new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, **5**:361-397.

Li, Z.W., Wang, B., Li, M.J., Ma, W.Y., 2005. A Probabilistic Model for Retrospective News Event Detection. Proc. SIGIR Conf. on Research and Development in Information Retrieval, p.106-113. [doi:10.1145/1076034.1076055]

Luxburg, U., 2007. A tutorial on spectral clustering. *Statist. & Comput.*, **17**(4):395-416. [doi:10.1007/s11222-007-9033-z]

Mei, Q.Z., Zhai, C.X., 2005. Discovering Evolutionary Theme Patterns from Text: An Exploration of Temporal Text Mining. Proc. 11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining, p.198-207. [doi:10.1145/1081870.1081895]

Topic Detection and Tracking Evaluation (TDT) Project, 2007. Available from http://www.itl.nist.gov/iad/mig//tests/tdt/ [Accessed on Aug. 8, 2009].

Vlachos, M., Meek, C., Vagena, Z., Gunopulos, D., 2004. Identifying Similarities, Periodicities and Bursts for Search Queries. Proc. ACM SIGMOD Int. Conf. on Management of Data, p.131-142. [doi:10.1145/1007568.1007586]

Xia, D.Y., Wu, F., Zhang, X.Q., Zhuang, Y.T., 2008. Local and global approaches of affinity propagation clustering for large scale data. *J. Zhejiang Univ.-Sci. A*, **9**(10):1373-1381. [doi:10.1631/jzus.A0720058]

Yang, Y.M., Pierce, T., Carbonell, J.G., 1998. A Study on Retrospective and On-line Event Detection. Proc. SIGIR Conf. on Research and Development in Information Retrieval, p.28-36. [doi:10.1145/290941.290953]

Yang, Y.M., Zhang, J., Carbonell, J., Jin, C., 2001. Topic-Conditioned Novelty Detection. Proc. 8th ACM SIGKDD Int. Conf., p.688-693. [doi:10.1145/775047.775150]

Yuan, Z.J., Yan, J., Yang, S.Q., 2007. Online Burst Detection Over High Speed Short Text Streams. Proc. 7th Int. Conf. on Computational Science, p.717-725. [doi:10.1007/978-3-540-72588-6_119]

Zhang, K., Li, J.Z., Wu, G., 2007. New Event Detection Based on Indexing-Tree and Name Entity. Proc. 30th Annual Int. ACM SIGIR Conf., p.215-222. [doi:10.1145/1277741.1277780]

Zhang, K., Li, J.Z., Wu, G., Wang, K.H., 2008. A new event detection model based on term reweighting. *J. Softw.*, **19**(4):817-828 (in Chinese). [doi:10.3724/SP.J.1001.2008.00817]

Zhu, Y., Shasha, D., 2002. Statstream: Statistical Monitoring of Thousands of Data Streams in Real Time. Proc. 28th Int. Conf. on Very Large Databases, p.358-369. [doi:10.1016/B978-155860869-6/50039-1]