Parameter Free Bursty Events Detection in Text Streams

Gabriel Pui Cheong Fung[†] Jeffrey Xu Yu[†] Philip S. Yu[‡] Hongjun Lu[††]

[†]The Chinese University of Hong Kong, Hong Kong, China, {pcfung,yu}@se.cuhk.edu.hk
[‡]T. J. Watson Research Center, IBM, USA, psyu@us.ibm.com
[††]The Hong Kong University of Science and Technology, Hong Kong, China, luhj@cs.ust.hk

Abstract

Text classification is a major data mining task. An advanced text classification technique is known as partially supervised text classification, which can build a text classifier using a small set of positive examples only. This leads to our curiosity whether it is possible to find a set of features that can be used to describe the positive examples. Therefore, users do not even need to specify a set of positive examples. As the first step, in this paper, we formalize it as a new problem, called hot bursty events detection, to detect bursty events from a text stream which is a sequence of chronologically ordered documents. Here, a bursty event is a set of bursty features, and is considered as a potential category to build a text classifier. It is important to know that the hot bursty events detection problem, we study in this paper, is different from TDT (topic detection and tracking) which attempts to cluster documents as events using clustering techniques. In other words, our focus is on detecting a set of bursty features for a bursty event. In this paper, we propose a new novel parameter free probabilistic approach, called feature-pivot clustering. Our main technique is to fully utilize the time information to determine a set of bursty features which may occur in different time windows. We detect bursty events based on the feature distributions. There is no need to tune or estimate any parameters. We conduct experiments using real life data, a major English newspaper in Hong Kong, and show that the parameter free feature-pivot clustering approach can detect the bursty events with a high success rate.

1 Introduction

In this paper, we study a new problem, called *hot bursty events detection* in a text stream, where a text stream is a sequence of chronologically ordered documents, and a hot bursty event is a minimal set of bursty features that occur together in certain time windows with strong support of documents in the text stream. For example, *SARS* (Special Severe Acute Respiratory Syndrome) is a bursty event that consists of a set of bursty features such as sars, outbreak, atypic, respire, pneumonia, inflect, etc. This bursty event was reported in four hot periods, in a major English newspaper, South China Morning Post, in Hong Kong: (1) from 3rd April 2003 to 26th June 2003, (2) on 20th July 2003, (3) on 2nd October 2003; and (4) on 11th January 2004. The first hot period was the period when it was identified as a dangerous new disease. The second hot period was the time that the director of Health of Hong Kong announced that she would resign her position and take up a senior position at the World Heath Organization. The third hot period was the period when an independent investigation report against *SARS* was disclosed. The fourth hot period was when there were some suspicious *SARS* cases identified in Guangdong province of China. The determination of such minimal set of bursty features, to specify a burst event, assists text classification, as one major step ahead of current research activities on text classification. It is because that the set of bursty features can be used as a set of features for positive examples, and therefore helps partially supervised text classification [10, 6], which is a text classification technique using positive examples only. In other words, with our techniques, users do not even need to specify a set of positive examples to build a text classifier. However, the focus of this paper is on the determination of bursty events, and is not on partially supervised text classification with a set of positive features.

The bursty events detection problem is given below.

**Proceedings of the 31st VLDB Conference,
Trondheim, Norway, 2005**

Consider a text stream $\mathcal{D} = \{d_1, d_2, \cdots\}$ where d_i is a document, and the length of $\mathcal{D}$ is $|\mathcal{D}|$. A document d_i consists of a set of features, $f_{i_1}, f_{i_2}, \cdots$, and is reported at time t_i. In the text stream $\mathcal{D}$, $t_i \leq t_j$ if $i < j$. Dividing the text stream, $\mathcal{D}$, into L non-overlapping time windows, W_i of the same length, say per day. The problem of *hot bursty events detection* is a problem to find a set of bursty events, where a bursty event consists of a minimal set of bursty features, in time windows $W_i, W_j, \cdots$ that together identifies the event with the largest number of documents that contain the bursty features.

Our problem is different from the existing event detection problems such as TDT (Topic Detection and Tracing) [2, 3, 14, 26, 21, 27, 25]. TDT is an unsupervised learning task (clustering) that finds clusters of documents matching the real events (sets of documents identified by human) by reducing the number of missing documents in the clusters found and reducing the possibility of false alarms. The key issue of our hot bursty events detection is to find the minimal sets of bursty features automatically. In other words, the emphasis of our problem is to identify sets of bursty features, whereas the emphasis of TDT is to find clusters of documents.

The hot bursty events detection can be possibly handled by clustering of documents followed by a step of selecting features from the clusters found. We call it a *document-pivot clustering* approach, because it first clusters similar documents into clusters, and then selects features as bursty events from the clusters. The related works include TDT [2, 3, 14, 18, 21, 26, 27], text mining [9, 13, 14, 17, 19, 20, 22], and visualization [7, 11, 24]. However, the main drawback of adapting these techniques for the new hot bursty events detection problem is that they require many parameters and it is very difficult to find an effective way to tune these parameters. For example, [26, 27] propose a divide-and-conquer version of the group-average clustering approach [23] for event detection using six parameters, bucket size, clustering threshold, reducing factor, number of iterations between re-clustering, features per vector, and feature weighting schema. These parameters are interrelated. Changing one parameter may have great impacts on the selection of other parameters. [19] proposes a χ^2 approach for extracting significant time varying features from text, where extracting different kinds of features requires different thresholds. It needs two χ^2-thresholds for extracting name entities and noun phrases in a text stream. [20] proposes a χ^2 based strategy for visualizing the major events in a text stream. Similar to [19], [20] needs different thresholds for different kinds of features including two additional parameters, namely, the grouping threshold and the stopping criteria, in order to identify different events. Without any prior knowledge about the events in the text stream, it would be rather difficult to estimate these parameters. None of the previous reported studies discussed in details how to estimate and tune the parameters, to our best knowledge. The task of tuning parameters is time-consuming and is difficult, because these parameters are sensitive and critical for event detection.

In this paper, we propose a new novel *feature-pivot clustering* approach for hot bursty events detection. By feature-pivot clustering, as a term to distinguish it from document-pivot clustering, we mean that we do not need to cluster documents in order to find bursty events. The uniqueness of our approach is as follows. First, as the first attempt, we identify hot bursty features by feature distribution, as a time-series in time windows. Second, we group bursty features into bursty events. Third, we identify the hot periods of burst events. The main advantage of our approach is parameter free. There are no parameters that need to be tuned, and there is no need to use any weighting schema as we do not need to weight the features. It is also important to note that our approach can in turn help TDT to select features for the existing event detection problem [2, 3, 14, 26, 21, 27, 25].

The rest of the paper is organized as follows. Section 2 discusses the document-pivot clustering and its problems. Section 3 presents our novel parameter free feature-pivot clustering approach. Section 4 shows that the parameter free feature-pivot clustering approach can detect the bursty events with a high success rate. The related works are discussed in Section 5. We conclude this work in Section 6.

2 Document-Pivot Clustering and Its Problems

In this section, we address the issues behind the document-pivot clustering approach which makes hot bursty events detection difficult. For detecting hot bursty events, the document-pivot clustering approach first assigns weights to the features based on the most widely-used *tf·idf* schema [15]. Second, it performs clustering to group similar documents into clusters. Third, it selects features, as bursty features, from the clusters of documents based on some feature selection approaches [16]. The first two steps are the main steps used in TDT [2, 3, 14, 26, 21, 27, 25]. The limitations of adopting this approach are given below.

- The task of hot bursty event detection is to find a minimal set of features that can represent a bursty event. However, in the document-pivot clustering approach, features as a whole need to be considered to measure the similarity between two documents. The similarity of documents can be biased to the noisy features. Our early study reported that the most similar documents often belong to different categories [6]. Therefore, with the document-pivot clustering approach, most simi-

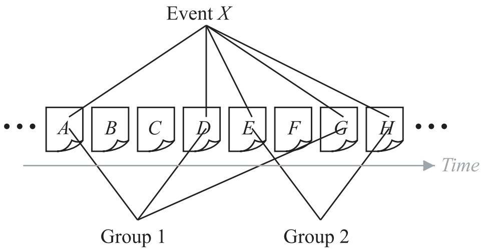


Figure 1: Document Clustering

lar documents do not necessarily report the same event.

- In the document-pivot clustering approach, the *tf·idf* schema [15] is used for feature weightings. However, the *tf·idf* schema is originally designed for information retrieval, not for clustering, even though it performs well in most text clustering problems. There are many *tf·idf* schema variations, but the basic idea is the same such as features that appear in a few documents are useful, and should be assigned higher weights. The *tf·idf* schema does not suit for our purposes for hot bursty events detection, because we need to find the features that appear in a large number of documents in certain hot periods, so as to distinguish the set of documents that contain the burst features from the other documents.

- The document-pivot clustering approach is not effective in handling the cases where the same events occurs as bursts several times in a long time period [20, 27]. The reason is that such an event will be broken into parts when some specific features do not occur frequently enough in consecutive time windows. Figure 1 illustrates an example. Suppose there are eight consecutive documents, from A to H, where the documents A, D, E, G and H support the same event X. During document clustering, suppose that the documents A, D and G are initially grouped together in a cluster, G_1, and the documents E and H are initially grouped together in another cluster, G_2. Recall a threshold (stopping criterion) is used to maintain high intra-similarity within a cluster, which prohibits further documents, that are not related to the cluster, being assigned to it. As a result, G_1 and G_2 may not be able to merge to form a single cluster. In other words, a long running event may be broken down into several separated events (several different clusters). It may result in many small clusters, and can make it difficult to find the major events which lasts long.

- Assume that the document-pivot clustering approach can cluster documents well as events. It is still difficult to determine bursty events, because it requires a ranking function to rank events. However, without any prior knowledge, it is difficult to formulate a good ranking function. In addition, it is difficult to determine its hot periods. Another threshold may need to be introduced to determine the hot periods, such that a hot period is defined as the number of documents that belong to the bursty event in a specific time period is larger than the predefined threshold. However, there does not exist a single threshold for all different events.

3 Bursty Event Detection: A Feature-Pivot Clustering Approach

All the above give us the motivation for considering the feature distributions directly rather than on document distributions during clustering, i.e. feature-pivot clustering.

Our framework is outlined in Figure 2. There are three major steps: (1) Bursty features identification, (2) Bursty features grouping, and (3) Hot periods of the bursty events determination. Note that no weighting schema is necessary in our framework. Details are given in the following sections.

3.1 Bursty Features Identification

Assume the number of documents that contain the feature f_j in a window W_i, denoted as $n_{i,j}$, follows a generative probabilistic model, which is a model based on an unknown probability distribution. With the generative probabilistic model, we can compute the probability of the number of documents that contain the feature f_j in the time window W_i, denoted as $P_g(n_{i,j})$.

$P_g(n_{i,j})$ can be modeled using a *hyper-geometric distribution*. Recall the definition of hyper-geometric distribution [12]: A sample of size n' objects is selected, at random (without replacement), from the N' objects, such that K' objects in n' are classified as success and $N' - K'$ objects are classified as failure, then the random variable X' that denotes the number of successes in the sample has a hyper-geometric distribution. In our problem, N' is the number of documents in the text stream, n' is the number of documents in a window, K' is the number of documents that contain the specific feature in the particular window, and $n' - K'$ is the number of documents that do not contain the specific feature in the particular time window. As a result, the probability that the feature f_j in the time window W_i can be modeled by a hyper-geometric distribution. Note: hyper-geometric distribution is computational expensive such that its computational time growth quadratically with $n_{i,j}$. We model the hyper-geometric distribution using the *binomial distribution*, since the computation of binomial distribution is far

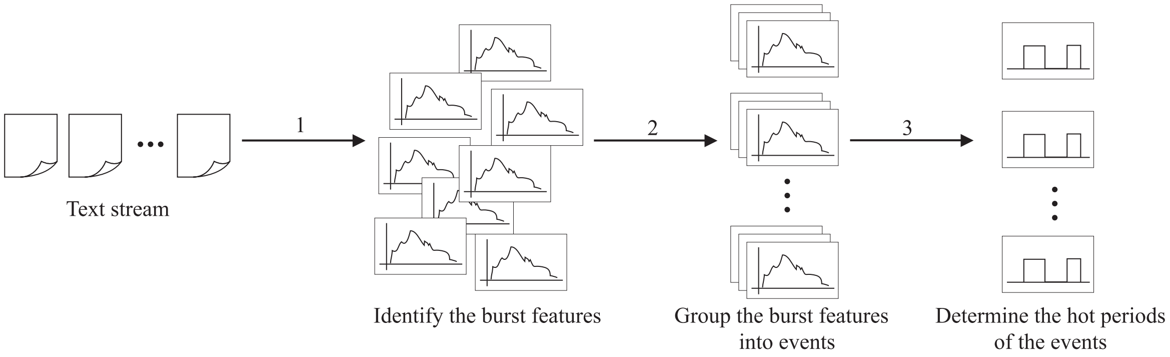

Text stream
1
Identify the burst features
2
Group the burst features into events
3
Determine the hot periods of the events

Figure 2: The Overview of Feature-Pivot Clustering Approach

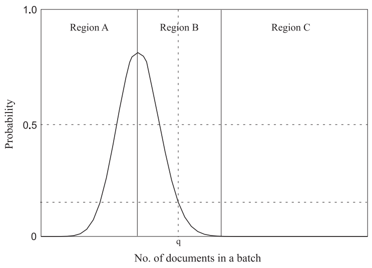


Figure 3: A typical binomial distribution

more efficient. Furthermore, both distributions will eventually be the same when the database is large [12]. Hence, $P_g(n_{i,j})$ is modeled by a binomial distribution, and is computed as follows.

$$P_g(n_{i,j}) = \binom{N}{n_{i,j}} p_j^{n_{i,j}} (1-p_j)^{N-n_{i,j}} \tag{1}$$

We explain N and p_j below.

N is the number of documents in a time window. It is worth noting that, although the number of documents, N_i, in each time window can be different, we can re-scale it in all time windows, such that all N_i become the same. We do it by adjusting the frequencies of features in all time windows. For example, suppose that there is a feature, f_j, in two time windows, W_1 and W_2. In W_1, the number of documents is $N_1 = 78$, and the number of documents that contain f_j is $n_{1,j} = 24$; and in W_2, the number of documents is $N_2 = 82$, and the number of documents that contain f_j is $n_{2,j} = 35$. We can re-scale N_i and N_j to 100 by rescaling $n_{1,j} = 31$ and $n_{2,l} = 43$ accordingly. Note that setting N does not affect the quality of bursty events detection, because the overall feature distributions are unaffected. As a result, N is not considered as a parameter in our scheme.

In Eq (1), p_j is the expected probability of the documents that contain the feature f_j in a random time window, and is therefore the average of the observed probability of f_j in all time windows containing f_j:

$$p_j = \frac{1}{L'} \sum_{i=0}^{L'} P_o(n_{i,j}) \tag{2}$$

$$P_o(n_{i,j}) = \frac{n_{i,j}}{N} \tag{3}$$

where L' is the number of time windows containing f_j. Note that $P_g(n_{i,j})$ becomes maximum if $n_{i,j}/N = p_j$.

Figure 3 shows a typical binomial distribution, $P_g(n_{i,j})$, for the feature f_j in a time window W_i. Note: binomial distribution is asymmetric except when $p_j = 0.5$. The shape of the binomial distribution depends on p_j only. A larger p_j would shift the burst to the right hand side. It is worth noting that if the frequency of a feature that appears in every window is very high, e.g. a stopword, then L' and $P_o(n_{i,j})$, in Eq. (2) and Eq. (3), would both be large, which result in a large p_j. In this case, the binomial distribution in a time window is similar to the one shown in Figure 4. The main difference between Figure 3 and Figure 4 is given below. In Figure 3, there are three regions along the x-axis (the number of documents): R_A, R_B and R_C. R_A is from 0 to the x value where $P_g(x)$ is the maximum, R_B is from the x value where $P_g(x)$ is the maximum to the x value where $P_g(x)$ becomes zero again, and R_C is the region followed by R_B. In Figure 4, the right hand side of the distribution can never reach 0. Therefore, a feature f_j will be taken as a stopword if its binomial distribution becomes the one as shown in Figure 4.

We discuss how likely an important feature f_j will be wrongly taken as a stopword below. Suppose that f_j is a bursty feature in a bursty event E_k, such that f_j only appears with high frequency in the hot periods of E_k. It implies $n_{i,j}$ is large in the time window W_i where E_k is a bursty event. If it occurs, all the observed probability in every sliding window, $P_o(n_{i,j})$ (Eq. (3)) will be large, whereas the number of time windows that contain f_j will be small, i.e. L' in Eq. (2) will be small. Hence, p_j will be large as well as shown in Figure 4. Therefore, it is possible in theory that f_j

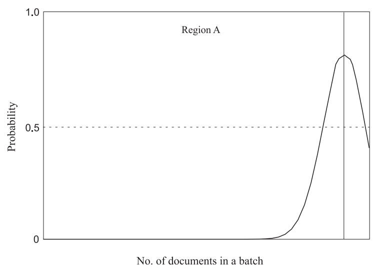


Figure 4: The binomial distribution of a stopword

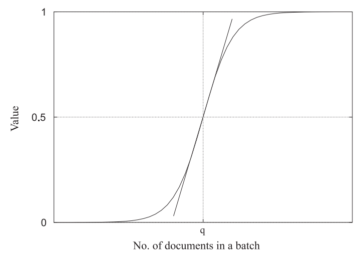


Figure 5: A Sigmoid Distribution

will be wrongly taken as a stopword. However, it is most unlikely that f_j will be wrongly taken as a stopword for the following reasons. Feature distributions are sparse in nature. Even though f_j may be only related to E_k as its bursty feature, it is most likely that f_j will appear in other time windows where E_k is not a bursty window. In other words, for the same feature f_j, the number of documents that contain it, $n_{i,j}$, will be large in some time windows and small in some other windows. On average, the observed probability, $P_o(n_{i,j})$, for such a feature f_j, will not be large as the one for a stopword, which is confirmed by our extensive experimental studies. We will further conduct analytical studies on this issue as our future work.

We decide the probability that a feature is bursty based on the binomial distributions, $P_g(n_{i,j})$, in Figure 3. Let $P_b(i, f_j)$ be the probability that the feature f_j is burst in the time window W_i. We consider three cases below.

- When $n_{i,j}$ is in R_A, it implies that $P_o(n_{i,j}) \leq p_j$. It suggests that the probability of the feature f_j in W_i is less than or equal to the probability that f_j is drawn randomly. We consider f_j as a non-bursty feature in W_i, and let $P_b(i, f_j) = 0$.
- When $n_{i,j}$ is in R_C, it implies that $P_o(n_{i,j})$ is noticeably higher than the prior probability of the feature f_j (p_j). It suggests that f_j exhibits an abnormal behavior in W_i. We consider f_j as a bursty feature in W_i, and let $P_b(i, f_j) = 1$.
- When $n_{i,j}$ is in R_B, there are further three cases. When $n_{i,j}$ approaches the boundary of R_B and R_C, the corresponding feature f_j will be a bursty feature; when $n_{i,j}$ approaches the boundary of R_B and R_A, f_j will be a non-bursty feature; and when $n_{i,j}$ is on the mid-point of region R_B (the point q in Figure 3), f_j can either be bursty or not bursty.

  Based on the case analysis, we use a *sigmoid function* to determine whether f_j is bursty or not when $n_{i,j}$ is in the region R_B.

$$P_b(i, f_j) = \frac{1}{1 + e^{-x}} \tag{4}$$

$$x = P_g(n_{i,j}) \cdot \theta - q \tag{5}$$

where q is the mid-point in the region R_B of Figure 3, and θ is the slope of the sigmoid function (Figure 5) which can be readily computed by referring to the range of R_B.

3.2 From Bursty Features To Bursty Events

Let the bursty features identified be $B = \{b_0, b_1, \ldots, b_{|B|}\}$. A bursty event is an event that consists of bursty features. The selection of minimal number of features to form a bursty event is formulated as follows. Let a bursty event $E_k = \{e_0, e_1, \ldots, e_{|B|}\}$ where $e_i = \{0, 1\}$. It suggests the following. When $e_i = 0$, the i-th bursty feature b_i does not contribute to the bursty event E_k; when $e_i = 1$, the i-th feature b_i is selected as the key feature for the bursty event E_k. For example, suppose $B = \{$`database`, `food`, `management`, `music`$\}$. $E_k = \{1, 0, 1, 0\}$ implies that the bursty event E_k contains two bursty features, `database` and `management`. Here, the problem of determining the minimal set of bursty features of a bursty event can be solved as finding the optimal E_k such that the probability of the bursty features grouped together is maximum for the text stream $\mathcal{D}$. Let $\mathsf{D} = \{D_0, D_1, \ldots, D_{|B|}\}$ be a set of sets of documents where D_i contains documents that contain the bursty feature b_i. Mathematically,

$$\max P(E_k|\mathsf{D}) = \frac{P(\mathsf{D}|E_k)P(E_k)}{P(\mathsf{D})} \tag{6}$$

Taking logarithm in Eq. (6), maximizing Eq. (6) is equivalent to minimize:

$$\begin{aligned} \min - \ln P(E_k|\mathsf{D}) = & - \ln P(\mathsf{D}|E_k) - \ln P(E_k) \\ & + \ln P(\mathsf{D}) \end{aligned} \tag{7}$$

Note: $\ln P(D)$ is independent of E_k. Thus, minimizing Eq. (7) is equivalent to minimize the following cost function:

$$\min c(E_k|\mathsf{D}) = - \ln P(\mathsf{D}|E_k) - \ln P(E_k) \tag{8}$$

We show how to compute $P(\mathsf{D}|E_k)$ and $P(E_k)$ below.

- **Computing** $P(E_k)$: Let the total number of time windows be L. We consider that a bursty feature b_j is a time-series of length L, such that the i-th value in the time-series is the bursty probability $P_b(i, b_j)$ in the time window W_i. We solve the problem of computing the probability of bursty features to be grouped together $(P(E_k))$ as to compute the probability of the corresponding time-series to be grouped together. This can be achieved by computing the similarity among a set of time-series given E_k [8, 5, 1]. In this paper, we take a simple yet efficient and effective approach to compute $P(E_k)$, that is, by computing the overlapping areas among different time-series.

$$P(E_k) = \frac{\bigcap_{j=0}^{|B|} |_{e_j=1} a_j}{\bigcup_{j=0}^{|B|} |_{e_j=1} a_j} \tag{9}$$

where a_j is the area covered by the feature b_j in the time-series.

- **Computing** $P(\mathsf{D}|E_k)$: We assume the feature distribution is independent, and formulate $P(\mathsf{D}|E_k)$ as follows. The idea behind it will be explained after introducing the formulation.

$$P(\mathsf{D}|E_k) = \prod_{j=0}^{|B|} \left(\frac{|D_j|}{|M|}\right)^{e_j} \left(1 - \frac{|D_j|}{|M|}\right)^{1-e_j} \tag{10}$$

$$M = \bigcup_{\substack{j=0 \\ e_j=1}}^{|B|} D_j \tag{11}$$

Here, M (Eq. (11)) is a set of documents that contain the bursty event E_k. In Eq. (10), the first component computes the probability of the documents that contain the bursty feature b_j in M, whereas the second component computes the probability of the documents that do not contain the bursty feature b_j in M. In other words, if $e_j = 1$, it implies that the bursty feature b_j belongs to the event E_k, and we compute the first component; if $e_j = 0$, we compute the second component. Hence, $P(\mathsf{D}|E)$ computes the production of the probability of D under M where M is constructed by the given E_k.

Finally, the cost function (Eq. (8)) can be computed as follows.

$$c(E_k|\mathsf{D}) = -\sum_{\substack{j=0 \\ e_j=1}}^{|B|} \ln(|D_j|) - \sum_{\substack{j=0 \\ e_j\neq 1}}^{|B|} \ln(|M| - |D_j|) + l\ln(|M|) - \ln \frac{\bigcap_{j=0}^{|B|} |_{e_j=1} a_j}{\bigcup_{j=0}^{|B|} |_{e_j=1} a_j} \tag{12}$$

Some observations can be made on Eq. (12). First, if the cost function c becomes smaller, then it suggests that the selected bursty features ($e_j = 1$) will be strongly related for the bursty event E_k. Second, if the bursty features are very similar, the cost function c becomes smaller, because the last component of Eq. (12) becomes smaller, which makes these bursty features to be grouped together. When the areas of two features completely overlapped, then the forth component becomes 0. Third, if the documents do not share a high degree of common bursty features, the cost function c becomes larger, because the third component becomes large. Fourth, if the bursty features are a subset of another set of bursty features, the cost function c becomes large, because the last component becomes large for the reason that the overlapping area of the two sets of bursty features becomes small.

We further address two important issues in grouping bursty features for a bursty event below.

The first issue is whether two bursty features f_j and f_l will be wrongly grouped together in a bursty event E_k, if the two features have the high similarity in their feature distribution. For example, the bursty feature *Sars* and the bursty feature *Iraq* are similar in the corresponding feature distributions (Figure 6 (a) and (g)). Is it possible that the two features will be grouped in the same bursty event? Below, we show that it is unlikely that the irrelevant bursty features will be grouped together. Consider the cost function (Eq. (8)) which has two components, $P(\mathsf{D}|E_k)$ and $P(E_k)$. If two bursty features have high similarity in their feature distributions, as time-series data, $P(E_k)$ (Eq. (8)) becomes large, because the common area of the two feature distributions becomes large. In other words, $P(E_k) \to 1$, and therefore $\ln P(E_k) \to 0$. It makes the cost (Eq. (8)) smaller as in favor of grouping these two bursty features. However, if the two bursty features are about two different stories (events), it is unlikely that they will appear in the same documents. For example, it is unlikely that many documents will discuss both *Sars* and *Iraq* together. Recall D_j is the set of documents that contain a bursty feature f_j, and D_l is the set of documents that contain a bursty feature f_l. If two bursty features appear in different documents, $P(\mathsf{D}|E_k)$ becomes smaller (Eq. (10)), because M (Eq. (11)) becomes larger, and therefore the cost becomes larger. Consequently, it is unlikely that the irrelevant bursty features will be wrongly grouped together. Detail information will be given in our experimental studies.

The second issue is whether the resulting set of bursty features for E_k will possibly include noises. The quality of the set of bursty features grouped together is guaranteed for the similar reasons we discussed for the above first issue. Consider $P(\mathsf{D}|E_k)$ again. If a bursty event E_k contains many features that appear in different sets of documents, $P(\mathsf{D}|E_k)$ becomes small, which

Algorithm 1 *HB-Event*(B, D)

```
Input: A set of bursty features, $B$, and the set of documents $\mathsf{D}$ that contains bursty features in $B$;
Output: A list of bursty events, $\{E_1, \ldots, E_k\}$;
$k \leftarrow 0$;
repeat
    $k \leftarrow k + 1$;
    compute $E_k$ by minimizing Eq. (12), using $B$ and $\mathsf{D}$;
    $B' \leftarrow \emptyset$;
    for each $e_j \in E_k$ do
        if $e_j = 1$ then
            $B' \leftarrow B - \{b_j\}$;
        end if
    end for
    $B \leftarrow B - B'$;
until $|B'| = 1$
return $\{E_1, \ldots, E_k\}$;
```

makes it unlikely to group them together. A set of bursty features are grouped under the condition that they are contained in the similar documents (Eq. (11)).

Our *HB-Event* algorithm, for Hot-Bursty-Event detection, is shown in Algorithm 1. The input is the set of bursty features B and the set of documents D that contains bursty features in B. The algorithm returns hot bursty events by repeatedly selecting the bursty events. Note: in Algorithm 1, a bursty feature only appears in one bursty event E_j. The main idea exhibited here can be extended to the cases where a bursty feature appears in multiple bursty events.

3.3 Hot Periods of the Bursty Events

The hot periods of a bursty event, E_k, are determined below. Let $H_k = \{h_0, h_1, \ldots, h_n\}$ for $h_i \in \{0, 1\}$ where $h_i = 1$ indicates that the bursty event E_k is hot in the time window W_i. Because we formalize the bursty features as time-series, we compute the probability of the hot bursty event E_k in W_i, denoted $\mathcal{P}_b$, by computing the expected probability of the bursty event based on the set of bursty features that belong to the bursty event E_k:

$$\mathcal{P}_b(i, E_k) = \frac{1}{|B_k|} \sum_{j=0}^{|B|} e_j \cdot P_b(i, f_j) \qquad (13)$$

where $|B_k|$ is the number of bursty features in E_k. In this paper, we say a bursty event E_k is hot in W_i, if $\mathcal{P}_b(i, E_k) > \beta$, where β is simply set as 2 times of the standard deviation above the expected value of $\mathcal{P}_b(i, E_k)$ for $i = 1, 2, \cdots$. We find that the setting of β value is effective in our experimental study using a real dataset. There is no need to tune β.

4 Experimental Studies

We have archived two-year news stories from a major English news paper in Hong Kong, South China Morning Post(`www.scmp.com.hk`), from 2003-01-01 to 2004-12-31. It consists of 66,300 news stories. We only conducted simple document pre-processing to remove punctuation, digits, stopwords, web page addresses and email addresses. All features are stemmed and converted to lower cases. The number of features after stemming is 93,807.

We implemented our framework using Java$^{\text{TM}}$ and conducted our testing on Solaris. In the experimental studies, we concentrated on our novel feature-pivot clustering approach, and do not show the results using document-pivot clustering, because there are no reported studies providing details for us to fine tune parameters for grouping bursty features. Simply tuning of parameters may result unfair results, and we are reluctant to include such results in this paper, and plan to study it more as our future work.

4.1 Identifying Bursty Features

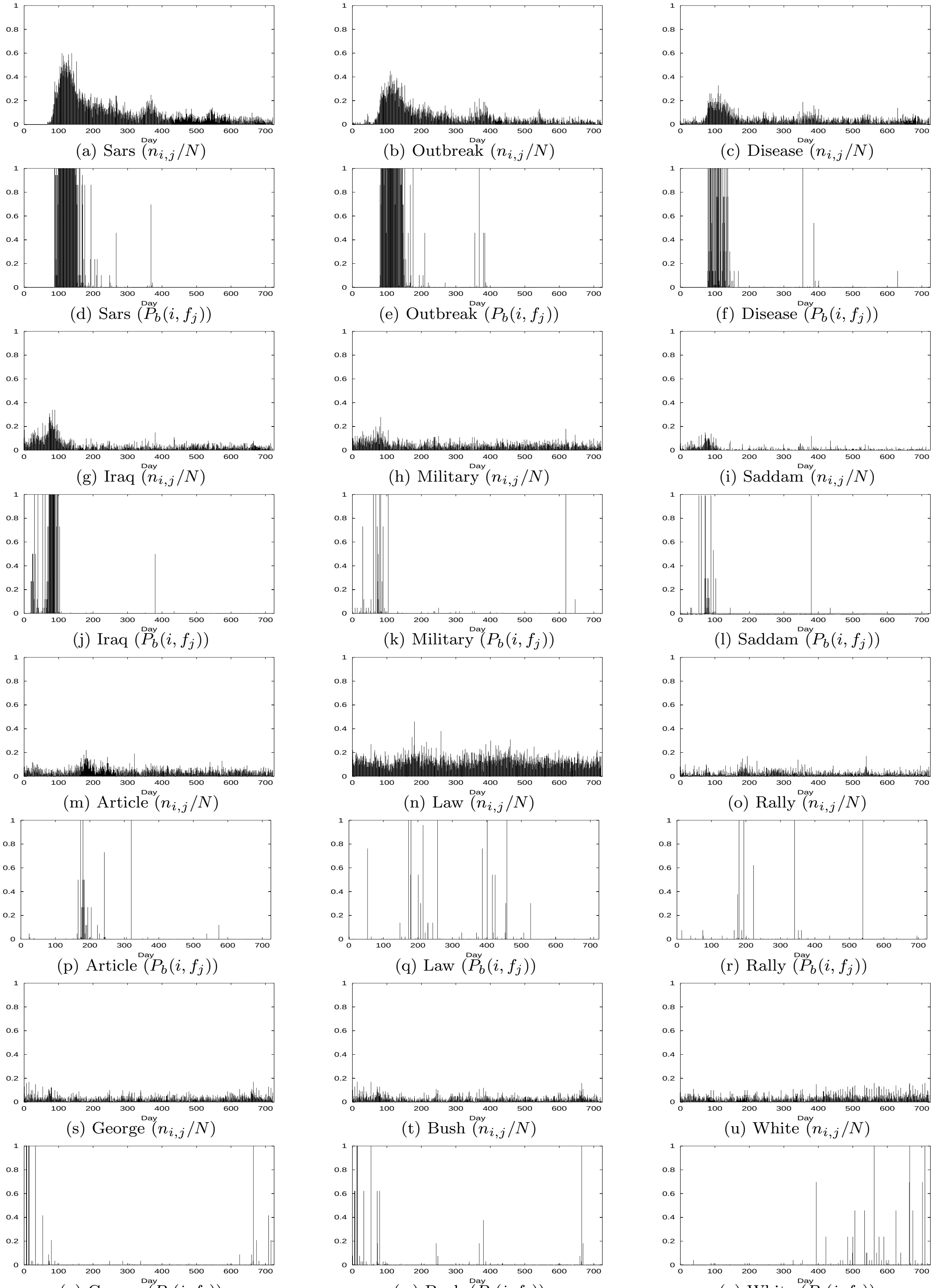
Among the 93,807 features, we found 373 features as bursty features in total. 12 bursty features are selected for detail discussions, including `Sars`, `Outbreak`, `Disease`, `Iraq`, `Military`, `Saddam`, `Article`, `Law`, `Rally`, `Gorge`, `Bush`, and `White`, as shown in Figure 6. In all the figures in Figure 6, the x-axis is the i-th date starting from Jan. 1st, 2003. A time window is a single day. There are two figures for each bursty feature, f_j, showing the percentages of news stories in a time window W_i that contain the bursty feature f_j, $n_{i,j}/N$, and its bursty feature probability, $P_b(i, f_j)$. The 1st, 3rd, 5th, and 7th rows show the percentage of news stories in a time window that contain the bursty feature in question, and the 2nd, 4th, 6th and 8th rows show the bursty feature probabilities.

As shown in Figure 6, there are some noticeable bursty features such as `Sars`, `Outbreak`, `Iraq`, and `Military`, if we compare (a) vs (d) for `Sars`, (b) vs (e) for `Outbreak`, and (g) vs (j) for `Iraq`. There are also some bursty features, like `Law` ((n) vs (q)), which appear everyday.

Our novel feature-pivot clustering approach can also find the hot periods where the bursty features occur. It is important to notice that no bursty features can be observed using document-pivot clustering approach, if they appear continuously like `Law` in our example. It shows the strength of the proposed feature-pivot clustering approach. More discussions will be given later when we discuss bursty events.

4.2 Identifying the Bursty Events and the Hot Periods

A bursty event contains a set of bursty features. Total 28 bursty events are found using the *HB-Event* algo-



(a) Sars ($n_{i,j}/N$)

(b) Outbreak ($n_{i,j}/N$)

(c) Disease ($n_{i,j}/N$)

(d) Sars ($P_b(i, f_j)$)

(e) Outbreak ($P_b(i, f_j)$)

(f) Disease ($P_b(i, f_j)$)

(g) Iraq ($n_{i,j}/N$)

(h) Military ($n_{i,j}/N$)

(i) Saddam ($n_{i,j}/N$)

(j) Iraq ($P_b(i, f_j)$)

(k) Military ($P_b(i, f_j)$)

(l) Saddam ($P_b(i, f_j)$)

(m) Article ($n_{i,j}/N$)

(n) Law ($n_{i,j}/N$)

(o) Rally ($n_{i,j}/N$)

(p) Article ($P_b(i, f_j)$)

(q) Law ($P_b(i, f_j)$)

(r) Rally ($P_b(i, f_j)$)

(s) George ($n_{i,j}/N$)

(t) Bush ($n_{i,j}/N$)

(u) White ($n_{i,j}/N$)

(v) George ($P_b(i, f_j)$)

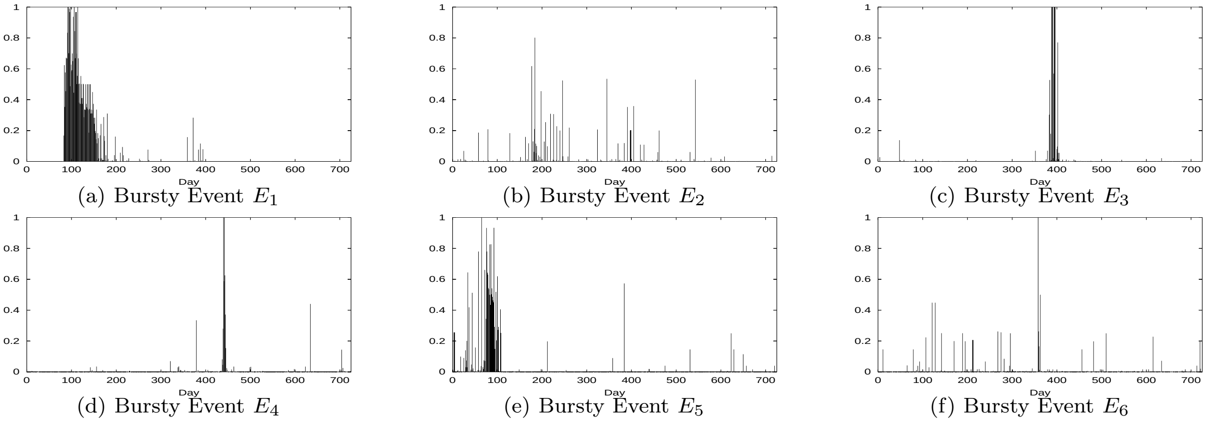
(w) Bush ($P_b(i, f_j)$)

(x) White ($P_b(i, f_j)$)

Figure 6: 12 Bursty Features (The percentages of news stories in a window W_i that contain the bursty feature f_j, $n_{i,j}/N$, are shown in (a)-(c), (g)-(i), (m)-(o), and (s)-(u), and the probabilities of bursty features, $P_b(i, f_j)$, are shown in (d)-(f), (j)-(l), (p)-(r), and (v)-(x).)

Bursty Events	Bursty Features
E_1 (*SARS*)	sars, outbreak, atypic, respire, pneumonia, inflect, ...
E_2 (*Legislation*)	article, Yip, law, rally, head
E_3 (*Bird-Flu*)	bird, flu
E_4 (*Taiwan Issue*)	Taiwan, Chen, Shu, Bian
E_5 (*Iraq-War*)	Iraq, war, military, Hussein, Saddam,
E_6 (*Gas*)	victim, might, accident, gas

Table 1: 6 Bursty Events



(a) Bursty Event E_1 (b) Bursty Event E_2 (c) Bursty Event E_3

(d) Bursty Event E_4 (e) Bursty Event E_5 (f) Bursty Event E_6

Figure 7: 6 Bursty Events (Probability for Bursty Events)

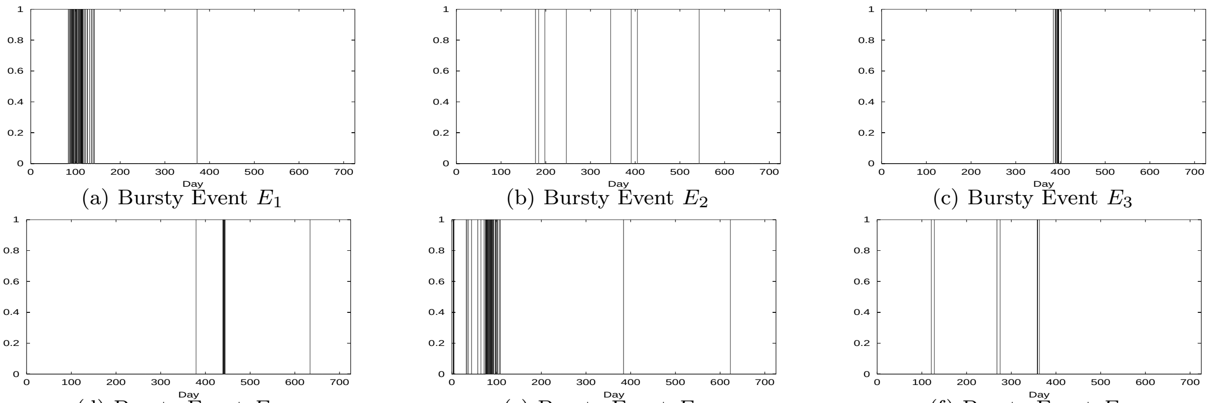
rithm (Algorithm 1) from the 373 bursty features. Table 1 gives the top 6 bursty features. Recall: with the *HB-Event* algorithm, the bursty event E_1 is first identified, from the total 373 bursty features. Those bursty features that appear in E_1 will be removed, and the second bursty event E_2 will be identified. The process repeats until all bursty events are identified. The maximum, minimum, and average size of bursty events are, 12, 2 and 3.46. Note: the names of the bursty events in Table 1 are named by human to match the real events. The 6 bursty events are shown in Figure 7.

As shown in Figure 6, `Sars`, `Outbreak` and `Iraq` have rather high similarity in their feature distributions (Figure 6 (d), (e) and (j)). However, `Sars` and `Outbreak` should be grouped together as bursty features for the bursty event *SARS*, and `Sars` and `Iraq` should not be grouped together for any bursty event. We explain why our feature-pivot clustering approach can correctly group `Sars` and `Outbreak` together, but not `Sars` and `Iraq`.

- Grouping bursty features `Sars` and `Iraq`: The total numbers of documents that contain the bursty feature `Sars` and `Iraq` during the bursty period are $|D_{Sars}| = 3,240$ and $|D_{Iraq}| = 2,404$, respectively. In total, there are 153 documents reporting both events at the same time, such as $|D_{Sars} \cap D_{Iraq}| = 153$, and there are 5,491 documents that contain either `Sars` or `Iraq` such as $|M| = |D_{Sars} \cup D_{Iraq}| = 5,491$ (Eq. (11)). Consider whether `Sars` and `Iraq` shall be grouped. If they are grouped together, with Eq. (10), $P(\mathsf{D}|E_k) = (3240/5491) \times (2404/5491) = 0.258$. The cost c becomes $0.190 + 0.588 = 0.778$ where $P(E_k) = 0.646$. If they are not grouped, then with Eq. (10), $P(\mathsf{D}|E_k) = (3240/5491) \times (1 - 2404/5491) = 0.332$. The cost c becomes $0+0.479 = 0.479$ where $P(E_k) = 1$. Therefore, the two bursty features should not be grouped together.

- Grouping `Sars` and `Outbreak`: The total numbers of documents that contain the bursty feature `Sars` and `Outbreak` during the bursty period are $|D_{Sars}| = 3,240$ and $|D_{outbreak}| = 2,254$, respectively. In total, there are 1,854 documents reporting both events at the same time, such as $|D_{Sars} \cap D_{Outbreak}| = 1,854$, and there are 3,640 documents that contain either `Sars` or `Outbreak` such as $|M| = |D_{Sars} \cup D_{Outbreak}| = 3,640$ (Eq. (11). If `Sars` and `Outbreak` are grouped together, $P(\mathsf{D}|E_k) = (3240/3640) \times (2254/3640) = 0.551$. The cost c becomes $0.043 + 0.259 = 0.302$ where $P(E_k) = 0.906$. If `Sars` and `Outbreak` are not grouped together, $P(\mathsf{D}|E_k) = (3240/3640) \times (1 - 2254/3640) = 0.338$. The cost c becomes $0 + 0.471 = 0.471$ where $P(E_k) = 1$. As a result, we should group `Sars` and `Outbreak` together.

Figure 8 shows the hot periods of the same bursty events (Figure 7). We highlight some observations in



(a) Bursty Event E_1 (b) Bursty Event E_2 (c) Bursty Event E_3

(d) Bursty Event E_4 (e) Bursty Event E_5 (f) Bursty Event E_6

Figure 8: 6 Bursty Events (Hot Periods)

Table 1 and Figure 8 in details as case studies, referring to Figure 6. The bursty event E_1 (*SARS*) includes the bursty features `Sars` (Figure 6 (d)), `Outbreak` (Figure 6 (e)) and `Disease` (Figure 6 (f)), because all have similar feature distribution. In a similar fashion, the bursty event E_5 (*Iraq-War*) includes the bursty features such as `Iraq` (Figure 6 (j)), `Military` (Figure 6 (k)) and `Saddam` (Figure 6 (l)) for having the high similarity among their feature distributions.

The bursty event E_2 (*Legislation*) includes the bursty features such as `Article` (Figure 6 (p)) and `Law` (Figure 6 (q)) in a less obvious way. Consider the cost function (Eq. (8)) which has two components, $P(\mathsf{D}|E_k)$ and $P(E_k)$. For E_2, although $P(E_k)$ is small, 0.3, as the overlapping area between these features is small), $P(\mathsf{D}|E_k)$ is large. Most documents contain all these features during the period when the features are bursty. Some details are given below for E_2. There was a massive demonstration against the Hong Kong Basic Law Article 23 legislation on 1st July 2003. In the aftermath of the demonstration, on 6th July 2003, the Hong Kong government announced that the second reading of the law was to be postponed, and the head of security (Mrs. Yip) resigned position on 16th July 2003 that political commentators attributed the resignation to the protests over the Basic Law Article 23 legislation. Therefore, `Article` and `Law` co-occurred together during the corresponding bursty periods. Apart from the aforementioned periods, there are two major bursts for E_2. One is on 5th September 2003 and the other is on 23rd November 2003. On 5th September 2003, the Chief Executive of Hong Kong announced that Article 23 legislation would be withdrawn and there was no timetable for its re-introduction. On 23rd, November 2003, it is the district council election. It offered the first opportunity for voters to express their opinions since July 1st. The bursty feature `Rally` shows the similarity to the bursty feature `Article`, because demonstration was usually associated with rally. The major difference between the feature distribution of `Rally` and `Article` is that `Rally` has a different burst period on 2nd July 2004, because on 1st July 2004, there was another massive demonstration, which included over 300,000 people. In short, all the bursty features are strongly interrelated to each others. The similar observations can be observed for E_3 (*Bird-Flu*), E_4 (*Taiwan-Issue*), and E_6 (*Gas*).

5 Related Work

Topic detection and tracking (TDT) is the major area that tackles the problem of discovering events from a stream of news stories [2, 18, 27, 26, 3, 4, 27, 26, 21]. They all use similar techniques for event detection, that is to cluster similar documents together to form events. We discussed in Section 2 that this approach cannot be directly applied to our hot bursty events detection. In addition to the quality issue whether it can find bursty events, there is an efficiency issue. The size of the corpus usually makes the clustering problem become difficult. The work in [21, 27, 26] attempted to improve the efficiency of clustering, however, it further introduces more parameters to be tuned.

[9] shows how to extract bursty features from text streams based on modeling the text stream using an infinite-state automaton, where bursts are modeled as state transitions. Our work is different, because we do not only attempt to extract bursty features, but also, as one step further, attempt to group the related bursty features into bursty events, as well as to determine the hot periods of bursty events. Note: for the state transition in [9], it needs to define the probability for each state, whereas our feature-pivot clustering approach is parameter free.

The work in [19, 20, 17, 14] studied bursty events in a text stream using a similar model formulation. For each feature, such as *name entity* and *noun phase*, in the corpus, they performed a χ^2 test to identify days on which the occurrences yield a value above a predefined threshold, and group the consecutive days that meet this criteria into events. Our approach is different. First, we do not need any complex parameter tuning, whereas [19, 20] need to predefine several thresholds by the user. Second, the authors showed that it is difficult to construct an event which lasts for a long period. The reason is that a period may be broken into parts because the specific feature does not occur frequently in every consecutive time windows. We do not need to explicitly define whether a feature is bursty or not in a time window. We model each of the bursty features as time-series of probability, and group the bursty features into bursty events.

[22] proposes methods for mining knowledge from the query logs of the MSN search engine by building a time-series for each query term, where the elements of the time-series are the number of times that a query is issued on a day. By observing the patterns of the time-series, [22] attempts to find the periods where time-series becomes bursty. In transforming the features into time-series, they adopted the techniques of moving average, which is sensitive to the length of time windows. They combined the features with similar patterns only in the time-series, but did not pay attention to the content. We find the bursty events based on both the time and content information.

The related works also include visualization techniques [7, 24, 11]. Their focus is on the visualization (how to present the information based on a set of given events), rather than on the detection side (how to identify a set of events).

6 Conclusion

In this paper, we studied a new problem, called hot bursty events detection in a sequence of chronologically ordered documents, where a bursty event is a set of bursty features appearing in certain time windows. Taken a set of bursty features as positive features in a set of positive examples (labeled by the corresponding bursty event). The new problem is important, because, as the first attempt, it attempts to find a complete solution to build text classifiers without any human assistance, along the line of (1) identifying positive features, (2) enlargement of positive features, (3) identification/enlargement of negative examples, and (4) text classification building. Except for the first issue, which is the focus of this paper, the other issues have been addressed in the recent papers, and are known as partially supervised text classification.

We proposed a parameter free probabilistic approach for effectively and efficiently identifying bursty events, called feature-pivot clustering approach. Our algorithm, proposed in this paper, is an off-line algorithm which has its potential to be extended to an on-line algorithm, because it mainly uses distributions, which can be handled using the up-to-date data streaming mining techniques. In the feature-pivot clustering approach, we utilize both time and content information in a very effective way. We identify bursty features by their distributions, and group strongly interrelated bursty features as bursty events. Our approach groups the interrelated bursty features together if they appear in the same documents frequently enough, because our approach also pays attention to the content. In other words, it is unlikely that a bursty event contains irrelevant bursty features together. It is important to know that it can be achieved without parameter tuning and estimation.

We conduct experimental studies using a two-year news stories archived from a major English news paper in Hong Kong, South China Morning Post. The testing results showed that the parameter free feature-pivot clustering approach can detect the bursty events with a high success rate.

References

[1] R. Agrawal, K.-I. Lin, H. S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceedings of 21th International Conference on Very Large Data Bases (VLDB'95)*, 1995.

[2] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR'98)*, 1998.

[3] T. Brants and F. Chen. A system for new event detection. In *Proceedings of the 26th ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR'03)*, 2003.

[4] M. Connell, A. Feng, G. Kumaran, H. Raghavan, C. Shah, and J. Allan. UMass at tdt 2004. In *2004 Topic Detection and Tracking Workshop (TDT'04)*, 2004.

[5] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data (SIGMOD'94)*, 1994.

[6] G. P. C. Fung, J. X. Yu, H. Lu, and P. S. Yu. Text classification without negative labeled examples. In *Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, 2005.

[7] G. J. F. Jones and S. M. Gabb. A visualisation tool for topic tracking analysis and development. In *Proceedings of the 25th ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR'02)*, 2002.

[8] E. J. Keogh and P. Smyth. A probabilistic approach to fast pattern matching in time series databases. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD'97)*, 1997.

[9] J. M. Kleinberg. Bursty and hierarchical structure in streams. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining (KDD'02)*, 2002.

[10] X. Li and B. Liu. Learning to classify texts using positive and unlabeled data. In *Proceedings of 2003 International Joint Conference on Artificial Intelligence (IJCAL'03)*, 2003.

[11] N. E. Miller, P. C. Wong, M. Brewster, and H. Foote. Topic islands – a wavelet-based text visualization system. In *Proceedings of the 9th IEEE Visualization*, 1998.

[12] D. C. Montogomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Inc., second edition, 1999.

[13] S. Morinaga and K. Yamanishi. Tracking dynamics of topic trends using a finite mixture model. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining (KDD'04)*, 2004.

[14] R. Papka and J. Allan. On-line new event detection using single pass clustering. Technical Report IR–123, Department of Computer Science, University of Massachusetts, 1998.

[15] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24 (5):513–523, 1988.

[16] F. Seabastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34 (1):1–47, 2002.

[17] D. A. Smith. Detecting and browsing events in unstructured text. In *Proceedings of the 25th ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR'02)*, 2002.

[18] M. Spitters and W. Kraaij. TNO at TDT2001: Language model-based topic detection. In *2001 Topic Detection and Tracking Workshop (TDT'01)*, 2001.

[19] R. C. Swan and J. Allan. Extracting significant time varying features from text. In *Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM'98)*, 1998.

[20] R. C. Swan and J. Allan. Automatic generation of overview timelines. In *Proceedings of the 23rd ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR'00)*, 2000.

[21] D. Trieschnigg and W. Kraaij. Hierarchical topic detection in large digital news archives. In *Proceedings of the 5th Dutch Belgian Information Retrieval workshop*, 2005.

[22] M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data (SIGMOD'04)*, 2004.

[23] P. Willett. Recent trends in hierarchic document clustering: A critical review. *Information Processing and Management*, 24 (5):577–597, 1988.

[24] P. C. Wong, W. Cowley, H. Foote, E. Jurrus, and J. Thomas. Visualizing sequential patterns for text mining. In *Proceedings of the 2000 IEEE Symposium on Information Visualization*, 2000.

[25] Y. Yang, T. Ault, T. Pierce, and C. W. Lattimer. A study on thresholding strategies for text categorization. In *Proceedings of the 23rd ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR'00)*, 2000.

[26] Y. Yang, J. Carbonell, R. Brown, T. Pierce, B. T. Archibald, and X. Liu. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems*, 14 (4):32–43, 1999.

[27] Y. Yang, T. Pierce, and J. Carbonell. A study on retrospective and on-line event detection. In *Proceedings of the 21st ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR'98)*, 1998.