Alarm Association Rules Mining in Flight Booking System Based on Sliding Time Window Model

1Jianli Ding, 1Chao Zhang, 1Jing Wang and 2Yao Wang
2College of Computer Science and Technology, Civil Aviation University of China, Tianjin, 300300, China
2TravelSky Technology Limited Company, Beijing, 100105, China

Corresponding Author: Jianli Ding, College of Computer Science and Technology, Civil Aviation University of China, Tianjin, 300300, China

ABSTRACT

Alarm association rules mining is an important task in system fault diagnosis and localization. Once the system fails, it will produce a large number of alarm information. By analyzing the characteristics of the booking system alarm data, this study puts forward alarm association rules mining algorithm based on sliding time window model to find the fault source and the correlation between fault factors in a large number of alarm information. The experiments show that the valuable alarm association rules can be acquired from the alarm data accurately and rapidly. These rules can provide support decision for the system maintenance personnel.

Key words: Sliding time window, flight booking system, fault alarm analysis, association rules mining

INTRODUCTION

Flight booking system is an important sub-system of the new generation passenger service information system of the National Civil Aviation Industry. It carries a lot of important aviation business. The system needs to maintain 24 h a day without interruption of continuous operation and it has the safety, reliability, high efficiency and other characteristics. Paralysis of the system or even an application service interrupted may lead to paralysis of the civil aviation business. Therefore, the state and Civil Aviation Authority make high demands for the safe and reliable operation of the civil aviation passenger service information system. Currently, the information of monitoring the whole process of the system and the application software is saved in the log files. However, the common phenomenon in civil aviation industry is the daily operation information of the business system that is recorded in large number of log files which contain very important operation data as well as the causes and consequences of failure events. In particular, there will be a lot of alarm information when the system fails. This situation makes system maintenance personnel that can not accurately find the root of causing alarm information and also makes them more difficult to find the relationship between alarm data, so system maintenance is a difficult and arduous task. So it is essential to scientifically analyze the relationship between alarm data and provide accurate association rules to maintenance personnel. Displaying the root of alarm information, filtering out the alarm information caused by the root of the fault, achieving fault diagnosis, prediction (Zhang and Hou, 2009) and intelligent alarm processing are important to maintain the system security.
TAM AND THE CHARACTERISTICS OF THE ALARM DATA

TAM monitoring system: TravelSky Application System (TAM) is a platform of monitoring all the sub-systems of the aviation passenger service information system. Flight booking system is one of sub-systems being monitored. The monitoring system sets agent on every application sub-system. Whenever the application sub-systems fail, they will produce massive alarm data. Agent endpoints collect the alarm data and send them to system maintenance personnel by a simple analysis. At the same time all the alarm datasets are saved in the log files.

Characteristics of the alarm data: The alarm data of the application sub-systems collected by the TAM has its own unique characteristics. It is stored in the log files and is a kind of unstructured data. Through, alarm data analysis and research, the characteristics of the alarm data are summarized as follows:

- **Massive data:** Due to the rapid development of the civil aviation industry, there are more and more application sub-systems. The relationship between systems has become increasingly complex. So, if a system fails, it may lead to other systems produce alarm data, thus the alarm data is massive
- **Sequent:** The generation of the alarm data is in time order. The root failure always occurs first and then it leads to some subsequent failures. They have a very strong correlation which is an important basis to determine the root fault of the alarm
- **Association:** Because of the association between sub-systems, there must be a valuable correlation between alarm data which is the focus of mining analysis and research
- **Unstructured:** Alarm data is stored in the log file which is a document format. It has no fixed structure and is only a certain standard format. While, the information stored in the log is varied, including the alarm information and other unnecessary information. Mining algorithm can not be applied directly to the original log file, so the alarm data in the log file must be preprocessed into structured data to facilitate mining

DESCRIPTION OF ALARM CORRELATION PROBLEM

Alarm data preprocessing: Alarm data is stored in the log files. The log files stored in the unstructured form make general mining algorithms can not be applied directly to this alarm data, so alarm data preprocessing is essential. Through the data preprocessing, the alarms become structured data and the mining algorithm can process them directly. Data preprocessing is the important and crucial step in the process of data mining. The quality of data preprocessing determines the effectiveness of association rule mining. Alarm correlation analysis model is shown in Fig. 1.

Use regular expressions to extract the alarm data with XML form in the log files and save it into XML formatted text. Then parse the XML text by using Dom4j frame. Extract the alarm information for each of the application sub-system and remove the noise data (alarm data of missing the key fields) and save it as structured data. The alarm data is stored in the database as the follow form:

\[ D = \{Event\_hostname, Event\_name, Event\_id, Source, Info, Time, Level, Agent\_name\} \]
Fig. 1: Alarm correlation analysis model

where, Source indicates the source where the alarm occurs, which is the IP address of the host, where the alarm occurs. Info indicates the alarm information. Time indicates the time when the alarm occurs, Level indicates a failure level.

**Related problem description**

**Alarm sequence:** Alarm sequence $S$ is composed of a plurality of ordered alarms (Xu *et al*., 2007; Li and Li, 2010). It is expressed as:

$$S = \{s, T_s, T_e\}$$

where, $T_s$ is the start time of the alarm sequence, $T_e$ is the end time of the alarm sequence. As shown in Fig. 2, alarm sequence is composed of a number of alarm events $(A, t)$. $A$ indicates alarm event, $t$ represents the time when the alarm occurs.

**Sliding time window and sliding step:** Zhu *et al.* (2007), Hou and Zhang (2008), Li and Chen (2008) and Chen *et al.* (2009) described that for a given alarm sub sequence of alarm sequence $S = \{s, T_s, T_e\}$ can be expressed as:

$$S_w = \{w, t_w, t_3\}$$

where, $w$ indicates the window width, $t_w > T_s, t_w < T_e, w = t_e - t_s$. In the Fig. 2, the alarm events that occur in the first alarm window are $<A, B, C>$. The $d$ indicates the sliding step which is the distance of time window slide backwards from the current.
Alarm association rule: Given an alarm sequence $S$, the alarm situation $\alpha$ is an alarm collection which is composed of alarm sequences. Set window width $w$ and sliding step $d$. Traverse the entire alarm database, if the frequency of the occurrence of the alarm situation $\alpha$ is greater than a given minimum support, then the alarm situation $\alpha$ is frequent:

$$sup(\alpha, S, W) = \frac{\sum_{W \in W(S, W)} |\alpha \subset S_w|}{|W(S, W)|}$$  \hspace{1cm} (1)

Confidence of the alarm situation $\alpha \Rightarrow \beta$ is defined as:

$$conf(\alpha \Rightarrow \beta) = \frac{sup(\beta, S, W)}{sup(\alpha, S, W)}$$  \hspace{1cm} (2)

If the alarm situation $\alpha = A$, $\beta = AB$, $conf(\alpha \Rightarrow \beta) = 90\%$, alarm situation $\alpha$ and $\beta$ are frequent and their confidence are greater than the minimum confidence, then acquire an association rule; if situation $\alpha$ occurs, the probability of occurrence of situation $\beta$ is 90% within one window time.

Alarm partial order: Liu and Li (2003) stated that here are two partial order relations between alarm sequences:

- **Serial**: According to the characteristics of alarm data, the occurrence of alarms between alarm sequences is successive. When an alarm event occurs, it will lead to more alarm events. As shown in Fig. 3a, alarm event A occurs, it will lead to the occurrence of the alarm event B. This partial order relation calls serial relation.
- **Parallel**: Figure 3b shows that alarm situation $\beta$ is composed of alarm event C and D and alarm both these events often occur together. But the order that they appear is uncertain, then the partial order of the event C and D is parallel.

---

422
ALARM ASSOCIATION RULES MINING ALGORITHM

This algorithm is divided into two main parts: Generate alarm associated datasets, Alarm association rule mining. In the part of generating alarm associated datasets, in order to improve the efficiency of mining association rules, use the idea of the effective time window to compresses alarm data.

Generate alarm associated datasets: First, set the time window width and sliding step, then traverse the preprocessed alarm datasets and generate alarm associated datasets. Because there are many alarm fields in the alarm datasets, in order to improve the efficiency of mining algorithm, extract only those fields that are closely related with the alarm association rule mining. Extract four main fields in this study: (1) The IP address of the host where alarm occurs, (2) Alarm information: a description of the kind of the alarm, including the type of the failure and causes, (3) Alarm time and (4) Alarm level. In the processing of generating alarm datasets, alarm events occurring in one window time are regarded as an item data. Each item includes alarm information, alarm time, the IP address of the host and the alarm level.

Because the systems do not produce alarm information all the time, the time difference between the alarm data is relatively large, so in the process of the time window sliding on the dimension of the alarm data, there are some empty time windows, there is no alarm event occurs within these time windows which are called invalid time windows. The number of effective windows on the entire alarm datasets is expressed in Eq. 3:

\[ |W(s, w)| = \left[ 1 + \frac{T_s - T_v}{d} \right] - |W(s, w)| \quad (3) \]

where, \( |W(s, W)| \) is the number of non effective windows. Obviously, in the process of generating frequent items, it needs to scan the entire alarm datasets, so the non effective windows waste a lot of time and take up a lot of memory. In this study, removing all the non effective windows and taking an experiment based on the sliding time window model, the efficiency of the association rules mining is greatly improved. From the Eq. 1, by decrease in the number of time windows, the support is also improved.

It is found that the same alarm event occurs many times at the same time and also occurs in different time within one time window by analyzing the alarm data. To compress the alarm data and avoid this extreme case of an alarm event occurring frequently in one time window and rarely in other time windows that is calculated as frequent item, this study uses a Boolean model. The same alarm event occurring many times within one time window is recorded once.

Alarm association rule mining: This study uses the MC_Apriori (Miao and Wang, 2001; He and Du, 2011; Zhang and Zhang, 2012) algorithm which is an improved Apriori algorithm based on the matrix compression. Using Boolean model also to facilitate the use of Boolean matrix here. In the Boolean matrix, each row represents alarm events occurring within one time window, columns represent all alarm events in the alarm datasets. In the Boolean matrix, 1 represents what alarm event happens and 0 represents that no alarm event happens. At the same time, increase a column sum_r to record the alarm events that occur in every time window and add a row sum_r to record the support of each item. Boolean matrix is expressed as follows:
Here are two important natures:

**Nature 1:** If \( A_k \) is \( k \)-item set which is useless in the next step of generating \((k+1)\)-frequent item sets, so remove it away. That is to say if the \( \text{sum}_c \) is \( k \) in the Boolean matrix, this row can be deleted in the next step of generating \((k+1)\)-item sets.

**Nature 2:** The \( L_k \) is the collection of frequent \( k \)-item sets. If \(|L_k| < k+1\), then the maximum frequent item set in the alarm datasets is \( k \)-item set. Here, \(|L_k|\) is the number of frequent \( k \)-item sets.

Nature 2 can be used as the end condition of loop of searching frequent item sets. Add up the support count and put it on the row \( \text{sum}_r \). Add up the events count and put it on the column \( \text{sum}_c \). Compress the matrix by using the nature 1 and 2.

The advantage of Boolean matrix is that just do “AND” operation when generate \( k \)-item sets, the support count is the number of 1.

Set \( \{I_i, I_j\} \) is defined as the \( D_{ij} \):

\[
D_{ij} = D_i \land D_j = \begin{bmatrix}
    a_{i1} \land a_{j1} \\
    a_{i2} \land a_{j2} \\
    \vdots \\
    a_{in} \land a_{jn}
\end{bmatrix}
\]

(5)

where, “\( \land \)” is the “AND” operation and the support count of the 2-itemsets is expressed as:

\[
\text{support\_count}(I_i, I_j) = \sum_{k=1}^{n} (a_{ik} \land a_{jk})
\]

(6)

\( K \)-itemsets is defined as the Eq. 7:

\[
D_{ik} = D_1 \land D_2 \land \cdots \land D_k = (D_1 \land D_2 \land \cdots \land D_{k-1}) \land D_k
\]

(7)

The support count of the \( k \)-itemsets can be expressed as follows:

\[
\text{support\_count}(I_1, I_2, \cdots, I_k) = \sum_{i=1}^{m} ((a_{i1} \land a_{i2} \land \cdots \land a_{i(k-1)}) \land a_{ik})
\]

(8)

Compress the matrix and mine the association rules until satisfy the nature 2 to exit the loop. The alarm association rules mining algorithm flow chart is shown in Fig. 4.
Fig. 4: Flowchart of the association rules mining algorithm

<table>
<thead>
<tr>
<th>IP</th>
<th>Alarm type</th>
<th>IP</th>
<th>Alarm type</th>
<th>Sup</th>
<th>Conference</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.6.141.16</td>
<td>RESTART</td>
<td>10.6.141.800</td>
<td>ERROR</td>
<td>0.142</td>
<td>0.912</td>
</tr>
<tr>
<td>10.6.141.23</td>
<td>NTFS</td>
<td>10.6.141.23</td>
<td>TPS</td>
<td>0.126</td>
<td>0.905</td>
</tr>
<tr>
<td>10.6.141.23</td>
<td>TPS</td>
<td>10.6.141.23</td>
<td>NTFS</td>
<td>0.126</td>
<td>0.905</td>
</tr>
<tr>
<td>10.6.141.5</td>
<td>AVCACTL</td>
<td>10.6.141.800</td>
<td>ERROR</td>
<td>0.103</td>
<td>0.887</td>
</tr>
<tr>
<td>10.6.141.14</td>
<td>MEM1</td>
<td>10.6.141.170</td>
<td>MEM2</td>
<td>0.113</td>
<td>0.905</td>
</tr>
<tr>
<td>10.6.141.7</td>
<td>SPACE</td>
<td>10.6.141.78</td>
<td>CORE</td>
<td>0.137</td>
<td>0.876</td>
</tr>
<tr>
<td>10.6.141.3/4</td>
<td>ERROR TPS</td>
<td>10.6.141.3/4</td>
<td>NTFS</td>
<td>0.123</td>
<td>0.901</td>
</tr>
</tbody>
</table>

EXPERIMENT AND ANALYSIS OF THE RESULTS

The experiment is done on the Intel processor 2140 machine with 2GB of main memory and the algorithm is written in JAVA. In the experiment set the time window \( w \) is 5 min, the sliding step \( d \) is 2 min, the minimum support is 7% and the minimum confidence is 85%. The total number of the alarm records is 40452 and these alarm records occurred in four weeks. The mined alarm association rules in this experiment are shown in the Table 1.

Set the first rule as an example to explain the meaning of the mining alarm association rules: when the alarm event “RESTART” occurred in the IP address 10.6.141.16, then within the time window \( w = 5 \) min the probability of the alarm event “ERROR” occurs in the IP address 10.6.141.8 is 91.2%.

In order to verify the accuracy of association rules, another experiment is done by using the alarm datasets occurred on December 2013. Through the experiment, the accuracy rate
Table 2: The accuracy of part of rules

<table>
<thead>
<tr>
<th>Alarm type</th>
<th>Then</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTART</td>
<td>ERROR</td>
<td>0.946</td>
</tr>
<tr>
<td>NTFS</td>
<td>TPS</td>
<td>0.899</td>
</tr>
<tr>
<td>TPS</td>
<td>NTFS</td>
<td>0.899</td>
</tr>
<tr>
<td>AVCACTL</td>
<td>ERROR</td>
<td>0.938</td>
</tr>
<tr>
<td>SPACE</td>
<td>CORE</td>
<td>0.872</td>
</tr>
<tr>
<td>ERROR TPS</td>
<td>NTFS</td>
<td>0.865</td>
</tr>
</tbody>
</table>

of part of the association rules is very high. This verifies the validity of the part of association rules. The result is shown as the Table 2.

According to the partial order relations in association rules mentioned above, analyze the meaning and application of the mining alarm association rules. For example, the first alarm association rule satisfies serial order, adding it to the alarm rule sets. This alarm association rule has two meanings; the first, trace the source of the fault and treat the rule antecedent as the fault source; the second, compress the redundant alarm data. Because the fault source produces alarm data, it will lead to another or more systems produce alarm data. If the fault source produces alarm data in the future, these alarm data caused by fault source can be filtered. Let the agent endpoints of the TAM only extract the alarm data from fault source. In Table 1, the second and the third association rules satisfy parallel order relation, when the alarm event of the rule antecedent occurs, then the alarm event can be used to predict the coming alarm data occurring on the other system. The maintenance personnel can take measures to deal with the fault in advance.

CONCLUSION

In this study, the alarm data has its own characteristics and the data preprocessing is essential. Many valuable association rules have been mined by using the sliding time window model according the characteristics of the alarm data. These alarm association rules have been verified in practice and play an important role in the decision support for the maintenance personnel.

ACKNOWLEDGMENTS

This study is supported by the Major Project of Civil Aviation Administration (MHRD20130106) and The Major Project of Aviation Technology (2013).

REFERENCES


