A Method to Divide Stream Data of Scores over Review Sites

Yuki Yamagishi[1], Seiya Okubo[1], Kazumi Saito[1], Kouzou Ohara[2],
Masahiro Kimura[3], and Hiroshi Motoda[4]

[1] University of Shizuoka, Shizuoka 422-8526, Japan
{j14505,s-okubo,k-saito}@u-shizuoka-ken.ac.jp
[2] Aoyama Gakuin University, Kanagawa 252-5258, Japan
ohara@it.aoyama.ac.jp
[3] Ryukoku University, Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[4] Osaka University, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. The word of mouth information over certain review sites affects various activities from person to person. In large-scale review sites, it can happen that evaluation tendency of a product changes in a large way by only a few reviews that were rated and posted by certain users. Thus, it is very important to be able to detect those influential reviews in social media analysis. We propose an algorithm that can efficiently divide stream data of review scores by maximizing the likelihood of generating the observed sequence data. We assume that the user's fundamental scoring behavior follows a multinomial distribution model and formulate a division problem.

Keywords: Data stream, Local improvement, Review site, Likelihood function.

1 Introduction

In recent years, reviews posted by users in review sites are explosively increasing and those affect directly user's purchase decisions and after-purchase actions. This implies that review sites are becoming one of the important social media, which has an influence on sales promotion of products and services. There have been a large number of studies on social media from various aspects. One typical direction is modeling how information propagates through a social network (Yang and Counts 2010; Yang and Leskovec 2010; Bakshy et al. 2011; Cui et al. 2011; Guille and Hacid 2012). At a more fundamental level, sentiment analysis tries to classify contents on social media for a certain topic (Melville et al. 2009; Pak and Paroubek 2010; Glass and Colbaugh 2011), which could allow some companies to know how their products are evaluated by consumers.

We are not only interested in knowing what is happening now and how it will develop in the future, but also in knowing what happened in the past and how it was caused, by some crucial changes in the distribution of the information. Such changes might involve changes in the number of reviews which are posted in a certain period,

D.-N. Pham and S.-B. Park (Eds.): PRICAI 2014, LNAI 8862, pp. 913–919, 2014.

i.e., changes in the posting interval and/or frequency, in which case existing burst detection techniques (Kleinberg 2002; Zhu and Shasha 2003; Sun et al. 2010) would be applicable. However, if no change in the time interval is involved, we would not be able to use these techniques because they do not focus on the change in the content. In other words, they intend to detect a burst for a single topic, and do not deal directly with multiple topics and the change of their distribution.

Although these time series events that happen over a review site are of high value to items reviewed, it is hard to guess them from the average score and recent reviews. For this reason, a study of detecting changes of score distribution from a huge number of reviews is important in the field of web intelligence. Therefore, we propose an interval division method for score time series data from a retrospective point of view in a way similar to the work of Kleinberg (2002), Swan and Allan (2000). The problem settings and the algorithm of the proposed method are similar to that of what Saito et al. (2013) have already proposed, but a newly introduced procedure which iteratively improves the solution quality is totally new.

2 Problem Settings

We focus on the stream data of scores over a review site where items are evaluated by scores with J-category and formally define the division point detection problem. As mentioned in Section 1, our goal is to detect scoring changes and how long these changes might have persisted. Let the score and the time of n-th review of an item be s_n and t_n, respectively. Then, let us denote the observed scoring vector of an item as $\mathcal{D} = \{(s_1, t_1), \cdots, (s_N, t_N)\}$, where $n \in \mathcal{N} = \{1, \cdots, N\}$ and $s_n \in \mathcal{J} = \{1, \cdots, J\}$. For convenience sake of our model description, we introduce a dummy function to convert s_n to a J-dimensional vector, *i.e.*, $s_{n,j} = \{1$ if $s_n = j$; 0 otherwise. Here, we assume that p_j which is a probability of giving the score j is a multinomial distribution. Accordingly, the log-likelihood for $\mathcal{D}$ is calculated by the parameter vector $\boldsymbol{p} = \{p_1, \cdots, p_J\}$ as $\mathcal{L}(\mathcal{D}; \boldsymbol{p}) = \sum_{n\in\mathcal{N}} \sum_{j\in\mathcal{J}} s_{n,j} \log p_j$. Thus, the maximum likelihood estimators of $\mathcal{L}(\mathcal{D}; \boldsymbol{p})$ is given by $\hat{p}_j = \sum_{n\in\mathcal{N}} s_{n,j}/|\mathcal{N}|$.

Let the time of the k-th division point be T_k $(t_1 < T_k < t_N)$. The parameter vector that the distribution follows switches from $\boldsymbol{p}_k$ to $\boldsymbol{p}_{k+1}$ at the k-th point T_k. Namely, we are assuming a series of step functions as a shape of parameter vector changes. Let the set comprising K division points be $C_K = \{T_1, \cdots, T_K\}$, and we set $T_0 = t_1$ and $T_{K+1} = t_N$ for the sake of convenience and $T_{k-1} < T_k$. Let the division of $\mathcal{D}$ by C_K be $\mathcal{D}_k = \{n; T_{k-1} < t_n \leq T_k\}$, *i.e.*, $\mathcal{N} = \{1, \cdots, N\} = \{1\} \cup \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_{K+1}$, and $|\mathcal{D}_k|$ represents the number of observed points in $(T_{k-1}, T_k]$. Here, we request that $|\mathcal{D}_k| \neq 0$ for any $k \in \mathcal{K} = \{1, \cdots, K+1\}$ and there exists at least one t_n and $t_n \in \mathcal{D}_k$ is satisfied. The problem of detecting division points is equivalent to a problem of finding a subset $C_K \subset \mathcal{T}$ where $\mathcal{T}$ is a set of the observed time points, *i.e.*, $\mathcal{T} = \{t_1, t_2, \cdots, t_N\}$.

The log-likelihood for the $\mathcal{D}$, given a set of division points C_K is calculated by defining the parameter vectors $\boldsymbol{P}_{K+1} = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_{K+1}\}$ as $\mathcal{L}(\mathcal{D}; \boldsymbol{P}_{K+1}, C_K) = \sum_{k\in\mathcal{K}} \mathcal{L}(\mathcal{D}_k; \boldsymbol{p}_k)$. Therefore, the maximum likelihood estimators of $\mathcal{L}(\mathcal{D}; \boldsymbol{P}_{K+1}, C_K)$ is given by $\hat{p}_{k,j} = \sum_{n\in\mathcal{D}_k} s_{n,j}/|\mathcal{D}_k|$ for $k = 1, \cdots, K+1$ and $j = 1, \cdots, J$. Substituting these estimators to $\mathcal{L}(\mathcal{D}; \boldsymbol{P}_{K+1}, C_K)$ leads to $\mathcal{L}(\mathcal{D}; \hat{\boldsymbol{P}}_{K+1}, C_K) = \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{D}_k} \sum_{j\in\mathcal{J}} s_{n,j} \log \hat{p}_{k,j}$. Thus, the division point detection problem is reduced to the problem of finding the division point set C_K that maximizes $\mathcal{L}(\mathcal{D}; \hat{\boldsymbol{P}}_{K+1}, C_K)$. However, $\mathcal{L}(\mathcal{D}; \hat{\boldsymbol{P}}_{K+1}, C_K)$ alone does not allow us to evaluate directly the effect of introducing C_K. It is important to evaluate how the log-likelihood improves over the one obtained without considering the parameter changes. Therefore, we reformulate the problem as the maximization problem of log-likelihood ratio. If we do not assume any changes, *i.e.*, $C_0 = \emptyset$, $\mathcal{L}(\mathcal{D}; \hat{\boldsymbol{P}}_{K+1}, C_K)$ is reduced to $\mathcal{L}(\mathcal{D}; \hat{\boldsymbol{P}}_1, C_0) = \sum_{n \in \mathcal{N}} \sum_{j \in \mathcal{J}} s_{n,j} \log \hat{p}_{1,j}$, where $\hat{p}_{1,j} = \sum_{n \in \mathcal{N}} s_{n,j}/|\mathcal{N}|$. Thus, the log-likelihood ratio of the two cases, one with K division points and the other with no division points is given by $\mathcal{LR}(C_K) = \mathcal{L}(\mathcal{D}; \hat{\boldsymbol{P}}_{K+1}, C_K) - \mathcal{L}(\mathcal{D}; \hat{\boldsymbol{P}}_1, C_0)$. In summary we consider the problem of finding the set of division points C_K that maximizes $\mathcal{LR}(C_K)$ defined above.

3 Division Point Detection Method

First, we describe the procedure of a greedy search, hereinafter referred to as *A1*. This is a progressive binary splitting without backtracking. We fix the selected set of $(k-1)$ division points C_{k-1} and search for the optimal k-th division point T_k and add it to C_{k-1}. Generally $2(\mathcal{LR}(C_k) - \mathcal{LR}(C_{k-1}))$ follows a χ^2-distribution when N is sufficiently large, thus we use the χ^2-test as the termination condition of this search. The algorithm is given below.

A1-1. Initialize $k = 1, C_0 = \emptyset$.
A1-2. Search for $T_k = \arg\max_{t_n \in \mathcal{T}} \{\mathcal{LR}(C_{k-1} \cup \{t_n\})\}$.
A1-3. Update $C_k = C_{k-1} \cup \{T_k\}$.
A1-4. If $2(\mathcal{LR}(C_k) - \mathcal{LR}(C_{k-1}))$ is lower than the critical value of χ^2, which was set according to the pre-specified significance level and the degree of freedom $J-1$, output C_K and stop.
A1-5. $k = k + 1$, and return to A1-2.

Here note that in A1-3 elements of the division point set C_k are reindexed to satisfy $T_{i-1} < T_i$ for $i = 2, \cdots, k$. Clearly, the time complexity of this simple algorithm is $O(NK)$ which is fast. Thus, it is possible to obtain the result within an allowable computation time even for a large N. However, since this is a greedy algorithm, it can be trapped easily in a poor local optimal.

Next, we describe the procedure of a local search, hereinafter referred to as *A2*. We start with the solution C_K obtained by A1, pick up a division point T_k from the list, fix the rest $C_K \setminus \{T_k\}$ and search for the better value T'_k of T_k, where $\cdot \setminus \cdot$ represents set difference. We repeat this from $k = 1$ to K. If no replacement is possible for all k $(k = 1, \cdots, K)$, *i.e.*, $T'_k = T_k$ for all k, no better solution is expected and the iteration stops. The algorithm is given below.

A2-1. Initialize $k = 1, h = 0$.
A2-2. Search for $T'_k = \arg\max_{t_n \in \mathcal{T}} \{\mathcal{LR}(C_K \setminus \{T_k\} \cup \{t_n\})\}$.
A2-3. If $T'_k = T_k$, set $h = h + 1$, otherwise set $h = 0$, and update $C_K = C_K \setminus \{T_k\} \cup \{T'_k\}$.
A2-4. If $h = K$, output C_K and stop.
A2-5. If $k = K$, set $k = 1$, otherwise set $k = k + 1$, and return to A2-2.

It is evident that this algorithm requires computation time several times larger than that of the greedy algorithm, but the solution quality is substantially improved over the greedy solutions.

3.1 Proposed Iterative Method

We refer to a basic procedure that obtain C_K by A1 and improve C_K by A2 as *Sequential Method*. The drawback of this procedure is that improvement starts only after all the division points K has been obtained. We can embed the improvement step within the first step. The following *Iterative Method* would be more credible in reaching the converged solution. The procedure is given below.

I1. Start the process from A1-1.
I2. If $k \geq 2$ at the beginning of A1-5, output C_k as C_K.
I3. Improve C_K by A2 and output C_K as C_k.
I4. Restart the process from A1-5, and return to I2.

4 Experimental Evaluation

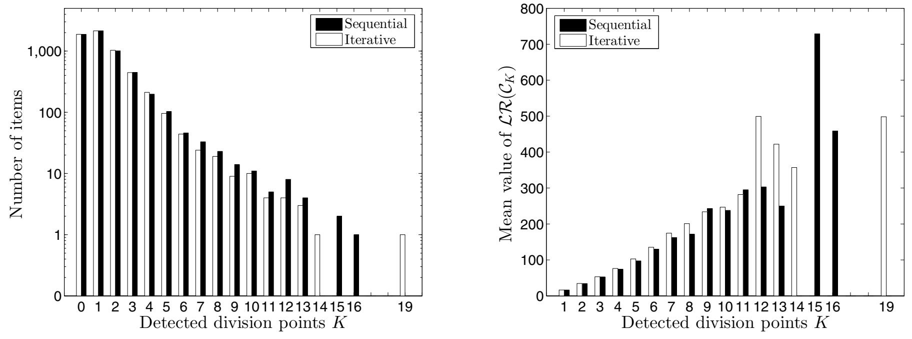
In what follows, we explain a real review dataset and show the results of the division point detection by our method. Here, we pre-specified the significance level $p = 0.005$, which determines when to terminate the A1 process.

We collected real stream data of scores from "@cosme" [1] which is a Japanese large-scale review site for cosmetics. Because the termination condition in our method assumes that there are sufficient samples, we targeted the items whose number of reviews $N = |\mathcal{D}|$ were greater than 200. In the collected @cosme dataset, the number of items is 5,924, the number of reviews is 4,329,702, the range of scores is 0 to 7 (used as $j = 1, \cdots, 8$).

First, we show two plots of statistics about division points K which are detected by Sequential Method and Iterative Method. Fig. 1(a) shows the frequency distribution of the number of detected division points K, and Fig. 1(b) shows the mean log-likelihood ratio $\mathcal{LR}(C_K)$ of the number of division points K. From Fig. 1(a), we can see the expected numbers of division points obtained by both methods are almost the same. Actually, the mean number of division points K per item for Sequential Method is 1.347 and that for Iterative Method is 1.316. From Fig. 1(b), we can see the expected value of the objective function obtained by Iterative Method is slightly greater than that obtained by Sequential Method. As a matter of fact, the mean log-likelihood ratio $\mathcal{LR}(C_K)$ per division point obtained by Sequential Method is 18.317 and that obtained by Iterative Method is 18.506.

Next, we explore the relation between solution quality and time complexity. Fig. 2(a) compares the improved values of the objective function over their simple greedy solution for each number of division points K in local search process (A2), and Fig. 2(b) compares the increased rates of time complexity (A1 and A2 vs. A1 alone) for each

[1] `http://www.cosme.net/`

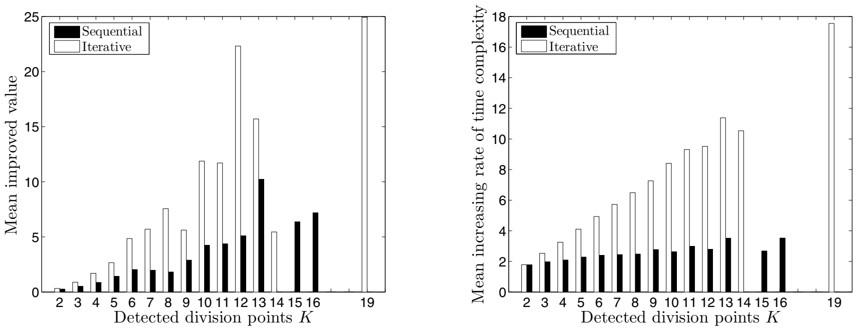


(a) Frequency distribution of K.

(b) Mean value of $\mathcal{LR}(C_K)$ for each K.

Fig. 1. Plots of the statistics about division points K

number of division points K. It can be seen clearly in Fig. 2(a) that the solution quality of Iterative Method is better than that of Sequential Method. In contrast, as shown in Fig. 2(b), the computation time of Sequential Method does not increase much as K increases, while that of Iterative Method does. Here, it should be noted that Iterative Method is also fast enough because time complexity in both methods is almost linear to the number of reviews.

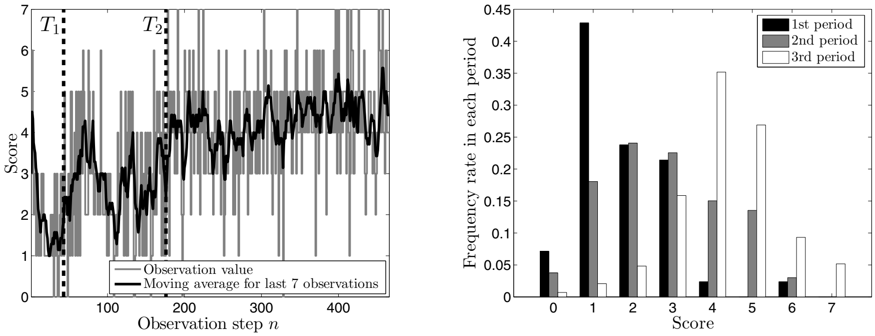


(a) Mean improved value of $\mathcal{LR}(C_K)$ for each K in A2.

(b) Mean increasing rates of time complexity (A1 and A2 vs. A1 alone) for each K.

Fig. 2. Plots of the relation between solution quality and time complexity

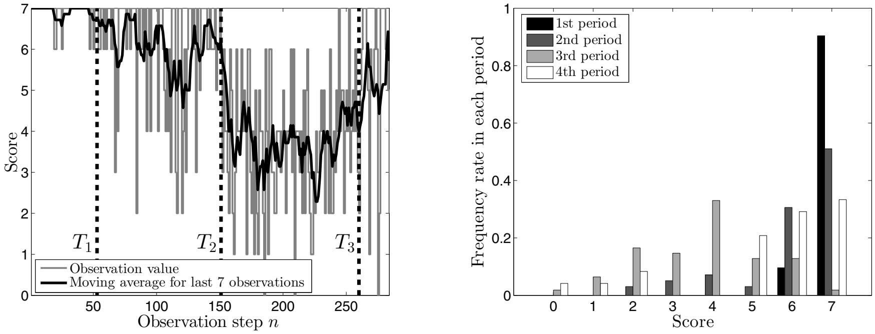
Finally, we show the details of individual items which are high in the ranking of the effective value $\mathcal{LR}(C_k) - \mathcal{LR}(C_{k-1})$. Here, we only show the results by Iterative Method that gives better solutions. For easier understanding the results, we selected items whose number of reviews $N = |\mathcal{D}|$ were less than 500. Figs. 3(a) to 4(b) show the detected division points T_k and the changes of the score distribution for two items which are the 1st and the 2nd place under the conditions mentioned above, respectively. As presented in these figures, we can say that our method could be applicable to divide such streaming data of scores as used in this study with a suitable number of periods according to change of a score distribution of reviews.



(a) Detected division points T_k

(b) Change of score distribution

Fig. 3. Item ID 2,512 (the 1st place under the conditions) in @cosme dataset



(a) Detected division points T_k

(b) Change of score distribution

Fig. 4. Item ID 4,298 (the 2nd place under the conditions) in @cosme dataset

5 Conclusion

This paper addressed the problem of detecting the division points from time series data of reviews by considering the change in the distribution of score. We formally defined the problem of detecting the division points as the problem of finding the model parameter values by maximizing the likelihood of the observed stream of score data being generated. We then devised efficient algorithms to search for the division points, whose time complexity is almost linear to the number of reviews. Our immediate future work is to evaluate experimentally the proposed method with synthetic stream data of scores in order to confirm whether or not it can detect the ground truth.

Acknowledgment. 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. 23500312).

References

Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone's an influencer: Quantifying influence on twitter. In: Proceedings of WSDM 2011, pp. 65–74 (2011)

Cui, P., Wang, F., Yang, S., Sun, L.: Item-level social influence prediction with probabilistic hybrid factor matrix factorization. In: Proceedings of AAAI 2011, pp. 331–336 (2011)

Glass, K., Colbaugh, R.: Estimating sentiment orientation in social media for business informatics. In: AAAI Spring Symposium, AI for Business Agility (2011)

Guille, A., Hacid, H.: A predictive model for the temporal dynamics of information diffusion in online social networks. In: Proceedings of WWW 2012, pp. 1145–1152 (2012)

Kleinberg, J.: Bursty and hierarchical structure in streams. In: Proceedings of KDD 2002, pp. 91–101 (2002)

Melville, P., Gryc, W., Lawrence, R.D.: Sentiment analysis of blogs by combining lexical knowledge with text classification. In: Proceedings of KDD 2009, pp. 1275–1284 (2009)

Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of LREC 2010, pp. 1320–1326 (2010)

Saito, K., Ohara, K., Kimura, M., Motoda, H.: Detecting Changes in Content and Posting Time Distributions in Social Media. In: Proceedings of ASONAM 2013, pp. 572–578 (2013)

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, 258–267 (2010)

Swan, R., Allan, J.: Automatic Generation of Overview Timelines. In: Proceedings of SIGIR 2000, pp. 49–56 (2000)

Yang, J., Counts, S.: Predicting the speed, scale, and range of information diffusion in twitter. In: Proceedings of ICWSM 2010 (2010)

Yang, J., Leskovec, J.: Modeling information diffusion in implicit networks. In: Proceedings of ICDM 2010, pp. 599–608 (2010)

Zhu, Y., Shasha, D.: Efficient elastic burst detection in data streams. In: Proceedings of KDD 2003, pp. 336–345 (2003)