5-26-2012

On Detecting Feasible Periodicity for Periodic Event in Binary Data Series

Rui Yang
School of Management and Economics, University of Electronic Science, aresyangrui@gmail.com

Hua Yuan
School of Management and Economics, University of Electronic Science

Yu Qian
School of Management and Economics, University of Electronic Science

Lun Hou
School of Management and Economics, University of Electronic Science

On Detecting Feasible Periodicity for Periodic Event in Binary Data Series

Yang Rui[1], *Yuan Hua*[1], *Qian Yu*[1], *Hou Lun*[11]
[1]School of Management and Economics, University of Electronic Science and Technology of China, China

Abstract: In this paper, we investigated the problem of discovering periodicity of a certain event in a binary data series and a new method basing on cross entropy is proposed. First, a series of rational partition methods for binary data series are introduced, which can divide the data series into different segments (partition). Then, we use cross entropy to calculate the partition periodicity, which could be the good measurements for the feasible of event periodicity. Finally, a periodicity evaluation method is proposed to obtain the feasible periodicity of the given event. The results of calculation example show that the method can be used to explore feasible event periodicity in binary data series.

Keywords: data mining, data series, cross entropy, periodicity

1. INTRODUCTION

The sequential data series is a series that is commonly used in presentation the events sequentially happened, such as e-business information query for an Internet user, transactions in a superstore, etc.

Let *x* be an event happened in relation with time, and all the appearances of *x* can be presented by a sequential data series as:

$$S = \{e_1; e_2; \dots; e_{|S|}\}, \tag{1}$$

in which $|S|$ is the length of time and e_i denotes the appearance of *x* at time *i*:

$$e_i = \begin{cases} 1, & \text{event } x \text{ has hanppened at time } i; \\ 0, & \text{Otherwise.} \end{cases} \tag{2}$$

Definition 1. The binary data series *S* is said to have *periodicity* if a certain event *x* is repeated periodically [1]. Note that, here we define that "*repeated*" means at least two periods.

Discovering the periodicity of each event happened in sequential data series is a valuable work for data analyzing. By identifying periodicity of a certain event in data series, it could reveal important observations about the behavior and future trends of the case represented by the data series, and hence would lead to more effective decision making [2, 3].

The traditional periodicity detection for a certain event is a process for finding temporal regularities within the data series. In this work, we address the problem of how to detect the feasible periods of event *x* in *S*. To this line, partition periodicity is introduced to investigate the problem of periodicity discovering in binary data series. First, $\pi(n)$-partition method for binary data series are introduced, which can divide binary data series into different segments. Then, we use cross entropy to measure the feasibility of all the real partitions for a certain event's appearances periodicity. Finally, we proposed an evaluation method to obtain the periodicity of periodic event.

* Corresponding author. Email: aresyangrui@gmail.com(Yang Rui)

The rest of this paper is organized as follows. Section 2 describes related work in the literature; Section 3 shows the whole methodology; A calculation example and some experiment results for the proposed method are shown in section 4; Finally, Section 5 concludes this paper.

2. RELATED WORK

The previous methods on periodicity patterns mining in large data sets are mainly on mining full periodic patterns and partial periodicity is very common in practice. In [4], Han et al. studied an interesting data mining problem of searching for partial periodic patterns in time-series databases. Promoted by this research, Cao et al. proposed a new structure, the abbreviated list table (ALT), and several efficient algorithms to compute the partial periods and patterns [5]. He et al. investigated an interesting type of periodic pattern, called partial periodic (PP) correlation in [6]. In [7], Yang et al. proposed a more flexible model of asynchronous periodic pattern that may be present only within a subsequence and whose occurrences may be shifted due to disturbance. Yang et al. proposed a new mining problem that is to find surprising periodic patterns in a sequence of data [8]. In [9], An efficient single-pass algorithm using a best-first search strategy without support threshold, called MTKPP (Mining Top-K Periodic-frequent Patterns), is proposed. To address "rare item problem", minimum constraint model has been extended to the basic model of periodic- frequent patterns [10]. In [11], J. Assfalg et al. presented a framework that provides similarity search in time series databases regarding specific periodic patterns. The common features of these researches are they assume that users either know the value of the period beforehand or are willing to try various period values until satisfactory periodic patterns emerge [1].

As for the periodicity detection in series data, Michail Vlachos et al. presented a non-parametric method for accurate periodicity detection and introduced a new periodic distance measures for time-series sequences [12]. Parthasarathy et al. have presented an algorithm for detecting periodicity in time series datasets, which leverages the frequency characterization and autocorrelation structure inherent in a time series to estimate its periodicity [13].

3. THE METHOD

3.1 Some basic definitions

Assume that a *periodic partition* $\pi = \{P_1 | P_2 | \ldots | P_k\}$, where $P_i = \{e_{(i-1)*\frac{|S|}{k}+1}, \cdots, e_{i*\frac{|S|}{k}}\}$, which divide the time interval [1, $|S|$] into k ($k \geq 2$) equal segments as follows:

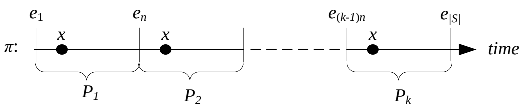


Figure 1. The partition over time interval [1, |*S*|]

Definition 2. The total appearances of event x in S is called the ***support*** of x, which is denoted by

$$supp(x) = \sum_{e_i \in S} e_i. \tag{3}$$

Accordingly, the total appearances of event x in P_j is denoted by

$$supp(x|P_j) = \sum_{e_i \in P_j} e_i. \tag{4}$$

With respect to periodic partition π, if $supp(x|P_j) > 0$, we say that x ***appears*** in P_j. Otherwise, event x ***never appears*** in period P_j.

Lemma 1. $supp(x) = \sum_{i=1}^{k} supp(x|P_i)$.

Lemma 2. *If π is a "good" partition to show the periodicity of event x in S, then the distribution of $supp(x|P_j)$ (j=1,..,k) will be equally.*

Proof. Since the periodic partition $\pi=\{P_1|P_2|\ldots|P_k\}$ divide time interval [1, |*S*|] into *k* equal segments, and *x* show periodicity w.r.t. partition π, then we can expect that $supp(x|P_i)\approx supp(x|P_j)$, where $i,j=1,..,k$, for any $i\neq j$. On the other hand, we can obtain the following result with *lemma* 1:

$$supp(x) = \sum_{i=1}^{k} supp(x|P_i) \approx k\times supp(x|P_i)_{i\in[1,\ldots,k]} \Rightarrow \frac{supp(x|P_i)}{supp(x)}\approx\frac{1}{k}.$$

That is, the probability of *x* appearing in P_j (j=1,..,*k*) is almost equally to 1/*k*. Along this line, the problem of detecting feasible periods of event *x* in *S* consists of two main steps: (1) discover a set of feasible partition $\pi=\{P_1|P_2|\ldots|P_k\}$; (2) find a function *f* to measure the closeness, i.e. feasibility, of distribution of $supp(x|P_j)$ to 1/*k*.

3.2 $\pi(n)$-partition method for binary series data

Assume that there exists a partition $\pi(n)$, which divide the time interval [1, |*S*|] as follows:

- Begin with the first event;
- Every continuous *n* appearances of *x*, i.e. e_i, are collected into a sub interval: $P_i=\{e_{(i-1)*n+1},\ldots,e_{i*n}\}$, $i\in[0,\lceil\frac{|S|}{n}\rceil]$, which means that events $\{e_{(i-1)*n+1},\ldots,e_{i*n}\}$ are deemed as appearing in the same period P_i by $\pi(n)$.

As a result, $\pi(n)=\left\{P_1|P_2|\ldots|P_{\lceil\frac{|S|}{n}\rceil}\right\}$ partitions the time interval [1, |*S*|] into $\lceil\frac{|S|}{n}\rceil$ continuous periods (segments). See Figure.2.

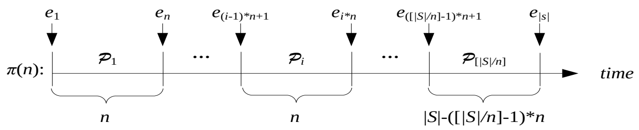


Figure 2. $\pi(n)$ partition method.

3.3 Cross entropy

Suppose *Y* is a discrete random variable that obtains values from a finite set $y_1, y_2, \ldots, y_n$, with probabilities $p_1, p_2, \ldots, p_n$. Shannon developed the concept of entropy to measure the uncertainty of discrete random variable [14] *Y* as:

$$H(Y) = -\sum_y p(y)\log p(y). \tag{5-1}$$

Further, given two probability distributions $p=\{p_1, p_2, \ldots, p_n\}$ and $q=\{q_1, q_2, \ldots, q_n\}$, the cross entropy or the Kullback-Leibler divergence [15] between *p* and *q* is defined by

$$D_{KL}(p||q)_n = \sum_y p(y)\log\frac{p(y)}{q(y)} \tag{5-2}$$

By convention, here $0\log 0=0$.

Theorem 1. $D_{KL}(p||q)_n=0$, if $p=q$ [15].

3.4 Partition periodicity detection

In this work, we will use cross entropy to measure the feasibility of a potential periodic event.

If $\pi(n)$ is a "good" partition to show the periodicity of event x in S, then it would divide the time interval [1, $|S|$] into $\left\lceil \frac{|S|}{n} \right\rceil$ continuous segments, and we can expect that the perfect distribution of x in each segment P_i can be referred as:

$$q_n = \left\{ \frac{1}{\left\lceil \frac{|S|}{n} \right\rceil}, \dots, \frac{1}{\left\lceil \frac{|S|}{n} \right\rceil} \right\} \tag{6-1}$$

That is to say, the partition periodicity reveals that how well-distributed the event x in S. Obviously, the more balanced appearances distribution of x in S, e.g., uniform distribution, the better periodicity for x in time interval [1, $|S|$].

Unfortunately, with the partition $\pi(n)$, the appearances of x in each segment P_i would be randomly in real data series. The posterior probability distribution can be calculated as:

$$p_n = \left\{ \frac{supp\,(x|P_1)}{supp\,(x)}, \frac{supp\,(x|P_2)}{supp\,(x)}, \dots, \frac{supp(x|P_{\lceil \frac{|S|}{n} \rceil})}{supp\,(x)} \right\} \tag{6-2}$$

We can obtain a value of $D_{\mathrm{KL}}(p||q)_n$ as:

$$D_{KL}(p||q)_n = \sum_{\lceil \frac{|S|}{n} \rceil} p_n \log \frac{p_n}{q_n} \tag{7}$$

In real, different partition $\pi(n)$ on sequence S will result different distribution p_n and q_n. Known from *theorem* 1, a smaller value of $D_{\mathrm{KL}}(p||q)_n$ means the posterior distribution p_n is more close to q_n. Let D_0 be the threshold measure of $D_{\mathrm{KL}}(p||q)_n$, all the "good" period partitions $\pi(n)$ for periodic event x should satisfy the condition that :

$$n^* = \arg_{1 \le n \le \lceil \frac{|S|}{2} \rceil,\, D_{KL}(p||q)_n \le D_0} \min \; \{D_{KL}(p||q)_n\}. \tag{8}$$

Relation (8) means that there may be more than one feasible period for x in S, in which, the definition of D_0 is more related to the management sense in real application. So, our goal is to finding a bunch of density p^*_n to minimizes the Kullback-Leibler distance. With this line, we propose the following greedy algorithm to calculate the feasibility of all the potential periods for x:

Algorithm 1 Calculation $D_{\mathrm{KL}}(p||q)$

Input: Binary time series data set S, D_0; **Output:** D_{KL}.
For i=1 to $\left\lceil \frac{|S|}{2} \right\rceil$ **do**
  Partition S into $\left\lceil \frac{|S|}{i} \right\rceil$ segments;
  Compute $p_k = \frac{supp(x|P_k)}{supp(x)}$, $k = 1, \dots, \left\lceil \frac{|S|}{i} \right\rceil$;
  $D_{KL}(p||q)_i = \sum_k p_k \log \left(p_k * \left\lceil \frac{|S|}{i} \right\rceil \right)$;
  If $D_{KL}(p||q)_i \le D_0$ **then** $D_{KL} = \cup \{D_{KL}(p||q)_i\}$;
End for
Return D_{KL}.

4. EXPERIMENTAL RESULTS

In this section, we present a case study to show the performance of the proposed method in finding periodicity of event in binary data series.

4.1 An example

Given two binary sequences BS_{x1}={10000 10000 10000 10000} and BS_{x2}={10000 01000 00100 00010}. The calculate results of $D_{\mathrm{KL}}(p||q)$ for each partition $\pi(n)_{2\leq n\leq 10}$ are shown in Figure 3. It is easy to know the feasible partition for symbol x_1 and x_2 from the above figure 3 are n^*_{x1} ={5, 10} and n^*_{x2} ={5, 6, 10} with threshold D_0=0.01.

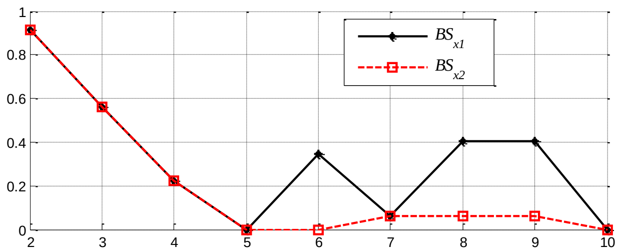


Figure 3. $D_{KL}(p||q)_n$ values for event x_1 and x_2.

4.2 Experimental setup

Two real-world data sets have been used in the experiments.

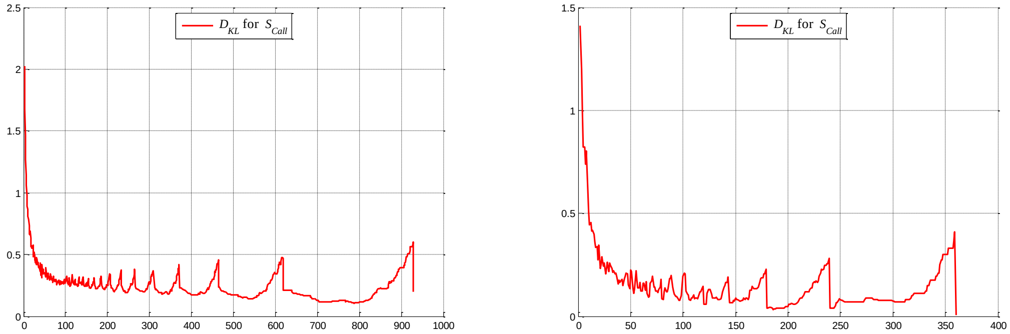
The first, *Amazon access samples data set* (AASDS), was downloaded from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/) which was created and donated by Amazon Corp in 2011 and has been cited over 1000 times. AASDS contains 30000 users' access history from 2005.8 to 2010.8. Actually, the time attribution is most important for studying the accessing periodicity of Amazon users, thus we generated the time data series S_{Amazon} mainly according to the attribution of REQUEST_DATE. The data processing is as following: user (ID#33400) with 716064 actions records , i.e., records of event {*x*=*accessing Amazon.com*}, is chose in the experiment; the first action time of user #33400 recorded in AASDS as the start of e_1 and the last action time as the end of the series $e_{|S|}$; all these actions are counted by 24-hours-day, for example, if the user #33400 had accessed Amazon.com more than 0 times in Sep. 2, 2005, then we marked *e* at the *position of day* Sep. 2, 2005 as "1" in S_{Amazon}, otherwise, "0" is marked.

Another data set is the *Lover's Call Data Set* (LCDS) collected from two lover's communications from Nov.1, 2011 to Dec. 1, 2011by cell phone. We generated data series S_{Call} as follows: the first hour of Nov.1, 2011is set as e_1 and the last hour of Nov.30, 2011 is set as $e_{|S|}$; if the two had called each in each hour, we marked *e* at the *position of this hour* as "1" in S_{Call}, or "0" is marked.

Table 1. Descriptions of the two data sets

Data set	Domain	#Users	#Records	#Recording Time	#|S|
AASDS	Web site accessing	1 selected: ID#33400	716064	2629440	1857 days
LCDS	Communication	1 selected	108	43200	720 hours

4.3 Experimental results



(a) Calculate $D_{KL}(p||q)_n$ for accessing Amazon.com (b) Calculate $D_{KL}(p||q)_n$ for Lover's call

Figure 4. $D_{KL}(p||q)_n$ values for user accessing Amazon.com and the Lovers' call.

The experiment results in Figure 4 indicates that the local minimum $D_{\mathrm{KL}}(p||q)_n$ goes down (more feasible) along with increasing partition n.

All the values of $D_{\mathrm{KL}}(p||q)$ for potential periodic partition, i.e., $\pi(n)_{2\leq n\leq|S|/2}$ will be calculated by the algorithm 1, so the series of $\{D_{\mathrm{KL}}(p||q)_n\}_{2\leq n\leq|S|/2}$ will also show some periodic patterns since x appears periodically in S. From this angle of view point, we can make a prejudgment for the periodicity of each event from the value series of $\{D_{\mathrm{KL}}(p||q)_n\}_{2\leq n\leq|S|/2}$.

Known from the results of figure 4, the events series of user #33400 for accessing Amazon.com are very regular and the period is about n*=43days. On the contrary, the lover's phone calling events shows weak periodicity, which means the two lovers called each other randomly, and the calling events ware not uniform distributed.

4.4 Penalty factor for π(n)-partition

An important issue for the proposed $\pi(n)$-partition method is that: a partition with bigger n will make it more easier to satisfy the appearance condition of x in each segment, i.e., $supp(x|P_i)>0$, $i=1,\ldots,\left\lceil\frac{|S|}{n}\right\rceil$. That is to say, the partitioned period with bigger n will gain more partition feasibility.

On the other hand, the rules with big periodicity may helpless in real application decision making, for example, assume that we know a periodic rule "*sold laptop every week*" about an e-commerce web site, so, the periodic rule "*sold laptop every month*" is also holds true, but the latter is helpless for marketing decision making. To meet this challenge, we introduce a ***penalty factor*** $log(n)$ to balance this bias and use the modified value of $log(n)*D_{\mathrm{KL}}(p||q)_n$ to measure the partition feasibility of $\pi(n)$ for the periodicity of periodic event x. The red lines in figure 6 show the new D_{KL} values with penalty factor.

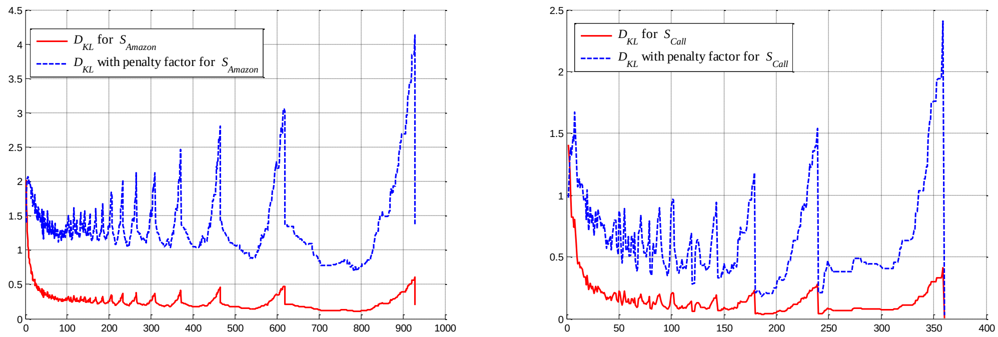


Figure 5. $D_{KL}(p||q)_n$ values with penalty factor for the events of user accessing Amazon.com and the Lovers' call.

Figure 5 indicates that the penalty factor can enlarge the difference between $D_{\mathrm{KL}}(p||q)_n$ and $D_{\mathrm{KL}}(p||q)_{n+1}$, especially, under circumstances that $\pi(n)$ is locally "good" periodic partition, the penalty factor will makes the values of $D_{\mathrm{KL}}(p||q)_n$ more significant to help us find the exact "good" periodic partition.

5. CONCLUSIONS

In this paper, we investigated the problem of discovering periodicity of a binary data series and a cross entropy based new method is proposed. First, a series of rational partition methods for binary data series S are introduced, which can divide S into different segments (partition). Then, we use cross entropy to calculate the partition feasibility, which could be a good measure for the feasible periodicity of event. Finally, according to the generated partition feasibility, we proposed a periodicity evaluation method to obtain the feasible periodicity of event. The results of calculation example show that the method can be used to explore feasible event periodicity in binary data series.

ACKNOWLEDGEMENT

This research was supported by the Specialized Research Fund for the Doctoral Program of Higher Education (20100185120024), the National Natural Science Foundation of China (71101018/71102055) and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Mohamed G. Elfeky, Walid G. Aref, and Ahmed K. Elmagarmid. Periodicity detection in time series databases. IEEE Transaction on Knowledge and Data Engineering, 17(7):875– 887, 2005.

[2] Zhenhui Li, Bolin Ding, Jiawei Han, and Roland Kaysand Peter Nye. Mining periodic behav- iors for moving objects. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1099–1108, 2010.

[3] Faraz Rasheed, Mohammed Alshalalfa, and Reda Alhajj. Efficient periodicity mining in time series databases using suffix trees. IEEE Transaction on Knowledge and Data Engineering, 23(1):79–94, 2011.

[4] Jiawei Han, Guozhu Dong, and Yiwen Yin. Efficient mining of partial periodic patterns in time series database. In Proc. Int. Conf. on Data Engineering, pages 106–115, 1999.

[5] Huiping Cao, David W. Cheung, and Nikos Mamoulis. Discovering partial periodic patterns in discrete data sequences. In In Advances in Knowledge Discovery and Data Mining, 8th Pacific-Asia Conference, PAKDD, pages 653–658, 2004.

[6] Zhen He, X. Sean Wang, Byung Suk Lee, and Alan C. H. Ling. Mining partial periodic correlations in time series.

Knowledge and Information Systems, 15:31–54, 2008.

[7] Jiong Yang, Wei Wang, and Philip S. Yu. Mining asynchronous periodic patterns in time series data. IEEE Transactions on Knowledge and Data Engineering, 15, No.3:613–628, 2003.

[8] Jiong Yang, Wei Wang, and Philip S. Yu. Mining surprising periodic patterns. Data Mining and Knowledge Discovery, 9:189–216, 2004.

[9] Komate Amphawan, Philippe Lenca, , and Athasit Surarerks. Mining top-k periodic-frequent pattern from transactional databases without support threshold. In The 3rd International Conference on Advances in Information Technology, pages 18–29, 2009.

[10] R.Uday Kiran and P.Krishna Reddy. Mining periodic-frequent patterns using multiple mini- mum supports. In International Conference on Management of Data, 2009.

[11] Johannes Assfalg, Thomas Bernecker, Hans-Peter Kriegel, Peer Kr ̈oger, and Matthias Renz. Periodic pattern analysis in time series databases. In Proceedings of the 14th International Conference on Database Systems for Advanced Applications, pages 354–368, 2009.

[12] Michail Vlachos, Philip S. Yu, and Vittorio Castelli. On periodicity detection and structural periodic similarity. In SDM'05, pages 449–460, 2005.

[13] S. Parthasarathy, S. Mehta, and S. Srinivasan. Robust periodicity detection algorithms. In Proceedings of the 15th ACM international conference on Information and knowledge man- agement, pages 874–875, 2006.

[14] T. Cover and J. Thomas. The elements of information theory. Plenum Press, New York, 1991.

[15] S. Kullback and R. A. Leibler. On information and sufficienvy. The Annals of Mathematical St atistics, 22:79–86, 1951.