J Intell Inf Syst (2015) 44:243–269
DOI 10.1007/s10844-013-0283-2

Change point detection for burst analysis from an observed information diffusion sequence of tweets

Kazumi Saito · Kouzou Ohara · Masahiro Kimura · Hiroshi Motoda

Received: 31 January 2013 / Revised: 13 September 2013 / Accepted: 4 October 2013 /
Published online: 24 October 2013

Abstract We propose a method of detecting the period in which a burst of information diffusion took place from an observed diffusion sequence data over a social network and report the results obtained by applying it to the real Twitter data. We assume a generic information diffusion model in which time delay associated with the diffusion follows the exponential distribution and the burst is directly reflected to the changes in the time delay parameter of the distribution. The shape

The Twitter data we used in this paper were provided by Prof. Fujio Toriumi of Tokyo University and Prof. Kazuhiro Kazama of Wakayama University. This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-13-4042, and JSPS Grant-in-Aid for Scientific Research (C) (No. 23500194).

K. Saito (✉)
School of Administration and Informatics, University of Shizuoka,
Shizuoka 422-8526, Japan
e-mail: k-saito@u-shizuoka-ken.ac.jp

K. Ohara
Department of Integrated Information Technology, Aoyama Gakuin University,
Kanagawa 229-8558, Japan
e-mail: ohara@it.aoyama.ac.jp

M. Kimura
Department of Electronics and Informatics, Ryukoku University,
Otsu 520-2194, Japan
e-mail: kimura@rins.ryukoku.ac.jp

H. Motoda
Institute of Scientific and Industrial Research, Osaka University,
Osaka 567-0047, Japan
e-mail: motoda@ar.sanken.osaka-u.ac.jp

H. Motoda
School of Computing and Information Systems, University of Tasmania,
Hobart, TAS 7005, Australia

of the parameter's change is approximated by a step function and the problem of detecting the change points and finding the values of the parameter is formulated as an optimization problem of maximizing the likelihood of generating the observed diffusion sequence. Time complexity of the search is almost proportional to the number of observed data points and has been shown to be very efficient. We first demonstrated that the proposed method can detect the burst using a synthetic data and showed that it performs better than one of the representative state-of-the-art methods, confirming that the proposed method covers a wider range of change patterns. Then, we extended our evaluation on synthetic data to show that it is efficient and effective comparing it with a naive exhaustive search and a simple greedy method. We then apply the method to the real Twitter data of the 2011 To-hoku earthquake and tsunami, and reconfirmed its efficiency and effectiveness. Two interesting discoveries are that a burst period detected by the proposed method tends to contain massive homogeneous tweets on a specific topic even if the observed diffusion sequence consists of heterogeneous tweets on various topics, and that assuming the information diffusion path to be a line shape tree can give a good approximation of the maximum likelihood estimator when the actual diffusion path is not known.

Keywords Social networks · Information diffusion · Change point detection · Burst detection

1 Introduction

Recent technological innovation and popularization of high performance mobile/smart phones has drastically changed our communication style and the use of various social media such as Twitter[1] and Facebook[2] has been substantially affecting our daily lives. In these social media, information propagates through the social network formed based on friendship relations. Especially, Twitter, micro-blog in which the number of characters is limited to 140, is now very popular among the young generation owing to its handiness and easiness of usage. Besides, it is fresh to our memory that Twitter played a very important role as the information infrastructure during the recent natural disaster, both domestic and abroad, including the 2011 To-hoku earthquake and tsunami in Japan.

In the domain of social network analysis, several measures, called centrality, have been proposed so far to characterize nodes in the network based on its structure (Bonacichi 1987; Katz 1953; Wasserman and Faust 1994). While such centrality measures can be used to identify those nodes that play an important role in diffusing information over the network, it has also been shown that measures based solely on the network structure are not good enough to such a problem of influence maximization (Kempe et al. 2003; Kimura et al. 2010) in which the task is to identify a limited number of nodes which together maximize the information spread and that explicit use of information diffusion mechanism is essential (Kimura

[1] https://twitter.com/

[2] https://www.facebook.com/

et al. 2010). In general, the mechanism is represented by a probabilistic diffusion model. Most representative and basic ones are the Independent Cascade (IC) model (Goldenberg et al. 2001; Kempe et al. 2003) and the Linear Threshold (LT) model (Watts and Dodds 2007; Watts 2002) including their extended versions that explicitly handle asynchronous time delay, Asynchronous time delay Independent Cascade (AsIC) model (Saito et al. 2009) and Asynchronous time delay Linear Threshold (AsLT) model (Saito et al. 2010). In fact, the nodes and links that are identified to be influential using these models are substantially different from those identified by the existing centrality measures.

In reality, we observe that the information on a certain topic propagates explosively for a very short period of time. Because such information affects our behaviour strongly, it is important to understand the observed event in a timely manner. This brings in an important and interesting problem, which is to accurately and efficiently detect the burst from the observed information diffusion data and to identify what caused this burst and how long it persisted. Any of the above mentioned probabilistic models cannot handle this kind of problem because they assume that information diffuses in a stationary environment, i.e. model parameters are stationary. Zhu and Shasha (2003) approached this problem without relying on a diffusion model. They detected a burst period for a target event by counting the number of its occurrences in a given time window and checking whether it exceeds a predetermined threshold or not. Zhang (2006) proposed a data structure called Shifted Aggregation Tree that allows to count the frequency of an event more efficiently. Ebina et al. (2011) extended this approach and devised a more compact form of the tree structure to avoid unfruitful aggregation operations. Araujo et al. (2006) introduced a stochastic model that generates an observed sequence of frequencies of an event recorded within a certain time unit, and estimated the model parameters that are hidden variables representing the true frequencies at individual time points using genetic algorithm. All of these methods focus on the frequency within a certain time period, which are different from our approach that directly deals with the change of time interval between occurrences of a target event. There are studies, similar to ours, that tried to solve this problem focusing on the time interval. Kleinberg (2002) challenged this problem using a hidden Markov model in which bursts appear naturally as state transitions, and successfully identified the hierarchical structure of e-mail messages. Sun et al. (2010) extended Kleinberg's method so as to detect correlated burst patterns from multiple data streams that co-evolve over time.

We handle this problem by assuming that parameters in the diffusion model have been changed due to unknown external environmental factors and devise an efficient algorithm that accurately detects the changes in the parameter values from a single observed diffusion data sequence. In particular, we note that the parameter related to the time delay is most crucial in the burst detection and focus on detecting the changes in the time delay parameter that defines the delay distribution. We modeled the time delay in AsIC and AsLT models by the exponential distribution, thus we do the same in this paper. This corresponds to associating the burst with the information diffusion with a shorter time delay. A typical burst is a phenomenon in which a parameter value changes abruptly for a short period of time and returns back to the normal value. In this paper we allow a more general change pattern of the parameter value. By focusing only on this time delay, we can devise a generic algorithm that does not depend on a specific information diffusion model, e.g. be it either AsIC or AsLT.

More precisely, we assume that time delay parameter changes are approximated by a step function and propose an optimization algorithm that maximizes the likelihood ratio that is the ratio of the likelihood of observing the data assuming the time delay parameter changes (change points and parameter values between the successive change points) to the likelihood of observing the data assuming that there is no changes in the time delay parameter. The algorithm relies on an iterative search based on a recursive splitting with a delayed backtracking, and requires no predetermined threshold. The time complexity is almost proportional to the number of observed data points (candidates of possible change points). We first demonstrate that the proposed method can detect the burst assuming two simple change patterns (narrow up-and-down and wide stepwise) on a synthetic data and compare the result with Kleinberg's method (Kleinberg 2002) which is considered to be the state-of-the-art technique for burst detection. The proposed method successfully detects both changes accurately whereas Kleinberg's method is found to have some problem with a wide stepwise change pattern. We then conduct experiments on synthetic data to show that the proposed method is efficient and effective comparing it with a naive exhaustive search and a simple greedy method. We further test that the number of change points can also be estimated by the proposed method. After confirming that the proposed method works satisfactorily on synthetic data, we apply it to the Twitter data observed during the 2011 To-hoku earthquake and tsunami and confirm that the proposed method can efficiently and accurately detect the change points. We further analyze the content of the tweets and report the discovery that even use of the diffusion sequence data of the same user ID (not necessarily the data on a specific topic) allows us to identify that a specific topic is talked intensively around the beginning of the period where the burst is detected, and the assumption we made that the information diffusion path is a line shape tree gives a good approximation of the maximum likelihood estimator in this problem setting. Finally, we discuss that although the detected change points do not correspond exactly to nodes in a social network that caused the burst period, the detected change points are useful to find such nodes because we can limit nodes to be considered by focusing on those around them.

The paper is organized as follows. Section 2 briefly describes the framework of information diffusion model on which our problem setting is based. Section 3 elucidates the problem setting, and Section 4 describes the change point detection method including two other methods that are used for comparison together with the model selection method. Section 5 demonstrates that the proposed method can detect bursts and compare the result with Kleinberg's method. Section 6 evaluates the proposed method on synthetic data. Section 6 reports experimental results using real Twitter data. Section 7 summarizes what has been achieved in this work and addresses issues of future work.

2 Information diffusion model framework

We consider information diffusion over a social network whose structure is defined as a directed graph $G = (V, E)$, where V and E ($\subset V \times V$) represent a set of all nodes and a set of all links, respectively. Suppose that we observe a sequence of information diffusion $\mathcal{C} = \{(v_0, t_0), (v_1, t_1), \cdots, (v_N, t_N)\}$ that arose from the information released

at the source node v_0 at time t_0. Here, v_n is a node where the information has been propagated and t_n is its time. We assume that the time points are ordered such that $t_{n-1} < t_n$ for any $n \in \{1, \cdots N\}$. We further assume, as a standard setting, that the actual information diffusion paths of a sequence $\mathcal{C}$ correspond to a tree that is embedded in the directed graph G representing the social network (Sadikov et al. 2011), i.e., the parent node which passed the information to a node v_n is uniquely identified to be $v_{p(n)}$.[3] Here, $p(n)$ is a function that returns the node identification number of the parent of the node v_n in the range of $\{0, \cdots, n-1\}$.

The information diffusion model we consider here is any model that explicitly incorporates the concept of asynchronous time delay such as AsIC model (Saito et al. 2009) and AsLT model (Saito et al. 2010) in contrast to the traditional IC model (Goldenberg et al. 2001; Kempe et al. 2003) and LT model (Watts and Dodds 2007; Watts 2002) that do not consider the time delay. Said differently, it is a model that allows any real value for the time t_n at which the information has been propagated to a node v_n and assumes a certain probability distribution for the time delay $t_n - t_{p(n)}$. In this paper, we use the exponential distribution for the time delay, but any other distribution such as power law is feasible exactly in the same way.

3 Problem settings

In this section we formally define the change point detection problem. As mentioned in Section 1, we assume that some unknown change took place in the course of information diffusion and what we observe is a sequence of information diffusion of some topic in which the change is encapsulated. Thus, our goal is to detect each change point and how long the change persisted from there. Note that we basically pay attention to a diffusion sequence of a certain topic. From our previous result that people's behaviors are quite similar when talking the same topic (Saito et al. 2009, 2010), we can assume that the time delay parameter $r_{u,v}$ which is in principle defined for each link $(u, v) \in E$ takes a uniform value regardless of the link it passes through. In other word, we set $r_{u,v} = r$ $(\forall (u, v) \in E)$ and thus, the time delay of information diffusion is represented by the following simple exponential distribution $p(t_n - t_{p(n)}; r) = r \exp(-r(t_n - t_{p(n)}))$.

With this preparation, we mathematically define the change point detection problem. Let's assume that we observe a set of time points of information diffusion sequence $\mathcal{D} = \{t_0, t_1, \cdots, t_N\}$. Let the time of the j-th change point be T_j $(t_0 < T_j < t_N)$. The delay parameter that the distribution follows switches from r_j to r_{j+1} at the j-th change point T_j. Namely, we are assuming a step function as a shape of parameter changes. Let the set comprising J change points be $\mathcal{S}_J = \{T_1, \cdots, T_J\}$, and we set $T_0 = t_0$ and $T_{J+1} = t_N$ for the sake of convenience $(T_{j-1} < T_j)$. Let the division of $\mathcal{D}$ by $\mathcal{S}_J$ be $\mathcal{D}_j = \{t_n; T_{j-1} < t_n \leq T_j\}$, i.e., $\mathcal{D} = \{t_0\} \cup \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_{J+1}$, and $|\mathcal{D}_j|$ represents the number of observed points in $(T_{j-1}, T_j]$. Here, we request that $|\mathcal{D}_j| \neq 0$ for any $j \in \{1, \cdots, J+1\}$ and there exists at least one t_n such that $t_n \in \mathcal{D}_j$ is satisfied.

[3] Observed sequence $\mathcal{C}$ does not tell which parent activated which child. Without this assumption, we have to introduce hidden variables.

These settings are fully illustrated in Fig. 1, in which an information diffusion path is depicted as a tree drawn with thick lines, where the diffusion starts from the root node v_0. The white and black nodes on the diffusion path represent inactive and active nodes, respectively. We observe the sequence of time points $t_0, t_1, \cdots, t_N$, each of which is the time the corresponding node was activated. The time interval $t_2 - t_1$ between two adjacent active nodes v_1 and v_2 can be expressed as $t_2 - t_{p(2)}$ with the function $p(\cdot)$. Since the time delay parameter r changes from r_1 to r_2 at the time point t_3 in this figure, the first change point T_1 is t_3, and then the first partition of the set of observed time points $\mathcal{D}$, i.e., $\mathcal{D}_1$, contains three time points t_1, t_2, and t_3. It is noted that t_0 is not contained in $\mathcal{D}_1$. Such changes of r shapes a step function as depicted at the bottom of the figure. Given a set of observed time points $\mathcal{D}$, our aim in this paper is to find a set of J change points $\mathcal{S}_J$ and corresponding $J + 1$ values of the time delay parameter r by solving the optimization problem described below.

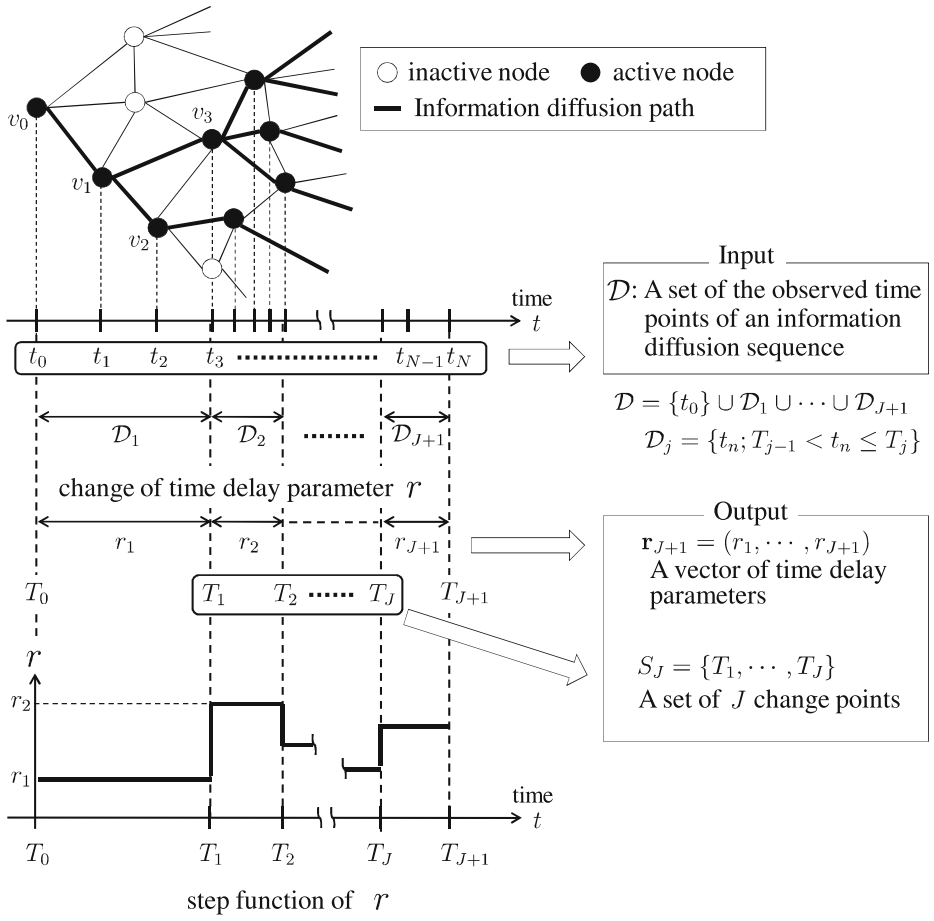

Fig. 1 An illustration of the problem setting

The log-likelihood for $\mathcal{D}$ for a given set of change points $\mathcal{S}_J$ is calculated, by defining the parameter vector $\mathbf{r}_{J+1} = (r_1, \cdots, r_{J+1})$, as follows.

$$L(\mathcal{D}; \mathbf{r}_{J+1}, \mathcal{S}_J) = \log \prod_{j=1}^{J+1} \prod_{t_n \in \mathcal{D}_j} r_j \exp(-r_j(t_n - t_{p(n)}))$$

$$= \sum_{j=1}^{J+1} |\mathcal{D}_j| \log r_j - \sum_{j=1}^{J+1} r_j \sum_{t_n \in \mathcal{D}_j} (t_n - t_{p(n)}). \quad (1)$$

Thus, the maximum likelihood estimate of the parameter of (1) is given by

$$\hat{r}_j^{-1} = \frac{1}{|\mathcal{D}_j|} \sum_{t_n \in \mathcal{D}_j} (t_n - t_{p(n)}), \quad j = 1, \cdots, J+1. \quad (2)$$

Further, substituting (2) to (1) leads to

$$L(\mathcal{D}; \hat{\mathbf{r}}_{J+1}, \mathcal{S}_J) = -N - \sum_{j=1}^{J+1} |\mathcal{D}_j| \log \left(\frac{1}{|\mathcal{D}_j|} \sum_{t_n \in \mathcal{D}_j} (t_n - t_{p(n)}) \right). \quad (3)$$

Therefore, the change point detection problem is reduced to the problem of finding the change point set $\mathcal{S}_J$ that maximizes (3). However, (3) alone does not allow us to directly evaluate the effect of introducing $\mathcal{S}_j$. We, thus, reformulate the problem as the maximization problem of log-likelihood ratio. If we do not assume any change point, i.e., $\mathcal{S}_0 = \emptyset$, then (3) is reduced to

$$L(\mathcal{D}; \hat{r}_1, \mathcal{S}_0) = -N - N \log \left(\frac{1}{N} \sum_{n=1}^{N} (t_n - t_{p(n)}) \right). \quad (4)$$

Thus, the log-likelihood ratio of the case where we assume J change points and the case where we assume no change points is given by

$$LR(\mathcal{S}_J) = L(\mathcal{D}; \hat{\mathbf{r}}_{J+1}, \mathcal{S}_J) - L(\mathcal{D}; \hat{r}_1, \mathcal{S}_0)$$

$$= N \log \left(\frac{1}{N} \sum_{n=1}^{N} (t_n - t_{p(n)}) \right) - \sum_{j=1}^{J+1} |\mathcal{D}_j| \log \left(\frac{1}{|\mathcal{D}_j|} \sum_{t_n \in \mathcal{D}_j} (t_n - t_{p(n)}) \right). \quad (5)$$

We consider the problem of finding the set of change points $\mathcal{S}_J$ that maximizes $LR(\mathcal{S}_J)$ defined by (5).

We note that, in general, it is conceivable that we are not able to acquire the complete tree structure of the diffusion sequence data. Thus, here, we consider two extreme cases, one in which the information spreads fastest (star shape tree) and the other in which the information spread slowest (line shape tree). The function which defines the parent node becomes $p(n) = 0$ for the former and $p(n) = n - 1$ for the latter. In case where there is no change point, the maximum likelihood estimator is $r^{-1} = (t_1 + \cdots + t_N)/N - t_0$ for the former and $r^{-1} = (t_N - t_0)/N$ for the latter. While we conjecture that in reality the optimal value lies in between these two extreme values, under the assumption that the actual tree structure of the diffusion

data is unknown, we consider to approximate the optimal value by using either one of them. Here, note that in the former case, the maximum likelihood estimator represents the average diffusion delay time between the source node v_0 and each node v_i which is assumed to be connected to v_0 by a direct link, while in the latter case, it represents the average time interval between successive observation time points. Considering that the burst period we want to detect is much shorter than the other non burst periods, the latter case (line shape tree) seems to be more suitable for our aim. Therefore, $LR(\mathcal{S}_J)$ defined by (5) becomes

$$LR(\mathcal{S}_J) = N \log\left(\frac{t_n - t_0}{N}\right) - \sum_{j=1}^{J+1} |\mathcal{D}_j| \log\left(\frac{T_j - T_{j-1}}{|\mathcal{D}_j|}\right). \tag{6}$$

We compared the bursts detected by using the two extreme values, and found that the use of line shape tree gave better results and decided to use (6) in our experiments.

4 Change points detection method

We consider the problem of detecting change points as a problem of finding a subset $\mathcal{S}_J \subset \mathcal{D}$ when the set of time points of information diffusion result $\mathcal{D} = \{t_0, t_1, \cdots, t_N\}$ is given. In other words, we estimate the number J of change points, and search for J time points $\mathcal{S}_J$ that are most likely to be the change points from a sequence of N observation points.

First, for a given J, we present three methods of finding $\mathcal{S}_J$. Here, the three methods are naive method (an exhaustive search), simple method (a greedy search), and the proposed method that is a combination of a greedy search and a local one. Next, we present a method of finding the value of J, and describe our proposed method for solving the change points detection problem. Finally, we have a brief discussion about methods of detecting change points from $\mathcal{D}$.

4.1 Naive method

The simplest method is to exhaustively search for the best set of J change points $\mathcal{S}_J$. Clearly the time complexity of this naive approach is $O(N^J)$. Thus, the number of change points detectable would be limited to $J = 2$ in order for the solution to be obtained in a reasonable amount of computation time when N is large enough.

4.2 Simple method

We describe the simple method which is applicable when the number of change points J is large. This is a progressive binary splitting without backtracking. We fix the already selected set of $(j-1)$ change points $\mathcal{S}_{j-1}$ and search for the optimal j-th change point T_j and add it to $\mathcal{S}_{j-1}$. We repeat this procedure from $j = 1$ to J.

The algorithm is given below.

Step1. Initialize $j = 1, \mathcal{S}_0 = \emptyset$.
Step2. Search for $T_j = \arg\max_{t_n \in \mathcal{D}} \{LR(\mathcal{S}_{j-1} \cup \{t_n\})\}$.
Step3. Update $\mathcal{S}_j = \mathcal{S}_{j-1} \cup \{T_j\}$.

Step4. If $j = J$, output $\mathcal{S}_J$ and stop.
Step5. $j = j + 1$, and return to Step2.

Here note that in Step3 elements of the change point set $\mathcal{S}_j$ are reindexed to satisfy $T_{i-1} < T_i$ for $i = 2, \cdots, j$. Clearly, the time complexity of the simple method is $O(NJ)$ which is fast. Thus, it is possible to obtain the result within a reasonable computation time for a large N. However, since this is a greedy algorithm, it can be trapped easily to a poor local optimal.

4.3 Proposed method

We propose a method which is computationally almost equivalent to the simple method but gives a solution of much better quality. We start with the solution obtained by the simple method $\mathcal{S}_J$, pick up a change point T_j from the already selected points, fix the rest $\mathcal{S}_J \setminus \{T_j\}$ and search for the better value T'_j of T_j, where $\cdot \setminus \cdot$ represents set difference. We repeat this from $j = 1$ to J. If no replacement is possible for all $j\,(j = 1, \cdots J)$, i.e. $T'_j = T_j$ for all j, then no better solution is expected and the iteration stops.

The algorithm is given below.

Step1. Find $\mathcal{S}_J$ by the simple method and initialize $j = 1, k = 0$.
Step2. Search for $T'_j = \arg\max_{t_n \in \mathcal{D}} \{LR(\mathcal{S}_J \setminus \{T_j\} \cup \{t_n\})\}$.
Step3. If $T'_j = T_j$, set $k = k + 1$, otherwise set $k = 0$, and update $\mathcal{S}_J = \mathcal{S}_J \setminus \{T_j\} \cup \{T'_j\}$.
Step4. If $k = J$, output $\mathcal{S}_J$ and stop.
Step5. If $j = J$, set $j = 1$, otherwise set $j = j + 1$, and return to Step2.

It is evident that the proposed method requires computation time several times larger than that of the simple method, but it is much less than that of the naive method. How much the computation time increases compared to the simple method and how much the solution quality increases await for the experimental evaluation, which we will report in Section 7.

4.4 Model selection

So far, we have fixed the number of change points J, and proposed a method of finding the optimal parameter vector $\hat{\mathbf{r}}_{J+1}$ and inferring the change points $\mathcal{S}_J$ for the observed data $\mathcal{D} = \{t_0, t_1, \cdots, t_N\}$. Now, we present a method of estimating the value of J from $\mathcal{D}$, and incorporate it into the proposed method in Section 4.3 for solving the change points detection problem. To this end, we employ the likelihood ratio test.

For any non-negative integer J, let $Y_N(J+1)$ be the log-likelihood ratio test statistic of the model of $J+1$ change points against the model of J change points; i.e.,

$$Y_N(J+1) \;=\; L(\mathcal{D}; \hat{\mathbf{r}}_{J+2}, \mathcal{S}_{J+1}) \;-\; L(\mathcal{D}; \hat{\mathbf{r}}_{J+1}, \mathcal{S}_J). \tag{7}$$

By definition, we have

$$Y_N(J+1) \;=\; LR(\mathcal{S}_{J+1}) \;-\; LR(\mathcal{S}_J). \tag{8}$$

Thus, $Y_N(J+1)$ can be easily calculated from (6) and (8). We note that the model of J change points is equipped with the $J+1$-dimensional parameter vector $\mathbf{r}_{J+1}$ and the J-dimensional parameter vector $\mathcal{S}_J$. It is well known that $2Y_N(J+1)$ asymptotically approaches to the χ^2 distribution with two degrees of freedom as N increases, since the difference in dimensionality of the parameter spaces of the two models is two. Thus, we first set a significance level α $(0 < \alpha < 1)$, say $\alpha = 0.05$. Next, by comparing $2Y_N(J+1)$ to $\chi_{2,\alpha}$, we evaluate whether the model of $J+1$ change points fits significantly better than does the model of J change points. Here, $\chi_{2,\alpha}$ is the upper α point of the χ^2 distribution of two degrees of freedom; i.e., the positive number defined by

$$\frac{1}{2}\int_0^{\chi_{2,\alpha}} \exp\left(-\frac{x}{2}\right) dx \;=\; 1-\alpha. \tag{9}$$

The proposed algorithm incorporating model selection is as follows:

Step1. Initialize $J = 0$, $\mathcal{S}_0 = \emptyset$.
Step2. Find $\mathcal{S}_{J+1}$ by the proposed method in Section 4.3.
Step3. Calculate $2Y_N(J+1)$ from (6) and (8).
Step4. If $2Y_N(J+1) \leq \chi_{2,\alpha}$, output $\mathcal{S}_J$ and stop.
Step5. Set $J = J+1$, and return to Step2.

Here, we note that for model selection, we can consider employing various methods other than the likelihood ratio test, that include AIC (Akaike's Information Criterion) (Akaike 1974) and MDL (Rissanen's Minimum Description Length) (Rissanen 1989), although we used the likelihood ratio test for simplicity. Our immediate future work is to extensively compare those methods for the task of detecting change points from $\mathcal{D}$.

4.5 Discussion

As described above, the change points detection method proposed in this paper is based on a top-down divisive approach. As an alternative strategy, we can consider employing a method based on a bottom-up merge (or agglomerate) approach. More specifically, an intuitive bottom-up method is designed as follows: for each information diffusion path from $v_{p(n)}$ to v_n, we consider assigning an individual initial cluster with the parameter value $r_n = 1/(t_n - t_{p(n)})$, and performing the merge steps with respect to the pairs of adjacent clusters. Then, by repeating this merge steps $(N-J)$ times, we can obtain the J change points as its solution.

However, in terms of computational load, the top-down method works more efficiently than the bottom-up one in case of $J \ll N$. This is because the former needs only J times of the division process, while the latter requires $(N-J)$ times of the merge process. Quantifying the differences of these two opposite methods is an interesting future study.

5 Detectability of change pattern

We have assumed that maximizing the log-likelihood is a good way to detect unknown change points, and proposed an efficient method to solve this optimization

problem. We show in this section that the proposed method indeed finds the change points at least as good as the state-of-the-art method that takes a different approach. We chose Kleinberg's method that uses a hidden Markov model (Kleinberg 2002) as one of the representative methods.

We tested both methods for two simple change patterns of parameter values on a line shape information diffusion tree. In both cases we set the true number J^* of change points to $J^* = 2$, and the period we considered is between $T_0 = 0$ and $T_3 = 3{,}000$. The first pattern is a narrow up-and-down change pattern, where the change points are set to $T_1 = 1{,}000$ and $T_2 = 1{,}200$ and the corresponding time-delay parameters are set to $r_1 = r_3 = 1$ and $r_2 = 2$. Namely, the normal time-delay parameter is set to $r_0 = 1$ except for the relatively short time interval between $T_1 = 1{,}000$ and $T_2 = 1{,}200$ with the parameter $r_2 = 2$ as the bursty one. This change pattern is considered to be the simplest and the most typical change for burst, and is referred to as a simple burst pattern. The second pattern is a wide stepwise change pattern, the change points are set to $T_1 = 1{,}000$ and $T_2 = 2{,}000$, and the corresponding time-delay parameters are set to $r_0 = r_1 = 1$, $r_2 = 2$ and $r_3 = 4$. Namely, the time-delay parameter increases step by step as time proceeds. This type of change patterns are likely to be seen when the observation time periods are relatively short, like Twitter's bursty information diffusion of the 2011 To-hoku earthquake and tsunami in our limited data. The second change pattern is referred to as a simple change pattern.

We first show the results of a simple burst pattern. Figure 2 shows the estimated change patterns by the proposed method with the settings $J = 1, 2$ and 3 from the pseudo observation time points $\{t_0, t_1, \cdots, t_N\}$ generated according to the simple burst pattern. Here, we plot three results, 1) $\{(t_1, \hat{r}_1), \cdots, (t_N, \hat{r}_N)\}$ by using blue dots, where $\hat{r}_n = 1/(t_n - t_{n-1})$, 2) the change patterns of true time-delay parameters by the wide green line and 3) the estimated ones by the thin red line. Note that $\hat{r}_n$ is the maximum likelihood estimator for time-delay parameter for the successive two observation time points $\{t_{n-1}, t_n\}$. As expected, we confirmed that by selecting the true number of change points, i.e., $J^* = 2$, our proposed method could successfully detect this change pattern with reasonable accuracy as shown in Fig. 2b, in which the wide and thin lines are indistinguishable from each other. Actually, under the setting $\alpha = 0.05$ which brings about $\chi_{2,\alpha} = 5.99$, the algorithm described in Section 4.4 selected the correct number of change points, $J = 2$, from the obtained log-likelihood ratio test statistics, $2Y_N(1) = 9.39$, $2Y_N(2) = 71.29$ and $2Y_N(3) = 3.84$, because $2Y_N(3) < \chi_{2,\alpha}$. On the other hand, as shown in Fig. 2a and c, we obtained

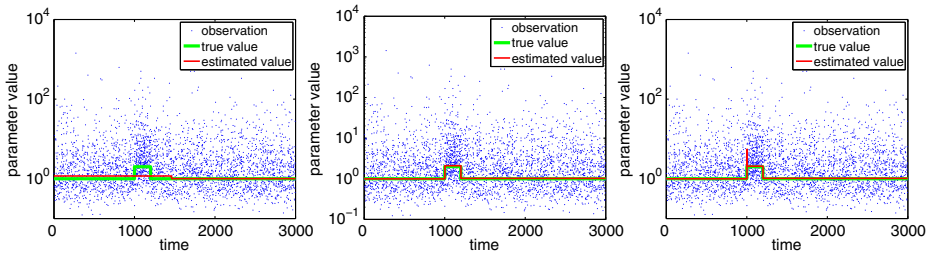

Fig. 2 Results of the proposed method for the simple burst pattern

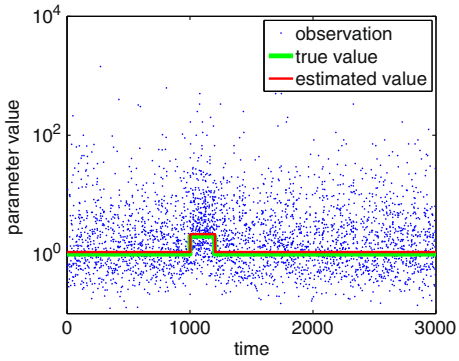

Fig. 3 Result of Kleinberg's method for the simple burst pattern

some under-fitting and over-fitting change patterns in case we set $J = 1$ and $J = 3$, respectively.

Figure 3 shows the estimated change pattern (state change trace) by Kleinberg's method from the same pseudo observation time points, where the scaling parameter s is set to $s = 2$ to reflect the change of our simple burst pattern, i.e., $r_0 = r_1 = r_3 = 1$ and $r_2 = 2$, and the cost $\tau(i, j)$ of moving from state i to j is defined by $\tau(i, j) = (j - i)\gamma \log(N)$ if $j > i$; otherwise $\tau(i, j) = 0$, where we employed $\gamma = 1$. Here, the scaling parameter s determines the delay parameter at the state j by $r_j = s^j r_0$ and the parameter r_0 is estimated by $r_0 = N/T_3$ as described in (Kleinberg 2002). As expected, we confirmed that Kleinberg's method could also successfully detect this change pattern with reasonable accuracy, as good as could our proposed method.

Next we show the results of a simple change pattern. Figure 4 shows the estimated change patterns by the proposed method with the settings $J = 1, 2$ and 3 from the pseudo observation time points generated according to the simple change pattern. Again, as expected, we confirmed that by selecting the true number of change points, i.e., $J^* = 2$, our proposed method could successfully detect this change pattern with reasonable accuracy as shown in Fig. 4b. From the obtained log-likelihood ratio test statistics, $2Y_N(1) = 854.19$, $2Y_N(2) = 166.58$ and $2Y_N(3) = 4.22$, our proposed method selected the correct number of change points, $J = 2$, as $2Y_N(3) < \chi_{2,\alpha}$. On the other hand, as shown in Fig. 4a and c, we obtained some under-fitting and over-fitting change patterns in case we set $J = 1$ and $J = 3$, respectively.

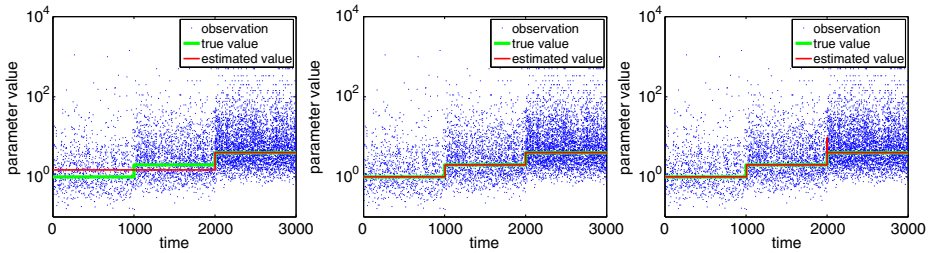

Fig. 4 Results of proposed method for the simple change pattern

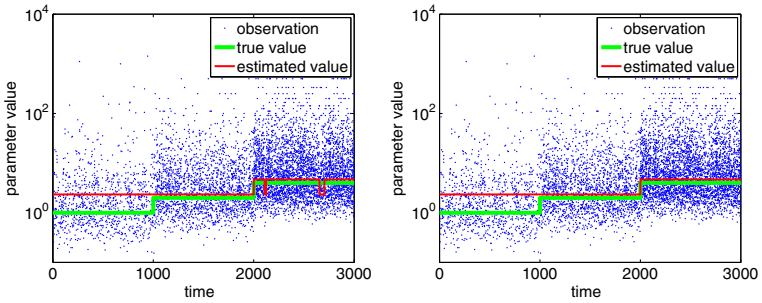

Fig. 5 Results of Kleinberg's method for the simple change pattern

Figure 5 shows the estimated change patterns by Kleinberg's method with the settings $\gamma = 1$ and 2 from the same pseudo observation time points. Here we employed the same experimental settings as in the case of the simple burst pattern. From Fig. 5a, we point out two shortcomings: (1) since the normal time delay parameter $r_0 = r_1$ was incorrectly estimated to be a significantly larger value, the method missed the change point $T_1 = 1,000$, and inaccurately estimated the time delay parameters r_2 and r_3; (2) the method produced a slightly over-fitted change patterns in the time periods between $T_2 = 2,000$ and $T_3 = 3,000$ in case of $\gamma = 1$. As shown in Fig. 5b, we can easily resolve the second shortcoming by increasing the cost coefficient γ to 2.

In order to more closely examine the effects of the incorrect estimation of r_0, we slightly modified Kleinberg's algorithm so that the normal time delay parameter is set to the correct one. Figure 6 shows the estimated change patterns by Kleinberg's method with the settings $\gamma = 1$ and 2 after this fix ($r_0 = 1$). As expected, we confirmed that Kleinberg's method could also successfully detect this change pattern with reasonable accuracy except for a slightly over-fitted change pattern around $T_2 = 2,000$ in case of $\gamma = 1$, as shown in Fig. 6a. Again, we can resolve the over-fitting problem by increasing the cost coefficient γ to 2, as shown in Fig. 6b.

In summary, Although Kleinberg's method is expected to work well for a typical burst pattern, as shown in Fig. 3, this method is likely suffer from an incorrect r_0 estimation problem in case of general change patterns, as shown in Fig. 5. In

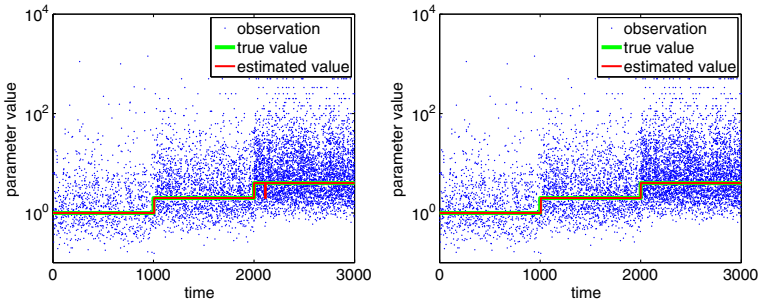

Fig. 6 Results of Kleinberg's method for the simple change pattern by fixing at $r_0 = 1$

addition, in order to improve the performance of the method, we might need some criteria to determine both the scaling parameter s and the cost coefficient γ, just like our proposed method employs the log-likelihood ratio test statistic for selecting the adequate number of change points. Here recall that the scaling parameter s of Kleinberg's method was preferably determined in our experiments, so as to reflect the change level of time delay parameters. Therefore, we consider that Kleinberg's method has some limitations, compared with our proposed method, to cope with general change patterns which are generally wider and more complex than a simple burst pattern.

6 Experimental evaluation on synthetic data

We experimentally compare the proposed method with the simple method described in the previous section in terms of how accurately they can detect change points and estimate the time delay parameters in an information diffusion sequence, using systematically generated observation sequence data. First, we conduct the experiments assuming that we know the true number of underlying change points. Then, we investigate whether the model selection method based on the likelihood ratio test shown in Section 4.4 can work well to detect the number of underlying change points in a given sequence.

6.1 Synthetic datasets

We systematically generated temporal sequences of pseudo observation time points of information diffusion in which J change points were embedded. First, we specified both the period $[0, T]$ and the number of change points J to be embedded in the period. Then, we equally partitioned the whole period into $J+1$ intervals of length $T/(J+1)$, i.e., $[0, T/(J+1)]$, $(T/(J+1), 2T/(J+1)], \cdots, (JT/(J+1), T]$, and generated observation time points in the j-th interval ($j = 1, \cdots J+1$) according to the exponential distribution with the parameter r_j, where r_1 was set to 1.0 and r_j for $j > 1$ was randomly chosen from either $2^{1/2} r_{j-1}$ or $2^{-1/2} r_{j-1}$. In fact, we considered the first observation time point in each period as a change point, and updated the value of the time delay parameter at that time point as mentioned above. Finally, we generated 10 datasets varying J from 1 to 10 with $T = 100{,}000$, each of which contains 1,000 sequences.

6.2 Experimental results using true number of underlying change points

First, we evaluate how the simple and the proposed methods can accurately detect the change points in a sequence when the true number of underlying change points J is given. To this end, we investigated the mean absolute error of detected change points $\mathcal{S}_J$ for each sequence, which is defined as follows:

$$MAE_{\mathcal{S}_J} = \frac{1}{J} \sum_{j=1}^{J} |T_j - \hat{T}_j|, \tag{10}$$

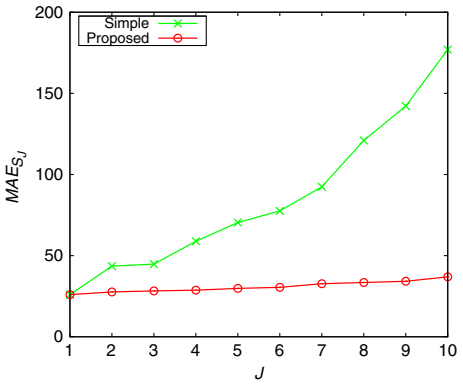

Fig. 7 Comparison between the simple and proposed methods in terms of the mean average error of detected change points

where T_j and $\hat{T}_j$ are the true and the detected change points, respectively. Figure 7 shows the average of MAE_{S_J} over the 1,000 sequences for each J. From this figure, we can clearly see that the proposed method can detect the change points more accurately than does the simple method. MAE_{S_J} for the proposed method shows little change even if the number of change points J increases, while MAE_{S_J} for the simple method almost linearly increases as J becomes larger. This is attributed to the fact that the number of observation time points in each period $(T_{j-1}, T_j]$, which serve as training examples to learn optimal change points, gets smaller as the number of underlying change points J gets larger. This result shows the local search employed by the proposed method is highly effective to reduce the error for a large J.

Next, to evaluate how these methods can accurately estimate the time delay parameters, we investigated the mean relative error of the estimated parameter values, $MRE_{\mathbf{r}}$, defined as follows:

$$MRE_{\mathbf{r}} = \frac{1}{J+1} \sum_{j=1}^{J+1} |r_j - \hat{r}_j| / r_j, \tag{11}$$

where r_j and $\hat{r}_j$ are the true and the estimated parameters, respectively. Figure 8 illustrates how the average of $MRE_{\mathbf{r}}$ over the 1,000 sequences changes according

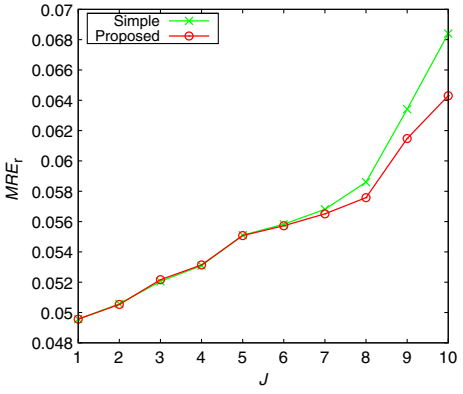

Fig. 8 Comparison between the simple and proposed methods in terms of the mean relative error of estimated time delay parameters

to the increase of the number of underlying change points J. It is found that the average of $MRE_{\mathbf{r}}$ increases in proportion to the value of J for both methods. This is because the parameter r_j is estimated based on the relation $r_j^{-1} = (T_j - T_{j-1})/|\mathcal{D}_j|$ as mentioned in Section 3. This means that, even if $\mathcal{D}_j$ is incorrectly estimated, it would not affect the estimation of r_j so much if J is small and the estimation error of $\mathcal{D}_j$ is limited, because $T_j - T_{j-1}$ is sufficiently large for a small J under the setting of our experiments. On the other hand, the estimation error of $\mathcal{D}_j$ has a larger influence on the estimation of r_j as J becomes large because $T_j - T_{j-1}$ gets smaller. Thus, considering the error for $\hat{T}_j$ observed in Fig. 7, it is interpretable that the errors of both methods are comparable to each other for a small J, while the proposed method is slightly better than the simple method even though the errors of both methods are getting large similarly as J increases.

6.2.1 Experimental results of model selection

We used the true number of underlying change points J for the experiments in the previous section. However, in reality, we never know it for a given observed sequence. Thus, we have to investigate whether the model selection method that is based on the likelihood ratio test, described in Section 4.4, can correctly detect the number of underlying change points. For this purpose, in Fig. 9, we present and compare the accuracy of the model selection method for the proposed method and the simple method, where the accuracy is defined as the ratio of the number of sequences for which the number of underlying change points is correctly detected by the model selection method over all 1,000 sequences for each J. In this experiment, we adopted 0.01 as the significance level α of the likelihood ratio test. This result demonstrates that combining the model selection method with the proposed method works well and achieves high accuracy. However, combining the model selection with the simple method does not work well and its accuracy gets worse as J becomes larger. Here, it is worth mentioning that these results are correlated to the results shown in Fig. 7, meaning that the ability of correctly detecting the change points for given J significantly affects the performance of the model selection method.

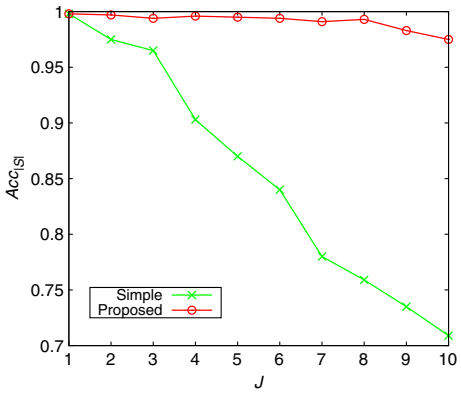

Fig. 9 Comparison between the simple and proposed methods in terms of the accuracy of the number of detected change points

Finally, we show the correctness of the selected model, more precisely the estimated step function $\hat{r}(t)$ that determines the value of the delay parameter r for a given time t, using the following relative measure:

$$Diff_{r(t)} = \frac{\int_0^{T_n} |\hat{r}(t) - r(t)| dt}{\int_0^{T_n} r(t) dt}, \tag{12}$$

where T_n is the final observation time point in a given sequence, and $r(t)$ is the true underlying step function that generated the synthetic sequence. Figure 10 shows the average of $Diff_{r(t)}$ over the 1,000 sequences for each J. From this figure, it is found that the average value increases for both the methods as J becomes larger. The reason of this tendency is the same as the reason why the mean relative error increases in Fig. 8. Namely, this is because the estimation error of change points causes the error of $|\mathcal{D}_||$, which has a larger impact on the estimation error of r_j as the period $T_j - T_{j-1}$ gets shorter due to J being larger. Interestingly the results obtained by using the simple method are comparable to those by using the proposed method although the proposed method detected the number of underlying change points more accurately than does the simple method as shown in Fig. 9. Actually, when the simple method makes a mistake, we observed that the simple method detects one or two more incorrect change points in addition to the correct ones, which worsen its accuracy as shown in Fig. 9. However, in many cases, those incorrect change points merely partition correct periods into sub parts. In other words, the error of the time delay parameters estimated by the simple method for the true number of change points J is mitigated by introducing additional more accurate change points although the accuracy in the number of detected change points reduces. It is noted that this does not mean the simple method can be an alternative to the proposed method since we need to know the change points as accurately as possible in order to investigate what caused the bursts of the information diffusion we observed in the real world. In that sense, these experimental results that demonstrate the proposed method that can more correctly detect the change points is more suitable for that purpose.

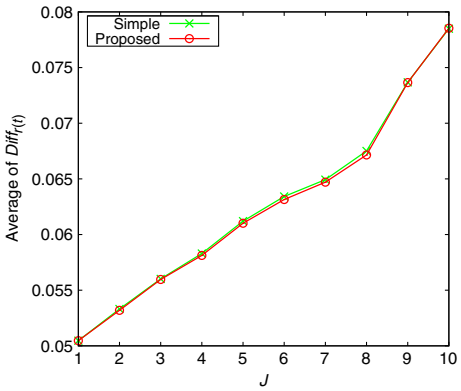

Fig. 10 Comparison between the simple and proposed methods in terms of the relative difference between the true step function $r(t)$ and the estimated step function $\hat{r}(t)$

7 Experimental evaluation by real data

Next, we experimentally evaluate the computation time and the accuracy of the change point detection using the real world Twitter information diffusion sequence data based on the methods we described in Section 4. We, then, analyze in depth the top 6 diffusion sequences in terms of the log-likelihood ratio based on the detected change points and burst periods. We also apply the model selection method introduced in Section 4.4 to the Twitter data and show its usefulness. Besides, we show that the line shape tree approximation is much better than the star shape based one, and investigate whether we are able to identify which node in a social network caused the burst from the detected change points.

7.1 Experimental Settings

The information diffusion data we used for evaluation are extracted from 201,297,161 tweets of 1,088,040 Twitter users who tweeted at least 200 times during the three weeks from March 5 to 24, 2011 that includes March 11, the day of 2011 To-hoku earthquake and tsunami. It is conceivable to use a retweet sequence in which a user sends out other user's tweet without any modification. But there exist multiple styles of retweeting (official retweet and unofficial retweet), and it is very difficult to accurately extract a sequence of tweets in an automatic manner considering all of these different styles. Therefore, in our experiments, noting that each retweet includes the ID of the user who sent out the original tweet in the form of "@ID", we extracted tweets that include @ID format of each user ID and constructed a sequence data for each user. More precisely, we used information diffusion sequences of 798 users for which the length of sequences are more than 5,000 (number of tweets). Note that each diffusion sequence includes retweet sequences on multiple topics. Since we do not know the ground truth of the change points for each sequence if there are changes in it, we used the naive method which exhaustively search for all the possible combinations of the change points as giving the ground truth. We had to limit the number of change points to 2 ($J = 2$) in order for the naive method to return the solution in a reasonable amount of computation time. The experimental results explained in the next subsection is obtained by using a machine with Intel(R) Xeon(R) CPU W5590 @3.33 GHz and 32 GB memory.

7.2 Main results

7.2.1 Performance evaluation

Figure 11 shows the computation time that each method needed to produce the results. The horizontal axis is the length of the information diffusion data sequences, and the vertical axis is the computation time in second. The results clearly indicate that the naive method requires the largest computation time. The computation time is quadratic to the sequence length as predicted. In contrast, the computation time for the simple and the proposed methods is much shorter and it increases almost linearly to the increase of the sequence length for both. The proposed method requires more computation time due to the extra iteration needed for delayed backtracking. In fact, the number of extra iteration is 2.2 on the average and 7 at most.

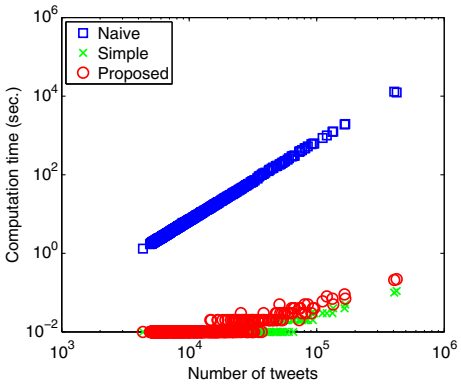

Fig. 11 Comparison of computation time among the three (*naive*, *simple*, and *proposed*) methods

Figure 12 shows the accuracy of the detected change points. We regarded that the solution obtained by the naive method is the ground truth. The horizontal axis is the sequence ranking of the log-likelihood ratio for the naive method (ranked from the top to the last), and the vertical axis is the logarithm of the likelihood ratio of the solution of each method. The results indicate that the simple method has lower likelihood ratio for all the range, meaning that it detects change points which are different from the optimal ones, but the proposed method can detect the correct optimal change points except for the low ranked sequences for which the likelihood ratio is small as is evident from the result in that the red curve representing the proposed method is indistinguishable from the blue curve representing the naive method. The reason why the accuracy of the proposed method for sequences with low likelihood decreases may be because the burst period is not clear for these sequences. In summary, out of the 798 sequences in total, the proposed method gave the correct results for 713 sequences (98.4 %), whereas the simple method gave the correct results for only 171 sequences (21.4 %). The average ratio of the likelihood ratio of the proposed method to that of the naive method (optimal solution) is 0.976, whereas the corresponding ratio for the simple method is 0.881, revealing that the proposed method gives much closer ratio to the optimal likelihood ratio. These results confirm that the proposed method can increase the change point detection accuracy to a great

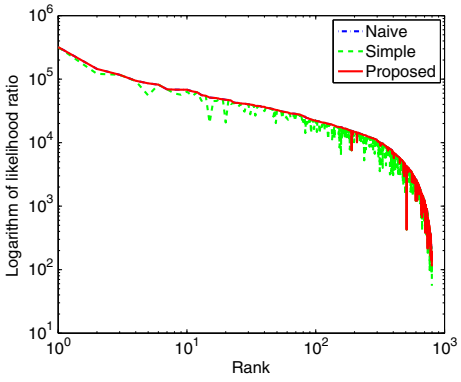

Fig. 12 Comparison of accuracy among the three (*naive*, *simple*, and *proposed*) methods

extent compared to the simple method with only a small penalty for the increased computation time.

7.2.2 In depth analysis of detected change points and burst periods

Next, we had a closer look at the top 6 diffusion sequences in terms of the log-likelihood ratios. Table 1 shows the total number of tweets included in the sequence, the starting and the ending time of the burst period, and the main topics that appeared near the beginning of the burst. Figure 13 shows how the cumulative number of tweets increases as time goes for each diffusion sequence. The horizontal axis is time and the vertical axis is the cumulative number of tweets. The two vertical lines indicated by small arrows in each graph are the change (starting and ending) points detected by the proposed method, and the interval between them is the burst period.

As is understood from Table 1, explosive retweeting of the information of urgent need about the earthquake for a short period of time triggered the start of the burst (with the exception of the 4th ranked sequence). The 4th ranked sequence is for the account called "ordinary timeline" which was set up for allowing to tweet everyday topics by adding "@itsumonoTL" at the beginning of the tweet when people are in voluntary restraint mood after the disastrous earthquake. We can say, with the exception of such a special case of "ordinary timeline", that we are able to detect efficiently a time period where tweets on a specific topic (of urgent need in this example) are intensively retweeted by looking at the change points detected by the proposed method even from the diffusion sequence that contains multiple topics.

Table 1 Major topics appearing at the beginning of the burst periods of the top 6 diffusion results in terms of log-likelihood ratio

Ranking	Length	Detected burst period		Major topics at the beginning of the burst period
		Start	End	
1	450,739	2011/3/11 14:48:13	2011/3/13 23:13:04	Retweets of the earthquake bulletin posted by the PR department of Japan Broadcasting Corporation, NHK (@NHK_PR).[a]
2	27,372	2011/3/11 15:13:57	2011/3/11 16:19:26	Retweets of the article on to-do list at the time of earthquake onset posted by a victim of the Great Hanshin-Awaji Earthquake.[b]
3	167,528	2011/3/12 00:18:19	2011/3/14 22:08:20	Retweets of the article on measures against cold at an evacuation site posted by the news department of NHK (@nhk_seikatsu).
4	423,594	2011/3/13 18:38:50	2011/3/19 02:20:58	Ordinary tweets irrelevant to the earthquake posted to a special account "@itsumonoTL".
5	63,485	2011/03/11 15:05:08	2011/03/12 01:52:13	Retweets of the earthquake bulletin posted by the Fire and Disaster Management Agency (@FDMA_JAPAN).
6	18,299	2011/3/11 15:45:17	2011/3/11 17:19:02	Retweets of a call for help posted by a user who seemed to be buried under a server rack (later found to be a false rumor).

[a]NHK is the government operated broadcaster.
[b]Great Hanshin-Awaji Earthquake occurred on January 17, 1995 in Kobe area and 6,434 people lost their lives

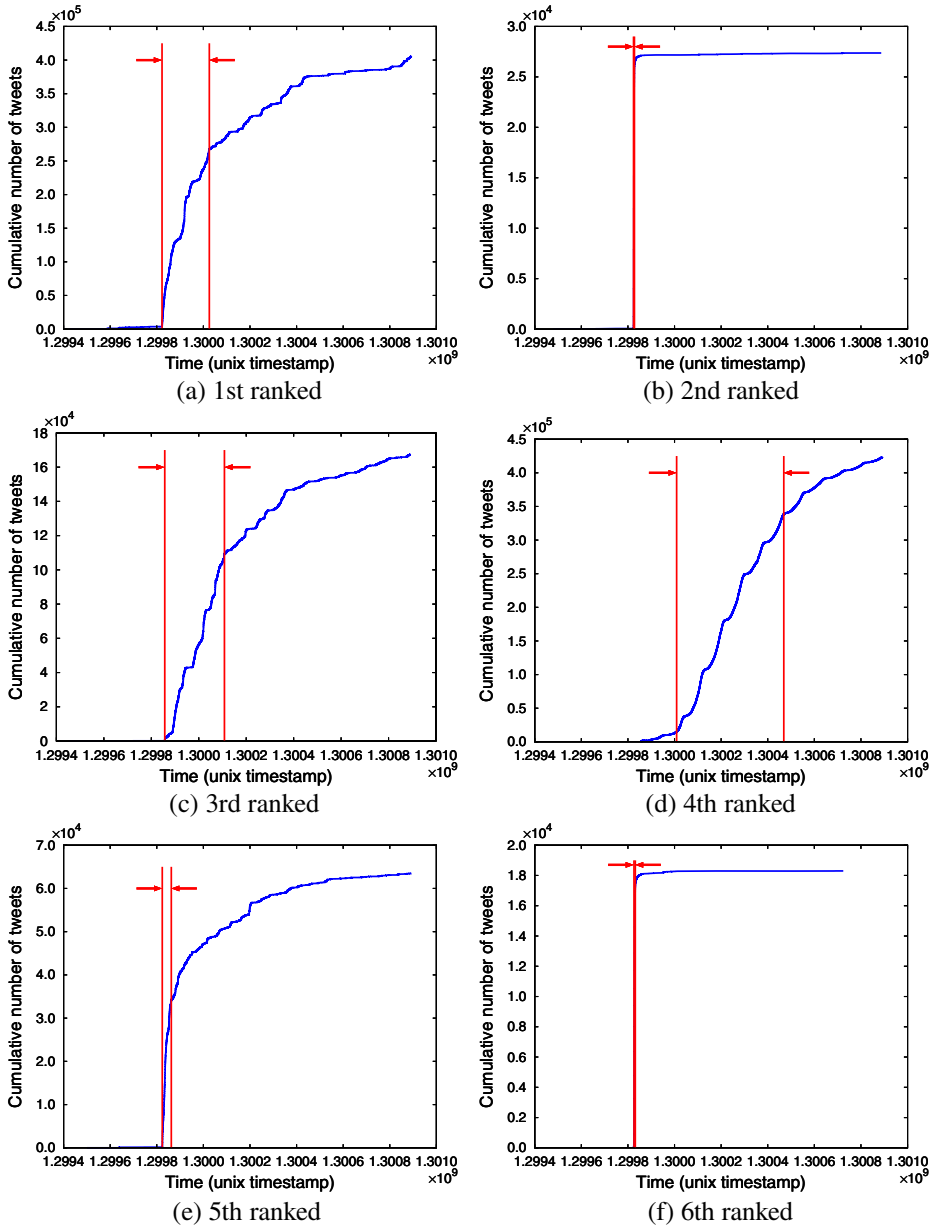

Fig. 13 Temporal change of cumulative number of tweets in the top 6 diffusion results in terms of the highest log-likelihood ratio

We note that the cumulative number of the tweets for the 2nd and 6th ranked diffusion sequences is smaller than the other 4 sequences from Table 1, and the burst period of these 2 sequences are much shorter than others and there is little changes in the number of tweets before and after the burst from Fig. 13. This difference is considered to come from whether the account is private or public.

Among these 4 sequences, except for the exceptional 4th one, the remaining 3 are all from the public organization accounts (1st and 3rd are NHK and 5th is FDMA). Information posted by these accounts tends to disseminate widely everyday. Thus, considering this situation, it is natural to observe that the cumulative number of tweets shows a relatively smooth increase as seen in Fig. 13 by adding multiple bursts of short periods about the earthquake-related information of urgent need as shown in Table 1. Figure 13e has only one smooth change during the burst period, which indicates that the earthquake bulletin in Table 1 is the only source of the burst. On the other hand, we see multiple smooth changes with discontinuity of the gradient at each boundary during the burst period in Fig. 13a and c. This implies that there can be other sources of the burst than shown in Table 1. Indeed, it is possible to identify these change points by increasing the value of J (an example explained later). On the other hand, Fig. 13b and f show that the information posted by an individual that is rarely retweeted in ordinary situations can be propagated explosively if it is of urgent need, e.g. timely information about earthquake.

Here, we report the result when we increase the number of change points. Figure 14 shows the result for the 3rd ranked sequence in Fig. 13c when J is set to 9. There are 9 vertical lines corresponding to each change point, but the first two change points are too close and indistinguishable. Note that horizontal axis is enlarged and the range shown is different from that in Fig. 13c. We see that the detected change points are located at the boundary points where the gradients of the curves change discontinuously. Those 4 broken lines in green are considered to indicate the end of the burst because the gradient change across each boundary is rather smaller. In fact, we investigated the most recent 10 tweets for these 4 change points and confirmed that no more than half of the retweets is talking about the same topic except the one second from the last in which 7 of them are on the same topic. The remaining 5 change points (red lines) all contain at least 7 retweets (10, 8, 7, 7, 9) that are on the same topic. From this fact, we can reconfirm that there appear many tweets on the same topic during the burst period.

7.2.3 Results of model selection

Next, we show the results obtained by applying the model selection method described in Section 4.4 to the Twitter data. Here, based on the results shown in Section 6.2.1,

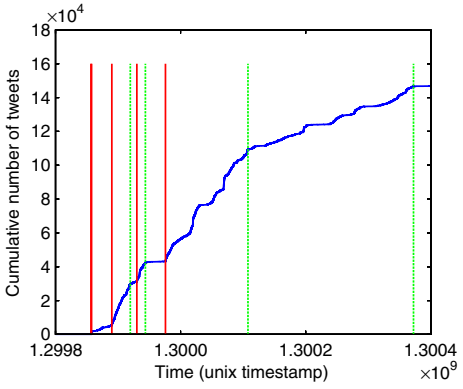

Fig. 14 Finer burst detection for the 3rd ranked sequence in Fig. 13c when J is set to 9

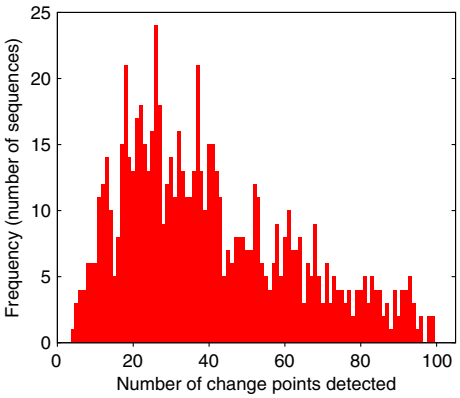

Fig. 15 Distribution of the number of detected change points for the Twitter data applying the model selection method to the proposed method ($\alpha = 0.01$)

we applied only the model selection method to the proposed method adopting the significance level $\alpha = 0.01$. Figure 15 shows the distribution of the number of detected change points for the retweet sequences. The horizontal axis means the number of change points detected, and the vertical axis means the frequency of sequences that have the corresponding number of change points. We aborted the algorithm described in Section 4.4 if it does not terminate even at $J = 100$. In fact we could not identify the number of change points for 36 out of a total 798 sequences. Figure 15 shows the distribution for the remaining 762 sequences. From this figure, we can observe many sequences that have about 10 to 40 change points. For example, 12 change points were detected for the 1st ranked sequence shown in Fig. 13a as illustrated in Fig. 16. This result demonstrates the model selection method works well even for the real Twitter data. On the other hand, we also observed that it detected 30 change points for the 2nd ranked sequence shown in Fig.13b as shown in Fig. 17, in which some change points are again too close and indistinguishable from each other. The number may sound too high, but actually these change points are consolidated into the 4 bursts in the corresponding step function of the time delay parameter r depicted in Fig. 18. The first biggest peak corresponds to the burst shown in Fig. 13b. Interestingly, the lowest peak corresponds to one of big afterquakes of the main quake. From this analysis, it can be said that even if the model selection method

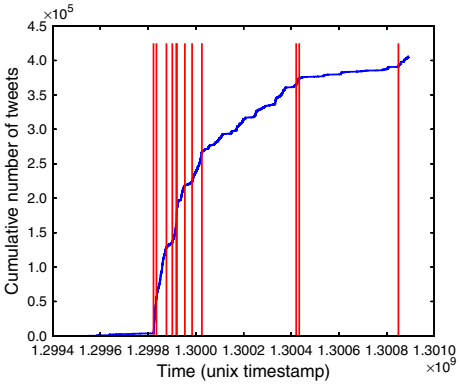

Fig. 16 Finer burst detection for the 1st ranked sequence in Fig. 13a when J is determined by the model selection method

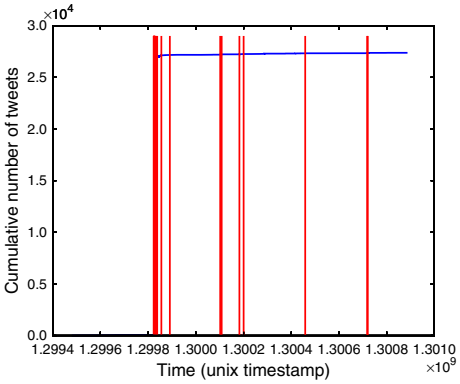

Fig. 17 Change points detected for the 2nd ranked sequence in Fig. 13b by the model selection method

detects many change points, we could find much fewer peaks in the corresponding step function of the time delay parameter.

7.2.4 Line hape tree vs. star shape tree

Note that all of these results were obtained by assuming that the information diffuses along the line shape tree as discussed in Section 3. Here, we show that use of line shape tree gives better results than use of star shaped tree. To this end, we compared the bursts detected for the 2nd and 6th ranked information diffusion sequences which include only one burst.

The results are illustrated in Fig. 19, where red solid and green broken vertical lines denote the change points detected by the naive method with the line shape and star shape settings, respectively. Only the time range of interest is extracted and shown in the horizontal axis. From these figures, we observe that use of line shape tree detects the change points more precisely as expected, which means that line shape tree gives a better approximation of the maximum likelihood estimator than star shape tree even if the actual tree shape of the diffusion path is not known to us.

7.2.5 Change points in a time line and nodes in a network

Remember that each observed time point corresponds to a node in a social network. In this sense, it can be said that the proposed method detects not only the change

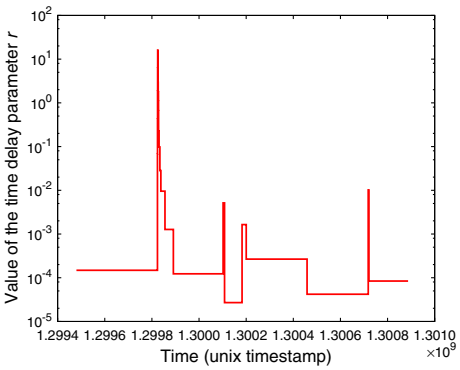

Fig. 18 Step function of the time delay parameter r corresponding to the results shown in Fig. 17

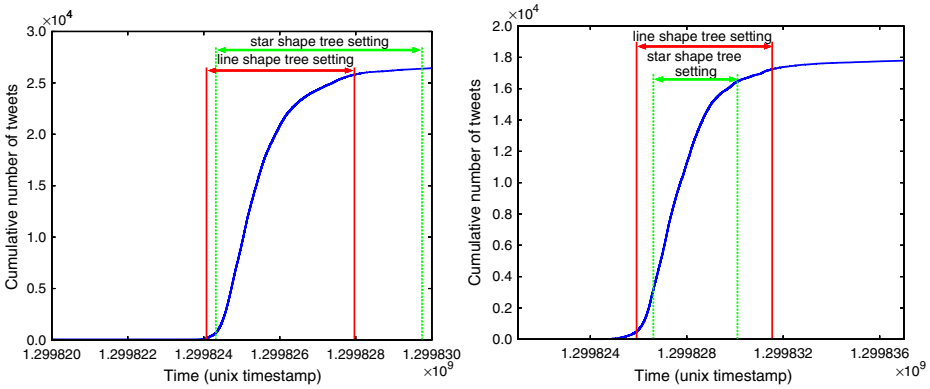

Fig. 19 Comparison of bursts detected by use of line shape tree and star shape tree for the 2nd and 6th ranked information diffusion sequences in Table 1

points in a time line, but also the change points in a network. However, unfortunately, those nodes do not necessarily correspond to those which actually caused the burst period. For example, in the second ranked sequence in Table 1, we observed at least 1 retweet of the article described in Table 1 per second after the start of the burst, 2011/3/11 15:13:57, while we observed at most 20 per minute before the burst started. This shows the accuracy of the detected change point, but it also means that the node that actually influenced nodes within the burst period could exist in the period before the change point. Indeed, we observed the first retweet at 2011/3/11 15:07:05 and 69 retweets thereafter before the change point. It is natural to think that some of them played an important role on the explosive diffusion of the article. We need to know the actual information diffusion path to find such important nodes, but detecting change points in a time line would significantly reduce the effort needed to do so because the search can be focused on the limited sub-sequences around the change points. Devising a method to find such important nodes is one of our future work.

8 Conclusion

We addressed the problem of detecting the period in which information diffusion burst occurs from a single observed diffusion sequence under the assumption that the delay of the information propagation over a social network follows the exponential distribution. To be more precise, we formulated the problem of detecting the change points and finding the values of the time delay parameter in the exponential distribution as an optimization problem of maximizing the likelihood of generating the observed diffusion sequence. We devised an efficient iterative search algorithm for the change point detection whose time complexity is almost linear to the number of data points, and presented the model selection method that determines the optimal number of change points to detect from the viewpoint of the likelihood ratio test. We tested the algorithm against the synthetic and the real Twitter data of the 2011 To-hoku earthquake and tsunami, and experimentally confirmed that the algorithm is much more efficient than the exhaustive naive search and is much more accurate

than the simple greedy search. By analyzing the real information diffusion data, we revealed that even if the data contains tweets talking about plural topics, the detected burst period tends to contain tweets on a specific topic intensively. We also observed that the model selection method detected many change points for some sequences, but at the same time, we confirmed that much fewer bursts could be detected from the step function derived from the estimated values of the time delay parameter. In addition, we experimentally confirmed that assuming the information diffusion path to be the line shape tree results in much better approximation of the maximum likelihood estimator than assuming it to be the star shape tree. This is a good heuristic to accurately estimate the change points when the actual diffusion path is not known to us. These results indicate that it is possible to detect and identify both the burst period and the topic diffused without extracting the tweet sequence for each topic and identifying the diffusion paths for each sequence, and the proposed method can be a useful tool to analyze a huge amount of information diffusion data. Our immediate future work is to compare the proposed method with other existing burst detection methods, especially with those which focus on the frequency within a certain time unit, since we have already empirically compared it with the representative method focusing on the time interval between occurrences of a target event. We also plan to quantify the differences between our top-down method and the bottom-up method mentioned in Section 4.5. Besides, we need to extensively test out and compare the other criteria for model selection including AIC and MDL, in addition to likelihood ratio test we adopted in this paper. We also plan to devise a method of finding nodes that caused the burst based on the change points detected, which evolves into a spatio-temporal analysis of the tree structure representing the information diffusion path.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*, 716–723.

Araujo, L., Cuesta, J.A., Merelo, J.J. (2006). Genetic algorithm for burst detection and activity tracking in event streams. In *Proceedings of the 9th international conference on Parallel Problem Solving from Nature (PPSN'06)* (pp. 302–311).

Bonacichi, P. (1987). Power and centrality: a family of measures. *American Journal of Sociology, 92*, 1170–1182.

Ebina, R., Nakamura, K., Oyanagi, S. (2011). A real-time burst detection method. In *Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 1040–1046).

Goldenberg, J., Libai, B., Muller, E. (2001). Talk of the network: a complex systems look at the underlying process of word-of-mouth. *Marketing Letters, 12*, 211–223.

Katz, L. (1953). A new status index derived from sociometric analysis. *Sociometry, 18*, 39–43.

Kempe, D., Kleinberg, J., Tardos, E. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2003)* (pp. 137–146).

Kimura, M., Saito, K., Nakano, R., Motoda, H. (2010). Extracting influential nodes on a social network for information diffusion. *Data Mining and Knowledge Discovery, 20*, 70–97.

Kleinberg, J. (2002). Bursty and hierarchical structure in streams. In: *Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2002)* (pp. 91–101).

Rissanen, J. (1989). *Stochastic complexity in statistical inquiry*. World Scientific.

Sadikov, E., Medina, M., Leskovec, J., Garcia-Molina, H. (2011). Correcting for missing data in information cascades. In *Proceedings of the 4th ACM international conference on Web Search and Data Mining (WSDM 2011)* (pp. 55–64).

Saito, K., Kimura, M., Ohara, K., Motoda, H. (2009). Learning continuous-time information diffusion model for social behavioral data analysis. In *Proceedings of the 1st Asian Conference on Machine Learning (ACML2009), LNAI* (Vol. 5828, pp. 322–337).

Saito, K., Kimura, M., Ohara, K., Motoda, H. (2010). Selecting information diffusion models over social networks for behavioral analysis. In *Proceedings of the 2010 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2010), LNAI* (Vol. 6323, pp. 180–195).

Sun, A., Zeng, D., Chen, H. (2010). Burst detection from multiple data streams: a network-based approach. *IEEE Transactions on Systems, Man, & Cybernetics Society, Part C, 40*, 258–267.

Wasserman, S., & Faust, K. (1994). *Social network analysis*. Cambridge, UK: Cambridge University Press.

Watts, D.J. (2002). A simple model of global cascades on random networks. *Proceedings of National Academy of Science USA, 99*, 5766–5771.

Watts, D.J., & Dodds, P.S. (2007). Influence, networks, and public opinion formation. *Journal of Consumer Research, 34*, 441–458.

Zhang, X. (2006). *Fast algorithms for burst detection*. PhD dissertation, New York University. http://pdf.aminer.org/000/301/507/better_burst_detection.pdf.

Zhu, Y., & Shasha, D. (2003). Efficient elastic burst detection in data streams. In *Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2003)* (pp. 336–345).