Research on Alarm Association Mechanism of Information System Based on FP-growth Algorithm

Yunrui He*, Yiying-yan, Yijie-dang, Yingwen-ren, Xinyi-le

Dispatch and Control Center, State Grid Information & Telecommunication Co., Ltd., Beijing, 100075, P.R. China

hyr52045@163.com, mango27@126.com, dangyijie@126.com, 18810669263@163.com, lexinyi97@163.com
*heyunrui@sgcc.com.cn

Abstract. With the continuous informatization advancement of State Grid Corporation, the scale of information networks has become increasingly complicated, therefore effective alarm filtering and correlation has become the focus of the operation and maintenance of information system. In this paper, FP-growth algorithm is used to mine the association rules of alarm data in information system. Aiming at the problems of short message gateway pressure, core alarm delay and even omission caused by large-scale network alarm, the calculation method of the confidence is optimized, which solves the problem of confidence calculation distortion caused by frequent alarm term. The experimental results show that the optimized FG-growth algorithm combines the alarm information effectively and has some reference significance for the actual operation and maintenance work.

1. Introduction

The information system of State Grid Corporation is huge. With the development of informatization, the number of alarms is increasing rapidly. Quickly mining the key information from such a large number of alarms is essential for the repairing the failures. Traditional association algorithms are mostly based on Apriori[1] algorithm, which is low in efficiency, as it needs to scan the database multiple times. To make up the flaws of Apriori algorithm, the FP-growth[2] algorithm mines frequent item-sets based on frequent pattern-trees. It only scans database twice without the need of candidate set, which has a small amount of computation and saves resources.

In order to optimize the FP-growth algorithm, some researchers weight the factors that influence the frequency signal of grid alarms with the combination of the weight model[3]. They highlight the factors with great weights when using the FP-growth algorithm to mine frequent patterns. Besides, to study changes in gene expression, an optimization method that uses support, credibility and improvement as the three indicators of the dynamic threshold is applied to the FP-growth algorithm[4]. Another optimization of the FP-growth algorithm to improve the process of traversing frequent item-sets, combining with the characteristics of feature words in the failure information of the CRHX EMU traction system[5].

The methods above improve the traversal process of the FP-growth algorithm given the characteristics of different application scenarios. However, there are few studies on the use of the FP-growth algorithm for alarm association and mining in information systems. The paper proposes an optimization method for FP-growth algorithm that introduces the sliding time window algorithm in the
data preprocessing process, which improves the efficiency when mining the association rules of the alarm data collected by the information network management system. Additionally, for the problem of confidence distortion when the network failures break out, the paper suggests an optimization method for the confidence algorithm in the FP-growth algorithm.

2. The Preprocessing of Alarm Information

Due to the large scale and complex structure of the network, the alarms caused by the same failure have a sequential relationship in time\[^6\]. The paper introduces the sliding time window algorithm. It starts from the time the monitoring started and selects the alarm information in the time window as an alarm transaction item, then studies the relevance of the alarm transaction items. The specific sliding time window algorithm is as follows.

Define two parameters, window width and sliding step length. The former is used to define the maximum time interval between two alarms related to each other, and the latter is used to prevent the successive alarms that occurred in a short time from being divided into different transactions due to the division of time windows.

The alarm sequence \(S = \{s, T_s, T_e\}\) consists of multiple ordered alarms in the alarm set \(E\), and is sorted according to the occurrence time with the time period \([T_s, T_e]\). \(S_v = \{s_v, t_s, t_e\}\) is a sub-item of the sequence \(S\), where \(t_s \geq T_s, t_e \leq T_e\) and \(t_e - t_s\) are the widths of the time window as \(W\). \(s_v\) is a sub-alarm sequence of the alarm sequence \(s\), which is composed of multiple ordered alarm vectors \((A, t)\), where \(A\) is the alarm monitoring item and \(t\) is the alarm occurrence time, \(A \in E\). If the sliding time window step is \(S\), the time window after sliding begins at \(t_e + S\) and ends at \(t_e + S\), as shown in Figure 1. It shows the alarm sequence \{A,C,......,F,E\} with the \(W\) being set to 5 and \(S\) to 2. It can be seen that and the alarm transaction in a certain time window is \{A,D,C,A,E\}.

![Fig.1 Sliding time window](image)

The sliding time window algorithm first sorts the alarm data according to the start time of the alarm occurrence. If the alarm occurrence time is within the time window \([t_{s1}, t_{e1}]\), record the alarm \([t_{s1}, t_{e1}]\) in the alarm transaction item-set corresponding to the window. Otherwise, slide the time window so that the alarm occurrence time is within the new time window \([t_{s1} + S_s, t_{e1} + S_s]\), where \(S_s\) is the sliding step.

3. FP-growth Algorithm and Optimization

3.1 Classical FP-growth Algorithm

The FP-growth algorithm applies the frequent pattern growth algorithm to compress the frequent item-sets in the transaction database. The main steps\[^7\] are as follows.

(1) Scan the transaction database to get the frequent item-set \(L\). Meanwhile, sort the items in \(L\) in descending order of support, building a FP tree with root node being null. After that, scan the transaction database again, and sort transaction data in the order of \(L\), deleting infrequent items. Then
create a branch path for each transaction, and now nodes on each branch path are the frequent item-sets in the transaction that has been sorted. For each transaction branch, if can, share the prefix path, and record the support of the transaction in the node. For the convenience of traversing the FP tree, it is necessary to create an item header table sorted by support, each row of which consists of a frequent item and identification pointer pointing to the position of the frequent item appeared for the first time.

(2) Dig out all frequent item-sets in the FP tree recursively. Traverse the item header table in reverse order, creating a conditional pattern base for each node. The conditional pattern base contains the frequent item and a set of prefix paths corresponding to it. After that, form a new frequent pattern tree, which is also called a conditional pattern tree, from the conditional pattern bases on the basis of the principle of pattern tree construction. Finally, mine the frequent pattern tree to obtain all frequent item-sets.

3.2 FP-growth Algorithm’s Optimization

In Section 3.1, items in frequent item-sets L that do not meet the support requirement are infrequent items and are needed to be deleted, to make sure that the frequent item-sets are obtained based on confidence mining and alarm association rules. Following the definition in Section 2, assume that there are two non-empty item-sets of alarm rules \( M \) and \( N \), \( M \subset S \), \( N \subset S \) and \( M \cap N \neq \emptyset \). The association rule \( M \Rightarrow N \) is a strong one only if the support and the confidence of it meet the minimum respectively\(^8\).

The support \( s \) of the association rule \( M \Rightarrow N \) is the percentage of transactions in \( E \) that contains \( M \cup N \), as follows.

\[
s = P(M \cup N) \tag{1}
\]

The confidence \( c \) of the association rule \( M \Rightarrow N \) is the percentage of transactions in \( E \) that contain \( M \) and \( N \) at the same time, as follows.

\[
c = P(N | M) = \frac{P(M \cup N)}{P(M)} = \frac{P(M) + P(N) - P(M \cap N)}{P(M)} \tag{2}
\]

Because the rules are generated by frequent item-sets, each rule automatically meets the requirement of the minimum support. \( M \Rightarrow N \) is a strong rule if the confidence \( c \) of the association rule is greater than or equal to the threshold \( \text{min}_\text{conf} \) of the minimum confidence.

By analyzing the expression of the confidence \( c \), it can be found that if the support of the alarm rule item-set \( N \) is very high while it of \( M \) is very low, the confidence will increase and the association rule is considered to be established. In fact, the probability of occurrence of the association rule \( M \Rightarrow N \) is very low, and there is no correlation between them. In this case, the confidence becomes distorted. In the actual operation and maintenance environment, network failures can cause a large scale of network alarms. The alarms caused by core network failure last longer, which leads to the corresponding support being higher. In order to avoid confidence distortion influenced by frequent alarms as the denominator, the paper adjusts the calculation of the confidence. The new confidence \( c_n \) is calculated as follows:

\[
c_n = \frac{P(M \cap N)}{P(M \cup N)} = \frac{P(M \cap N)}{P(M) + P(N) - P(M \cap N)} \tag{3}
\]

By analyzing the expression of computing the new confidence, it can be found that the new formula uses the set of the two as the denominator, which can effectively avoid the confidence distortion caused by either one of them being too large or too small. In actual operation and maintenance environment, corresponding thresholds can be set according to the support of the alarm rule item-set, and the calculation method of the confidence can be switched dynamically.
4. The Experiments
The experiment is based on the alarm data collected by the information network management system. Firstly, preprocess the alarm data in Section 2 to get the data in Table 1.

| W number | Time window (W) | Alarm monitoring item | W number | Time window (W) | Alarm monitoring item |
|----------|-----------------|-----------------------|----------|-----------------|-----------------------|
| 1        | (30, 0)         | A, B, C               | 6        | (30, 150)       | C, D                  |
| 2        | (30, 30)        | D, E                  | 7        | (30, 180)       | A                     |
| 3        | (30, 60)        | B                      | 8        | (30, 210)       | D                     |
| 4        | (30, 90)        | A                      | 9        | (30, 240)       | C, A                  |
| 5        | (30, 120)       | D, E                   | 10       | (30, 270)       | D, E                  |

Take the time window as a time unit, for example, as for A→D, W1→W2, W4→W5, W7→W8 and W9→W10 are four adjacent time windows that support the association relationship of A→D. Thus, the support of A→D is 4. Table 2 presents that the support of each association rule with time window calculated according to this rule.

| A     | B     | C     | D     | E     |
|-------|-------|-------|-------|-------|
| A     | -     | 1     | 1     | 4     | 3     |
| B     | 2     | -     | 1     | 1     | 1     |
| C     | 3     | 1     | -     | 3     | 2     |
| D     | 1     | 1     | 2     | -     | 3     |
| E     | 0     | 1     | 1     | 3     | -     |

Then, conduct association relationship mining. Firstly, traverse Table 1 to get the support of each rule. Then divide the number of occurrences of each rule by it of all the rules to get the support of A, B, C, D, and E, which are 0.25, 0.125, 3/16, 0.25 and 3/16, respectively. After that, the confidence shown in Table 3 can be computed according to the new confidence formula.

| A     | B     | C     | D     | E     |
|-------|-------|-------|-------|-------|
| A     | -     | 0.16  | 1     | 0.75  |
| B     | 0.5   | -     | 0.25  | 0.2   | 0.25  |
| C     | 0.16  | 0.25  | -     | 0.75  | 0.33  |
| D     | 0.14  | 0.2   | 0.6   | -     | 0.75  |
| E     | 0     | 0.25  | 0.2   | 0.75  | -     |

For each non-empty subset of $S$, the rule $A \Rightarrow B$ exists if the confidence $c_n$ of it is greater than or equal to $\text{min\_conf}$.

Finally, perform data testing according to the association relationship obtained by training. Converge an alarm item with an association relationship into an alarm message and compute the convergence rate.

5. Results and Analysis
The paper uses the original alarm data collected by the information network management system within half a year as the test data, with the parallel computation of FP-growth implemented by Python, compute the convergence of alarms based on Hadoop. The results are shown in Table 4.
Table 4. The convergence of alarms

| Minimum confidence threshold | Minimum support threshold | Number of combine alarms | Initial number of alarms | Combination rate |
|------------------------------|---------------------------|--------------------------|--------------------------|-----------------|
| 0.25                         | 6                         | 1077501                  | 8450989                  | 87.25%          |

The original alarms are effectively merged after the alarms being associated through the FP-growth algorithm. Furthermore, it converges alarms, and has a positive effect on locating core alarms that cause failures. Particularly, selecting minimum confidence threshold is important, which needs to be combined with the actual situation, since too large threshold may cause associated alarms being ignored while too small threshold may merge the unrelated alarms and leads to inaccuracy.

6. Conclusion

The paper discusses the application of the FP-growth algorithm in alarm association of the information system. Moreover, in order to reduce the confidence distortion caused by frequent term being the denominator, it optimizes the confidence algorithm. The optimized FP-growth algorithm can be used in the scenarios of a large-scale of network alarms caused by network failures, reducing the number of redundant alarms, as well as reducing the pressure on the SMS gateway used for alarm notification. In addition, it effectively helps locating the root cause of the alarm information in the massive alarms, providing the reference for the fast restoration of the system. In the future, the method of traversing to generate frequent item-sets can be optimized further, combining with the network topology of the information system network topology and the alarm association contained in the system architecture.

Acknowledgments

This work was supported by State Grid Information &Telecommunication Branch for the project environment. The authors would like to thank our leaders and collaborators.

References

[1] Agrawal R, Imielinski T, Swami A N. Mining Association Rules Between Sets of Items in Large Databases, SIGMOD Conference[M]// Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. 1993:207--216.

[2] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation[J]. Acm Sigmod Record, 2000, 29(2):1-12.

[3] Xiang. C, Xianghua. J, Feng. Z et al. Intelligent analysis of grid alarms and research and application of comprehensive displays [J]. Electric Engineering, 2018(5).

[4] Mallik S, Bhadra T, Mukherji A. DTFP-Growth: Dynamic Threshold-Based FP-Growth Rule Mining Algorithm Through Integrating Gene Expression, Methylation, and Protein-Protein Interaction Profiles[J]. IEEE Transactions on Nanobioscience, 2018, 17(2):117-125

[5] Yanhui W, Shuijun W, Man L et al. Research on correlation failure model of CRHX EMU traction system based on the improved FP-growth algorithm[J]. Journal of the China Railway Society. 2016, 38(9):72-80.

[6] Ting B. Research on mining alarm association rules of business support network based on parallel FP-growth algorithm[D]. Nanjing University of Posts and Telecommunication, 2015.

[7] Jie Y, Wenjuan Q. Research based on aprior & FP-growth algorithm[J]. Computer Systems & Application, 2013, 22(5):122-125.

[8] Weijie C, Xiaohui Z, Jianqiu Z et al. Survey of association rule generation. Computer Engineering. 2001, 27(5):31-33.