Change point detection for information diffusion tree

Kouzou Ohara[1], Kazumi Saito[2], Masahiro Kimura[3], and Hiroshi Motoda[4,5]

[1] Department of Integrated Information Technology, Aoyama Gakuin University
ohara@it.aoyama.ac.jp
[2] School of Administration and Informatics, University of Shizuoka
k-saito@u-shizuoka-ken.ac.jp
[3] Department of Electronics and Informatics, Ryukoku University
kimura@rins.ryukoku.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
motoda@ar.sanken.osaka-u.ac.jp
[5] School of Computing and Information Systems, University of Tasmania
hmotoda@utas.edu.au

Abstract. We propose a method of detecting the points at which the speed of information diffusion changed from an observed diffusion sequence data over a social network, explicitly taking the network structure into account. Thus, change in diffusion is both spatial and temporal. This is different from most of the existing change detection approaches in which all the diffusion information is projected on a single time line and the search is made in this time axis. We formulate this as a search problem of change points and their respective change rates under the framework of maximum log-likelihood embedded in MDL. Time complexity of the search is almost proportional to the number of observed data points and the method is very efficient. We tested this using both a real Twitter date (ground truth not known) and the synthetic data (ground truth known), and demonstrated that the proposed method can detect the change points efficiently and the results are very different from the existing sequence-based (time axis) approach (Kleinberg's method).

Keywords: Social networks, Information diffusion, Change point 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. 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 reality, the way information diffuses depends on both the content and the interest of the people. Being able to detect changes in the way information propagates allows us

[1] https://twitter.com/

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

to analyze peoples behavior, e.g. finding a community of people with a similar interest, and deepens our understanding of the world around us. This brings in an important and interesting problem, which is to accurately and efficiently detect the change points (where in the network the changes take place and how big the respective changes in the diffusion speed are) from the observed information diffusion data.

There are substantial number of studies on change detection in information diffusion process. Most of them treat change detection along the time axis alone in which all the diffusion information is projected on a single time line and the detection is formulated as a search problem in this time axis. These include [9], [8], [2], [1], [3], [7]. We have also approached this problem by directly dealing with the change of time interval between occurrences of a target event [6], and showed that our method outperformed Kleinberg's method [3] which is considered to be the state of the art. However, in reality information diffusion takes place along a diffusion path. Each path has multiple descendants (child nodes) and new paths start only from the children that are in the observed data. Thus, change in diffusion is both spatial and temporal. The above traditional sequence-based (time axis) approaches may be good enough to know a global trend over a long period of time, but is definitely not good enough to detect the correct change points. Information diffuses differently within different communities just as the sound velocity changes within different substances. Thus it is important to take both spatial and temporal factors into account in detecting changes, *i.e.*, where and when the change takes place.

We model these changes as changes in the time-delay parameter, where the delay is assumed to follow an exponential distribution. More precisely, we assume that the parameter changes are approximated by a step function along each diffusion path and propose an optimization algorithm that maximizes the likelihood of generating the observed diffusion sequence, and the number of change points are determined by MDL principle. The time complexity of the algorithm 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 bursts using a real Twitter data quite efficiently. The results were very different from Kleinberg's method [3] which is considered to be the state of the art for burst detection along the time axis. This confirmed the need to explicitly use the network structure. Since we do not know the ground truth for the Twitter data, we generated synthetic data and embedded the change points of varying number using the same network structure with the Twitter data. The proposed method could successfully detect the correct change points for all cases with one very minor mis-detection, while Kleinberg's method again performed very poorly and the detected many incorrect change points.

2 Proposed Method

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 $C = \{(v_0, t_0), (v_1, t_1), \cdots, (v_N, t_N)\}$ that arose from the information released at the source node v_0 at time t_0. Here, v_n is an *active* node where the information has been propagated and t_n is its time. We assume, as a standard setting, that the actual information diffusion paths of a sequence C correspond to a tree T_C that is embedded in the directed graph G representing the social network [5], i.e., the parent node which passed the information to a node v_n is uniquely identified to be $v_{p(n)}$ if $n > 0$. Here, $p(n)$ is a function that returns the node identification number of the parent of v_n in the range of $\{0, \cdots, n-1\}$.

By setting that the time delay of information diffusion is represented as the simple exponential distribution $p(t_n - t_{p(n)}; r) = r \exp(-r(t_n - t_{p(n)}))$, we mathematically define the change point detection problem. For the actual information diffusion paths of a sequence C, we consider the corresponding set of integers defined by $\mathcal{D} = \{0, 1, \cdots, N\}$. Let the node of the j-th change point be $n(j) \in \mathcal{D}$, then we assume that the delay parameter switches from r_j to r_{j+1} for the descendant nodes of $v_{n(j)}$ until another change took place. Namely, we are assuming a step function as a shape of parameter changes. Let the set comprising J change points be $\mathcal{S}_J = \{n(1), \cdots, n(J)\}$, and we set $n(0) = 0$ for the sake of convenience ($t_{n(j-1)} < t_{n(j)}$). Let the division of $\mathcal{D}$ by $\mathcal{S}_J$ be $\mathcal{D}_j$, i.e., $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_J$, where $\mathcal{D}_j$ is a set of the descendant nodes of $v_{n(j)}$ until another change happens, and $|\mathcal{D}_j|$ represents the number of observed points in $\mathcal{D}_j$. Here, we request that $|\mathcal{D}_j| \neq 0$ for any $j \in \{0, \cdots, J\}$.

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 nodes of information diffusion result C is given. For this purpose, we consider maximizing the following objective function.

$$L(C; \hat{\mathbf{r}}_{J+1}, \mathcal{S}_J) = -N - \sum_{j=0}^{J} |\mathcal{D}_j| \log \left(\frac{1}{|\mathcal{D}_j|} \sum_{n \in \mathcal{D}_j} (t_n - t_{p(n)}) \right). \tag{1}$$

Here, as shown in [6], we can obtain this objective function by substituting the maximum likelihood estimate of the parameter $\hat{\mathbf{r}}_{J+1}$ to the log-likelihood for C for a given set of change points $\mathcal{S}_J$. We first describe the simple method which is applicable when the number of change points J is large. This is a progressive binary splitting without backtracking. Below we describe the details of this algorithm **A**: after initializing $j \leftarrow 1$ and $\mathcal{S}_0 \leftarrow \emptyset$ (step **A1**), we fix the already selected set of $(j-1)$ change points $\mathcal{S}_{j-1}$ and search for the optimal j-th change point $n(j)$ (step **A2**), and add it to $\mathcal{S}_{j-1}$ (step **A3**). We repeat this procedure from $j = 1$ to J. Here note that in the step **A3** elements of the change point set $\mathcal{S}_j$ are reindexed to satisfy $t_{n(i-1)} < t_{n(i)}$ 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.

By inheriting the basic idea of our previous method [6], we propose a method which is computationally almost equivalent to the simple method but gives a solution of much better quality. Below we describe the details of this algorithm **B**: We start with the solution obtained by the simple method $\mathcal{S}_J$ (step **B1**), pick up a change point $n(j)$ from the already selected points, fix the rest $\mathcal{S}_J \setminus \{n(j)\}$ and search for a better value $n(j)'$ (step **B2**), 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. $n(j)' = n(j)$ for all j, then no better solution is expected and the iteration stops.

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 C. Now, we present a method of estimating the value of J from C for solving the change points detection problem. To this end, we employ MDL (Rissanen's Minimum Description Length) [4]. More specifically, in order to describe the information diffusion model based on the obtained result $\mathcal{S}_J$, we need the set of $J+1$ time-delay parameters $\hat{\mathbf{r}}_{J+1}$, as well as the set of J change points $\mathcal{S}_J$, which amounts to $2J+1$ parameters. Thus we can consider the following MDL formula for the case of J change points:

$$MDL(J) \;=\; -L(C;\hat{\mathbf{r}}_J, \mathcal{S}_{J+1}) \;+\; \frac{1}{2}(2J+1)\log(N). \tag{2}$$

Below we describe the details of this algorithm **C**: after initializing $J \leftarrow 0$ and $\mathcal{S}_0 \leftarrow \emptyset$ (step **C1**), we compute $\mathcal{S}_{J+1}$ by the proposed algorithms **A** and **B**, and Calculate $MDL(J+1)$ from Equation (2) (step **C2**). We repeat this procedure from $J = 0$ by setting $J \leftarrow J+1$ until $MDL(J+1) \leq MDL(J)$. Here, we note that for model selection, we can consider employing various methods other than the MDL criterion and the likelihood ratio test, although we used the MDL criterion as a candidate.

3 Experiments

We applied the proposed method to the real-world information diffusion sequence which takes a form of tree and investigated how it can detect reasonable change points on the tree by visualizing the resulting change points and corresponding time-delay parameters estimated by it. To this end, we used a sequence of retweets extracted from Twitter [3], and formed a corresponding diffusion tree that has 477 nodes (tweets) and 476 edges (retweet actions). We refer to this dataset as the Retweet dataset.

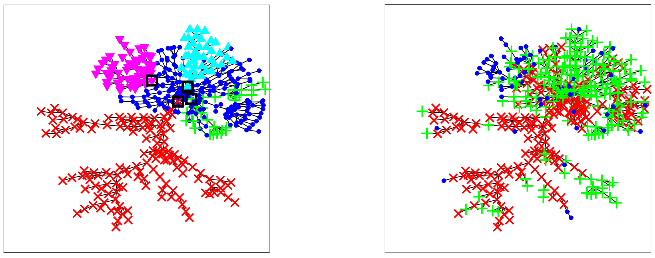
3.1 Results for Real Data

We applied our proposed method to the Retweet dataset, and obtained the result that the number of change points underlying in the tree is 4. Actually, the log-likelihoods for $J = 4$ and 5 are -33596.4 and -3353.9, respectively, and the corresponding MDL values are 3387.4 and 3387.9, respectively. We can observe that those values do not change significantly between $J = 4$ and 5, but it does not hold if J is smaller. Figure 1(a) visualizes the result for $J = 4$, in which nodes of the diffusion tree are denoted by different colors and different markers according to the estimated time-delay parameter values associated to them and the four change points detected are indicated with squares.

From these results, we can find that the given diffusion tree is clearly divided into 5 subtrees which have a certain number of nodes and whose root nodes are either the root node of the whole tree or change points detected by the proposed method. In addition, it can be observed that the diffusion speed clearly changes between different subtrees. Thus, these subtrees are likely to be considered as different communities in which information diffusion speed of a certain topic is different. Analyzing these subtrees more in depth is one of the future directions of this work.

Next, we compared the proposed method with conventional sequence-based methods [3, 6] that detect change points by considering only a time series diffusion sequence

[3] https://twitter.com/



(a) Proposed

(b) KLBG1

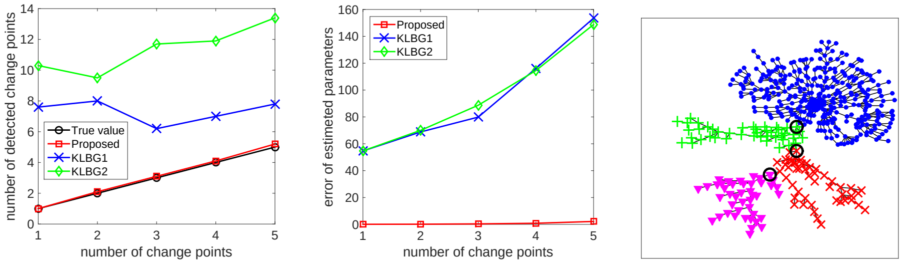
Fig. 1. Visualization of changes of diffusion time on the information diffusion tree by the proposed and KLBG1 methods.

without using any structural information of the network behind the diffusion. In this paper, we chose Kleinberg's method [3] as a representative one among them. It is based on hidden Markov model and has two parameters, γ and s. The parameter γ is used in its cost function, and we employed $\gamma = 1$ in this experiment. The parameter s is a scaling parameter and determines the delay parameter at the state j by $r_j = s^j r_0$ where the parameter r_0 is estimated by $r_0 = N/t_N$ as described in [3]. We set the scaling parameter s to 5 based on the observations obtained by applying our proposed method to the original dataset. Hereafter, we refer to Kleinberg's method with this setting as the KLBG1 method. In addition, we consider an alternative Kleinberg's method with another setting in which r_0 is fixed to 1.0, and refer to it as the KLBG2 method, which is used only for the experiments on the synthetic datasets discussed below.

Figure 1(b) shows, in the same manner as in Fig. 1(a), the result obtained by applying the KLBG1 method to the Retweet dataset. Comparing to Fig. 1(a), it is found that the number of change points detected by the KLBG1 method is substantially larger than the one by the proposed method. In addition, there are multiple small subtrees with an identical time delay parameter and they spread across a wide range of the diffusion tree. This is because the sequence-based methods use only a sequence of time stamps projected on a single time axis and do not take into account any structural information behind the diffusion process. Consequently, we cannot utilize this result to extract meaningful node groups or communities that could affect the information diffusion speed, which is possible by the proposed method.

3.2 Results for Synthetic Data

We constructed a synthetic sequence of information diffusion by utilizing the Retweet dataset. More specifically, to systematically regenerate the observation time points in which J change points are embedded, we divided $\mathcal{D}$ of the Retweet dataset into $J+1$ subsets $\mathcal{D}_0, \cdots, \mathcal{D}_J$ so that the original diffusion tree is decomposed into $J+1$ subtrees each of which has at least 20 nodes. Then, we set the time-delay parameter r_j to 1.0 for $j = 0$ and $5 \times r_{pt(j)}$ for $j > 0$, where $pt(j)$ means the index such that $p(n(j)) \in \mathcal{D}_{pt(j)}$. It is noted that this coefficient of 5 is equivalent to the value of the scaling parameter of the KLBG1 and KLBG2 methods. After that, we generated observation time of nodes

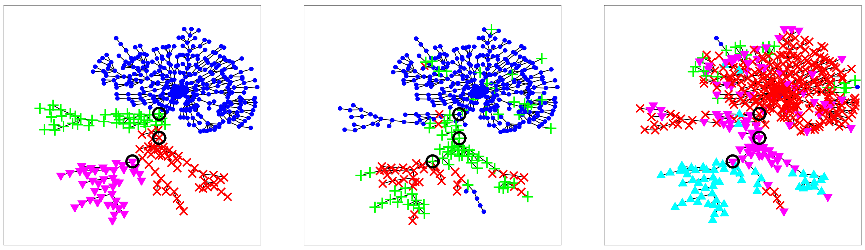


(a) Number of detected change points

(b) Error of estimated parameters

(c) True change points ($J = 3$)

Fig. 2. Learning performance by the proposed, KLBG1, and KLBG2 methods for $J = 1$ to 5.



(a) Proposed

(b) KLBG1

(c) KLBG2

Fig. 3. Visualization results using the true change points and those estimated by the proposed, KLBG1, and KLBG2 methods for a synthetic dataset having $J = 3$ change points.

in each $\mathcal{D}_j$ according to the exponential distribution with the parameter r_j, varying J from 1 to 5, and generated 10 different datasets for each value of J.

To quantitatively evaluate the proposed method, we applied the proposed, KLBG1, and KLBG2 methods to the synthetic datasets, and compared their learning performance in terms of two criteria: the number of detected change points and the estimation error of the time-delay parameter. For each value of J, we applied each method to the 10 different synthetic datasets, each embedded with J change points, and computed an average over these 10 trials for each criterion. Figure 2(a) shows the number of change points detected by each method. It is obvious that the proposed method can almost exactly detect the number of embedded change points regardless of the value of J. In contrast, both the KLBG1 and KLBG2 methods overestimated the number of embedded change points. The KLBG2 method detected much more change points than the KLBG1 method did although the KLBG2 method used the true value of r_0 in addition to the true scale parameter s that was available for the KLBG1 method.

Next, we investigated the error E between the estimated time-delay parameter and the true one, defined as $E = N^{-1} \sum_{n=1}^{N} |\hat{r}(n) - r(n)|$, where $\hat{r}(n)$ is a parameter value that is estimated to have generated the time delay $t_n - t_{p(n)}$, and $r(n)$ is its true value. Since both the KLBG1 and KLBG2 methods do not use any structural information of the diffusion tree, we defined $p(n)$ as $p(n) = \arg\max_{m \in D}\{t_m | t_m < t_n\}$ for these two methods so that E

gets to a small value if they exactly detect change points and estimate the corresponding parameter values within small deviations. The results for each value of J are shown in Fig. 2(b), from which it is clear that the proposed method achieved extremely small errors, and thus can accurately estimate the parameter value for any value of J. On the other hand, the errors for the KLBG1 and KLBG2 methods are extremely large and increase in proportion to the number of embedded change points J.

Figure 3 visualizes, in the way similar to the case of Fig. 1(a), results for a synthetic dataset in which $J = 3$ change points were embedded. Figures. 3(a) to 3(c) show the results of the proposed, KLBG1, and KLBG2 methods, respectively. The same 3 true change points illustrated by circles in Fig. 2(c) were used for the three methods. Comparing Figs. 2(c) and 3(a), we can see that the proposed method almost exactly detected the 3 true change points in the tree. In contrast, from Figs. 3(b) and 3(c), we see that they are much different from Fig. 2(c), and the diffusion speed changes at many nodes other than the true change points. The KLBG1 method is slightly better than the KLBG2 method, but the number of states it detected is 3 that is one less than the true value 4. The KLBG2 method that uses the true value of r_0 detected 5 states and there are many more change points than the true ones.

4 Conclusion

We addressed the problem of detecting the points at which the speed of information diffusion changed from a single observed diffusion sequence under the assumption that the delay of the information propagation follows the exponential distribution. Most of the existing change detection methods focus on changes in the time axis, ignoring the path along which information diffuses within the network. The proposed method is different and unique in that it explicitly takes the underlying network structure into account. It can deal with both spatial and temporal changes in information diffusion.

We formulated this problem as an optimization problem of maximizing the likelihood of generating the observed data. In doing so the change detected at a node is passed only to its descendants, and different information diffusion paths are handled in parallel. We devised an efficient iterative search algorithm whose time complexity is almost linear to the number of data points, and determined the optimal number of change points using MDL criterion. We tested the algorithm against the real Twitter data for which we do not know the ground truth and a synthetic data for which we know the ground truth.

The results for the real Twitter data revealed that the proposed method can detect change points efficiently. We also tested the other method that does not use the network structure data, choosing Kleinberg's burst detection method as one of the representative methods of this kind. The results were very different, which confirmed the need to explicitly use the network structure. The results for the synthetic data reveled that the proposed method could successfully detect the correct change points for all cases with one very minor mis-detection, while Kleinberg's method again performed very poorly and the detected many incorrect change points.

Acknowledgments

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. 26330261).

References

1. Araujo, L., Cuesta, J.A., Merelo, J.J.: 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 (2006)
2. Ebina, R., Nakamura, K., Oyanagi, S.: A real-time burst detection method. In: Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI). pp. 1040–1046 (2011)
3. Kleinberg, J.: 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 (2002)
4. Rissanen, J.: Stochastic Complexity in Statistical Inquiry. World Scientific (1989)
5. Sadikov, E., Medina, M., Leskovec, J., Garcia-Molina, H.: 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 (2011)
6. Saito, K., Ohara, K., Kimura, M., Motoda, H.: Change point detection for burst analysis from an observed information diffusion sequence of tweets. Journal of Intelligent Information Systems (JIIS) 44, 243–269 (2015)
7. Sun, A., Zeng, D., Chen, H.: Burst detection from multiple data streams: A network-based approach. IEEE Transactions on Systems, Man, & Cybernetics Society, Part C pp. 258–267 (2010)
8. Zhang, X.: Fast Algorithms for Burst Detection. PhD dissertation (New York University) (2006)
9. Zhu, Y., Shasha, D.: 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 (2003)