ATLANTIS PRESS
Advances in Engineering Research (AER), volume 131
3rd Annual International Conference on Electronics, Electrical Engineering and Information Science (EEEIS 2017)

Pruning and Summarizing the Discovered Time Series Association Rules from Mechanical Sensor Data

Qing YANG[1,a,*], Shao-Yu WANG[1,b], Ting-Ting ZHANG[2,c]

[1]School of Computer Science and Technology, Donghua University, China

[2]Department of Information Technology and Media, Mid Sweden University, Sweden

[a]yqij2929@163.com, [b]sywang@dhu.edu.cn, [c]tingting.zhang@miun.se

[*]Corresponding author

Keywords: Sensor Time Series, Association Rules, Rules Pruning, Rules Summarizing, BIGBAR.

Abstract. Sensors are widely used in all aspects of our daily life including factories, hospitals and even our homes. Discovering time series association rules from sensor data can reveal the potential relationship between different sensors which can be used in many applications. However, the time series association rule mining algorithms usually produce rules much more than expected. It's hardly to understand, present or make use of the rules. So we need to prune and summarize the huge amount of rules. In this paper, a two-step pruning method is proposed to reduce both the number and redundancy in the large set of time series rules. Besides, we put forward the BIGBAR summarizing method to summarize the rules and present the results intuitively.

Introduction

Rule discovery is one of the central tasks of data mining [1]. Association rule mining has a capability to find hidden correlations among different items within a data set [2]. Existing researches have proposed different algorithms for mining association rules from time series data. However, the problem is the number of discovered rules are too many and the huge amount of rules may include many redundant rules. We can hardly use those rules directly or present the huge amount of rules to users.

In this paper, we propose two methods to analyze a large dataset of discovered time series association rules and give a summary of the rules. Firstly, a pruning method of redundant rules has been applied to cut down the redundant rules and then we introduce BIGBAR, a bipartite graph based association rules summarizing method to summarize the rest of rules and find the interesting rules.

The rest of the paper are organized as follows. We introduce the state-of-art methods for associate rules pruning and summarizing in related work. In the method section, we explain the methods and algorithms used in this paper. After that, we show the experiments and results. Finally, we summarize our work and explain the future work.

Related Work

Pruning methods can be used to reduce the number of rules and eliminate insignificant rules. Interestingness measure is an important technique for pruning methods. H.Toivonen et al. [3] use confidence as interestingness measure and Bing Liu et al. [4] use the correlation by testing the chi-square between rules. Szymon Jaroszewicz et al. [5] introduced the maximum entropy principle to pruning rules. In, S.Kannan et al. [2] give a detailed summary of more than 40 interestingness measures. In addition, the researcher or user can define the interesting or redundant rules by themselves [6]. Another technique is called close item set or rule cover. These works gave a subset of rules that can cover all of the database transactions or important information [7].

Usually, there are still many rules after pruning. And we need to summarize the remaining rules and extract useful rules from them. It's commonly to use clustering methods [8] to group these rules

and give some representative rules for each cluster. Besides, in [9], the paper introduced a rule template method trying to conclude the templates for different types of rules. It's useful to present the general idea of rules.

Methods

Pruning Redundant Rules

In most cases, number of redundant rules is significantly larger than that of essential rules [10]. It's necessary to prune redundant rules before we use the rules or present them to users.

When we group time series association rules with the same left item or with the same right item, there are a lot of rules in the same group which is confusing when we want to visualize rules or use them for prediction. Our proposed pruning method is based on the two cases.

Let's start from the first case: pruning rules in the group of the same left items. For example, there are two rules R_1 *and* R_2 mined from time series A and B having the same left item: $R_1: p \rightarrow b \ \ [\text{confidence} = c_1]$, $R_2: p \rightarrow cb \ \ [\text{confidence} = c_2]$.

According to the definition of confidence in, we can get:

$$c_1 = \frac{\text{support}(p \rightarrow b)}{\text{support}(p)}, \tag{1}$$

$$c_2 = \frac{\text{support}(p \rightarrow cb)}{\text{support}(p)} \ . \tag{2}$$

where $\text{support}(p \rightarrow b)$ means 'p' happened in time series A and 'b' happened in time series B and $\text{support}(p \rightarrow cb)$ means 'p' happened in time series A and 'cb' happened in time series B within a period time. It's easy to find that $\text{support}(p \rightarrow cb) \leq \text{support}(p \rightarrow b)$. So c_1 is greater than or equals to c_2. If c_1 equals to c_2, we will define R_1 is a redundant rule. For the same pattern 'p' in time series A, R_1 indicates that there will be a 'b' in time series B with confidence of $c_1$but R_2 tells us there will be a 'cb' with the same confidence which gives us more information. We choose to keep the rule which provides more information.

Another case is pruning rules in the group with the same right item. For example, the two rules R_1 and R_2 are mined from time series A and B: $R_1: ab \rightarrow t \ \ [\text{confidence} = c_1]$, $R_2: a \rightarrow t \ \ [\text{confidence} = c_2]$.

According to the definition of confidence, we can get:

$$c_1 = \frac{\text{support}(ab \rightarrow t)}{\text{support}(ab)}, \tag{3}$$

$$c_2 = \frac{\text{support}(a \rightarrow t)}{\text{support}(a)} = \frac{\text{support}(a* \rightarrow t)}{\text{support}(a*)}, \tag{4}$$

where '*' represents any time series data. $a* \rightarrow t$ means 'a' follows any data in time series A leads to 't' happening in time series B which is equal to $a \rightarrow t$. This is different from general association rules because 'a' and 't' happened in different time series.

If c_1 equals to c_2, we regard R_1 as a redundant rule because if there is a 'a' followed by * (any data) in time series A, with the confidence of c_1 we can know there will be a 't' in time series B, but if there is a 'a' followed by 'b' in time series A, the same confidence we can know there will be a 't' in time series B. 'b' in R_1 is redundant because it don't carry more information.

The formalized definitions of the two cases are given below:

Definition 1: For rules R_1, R_2 with the same right item, if $R_1.\text{leftitem}$ is a substring of $R_2.\text{leftitem}$ and $R_1.\text{confidence} \geq R_2.\text{confidence}$, R_2 is a redundant rule.

Definition 2: For rules R_1, R_2 with the same left item, if $R_1.\text{rightitem}$ is a substring of $R_2.\text{rightitem}$ and $R_1.\text{confidence} \leq R_2.\text{confidence}$, R_1 is a redundant rule.

BIGBAR Summarizing Method

Pruning is one way to cut down redundant rules. However, there is no guarantee that the result of pruning can be presented to users because the number of rules could still be very large and hard to understand. What needs to be done next is to analyze and summarize the rules and extract useful information.

In this paper, we introduce a new method to find the interesting clusters of rules and then extract interesting rules within each clusters. This method is bipartite graph based association rule (BIGBAR) summarizing method which presents the association rules in a bipartite graph.

Bipartite graph has two independent sets of nodes and a set of edges linked between the two sets of nodes as showed in Fig. 1.We denote one set of nodes as the clusters of left items and the other set as the right items. The nodes in the same set have no links. One edge between two nodes denotes there is at least one rule whose left item is in one node, and right item is in the other node. We will record the average confidence and number of rules on the edge. After we finish this bipartite graph, we can easily see how different clusters of rules are distributed according to average confidence and number of rules on each edges.

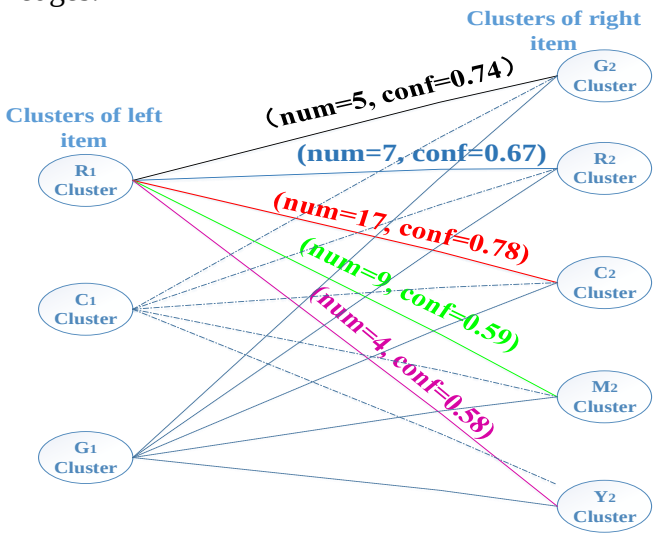

Fig. 1. Bipartite Graph of Rule Item Clustes

Before we draw a bipartite graph, the first thing is to cluster the left items and right items. We use hierarchical clustering for this purpose: as it provide us a simple and practical way to capture the similarity structure of the items. It combines two closest nodes as one cluster and the new cluster is considered as a new node. This process is done iteratively until there is one cluster of all the nodes. Detailed information of hierarchical clustering is introduced in [8]. The core of the algorithm is the definition of the distance between two items.

Definition 3: The item distance between two items is defined as follows:

$$\mathrm{dis}(\mathrm{item}_1, \mathrm{item}_2) = 1 - \frac{\mathrm{len}(\mathrm{lcs}(\mathrm{item1}, \mathrm{item2}))}{\mathrm{len}(\mathrm{item1}) + \mathrm{len}(\mathrm{item2})}, \quad (5)$$

where dis() is the distance of items, lcs() is longest common subsequence and len() is the length of sequence. We use the longest common subsequences to describe the similarity of the two items. We consider the factor of the length of the two items. Besides, the distance value should be smaller if they are closer.

The second step is to draw the bipartite graph. Algorithm 1 summarizes the process.

Algorithm 1 BIGBAR

Input: ItemClusterlist: leftclst, rightclst

Rulelist: rules

```
Output: a bipartite graph
for rule in rules
    if  rule.left in leftclst [i] and rule.right in rightclst [j]
        if  edge linked with leftcluts [i] and rightclst [j]
            tempconf = edge.conf * edge.num
            edge.num = edge.num + 1
            edge.conf = (tempconf+rule.conf) / edge.num
        else
            draw an edge from leftclst [i] to rightclst [j]
            edge.num = 1
            edge.conf = rule.conf
        end if
    end if
```

When the graph is built, we can find the interesting rules from it. Before we explain how to find out interesting rules, we need find the interesting rule clusters first.

Definition 4: A rule cluster is an abstract rule whose left item is a cluster of left items and right item is a cluster of right items. A rule cluster's confidence is the average confidence of all the rules in this cluster.

The last thing is to find interesting rule clusters and choose interesting rules in each clusters. There are three measures that can be considered to find interesting rule clusters: confidence, number of rules and both. For selecting representative rules, we can simply choose rules with higher confidence in each rule clusters.

Experiment and Results

Our experiment data is a large set of time series association rules from [11]. There are total 198,405 association rules mined from time series data from 23 sensors deployed on different parts of the industrial machine including motors, coolers, pumps, drives and tanks.

Firstly, we preprocess the rules and prune the redundant rules from them. After pruning, we summarize the remaining rules using BIGBAR algorithm and extract the interesting rules in each rule clusters. We show the results of 5 different rule sets in Table1.

Each line in Table1 show the result of one rule set. The first item is the name of time series pairs of the rules. For example, the first line show the result of rules mined from P2 time series and T2 time series. We pruned 226 rules from total 332 rules. Using hierarchical clustering on left and right items, we can get 3 and 5 clusters respectively. Finally, we extract 45 interesting rules with higher confidence after BIGBAR summarizing method. The pruning rate and reducing rate (including pruning and summarizing) are 68% and 86% respectively. Besides the results of the above 5 rule sets. The last line is the final results of all the rule sets. We extract 21825 interesting rules from 198405 rules and the reducing rate is nearly 89%.

Table 1. Results of pruning and summarizing.

Time series pair	Total No. of rules	No. of pruning rules	Rules pruning rate	No. of rule items clusters	No. of interesting rules	Rules reducing rate
[P2->T2]	[332]	[226]	[68%]	[Left=3, right=5]	[45]	[86%]
[C2->C1]	[352]	[236]	[67%]	[Left=3, right=6]	[54]	[84%]
[D2->C1]	[520]	[493]	[95%]	[Left=2, right=3]	[18]	[97%]
[C1->D1]	[100]	[82]	[82%]	[Left=2, right=2]	[12]	[88%]
[T1->P1]	[1043]	[490]	[47%]	[Left=6, right=6]	[108]	[90%]
[All]	[198,405]	[130,947]	[66%]	—	[21,825]	[89%]

Summary

Time series association rules mining leads us into a new world of association rules in big data field. However we are still facing many challenges. One of the biggest challenges is to understand the huge amount of discovered time series association rules.

In this paper, we introduced a two-step way to interpret the huge amount of rules to be understandable. The first pruning step is to find those rules that can represent other rules or carry much information than other rules. The number of rules can be reduced a lot.

The second step is summarizing the remaining rules using bipartite graph based association rules summarizing method which can show the distribution of the rule clusters and summarize the interesting rules.

Time series association rules can be mined between multiple time series. It's more complex to prune and summarize the multi-item rules. This is a problem needs to be solved in the future.

References

[1] Liu, Bing, Yiming Ma, and Ronnie Lee. "Analyzing the interestingness of association rules from the time series dimension." Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on. IEEE, 2001.

[2] Liu, Bing, Wynne Hsu, and Yiming Ma. "Pruning and summarizing the discovered associations." Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1999.

[3] Toivonen, Hannu, et al. "Pruning and grouping discovered association rules." (1995).

[4] Liu, Bing, Wynne Hsu, and Yiming Ma. "Pruning and summarizing the discovered associations." Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1999.

[5] Kannan, S., and R. Bhaskaran. "Association rule pruning based on interestingness measures with clustering." arXiv preprint arXiv:0912.1822 (2009).

[6] Ashwini Batbarai1, Devishree Naidu2. "Approach for Rule Pruning in Association Rule Mining for Removing Redundancy" International Journal of Innovative Research in Computer and Communication Engineering.Vol. 2, Issue 5, May 2014.

[7] Cristofor, Laurentiu, and Dan Simovici. "Generating an informative cover for association rules." Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on. IEEE, 2002.

[8] Jorge, Alipio. "Hierarchical clustering for thematic browsing and summarization of large sets of association rules." Proceedings of the 2004 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2004.

[9] Klemettinen, Mika, et al. "Finding interesting rules from large sets of discovered association rules." Proceedings of the third international conference on Information and knowledge management. ACM, 1994.

[10] Ashrafi, Mafruz Zaman, David Taniar, and Kate Smith. "A new approach of eliminating redundant association rules." International Conference on Database and Expert Systems Applications. Springer Berlin Heidelberg, 2004.

[11] Xue, Ruidong, et al. "Sensor time series association rule discovery based on modified discretization method." Computer Communication and the Internet (ICCCI), 2016 IEEE International Conference on. IEEE, 2016.