Location-Based Burst Detection Algorithm in Spatiotemporal Document Stream

Keiichi Tamura and Hajime Kitakami
Graduate School of Information Sciences, Hiroshima City University, Hiroshima, Japan

Abstract— *The recent increasing interest in consumer generated media has resulted in numerous studies on extracting topics from documents in micro blogs. These documents are usually arranged in a temporal order and hence are represented as a document stream. This study focuses on a document stream that consists of documents containing creation time and location information. This type of document stream is referred to as a spatiotemporal document stream. We propose a novel algorithm for detecting location-based bursts in a spatiotemporal document stream. To evaluate the new location-based burst detection algorithm, we use an actual spatiotemporal document stream composed of crawling tweets on Twitter. Experimental results show that the algorithm can detect location-based bursts that vary with user location.*

Keywords: spatiotemporal data; text mining; burst detection; consumer generated content; topic detection and tracking;

1. Introduction

With the recent increasing interest in consumer generated media, a large number of documents are continuously created on the Internet. The number continues to exponentially increase specially owing to the widespread use of micro blogs (e.g., Twitter, Facebook, and Google+) for creating online documents [1]. The documents on the Internet are usually arranged a temporal order and hence are represented as a document stream. Topic extraction from a document stream has recently been gaining increasing attention and numerous studies on the text mining domain have been conducted [2], because the contents of these documents include variety types of hot topics such as news, social events, geographical topics, hobbies, and daily happenings.

Kleinberg's burst detection algorithm [3], [4] is one of the most effective techniques to extract topics from a document stream. Kleinberg defines a bursty word as a word that increasingly occurs in a document stream. Some words are highly bursty in the sense that the frequency of their occurrences spike when a particular event attracts public attention. Kleinberg's burst detection algorithm aims to find certain time periods in which there is a high frequency of the occurrence of words. When a word related to an attention-attracting event becomes highly bursty, the interarrival time between documents that include the word becomes smaller during particular time period. Therefore, this time period when a word becomes highly burst can be detected using the interarrival time between the documents.

Recently, the widespread use of smart devices with a global positioning system and geographical applications have resulted in an increase in the number of documents with location information (e.g., geotag). Consequently, many documents in a document stream not only have a creation time but also contain location information. In other words, documents in a docment stream have a spatiotemporal order. The contents of these documents include topics that are closely related to a particular location. Therefore, we need to detect burstiness while considering location information. However, there have been no attention on location-based burst detection algorithms.

If topic "A" is a hot topic in a particular region "B," then it contains useful information in the vicinity of region "B." However, topic "A" is not useful for users far away from region "B." In this case, we need to detect burstiness by considering location. While topic "A" has to be presented as a highly bursty topic for users in the vicinity of region "B," it has to be presented as not highly bursty for users far away from region "B." To satisfy this requirement, it is necessary to integrate location information into burst detection algorithms.

This study focuses on a document stream that consists of documents containing creation time and location information. We call this type of a document stream spatiotemporal document stream (SDS). In this paper, we propose a novel method for detecting location-based bursts in SDS. The main contributions of our study are as follows:

- To enable easy handling of SDS, we define the data model of a document in SDS.
- To detect location-based bursts in SDS, we extend Kleinberg's burst detection algorithm. In our extension, the influence factor of a document is defined as the distance between a user and the location where the document was created. The location-based burst detection algorithm adjusts the burst using the influence factor of the document.
- To evaluate the new location-based burst detection algorithm, we use an actual SDS composed of crawling tweets on Twitter. The experimental results show that the algorithm can detect location-based bursts that vary with user location.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 presents a brief explanation on Kleinberg's burst detection algorithm. Section 4 presents the problem definition of location-based burst detection and a novel method for burst detection in

SDS. Section 5 shows the experimental results. Finally, section 6 concludes this paper.

2. Related Work

Since the Internet gained widespread use, topic detection and tracking [5] has been the most important research area in the text mining domain. In particular, because of the wide spread creation of various online documents on the Internet, there have been many studies on topic detection and tracking in document streams. To detect topics in a document stream that have attracted many people, burstiness is the simplest but the most effective criterion. Therefore, with the increased interest in extracting topics from online documents, such as news, message boards, blogs, micro blogs, several algorithms have been developed to detect bursts in document streams [3], [4], [6], [7], [8], [9], [10], [11].

There are a number of studies on burst detection algorithms. The most significant impact on many studies is Kleinberg's burst detection algorithm [3], [4]. It is based on a queuing theory for bursty network traffic. The details of the algorithm are explained later. It is used for various document streams such as e-mail [3], blogs [11], online publications [12], bulletin board, and social tags [13]. Moreover, there are some studies about the extension of Kleinberg's burst detection algorithm. In particular, Qi He et al. [14] proposed a clustering algorithm for documents in a document stream that uses bursty feature representation as a feature vector for clustering. Leskovec et al. [15] formulated memes as patterns of words by using a scalable clustering approach.

Recently, geographical topic detection and tracking [16], [17], [18], [19] has been attracting increasing attention, because the number of geographical documents have been increasing on the Internet. Sakaki et al. [16] proposed a model for real-time event detection using tweets on Twitter. To detect the location where an event has occurred, they used Kalman filtering and particle filtering, which are widely used for location estimation in ubiquitous computing. Cheng et al. [17] developed a classification method that uses words in tweets with a strong local geo-scope and a lattice-based neighborhood-smoothing model for refining the estimation of a user's location. Yin et al. [18] proposed a method to discover different topics in geographical regions. Furthermore, Yang et al. [19] developed a method to reveal the appearance and disappearance of topics in different regions.

There are numerous studies on burst detection and geographical topic detection and tracking. However, to the best of our knowledge, until now, there is no study that attempts to detect location-based bursts in SDS. This paper describes a data model for SDS and proposes a method for detecting location-based bursts. If location-based bursts can be detected in SDS, we can provide topics that are accurate and helpful for users who want to know local information.

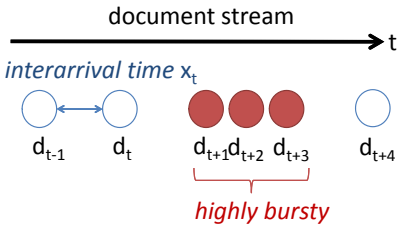


Fig. 1: Example of a Document Stream.

3. Preliminaries

This section presents the definition of a document stream and a burst, and briefly explains Kleinberg's burst detection algorithm.

3.1 Document Stream

A document stream is similar to a data stream. It is defined as a sequence of documents arranged in a temporal order. Fig. 1 shows an example of a document stream. In Fig. 1, the documents are posted in temporal order. The time interval x_t between document d_{t+1} and document d_t is called the interarrival time. Examples of a document stream include, but are not limited to, tweets on Twitter. Tweet i is represented as document d_i. The interarrival time x_i is defined as the time interval between the posting time of tweet $i+1$ and that of tweet i.

3.2 Burst

As the number of documents that include a word related to a particular event increases in a document stream, the interarrival time between these documents becomes smaller. A word is considered highly bursty during a period in which the interarrival time is shorter than usual. In addition, the period is described as bursty. For example, in Fig. 1, the interarrival time between d_{t+1} and d_{t+2}, and that between d_{t+2} and d_{t+3} are smaller than the other interarrival times. In this case, we can observe that this period is highly bursty.

3.3 Kleinberg's Burst Detection Algorithm

Kleinberg defined a model with an infinite-state automaton in which bursts are represented as state transitions. Assuming that there are m states in the automaton, each interarrival time is a probabilistic output that depends on the internal states of the infinite-state automaton. In the model, a state is associated with a burstiness state and a higher state indicates higher burstiness.

Let the sequence of interarrival times between document postings be $x = (x_1, x_1, \cdots, x_n)$. The problem is defined to find an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize the following cost function:

$$C(s|x) = \sum_{i=1}^{n-1} \tau(s_i, s_{i+1}) + \sum_{i=1}^{n} (-\ln f_{s_i}(x_i)). \quad (1)$$

The function $\tau(i, j)$ returns a state-transition cost from state i to state j. It is defined as

$$\tau(i,j) = \begin{cases} (j-i)\gamma \ln n, & \text{if } j > i, \\ 0, & \text{otherwise}, \end{cases} \quad (2)$$

where $\gamma(>0)$ is a user-given parameter and n is the number of documents in the document stream being observed. Equation 2 indicates that moving to a higher state incurs a cost and moving to a lower state incurs no cost.

The function $f_k(x_i)$ is the exponential density function for the probability of outputting the interarrival time x_i in state k and defined as

$$f_k(x_i) = \lambda_k e^{-\lambda_k x_i}, \tag{3}$$

where λ_k is the arrival rate of documents associated with state k and is defined as

$$\lambda_k = \frac{n}{T}\beta^k, \tag{4}$$

where n is the number of documents, T is the entire time range and $\beta(>1.0)$ is a user-given parameter. Equation 4 indicates that a higher state has a higher arrival rate.

The Viterbi algorithm for hidden Markov models, which is a dynamic programming approach, is the most effective solution for determining an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize Equation 1. First, we calculate the following cost $C_j(i)$:

$$C_j(i) = -\ln f_j(x_i) + min_l(C_l(i-1) + \tau(l,j)), \tag{5}$$

where $C_j(i)$ is the minimum cost of a state-transition sequence that ends with state j at the i-th time-interval in the document stream. Equation 5 can be calculated using the previous $(i-1)$-th $C_l(i-1)$ $(0 \le l \le m-1)$. Second, we find the minimum cost in $C_j(n)(0 \le j \le m-1)$. Suppose that the minimum cost in $C_j(n)(0 \le j \le m-1)$ is $C_{min}(n)$. Finally, we trace back with $C_{min}(n)$ as the starting point.

4. Location-based Burst Detection

This section presents the problem definition and a novel burst detection algorithm for spatiotemporal document stream (SDS).

4.1 Model and Problem Definition

Suppose that there are n documents in SDS. Let d_i denote the i-th document in SDS; then d_i consists of four items;

$$d_i = <id_i, content_i, ctime_i, cposition_i>, \tag{6}$$

where id_i is the identifier of the document, $content_i$ is the content (e.g., title, textdata, and tags), $ctime_i$ is the creation time of the document, and $cposition_i$ is the location where d_i was created or is located (e.g., latitude and longitude).

Fig. 2 shows an example of SDS comprising six documents. Each document d_i has its own creation time in the time line and a location on the geographical coordinate space.

Let W be a set of all words appearing in SDS. The word time-series data w_i is defined as $w_i = < word_i, CTT_i, CTP_i >$, where $word_i \in W$ is string data, CTT_i is the creation time sequence of the documents that include $word_i$ in their content, and CTP_i is the location sequence of the documents that include $word_i$.

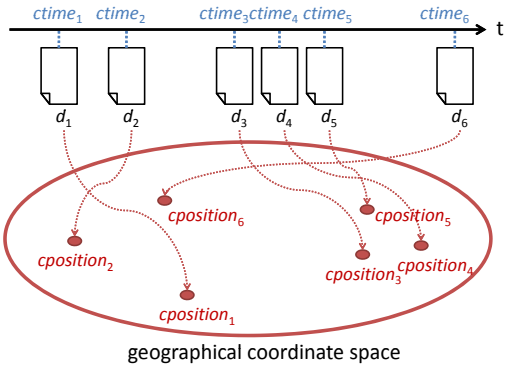


Fig. 2: Example of a Spatiotemporal Document Stream.

$$CTT_i = (ctt_{i,1}, ctt_{i,2}, \cdots, ctt_{i,tnum(i)}), \tag{7}$$

$$CTP_i = (ctp_{i,1}, ctp_{i,2}, \cdots, ctp_{i,tnum(i)}), \tag{8}$$

where $tnum(i)$ returns the number of documents that include $word_i$. The j-th element of CTT_i is represented as $CTT_i[j](= ctt_{i,j})$.

For example, in Fig. 2, suppose that $word_k$ is included in three documents $\{d_3, d_4, d_5\}$. In this case, the creation time sequence of $word_k$ is $CTT_k = (ctt_{k,1}, ctt_{k,2}, ctt_{k,3})$, where $ctt_{k,1} = ctime_3$, $ctt_{k,2} = ctime_4$, and $ctt_{k,3} = ctime_5$. Similarly, the location sequence of $word_k$ is $CTP_k = (ctp_{k,1}, ctp_{k,2}, ctp_{k,3})$, where $ctp_{k,1} = cposition_3$, $ctp_{k,} = cposition_4$, and $ctp_{k,3} = cposition_5$.

Here, let the interarrival time sequence of $word_i$ be $IAT_{CTT_i} = (iat_{i,1}, iat_{i,2}, \cdots, iat_{i,tnum(i)})$, where each $iat_{i,j}$ indicates an interarrival time:

$$iat_{i,j} = \begin{cases} ctt_{i,j} - stime, & \text{if } j = 1, \\ ctt_{i,j} - ctt_{i,j-1}, & \text{otherwise,} \end{cases} \tag{9}$$

$stime$ is the start time of SDS.

The goal of this study is to detect the location-based burst that varies with the user position up. In other words, for each $w_i \in W$, find an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize the $C(s|IAT_{CTT_i})$ associated with up.

For instance, suppose that d_3, d_4, and d_5 include the k-th word $word_k$ associated with an topic, and $word_k$ is highly bursty from $ctime_3$ to $ctime_4$ as defined by Kleinberg's burst detection algorithm. Then we need to show users located at a distance from the document creation location that $word_k$ is not highly bursty. This is because distant users are not interested in the topic. In contrast, $word_k$ is highly bursty for users in the vicinity of the document creation location because nearby users would be interested in the topic.

4.2 Main Concept

The simplest intuitive way to find location-based bursts in SDS is to detect bursts from documents that exist around users. Fig. 3 shows an example. There are two users;

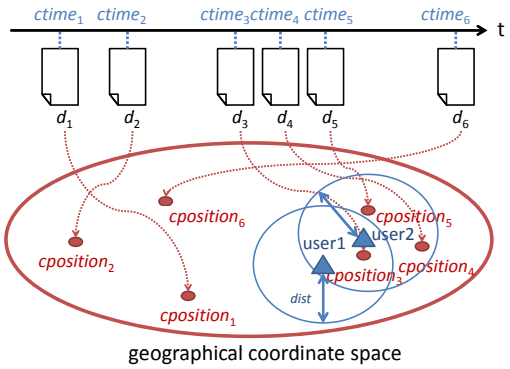


Fig. 3: Cutoff-Distance-Based Burst Detection.

$user1$ and $user2$. Each user uses only the documents that satisfy with $distance(d_i, user) \leq dist$, where the function $distance$ returns the distance between d_i and the $user$. The value of $dist$ is a cutoff distance given as a user-specific parameter. In Fig. 3, there is one document within distance $dist$ from $user1$. Moreover, there are three documents within distance $dist$ from $user2$.

This simple approach using cutoff distance is called cutoff-distance-based burst detection. In this approach, for each $w_i \in W$, we find an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize $C(s|IAT_{CTT_i'})$, where CTT_i' is the creation time sequence of documents that satisfy $calc\_distance(ctp_{i,j}, up) \leq dist$. Function $calc\_distance$ is returns the distance between the location of document $ctp_{i,j}$ and the user position up.

Algorithm 1 shows the cutoff-distance-based burst detection algorithm. First, we determine CTT_i' from CTT_i by filtering using the cutoff distance $dist$. Function $append\_sequence(S, item)$ appends $item$ to the tail of sequence S. Second, we generate the interarrival time sequence $IAT_{CTT_i'}$. Finally, we find an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize $C(s|IAT_{CTT_i'})$ using function KBD.

Although the cutoff-distance-based burst detection is the easiest way to detect bursts around users, it is largely dependent on the cutoff distance. For example, suppose that word k is highly bursty from $ctime_3$ to $ctime_5$ as shown in Fig. 3, and $user1$ and the $user2$ are close. However, the cutoff-distance-based burst detection shows $user1$ that word k is not highly bursty because there is only one document within $dist$ that include word k. This issue can be avoided by setting a large value for the cutoff distance $dist$. This results in another issue: burst detections are visibly affected by documents far away from users.

To address this issue, we integrate the influence factor of a document into Kleinberg's burst detection algorithm. The influence factor of a document is defined as the distance between a user and the location. The interarrival times are corrected using by the influence factors of documents. Interarrival time is the main factor for state transitions in Kleinberg's burst detection algorithm. Therefore, we correct the sequence of inter-arrival time $x = (x_1, x_2, \cdots, x_n)$ in accordance with the influence factors of documents.

Algorithm 1: Cutoff-Distance-Based Burst Detection

input : cutoff distance $dist$, position of the user up, word time-series data w_i, and parameter list for burst detection $params$
output: optimal state-transition sequence S

$CTT_i' \leftarrow ()$ /* make a empty sequence */
for $j \leftarrow 1$ **to** $|w_i{->}\ CTT_i|$ **do**
  $ctp \leftarrow w_i{->}\ CTP_i[j]$
  if `calc_distance` $(ctp,up) \leq dist$ **then**
    $CTT_i' \leftarrow$ `append_sequence` (CTT_i', ctp)
$IAT_{CTT_i'} \leftarrow ()$ /* make a empty sequence */
for $j \leftarrow 1$ **to** $|CTT_i'|$ **do**
  if $j = 1$ **then**
    $pctt \leftarrow stime$
  else
    $pctt \leftarrow CTT_i'[j-1]$
  $iat \leftarrow CTT_i'[j] - pctt$
  $IAT_{CTT_i'} \leftarrow$ `append_sequence` $(IAT'_{CTT_i'}, iat)$
$s \leftarrow$ `KBD` $(IAT_{CTT_i'}, params)$
return s

4.3 Algorithm

The location-based burst detection algorithm, unlike the cutoff-distance-based approach, does not filter documents according to distance. It corrects the sequence of interarrival time IAT_{CTT_i} by using the influence factors of documents including $word_i$. To correct the interarrival time sequence $x = (x_1, x_2, \cdots, x_n)$, time is added to each interarrival time x_i in accordance with the distance between document d_i and the user. As a result, the interarrival times of documents created far away from the user become longer than their actual interarrival times.

We define the corrected interarrival time as follows:

$$iat'_{i,j} = \begin{cases} ctt_{i,j} - stime + \delta(ctp_{i,j}, up), & \text{if } j = 1, \\ ctt_{i,j} - ctt_{i,j-1} + \delta(ctp_{i,j}, up), & \text{otherwise,} \end{cases} \quad (10)$$

where function δ returns a correction value.

Algorithm 2 shows the algorithm for location-based burst detection. The algorithm uses all the documents that include $word_i$. First, we generate the interarrival time sequence IAT'_{CTT_i} using by Equation 10. Second, we find an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize $C(s|IAT'_{CTT_i})$ using function KBD.

There are two methods for interarrival time correction. One is time-difference-based correction and the other is forgetting-factor-based correction. These two correction methods are described as follows:

Time-Difference-based Correction

In this correction, time difference is used for the calculation of correction value. The function $calc\_distance$

Algorithm 2: Location-Based Burst Detection

input : cutoff distance $dist$, user position up, word time-series data w_i, and parameter list for burst detection $params$
output: optimal state-transition sequence S

$IAT'_{CTT_i} \leftarrow ()$ /* make a empty sequence */
for $j \leftarrow 1$ **to** $|w{->}\, CTT_i|$ **do**
  if $j = 1$ **then**
    $pctt \leftarrow stime$
  else
    $pctt \leftarrow w{->}\, CTT_i[j-1]$
  $iat \leftarrow w{->}\, CTT_i[j] - pctt + \delta(w_i{->}\, CTP_i[j], up)$
  $IAT'_{CTT_i} \leftarrow$ `append_sequence` (IAT'_{CTT_i}, iat)
$s \leftarrow$ `KBD` $(IAT'_{CTT_i}, params)$
return s

returns the distance between location dp of the document and user up and SP is Earth's rotation rate.

$$\delta(dp, up) = \frac{calc\_distance(dp, up)}{SP} \tag{11}$$

Forgetting-Factor-based Correction

In this correction, a forgetting factor[20], [21] is used to calculate the correction value. Documents gradually lose their weight (or memory) according to distances.

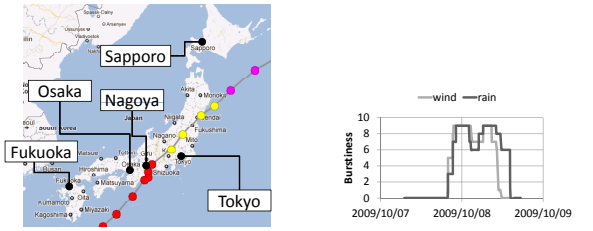
$$\delta(dp, up) = total\_time \times \alpha^r,$$
$$r = \frac{calc\_distance(dp, up) - d_{min}}{d_{max} - d_{min}}, \tag{12}$$

where α is a forgetting factor, $total\_time$ is elapsed time between the start time and current time, d_{max} is the maximum distance between the user and the locations where the documents were created, and d_{min} is the minimum distance between the user and the locations where the documents were created.

5. Experimental Results

To evaluate the location-based burst detection algorithm, we used an actual SDS that is composed of crawling tweets on Twitter about typhoon Melor in 2009. The number of tweets is 504. The time period is from 07:00:11 October 7, 2009 to 13:35:01 October 9, 2009. Typhoon Melor resulted in landfall at the Chita Peninsula in Japan on October 8 after 5 a.m. (JST). Rainfall increased at many places; in particular heavy rains were observed in Osaka, Mie, Tokyo and the Saitama Prefecture. Fig.4(a) shows the path of typhoon Melor.

In the experiments, we select two words; "wind" and "rain," which are the first and the second score in tf*idf results respectively. When the typhoon was on its way toward users, these two words generates the most interest.



(a) Typhoon track. (b) Burst of "wind" and "rain."

Fig. 4: Typhoon Melor.

The five major cities of Japan, Fukuoka, Osaka, Nagoya, Tokyo, and Sapporo are set as the users' positions (Fig. 4(a)). Since the typhoon was headed toward areas near Nagoya, it was a topic concern in the nearby cities of Osaka and Tokyo at that time. The effects of the typhoon began to first appear in Fukuoka because the typhoon went from southwest to northeast. Futhermore, the effects of the typhoon were felt last in Sapporo.

Fig. 4(b) shows the bursts of "wind" and "rain" which are extracted using Kleinberg's burst detection algorithm. Parameter β is set to 1.1 and γ is set to 0.05. The typhoon landed at 5:00 a.m. on October 8 and left the Japanese islands in the evening. Both words are highly bursty between the night of October 7 and the evening of October 8. The degree of burstiness for the word "rain" remained high until the end of the time period. The damage to the Japan islands from the typhoon not only due to wind but also rain. Therefore, concerns about rain continued until the end of the time period, where as concerns about wind decreased earlier in the time period. This result indicates that Kleinberg's burst detection can extract the bursts of words.

Fig. 5 shows the bursts of the word "wind" extracted using the location-based burst detection algorithm. In the graphs, TDC and FFC are the proposed method. TDC indicates that time-difference-based correction is used and FFC indicates that forgetting-factor-based correction is used. CDBD indicates the Distance-Cutoff-based Burst Detection method. In CDBD, the cutoff distance $cutoff$ is set to $150km$. Fig. 5(a), Fig. 5(b) and Fig. 5(c) are the results at Fukuoka. Fig. 5(d), Fig. 5(e) and Fig. 5(f) are the results at Osaka. Fig. 5(g), Fig. 5(h) and Fig. 5(i) are the results at Nagoya. Fig. 5(j), Fig. 5(k) and Fig. 5(l) are the results at Tokyo. Fig. 5(m), Fig. 5(n) and 5(o) are the results at Sapporo. Similarly, Fig. 6 shows the bursts of the word "rain" extracted using the location-based burst detection algorithm.

The graphs shows that the words "wind" and "rain" are bursty in Fukuoka during the initial time period. Then, the degree of burstiness quickly reduces. Since Fukuoka is the most west of five cities, the attention paid to the typhoon had risen there earlier than other locations. Moreover, since the typhoon left far away from Fukuoka, the words "wind" and "rain" became less interesting topics in Fukuoka. Therefore, these results provide accurate information for users in Fukuoka. Similarly, in Osaka, the burst appeared from the time when only a few is late compared with Fukuoka. Since it is closer to the typhoon than Fukuoka,

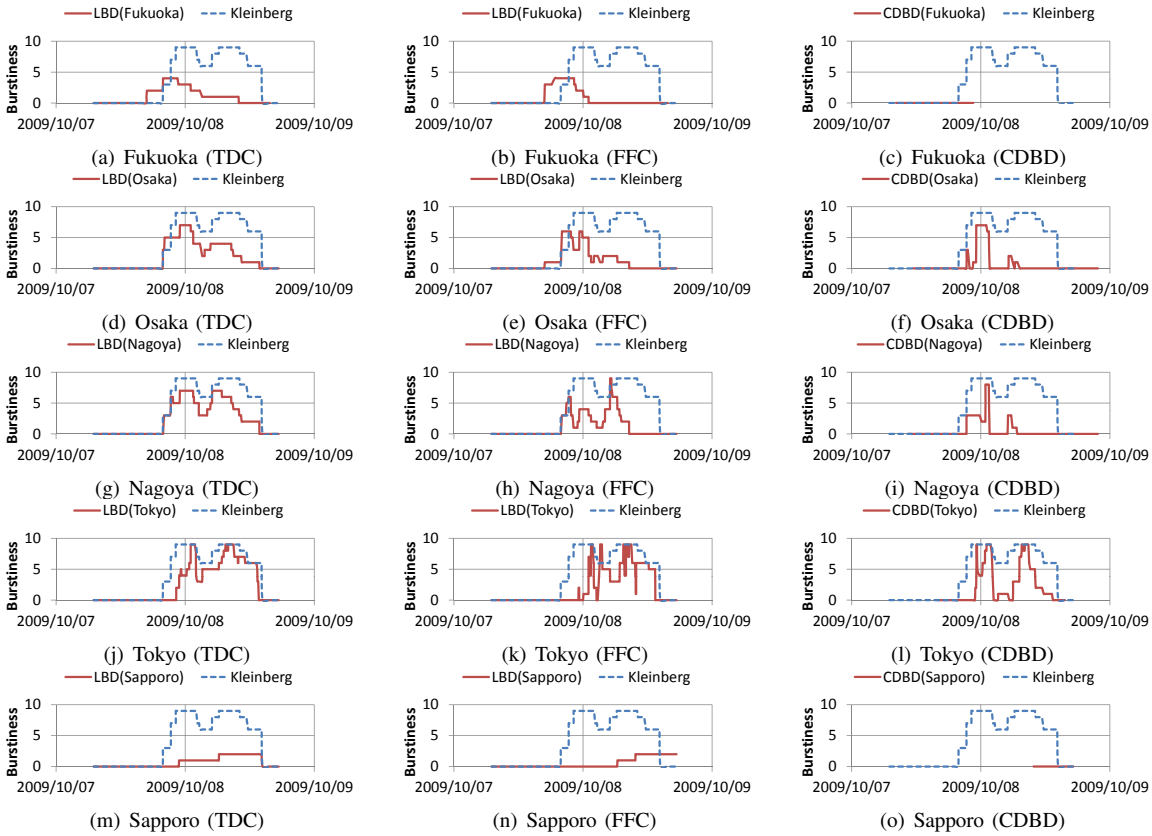


Fig. 5: Results of Word "wind."

the burst state of Osaka continued longer than that of Fukuoka. Nagoya is the closest to where the typhoon resulted in landfall. This resulted in those words being the most bursty around the time of landfall. Tokyo is east of Nagoya. A burst was initiated there slightly after Nagoya. A burst appeared the latest in Sapporo as the typhoon approached it last. Futhermore, the results accurately reflected the typhoon's minimal influence in Sapporo.

On the other hand, CDBD detected the words "wind" and "rain" are not bursty in Fukuoka and Supporo (Fig. 5(c), Fig. 5(o), Fig. 6(c) and Fig. 6(o)). This is because CDBD only considers documents within the cutoff distance $cutoff$. Moreover, in CDBD, burst of "wind" and "rain" appear in Tokyo when the typhoon made landfall. The landfall location is not located within $150km$. Therefore, almost all documents are located in more than $150km$ from Tokyo. This resulted in no bursty appearance in Tokyo at that time.

Fig. 7 shows the results of the word "wind" using CDBD with four different cutoff distances in Tokyo. If the cutoff distance is small, the period of burst is short, whereas, if the cutoff distance is large, the period of burst is long. In CDBD, it is difficult for users to select the best cutoff-distance. The proposed location-based burst detection algorithm dose not need any cutoff-distance. Therefore, our algorithm can detect location-based bursts easier and more correct than CDBD.

6. Conclusion

This study focuses on a document stream that consists of documents containing creation time and location information. We call this type of document stream a spatiotemporal document stream (SDS). We propose a novel algorithm for detecting location-based bursts in SDS. To evaluate the new location-based burst detection algorithm, we use an actual spatiotemporal document stream composed of crawling tweets on Twitter. The experimental results show that the algorithm can detect location-based bursts that vary with user location. In future work, we need more performance evaluations and comparsions with other work.

Acknowledgment

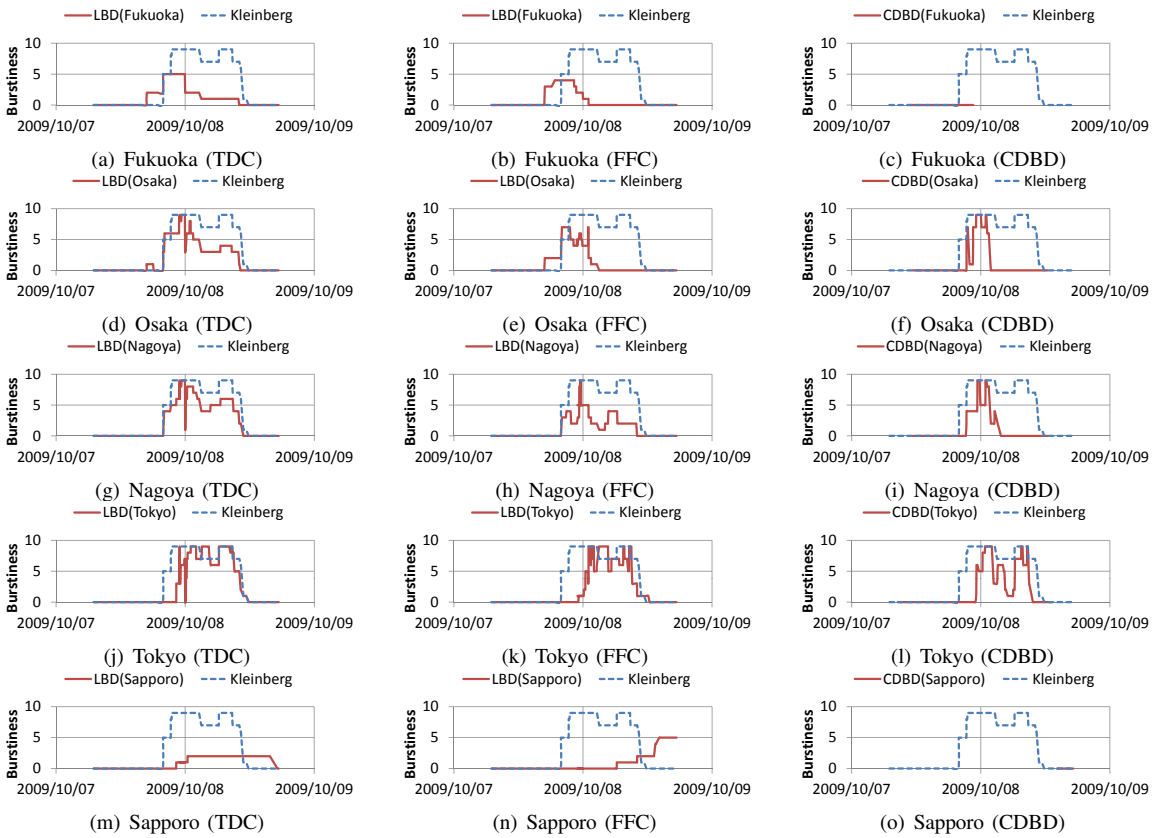
This work was supported in part by a Grant-in-Aid for Young Research (B) (No.23700124) from Ministry of Education, Culture, Sports, Science and Technology in Japan and a Grant-in-Aid for Scientific Research (C) (2) (No.20500137) from the Japanese Society for the Promotion of Science, Japan.

References

[1] M. Ebner and M. Schiefner, "Microblogging - more than fun," in *In Proceedings of the IADIS Mobile Learning Conference 2008*, 2008, pp. 155–159.

[2] J. M. Kleinberg, *Temporal Dynamics of On-Line Information Streams*. Springer, 2006.

[3] J. M. Kleinberg, "Bursty and hierarchical structure in streams," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 91–101.



(a) Fukuoka (TDC) (b) Fukuoka (FFC) (c) Fukuoka (CDBD)

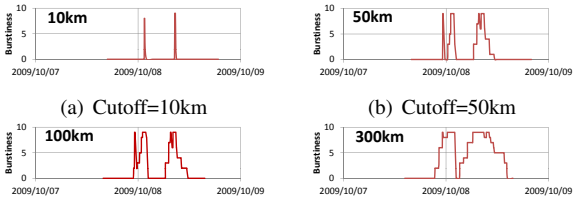
(d) Osaka (TDC) (e) Osaka (FFC) (f) Osaka (CDBD)

(g) Nagoya (TDC) (h) Nagoya (FFC) (i) Nagoya (CDBD)

(j) Tokyo (TDC) (k) Tokyo (FFC) (l) Tokyo (CDBD)

(m) Sapporo (TDC) (n) Sapporo (FFC) (o) Sapporo (CDBD)

Fig. 6: Results of Word "rain."



(a) Cutoff=10km (b) Cutoff=50km

(c) Cutoff=100km (d) Cutoff=300km

Fig. 7: Comparisons of Results of Word "wind" in Tokyo.

[4] J. M. Kleinberg, "Bursty and hierarchical structure in streams," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 373–397, 2003.

[5] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 37–45.

[6] Y. Zhu and D. Shasha, "Efficient elastic burst detection in data streams," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 336–345.

[7] X. Zhang and D. Shasha, "Better burst detection," in *Proceedings of the 22nd International Conference on Data Engineering*, 2006, pp. 146–149.

[8] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," in *Proceedings of the 31st international conference on Very large data bases*, 2005, pp. 181–192.

[9] X. Wang, C. Zhai, X. Hu, and R. Sproat, "Mining correlated bursty topic patterns from coordinated text streams," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 784–793.

[10] D. He and D. S. Parker, "Topic dynamics: an alternative model of bursts in streams of topics," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 443–452.

[11] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, "On the bursty evolution of blogspace," in *Proceedings of the 12th international conference on World Wide Web*, 2003, pp. 568–576.

[12] K. K. Mane and K. Börner, "Mapping topics and topic bursts in pnas," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101 Suppl 1, pp. 5287–5290, 2004. [Online]. Available: http://arxiv.org/abs/cs/0402029

[13] J. Yao, B. Cui, Y. Huang, and X. Jin, "Temporal and social context based burst detection from folksonomies," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010)*, 2010.

[14] Q. He, K. Chang, E.-P. Lim, and J. Zhang, "Bursty feature representation for clustering text streams," in *Proceedings of the Seventh SIAM International Conference on Data Mining*, 2007.

[15] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 497–506.

[16] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 851–860.

[17] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010, pp. 759–768.

[18] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, "Geographical topic discovery and comparison," in *Proceedings of the 20th international conference on World wide web*, 2011, pp. 247–256.

[19] H. Yang, S. Chen, M. R. Lyu, and I. King, "Location-based topic evolution," in *Proceedings of the 1st international workshop on Mobile location-based service*, 2011, pp. 89–98.

[20] "Online data mining for co-evolving time sequences," in *Proceedings of the 16th International Conference on Data Engineering*, 2000, pp. 13–22.

[21] Y. Ishikawa, Y. Chen, and H. Kitagawa, "An on-line document clustering method based on forgetting factors," in *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, 2001, pp. 325–339.