Mining Closed Strong Association Rules by Rule-growth in Resource Effectiveness Matrix

Lihua Zhang[1,2], Miao Wang[2,3,*], Zhengjun Zhai[1], Guoqing Wang[1,2,3]
[1]School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, China, 710072
[2]Science and Technology on Avionics Integration Laboratory, Shanghai, China, 200233
[3]China National Aeronautical Radio Electronics Research Institute, Shanghai, China, 200233
Email: {zhang_lihua, wang_miao,wang_guoqing}@careri.com
[*]Corresponding author

***Abstract*—Association rules mining approach can find the relationship among items. Using association rules mining algorithm to mine resource fault, can reduce the number of wrong alarm resources to be replaced. This paper proposed an efficient association rules mining algorithm: *CSRule*, for mining closed strong association rules based on association rule merging strategies. *CSRule* algorithm adopts several pruning strategies to mine closed strong association rules without storing the candidate set. To improve the mining efficiency, *CSRule* algorithm adopts effective pruning strategies to mine closed strong association rules in real time, instead of secondary mining only through the definition. The experimental results show our algorithm is more efficient than traditional algorithm.**

***Index Terms*—frequent pattern, closed, resource**

I. Introduction

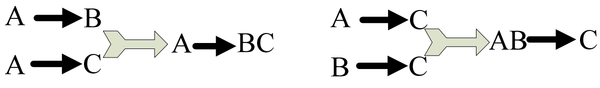
The resource is physical support of the avionics system. Normal operation in the system is achieved through reasonable scheduling of numerous resources. Every resource has certain capacity, while the resource capability is mainly reflected in the degree of satisfying specified demand. In the avionics system, different resource demands have different resource capability combinations; different resource capability combinations have different capacity integration approaches, while different capacity integration approaches give rise to resource capability relevance. Such capacity relevance makes as long as one resource goes wrong, other normal resources will warn the failure simultaneously. So, the analysis of association rules of capacity-related resources from the resource effectiveness data contributes to quick positioning of main fault resource, improving the rate of resource utilization, reducing system resource allocation, lowering complexity influence and enhancing system resource efficiency.

Since the number of resources in the system is large, the scale of resource effectiveness matrix collected in a sampling time bucket is huge. The influence of how to efficiently mine resource mode from these data on system prognostics and health management [1] of system efficiency is very pivotal. Data mining technology widely applied currently can mine knowledge from large quantities of data. Association rule mining [2-5] is an important branch of data mining technology. This method can mine the resource mode with some derivation relationship from large quantities of data. The form of association rule is denoted as $X \Rightarrow Y$, where X and Y represent one or more items in the dataset. The occurrence of X results in the occurrence of Y. In the analysis of resource effectiveness matrix, the form of association rules includes $R_1 \Rightarrow R_2$ (support: 80%, confidence: 100%), where R_1 and R_2 are two different resources or a group of resources. The meaning of this rule is that when R_1 is very healthy, R_2 is also 100% healthy; meanwhile, the proportion of common health for R_1 and R_2 in the whole resource effectiveness is 80%. The resources which go wrong and need to be replaced, can be gained according to the result of association rules. Assuming such association rule that $-R_3 \Rightarrow -R_4$ (support: 80%, confidence: 100%) exists, "-" means the occurrence of failure. It can be seen from the above association rule that after R_3 goes wrong, the probability of R_4 failure is 100%. So, R_4 failure may result from R_3 cascading delivery. From the perspective of frequent pattern mining, R_4 resource may be regarded as failure resource. However, from the perspective of association rules, after R_3 resource is replaced, R_4 resource may be normal. Therefore, association rules used to mine resource failure can reduce the number of wrong alarm resources to be replaced. From the economic perspective, association rules analysis is very important.

Traditional association rules mining approaches generally adopt the basic method, i.e. first mine frequent items [6-19] and then produce association rules. The association rules with high confidence can be gained through mining frequent items. But the duration of association rules mining and the internal storage consumption are very huge. For example, assuming $R_1R_2R_3$ is a frequent item, a total of 6 association rules can be produced, including $R_1 \Rightarrow R_2R_3$, $R_2 \Rightarrow R_1R_3$, $R_3 \Rightarrow R_1R_2$, $R_1R_2 \Rightarrow R_3$, $R_1R_3 \Rightarrow R_2$ and $R_2R_3 \Rightarrow R_1$. The number of resources involved in the system is often very numerous, so it will be time-consuming to produce association rules based on frequent pattern mining. Meanwhile, there is no judgment of confidence in the process of frequent pattern mining in the first step, so the results mined in the first

doi:10.4304/jsw.9.9.2417-2426

step may be pruned in the second step. This will greatly influence the efficiency of the algorithm. Thus, *SAW* algorithm [20] proposed by Wang et al. can directly mine inter-genic association rules from gene chip data. This algorithm is different from traditional method (first mine frequent pattern and then use confidence to produce association rules. *SAW* algorithm first mines all 2-length association rules that two genes meet support and confidence simultaneously; then all gene association rules are produced through merging these rules. Rule merging strategy contains two ways: forward merging and backward merging, as shown in Fig.1. But, this algorithm has no efficient mining strategy and just mines closed strong association rules through the definition.



(a) Backward combined (b) Forward combined

Figure 1. Two merging strategies in SAW algorithm

On the basis of the above analysis, this paper proposes closed strong association rules mining algorithm: *CSRule*, which is based on association rules merging strategies. This algorithm adopts several pruning strategies and thus can mine closed strong association rules without storing the candidate set. Different from *SAW* algorithm, *CSRule* algorithm adopts effective pruning strategies to mine closed strong association rules in real time, instead of conducting secondary mining only through the definition.

II. Problem Description

Resource effectiveness matrix is defined as a two-dimensional real matrix $D = R \times S$. Here, row set R represents the resource name; column set S refers to different sampling sites. Element D_{ij} of matrix D is a real number which refers to the effective value (e.g., BIT value) of resource i under sampling j. $|R|$ is the number of resources in data set D and $|S|$ is the number of sampling sites in data set D. For the convenience of mining, the original effective value in resource effectiveness matrix is usually discretized into 1, -1 and 0, where 1 means resource health; 0 refers to resource sub-health; -1 means resource failure, as shown in Table 1.

Definition 1: the relationship of resource R_1 and R_2 can be defined as follows: (1) If R_1 and R_2 are very effective simultaneously (both values are 1), R_1 and R_2 are of effective positive correlation, expressed as R_1R_2 with the support of $sup(R_1R_2) = \frac{|R_1R_2|}{|S|}$, where $|R_1R_2|$ is the number of sampling sites when R_1 and R_2 are very effective simultaneously (similarly hereinafter); (2) If R_1 and R_2 go wrong simultaneously (both values are -1), R_1 and R_2 are of failure positive correlation, expressed as $-R_1$-R_2 with the support of $sup(-R_1 - R_2) = \frac{|-R_1 - R_2|}{|S|}$; (3) If R_1 is very effective and R_2 goes wrong, R_1 and R_2 are of effective negative correlation, expressed as R_1-R_2; if R_2 is very effective and R_1 goes wrong, R_1 and R_2 are of effective negative correlation, expressed as $-R_1R_2$. When R_1 and R_2 are of effective negative correlation, the support is $sup(R_1 - R_2) = \frac{|R_1 - R_2|}{|S|} + \frac{|-R_1R_2|}{|S|}$.

According to the above support definition, confidence of association rules can be calculated, i.e. *conf*（R_1->R_2）=$sup(R_1R_2)/sup(R_1)$, where R_1 and R_2 can be a group of resources. So, three association rule modes aiming at the three modes of support can be gained through the support of three relations among resources, i.e. corresponding resource association rules can be gained through calculating resource frequent mode according to the definition of association rule. In this paper, resource association rules mined by *CSRule* algorithm are four types described in Definition 2.

TABLE I.
Discrete Resource Effectiveness Matrix

	S_0	S_1	S_2	S_3	S_4
R_1	1	-1	1	-1	1
R_2	1	-1	1	-1	-1
R_3	0	1	-1	1	1
R_4	1	1	0	1	1
R_5	-1	-1	0	-1	1

Definition 2: There are four types of resources association rules mined by *CSRule*: (1) $-R_m$->$-R_n$ means R_m resource failure will result in R_n resource failure, where R_m and R_n can be a group of resources (similarly hereinafter), *conf* ($-R_m$->$-R_n$)=*sup* ($-R_m$-R_n)/*sup*($-R_m$); (2) $-R_m$->R_n means R_m resource failure will result in normal operation of R_n. At this moment, R_m and R_n may have some kind of coupling. If R_m is changed, R_n failure may be caused. *conf* ($-R_m$->R_n)=*sup* ($-R_mR_n$)/*sup*($-R_m$); (3) R_m->$-R_n$ means normal operation of R_m will lead to R_n failure. At this moment, R_m and R_n may have some kind of coupling. If R_n is changed, R_m failure may be caused, *conf* (R_m->$-R_n$)=*sup* (R_m-R_n)/*sup*(R_m); (4) R_m->R_n means when R_m normally operates, R_n also normally operates. The significance is that when R_m is used, the reliability of the use of R_n is higher, *conf* (R_m->R_n)=*sup* (R_m-R_n)/*sup*(R_m).

Definition 3: if an association rule has not the superset with the same support threshold and confidence threshold, this association rule is closed strong association rule.

The purpose of *CSRule* algorithm is to mine all closed strong association rules meeting support and confidence thresholds as well as four modes in Definition 2 and restraints in Definition 3 from resource effectiveness matrix. Unlike traditional association rules mining, *CSRule* algorithm mines in the way of association rule extension. The specific mining process will be introduced in next section.

III. The CSRULE Algorithm

The mining steps of *CSRule* algorithm can be divided into two steps: firstly, produce all association rules (for the convenience of expression, association rules are called association rule pair for short) which satisfy the support and confidence thresholds and whose precursor and successor have only one resource; then, merge association rules according to “forward merging” and “backward merging” used in *SAW* algorithm to mine all closed strong association rules.

A. Association Rule Pair Generation

Generation of all association rule pairs can delete association rules which dissatisfy support threshold and confidence threshold so as to avoid the shortcoming of traditional generation of association rules based on frequent pattern: frequent pattern only satisfies support threshold and possible item sets dissatisfy confidence threshold. Meanwhile, to improve the emerging efficiency in next step of the algorithm, the sample information of each rule is saved while association rule pairs are generated. During generation of closed strong association rules, the sample information can be directly used to calculate and generate support and confidence of association rules so as to avoid scanning of original data. For example, in the resource effectiveness matrix shown in Table 1, for the rule R_3->-R_2, the support is 0.6 and the confidence is 100%; the frequent sample of R_3-R_2 recorded is $S_1S_3S_4$. For R_3 and R_4, the expression mode of the association rule is R_3->R_4. The support is 0.6 and the confidence is 100%; the frequent sample recorded is $S_1S_3S_4$. In order to boost merging and extension efficiency, for each association rule, not only frequent sample information but also the sample information of rule precursor should be recorded. For instance, precursor frequent sample information of the rule R_3->-R_2 is frequent sample set of R_3, i.e. $S_1S_3S_4$. The purpose of recording precursor frequent sample information is to facilitate calculation of confidence of association rules newly generated in follow-up “merging”. How to use the information will be stated in the next section.

B. Merging Strategy

After all association rules whose confidence and successor have only one resource are produced, the above association rules can be merged in line with “forward merging” or “backward merging” strategies. In accordance with “forward merging” or “backward merging” strategies described in *SAW* algorithm, the form of closed strong association rules finally gained is R_i->R_m…R_n or R_m…R_n->R_j. That is to say, the number of resources in precursor set or successor set in the rules cannot exceed 1 simultaneously, i.e. either the precursor has only one resource or the successor has only one resource; the precursor and the successor cannot have 2 resources or above. So, it is known according to the definitions of four association rules in Definition 2 that association rules in the first mode and the second mode comply with “backward merging” strategy; association rules in the third mode and the fourth mode comply with “backward merging” strategy; association rules in the first mode and the third mode comply with “forward merging” strategy; association rules in the second mode and the fourth mode comply with “forward merging” strategy.

The establishment of the above merging strategies can boost the efficiency of *CSRule* algorithm to generate candidate association rules: (1) the association rule extended currently is in the first mode. During “forward merging”, its candidate association rules can only be in the first mode and the third mode; during “backward merging”, its candidate association rules can only in the first mode and the second mode; (2) the association rule extended currently is in the second mode. During “forward merging”, its candidate association rules can only be in the second mode and the fourth mode; during “backward merging”, its candidate association rules can only in the first mode and the second mode; (3) the association rule extended currently is in the third mode. During “forward merging”, its candidate association rules can only be in the first mode and the third mode; during “backward merging”, its candidate association rules can only in the third mode and the fourth mode; (4) the association rule extended currently is in the fourth mode. During “forward merging”, its candidate association rules can only be in the second mode and the fourth mode; during “backward merging”, its candidate association rules can only in the third mode and the fourth mode. Therefore, during candidate judgment, it is necessary to first judge the mode type. If the mode of association rule currently extended and the mode of follow-up association rules comply with one of the above four requirements, support and confidence thresholds can be calculated.

We will briefly introduce how to use frequent sample information of association rules and precursor frequent sample information to calculate support and confidence. For instance, during “backward merging” of R_3->-R_2 and R_3->R_4, frequent sample information of R_3-R_2R_4 can be obtained through calculating the intersection of frequent samples of the above two association rules, i.e. $S_1S_3S_4$. Thus, the support of R_3-R_2R_4 is 0.6. This avoids scanning of original data set. For association rule R_3->-R_2R_4 generated through “backward merging”, it is necessary to know the support of precursor R_3 during calculation of the confidence. So, it is necessary to scan storage linked list only once where support values if each resource is stored. Thus, the support of each resource can be gained. Therefore, during “backward merging”, it is only necessary to record support of single precursor resource, without the need of recording frequent sample information.

For “forward merging”, since precursor resources of association rules generated grow dynamically, it is necessary to know the support of the precursor during calculation of confidence. So, during extension, it is necessary to record precursor frequent sample information of the association rules generated. For example, for -R_2->R_3, its support is 0.6 and confidence is 100%; frequent sample of -R_2R_3 needs recording, i.e. $S_1S_3S_4$; precursor frequent sample of -R_2 needs recording, i.e. $S_1S_3S_4$; for R_4->R_3, its support is 0.6 and confidence is 75%; frequent sample of R_1-R_4 needs recording, i.e. $S_1S_3S_4$; precursor frequent sample of R_4 needs recording, i.e. $S_0S_1S_3S_4$. -R_2R_4->R_3 can be generated through “forward merging” of the above two rules. Frequent sample of $-R_2R_4R_3$ is the intersection of the sample information of the above two association rules: $S_1S_3S_4$, so the support is 0.6. The precursor frequent sample of $-R_2R_4\text{->}R_3$ can be obtained through calculating the intersection of precursor frequent sample of $-R_2$ in $-R_2\text{->}R_3$ and precursor frequent sample of R_4 in $R_4\text{->}R_3$: $S_1S_3S_4$. Therefore, confidence of $-R_2R_4\text{->}R_3$ is 100%. Thus, the support and confidence of the association rules generated after the merging can be directly calculated according to sample information and precursor frequent sample information of merged association rules, which avoids scanning of original data set and improving mining efficiency of the algorithm.

C. Pruning Strategy

We are about to introduce how to design pruning strategies to mine closed strong association rules in real time without storing the candidate set. According to the previous analysis, there are two ways for association rule extension: "forward merging" and "backward merging". Extension of initial association rules can be "forward merging" or "backward merging". For the association rule with middle extension, if this rule is gained through "forward merging" extension of initial association rule, all candidate association rules of this association rule can only be extended through "forward merging". If this rule is gained through "backward merging" extension of initial association rule, all candidate association rules of this association rule can only be extended through "backward merging". For "backward merging", it is known from the following Lemma 1 that the support and confidence satisfy anti-monotonicity; for "forward merging", it is known from the following Lemma 2 that the support meets anti-monotonicity, but the confidence dissatisfies anti-monotonicity. So, during design of pruning strategies, the pruning ways of "forward merging" and "backward merging" are slightly different.

Lemma 1: when association rules are in accordance with "backward merging" strategy, the support and the confidence satisfy anti-monotonicity.

Proof: when association rules are in accordance with "backward merging" strategy, the number of resources in the rules newly generated will increase, so the support of resource set will not increase. Thus, the support meets anti-monotonicity during "backward merging" of association rules. According to the definition of confidence, the numerator is the support of al resources in the rules newly generated and the denominator is the support of precursor resources. The number of precursor resources is unchanged during "backward merging" of association rules", and the number of successor recourses increases. Thus, the support of the denominator in the computational formula of confidence is unchanged and the support of numerator will not increase. Therefore, the confidence also meets anti-monotonicity during "backward merging" of association rules".

Lemma 2: when association rules are in accordance with "forward merging" strategy, the support satisfies anti-monotonicity, but the confidence dissatisfies anti-monotonicity.

Proof: when association rules are in accordance with "forward merging" strategy, the number of resources in the rules newly generated will increase, so the support of resource set will not increase. Thus, the support meets anti-monotonicity during "forward merging" of association rules. According to the definition of confidence, the numerator is the support of al resources in the rules newly generated and the denominator is the support of precursor resources. The number of precursor resources increases during "forward merging" of association rules", and the number of successor recourses is unchanged. Thus, the support of the denominator in the computational formula of confidence will not increase and the support of numerator will not increase, either. Therefore, the confidence also meets anti-monotonicity during "backward merging" of association rules". Consequently, the following conclusion cannot be gained: with the execution of "forward merging", the confidence of association rules newly generated shows less or equal variation trend. Thus, the confidence dissatisfied anti-monotonicity during "forward merging" of association rules.

According to Lemma 1, for "backward merging", since both support and confidence meet anti-monotonicity, as long as strong association rules currently extened do not meet the support threshold or confidence threshold, the extension can stop. *CSRule* algorithm adopts precursor detection to design pruning strategies. In other words, if frequent sample information of candidate association rules to be extended currently is the subset of frequent sample information of a precursor candidate association rule (i.e. all closed strong association rules which can be obtained through extension of current candidate association rules can be gained through extension of this precursor candidate association rule), current candidate association rules can be pruned. For example, as shown in Table 1, assuming the association rule currently extended is $-R_1\text{->}-R_2$, and $-R_1\text{->}R_3$ has been extended, the candidate association rule to be extended currently is $-R_1\text{->}R_4$. At this moment, relative to $-R_1\text{->}-R_2$, frequent sample of candidate association rule $-R_1\text{->}R_4$ is S_1S_3 (in other words, calculating the intersection of frequent sample S_1S_3 of $-R_1\text{->}-R_2$ and frequent sample S_1S_3 of $-R_1\text{->}R_4$, i.e. frequent sample of $-R_1\text{->}-R_2R_4$). Since $-R_1\text{->}R_3$ has been extended, $-R_1\text{->}R_3$ is the precursor candidate sample of $-R_1\text{->}-R_2$. And, relative to $-R_1\text{->}-R_2$, frequent sample of $-R_1\text{->}R_3$ is S_1S_3 (in other words, calculating the intersection of frequent sample S_1S_3 of $-R_1\text{->}-R_2$ and frequent sample S_1S_3 of $-R_1\text{->}R_3$, i.e. frequent sample of $-R_1\text{->}-R_2R_3$). At this moment, frequent sample S_1S_3 of $-R_1\text{->}-R_2R_4$ is the subset of frequent sample of $-R_1\text{->}-R_2R_3$. So, all closed strong association rules which can be gained through extension of $-R_1\text{->}-R_2R_4$ can be obtained through extension of $-R_1\text{->}-R_2R_3$. At the same time, the latter is the superset of the former, and they have the same support and confidence. Thus, $-R_1\text{->}-R_2R_4$ can be pruned.

According to Lemma 2, for "forward merging", when the support meets closed judgment conditions (i.e. as the scale of association rules continuously extends, the support of strong association rules newly generated is unchanged), the confidence of the extended rules is on the rise. This is because the support of the numerator is unchanged, if the support of the denominator is also remains unchanged, the definition of closed strong association rules is satisfied. So, during "forward merging", not just frequent sample of current candidate association rules should satisfy the pruning strategy in the last "backward merging", but also precursor frequent sample of candidate association rules should also meet that: precursor frequent sample of candidate association rules is the subset of precursor frequent sample of a precursor candidate sample. For example, as shown in Table 1, assume the association rule currently extended is $-R_1$->R_3; $-R_2$->R_3 has been extended; candidate association rule to be extended currently is R_4->R_3. At this moment, frequent sample set of $-R_1R_4$->R_3 is the subset of frequent sample set of $-R_1$-R_2->R_3 (from the perspective of support). Meanwhile, precursor frequent sample set of $-R_1R_4$->R_3 is the subset of precursor frequent sample set of $-R_1$-R_2->R_3 (from the perspective of confidence). Then, current candidate association rule R_4->R_3 can be pruned.

Lemma 3: assuming that P is a strong association rule extended currently, M is the candidate association rule set of P and N is the prior candidate association rule set of P. For a candidate sample $M_i(M_i \in M)$, if there exists a prior candidate association rule $N_j(N_j \in N)$ making the frequent sample set of PM_i be the subset of frequent sample set of PN_j, association rules which can be gained through extension of PM_i is the subset of the association rules gained through extension of PM_iN_j.

Proof: proof by contradiction is adopted. Assume when frequent sample set of current candidate association rule M_i is not the subset of frequent sample set of precursor candidate association rule N_j, M_i can be pruned. According to the assumption, the sample set which does not belong to PN_j exists in PM_i. Since closed association rule mining adopts depth-first way and N_j prior to M_i for extension, another association rule R_m making frequent sample set of PM_iR_m unequal to frequent sample set of $PM_iN_jR_m$ may exist. So, M_i cannot be pruned. This contradicts the assumption, so the original proof is established.

Lemma 4: assuming that P is a strong association rule extended currently, M is the candidate association rule set of P and N is the prior candidate association rule set of P. For a candidate sample $M_i(M_i \in M)$, if there exists a prior candidate association rule $N_j(N_j \in N)$ making frequent sample set of PM_i be a subset of the frequent sample set of PN_j and the prior frequent sample set of PM_i is the subset of prior frequent sample set of PN_j, then association rules which can be gained through extension of PM_i, is the subset of the association rules gained through extension of PM_iN_j.

Proof: proof by contradiction is adopted. (1) Assume when frequent sample set of current candidate association rule M_i is not the subset of frequent sample set of precursor candidate association rule N_j, M_i can be pruned. This process of proof is the same with that of Lemma 3, so it is omitted here; (2) assume when precursor frequent sample set of current candidate association rule M_i is not the subset of precursor frequent sample set of precursor candidate association rule N_j, M_i can be pruned. According to the assumption, another association rule R_m making precursor frequent sample set of PM_iR_m unequal to precursor frequent sample set of $PM_iN_jR_m$ (i.e. the confidence of PM_iR_m may be unequal to that of $PM_iN_jR_m$) may exist. In accordance with the definition of closed association rules, PM_iR_m may be a closed association rule, so it cannot be pruned. This contradicts the assumption, so the original proof is established.

Based on the above lemmas, *CSRule* algorithm uses the following four pruning strategies to prune candidate samples so as to improve mining efficiency of the algorithm.

Pruning strategy 1: when association rules are in accordance with "backward merging" strategy, if the support or confidence of the association rule currently extended dissatisfies threshold requirement users define, extension of current association rule will stop.

Pruning strategy 2: when association rules are in accordance with "forward merging" strategy, if the support of the association rule currently extended dissatisfies threshold requirement users define, extension of current association rule will stop.

Pruning strategy 3: assuming that P is the strong association rule currently extended, M is candidate association rule set of P, and N is precursor candidate association rule set of P, if a precursor candidate association rule N_j ($N_j \in N$) making frequent sample set of PM_i is the subset of frequent sample set of PN_j exists for candidate association rule M_i ($M_i \in M$), PM_i will be pruned.

Pruning strategy 4: assuming that P is the strong association rule currently extended, M is candidate association rule set of P, and N is precursor candidate association rule set of P, if a precursor candidate association rule N_j ($N_j \in N$) making frequent sample set of PM_i is the subset of frequent sample set of PN_j exists for candidate association rule M_i ($M_i \in M$), and precursor frequent sample set of PM_i is the subset of precursor frequent sample set of PN_j, PM_i will be pruned.

D. Algorithm Procedure and Example

Based on the previous descriptions, *CSRule* algorithm uses association rule merging to directly mine all closed association rules without storing the candidate association rules. The following algorithm description provides specific mining process. Figures 1-4 illustrate the mining process of *CSRule* algorithm: Fig.1 describes precursor resource as the normal resource and adopts "backward merging" extension way; Fig.2 describes precursor resource as the failure resource and adopts "backward merging" extension way; Fig.3 describes successor resource as the normal resource and adopts "forward merging" extension way; Fig.4 describes successor resource as the failure resource and adopts "forward merging" extension way, in the example, the data are shown in Table 1; the support threshold is 0.4; the confidence threshold is 70%.

Algorithm: CSRule algorithm

Input: support threshold: r_{min}, confidence threshold: $conf_{min}$, resource effectiveness data: D

Output: all closed strong association rules meeting the threshold

Initial value: association rule pair: G =Null, association rules to be extended currently: Q =Null, G_i=Null, G_j=Null

Algorithm description: CSRule(r_{min},$conf_{min}$, D, Q, G_i, G_j)

(1) If G is null, scan data set D to mine all strong association rule pairs meeting support and confidence thresholds and save in G and G_i as the first association rule in *G;*

(2) If Q is null, Q=G_i

(3) For every association rule G_j in G

(4) If QG_j is forward merging and satisfies pruning conditions of forward merging

(5) continue;

(6) else if QG_j is backward merging and satisfies pruning conditions of backward merging

(7) Continue;

(8) else

(9) Q=QG_j

(10) if QG_j is backward merging, then calculate the intersection of frequent sample set of Q and frequent sample set of G_j as frequent sample set of QG_j;

(11) if QG_j is forward merging, then calculate the intersection of frequent sample set of Q and frequent sample set of G_j as frequent sample set of QG_j, and calculate the intersection of precursor frequent sample set of Q and precursor frequent sample set of G_j as precursor frequent sample set of QG_j;

(12) CSRule (r_{min}, $conf_{min}$, D, Q, G_i, G_j->next);

(13) endif

(14) endfor

(15) If Q meets the definition of closed association rule, then

(16) Output Q

(17) endif;

(18) G_i = G_i->next;

(19) return

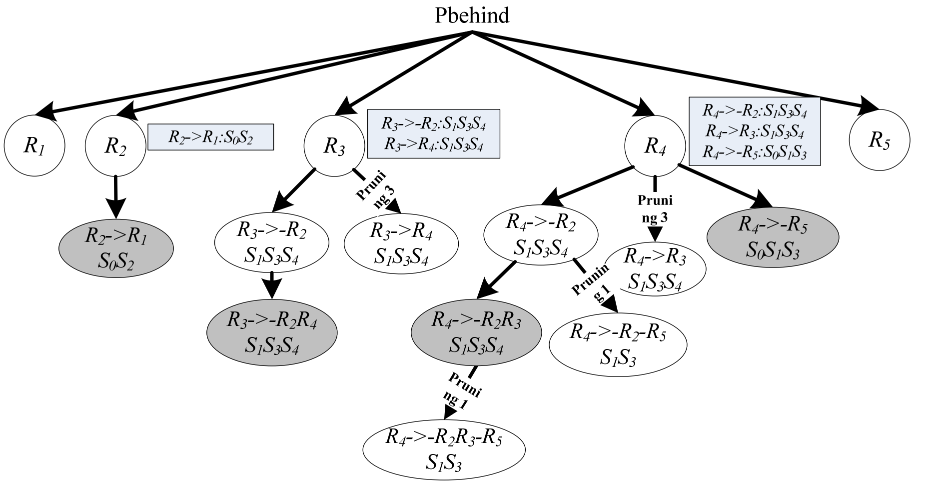


Figure 2. Example for mining precursor of CSRule algorithm as normal resource with “backward merging” extension

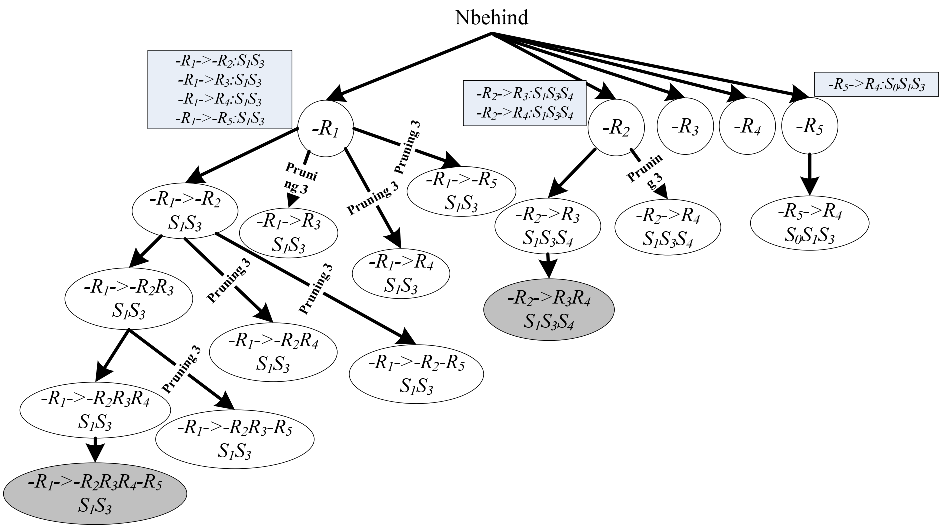


Figure 3. Example for mining precursor of CSRule algorithm as failure resource with “backward merging” extension

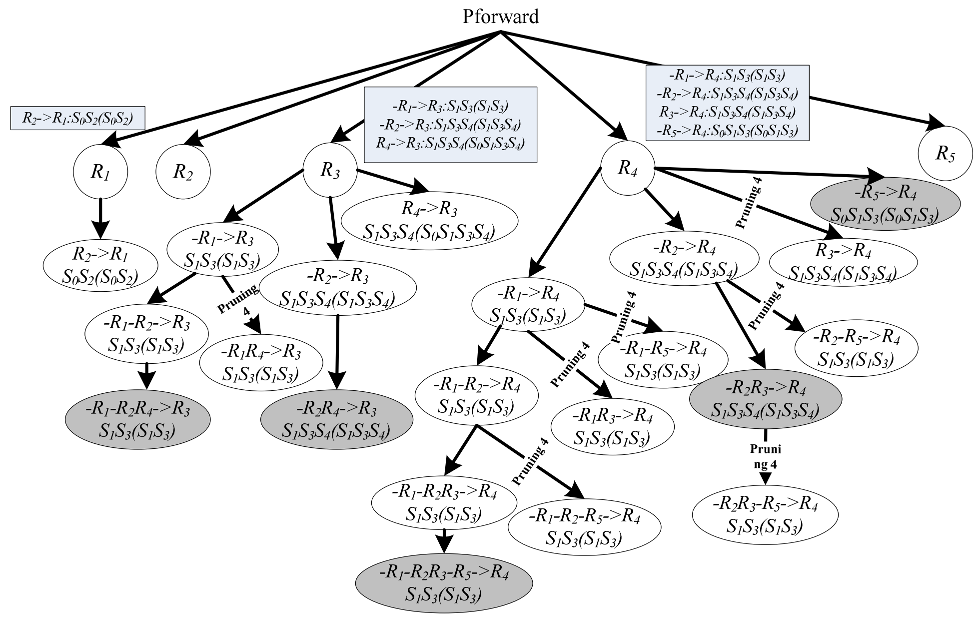


Figure 4. Example for mining successor of CSRule algorithm as normal resource with "forward merging" extension

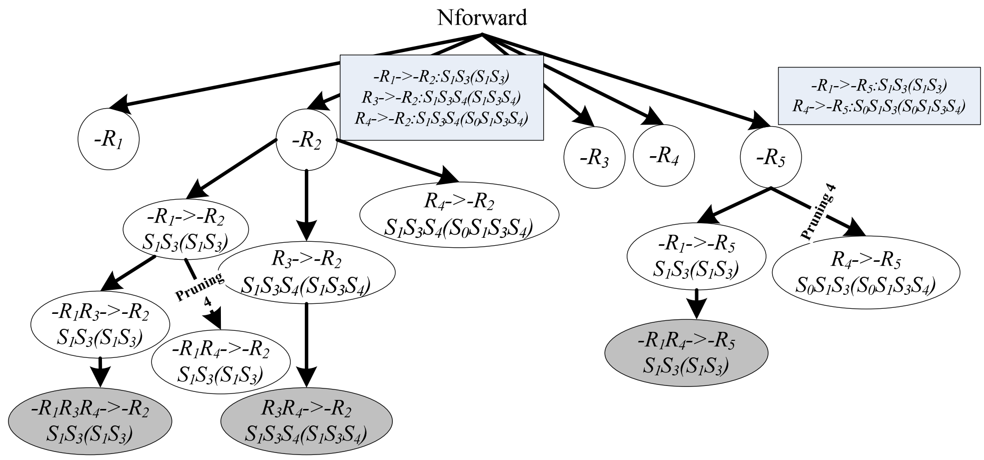


Figure 5. Example for mining successor of CSRule algorithm as failure resource with "forward merging" extension

IV. Experimental Results

In this section, we will make an experimental comparison on the mining efficiency and result of the algorithm above and existing algorithms. The hardware environment of the experiment is Intel(R) Core(TM)2 Duo 2.53GHz CPU and 4G internal memory; the software environment is Microsoft Windows 7 SP1 operating system; the algorithm programming and operating environment is Microsoft Visual C++ 6.0 SP6. Experimental data used in this paper are simulation data. To fully test the performance of the algorithm, we produce six data sets randomly, each of which contains 20 sampling sites and 800 resources. Table 2 describes proportions of 1, 0 and -1 in each row in each data set.

TABLE II.
PROPORTIONAL DISTRIBUTION OF NUMERICAL VALUES IN SIX DATA SETS

	1	0	-1
D_1	0.4	0.2	0.4
D_2	0.4	0.3	0.3
D_3	0.5	0.2	0.3
D_4	0.4	0.4	0.2
D_5	0.5	0.3	0.2
D_6	0.5	0.25	0.25

In this section, mining efficiency of *CSRule* algorithm and *SAW* algorithm is compared. *SAW* algorithm adopts the mining algorithm described in [20] for mining. Similar to *CSRule* algorithm, *SAW* algorithm also adopts association rule extension to mine all closed strong association rules. Different from *CSRule* algorithm, *SAW* algorithm does not adopt any pruning strategy for mining. To fully compare the extendibility of the algorithms, we produce multiple groups of data sets with different resources quantities. The selection of recourses is subject to resource sequence in the data sets.

Figures 6(a)-6(c) provide the comparison of performance duration of the above two algorithms under different data sets when the number of resources is 200 with the support of 0.25 and the confidence of 0.8, 0.7 and 0.6 respectively. It can be seen from these figures that with the decrease in the confidence, the mining efficiency of the two algorithms also decreases; but under almost all data sets with different scales, the mining efficiency of *CSRule* algorithm is higher than that of *SAW* algorithm. However, under D_4 data set in Fig. 5(c), the efficiency of *CSRule* algorithm is slightly lower than that of *SAW* algorithm. This is because under low confidence, the algorithm will produce many association rule pairs with the length of 2. So, in the mining process, it is necessary to make frequent closure judgment for association rules generated through merging. If the judgment shows the rule will not be pruned, this will seriously influence the pruning efficiency of the algorithm. Since the proportion of "1" in D_3, D_4 and D_6 is high, the data set is dense. Thus, more association rule pairs will be produced. Therefore, under the condition where association rule pairs initially merged are many, the mining efficiency of the two algorithms is not high. When the confidence is 0.7 and 0.6, the above mining process under three data sets cannot be completed in limited internal storage restraint.

In order to further compare the extendibility of the two algorithms, figures 7(a)-7(c) provide the comparison of performance duration of the above two algorithms under different data sets when the number of resources is 500 with the support of 0.45 and the confidence of 0.8, 0.7 and 0.6 respectively. Figures 8(a)-8(c) provide the comparison of performance duration of the above two algorithms under different data sets when the number of resources is 800 with the support of 0.45 and the confidence of 0.8, 0.7 and 0.6 respectively. It can be seen from the figures that under almost all data sets, the mining efficiency of is higher than that of *SAW* algorithm.

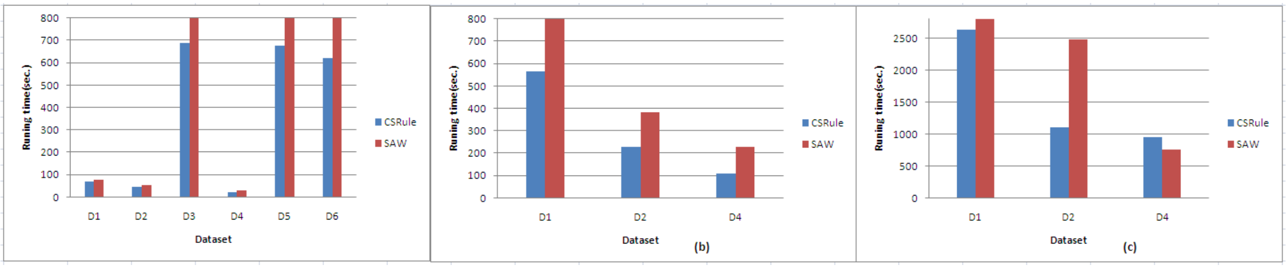


Figure 6. Comparison of performance duration of two algorithms under different data sets when the number of resources is 200: (a) confidence: 0.8; (b) confidence: 0.7; (c) confidence: 0.6

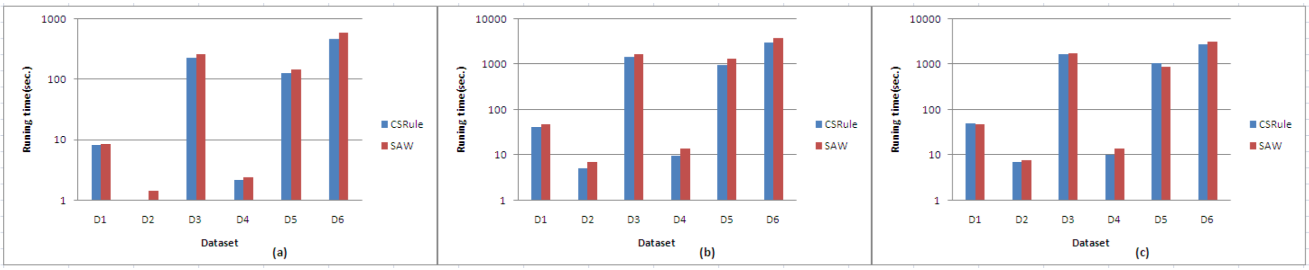


Figure 7. Comparison of performance duration of two algorithms under different data sets when the number of resources is 500: (a) confidence: 0.8; (b) confidence: 0.7; (c) confidence: 0.6

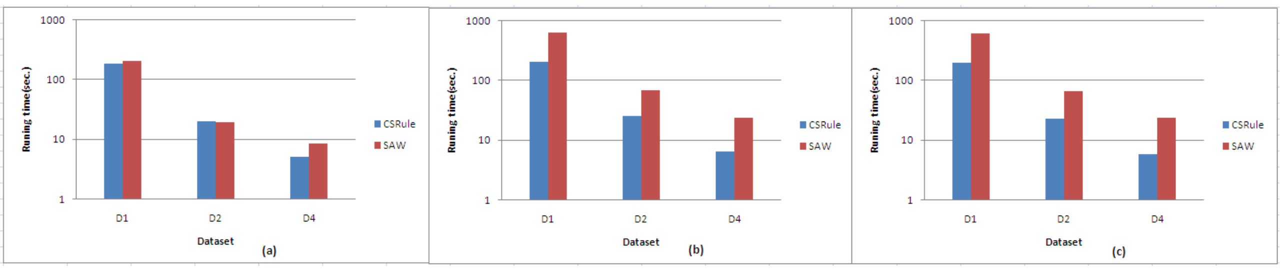


Figure 8. Comparison of performance duration of two algorithms under different data sets when the number of resources is 800: (a) confidence: 0.8; (b) confidence: 0.7; (c) confidence: 0.6

V. CONCLUSION

This paper proposed an efficient algorithm for mining closed strong association rules based on association rule merging strategies: *CSRule*. This algorithm adopts multiple pruning strategies to mine closed strong association rules without storing the candidate set. To improve the mining efficiency of the algorithm, *CSRule* algorithm adopts effective pruning strategies to mine closed strong association rules in real time, instead of secondary mining only through the definition. However, original data information will be lost if the mining process is conducted in the discrete data. Our future research object is to mine closed strong association rules related to resource health from true resource effectiveness data.

ACKNOWLEDGMENT

This paper is Supported by National Key Basic Research Program of China under Grant No. 2014CB744900.

REFERENCES

[1] Michael Pecht,et al.. A prognostics and health management roadmap for information and electronics-rich systems. Microelectronics Reliability, 2010:317–323.
[2] Becquet, C. Blachon, S., Jeudy, B., Boulicaut, J.F. and Gandrillon, O. Strong-association-rule mining for large-scale gene-expression data analysis: a case study o human SAGE data. Genome Biol, 2003, 12: 1-16.
[3] Creighton, C, Hanash,S. Mining gene expression databases for association rules. Bioinformatics, 2003, 19: 79-86.
[4] T. McIntosh, S. Chawla. High confidence rule mining for microarray analysis. IEEE/ACM TCBB, 2007, 4(4):611–623.
[5] G. Cong, A. Tung, X. Xu, F. Pan, and J. Yang, "FARMER: Finding Interesting Rule Groups in Microarray Datasets," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2004, p: 143-154.
[6] Wang, J, Han, J. BIDE: Efficient Mining of Frequent Closed Sequences, Data Engineering, 2004. Proceedings. p: 79 – 90.
[7] Saito T, Okada Y. Bicluster-Network Method and Its Application to Movie Recommendation. Knowledge and Systems Engineering. Springer International Publishing, 2014: 147-153.
[8] De Sousa Filho G F, dos Anjos F Cabral L, Ochi L S, et al. Hybrid Metaheuristic for Bicluster Editing Problem. Electronic Notes in Discrete Mathematics, 2012, 39: 35-42.
[9] Miao Wang, Xuequn Shang, Jingni Diao, Zhanhuai Li: WIBE: Mining Frequent Closed Patterns Without Candidate Maintenance in Microarray Dataset. DMIN 2010: 200-205.
[10] Pandey, G., Atluri, G., Steinbach, M., Myers, C. L., & Kumar, V.: An association analysis approach to biclusting. In Proc. ACM Conf. on Kownlege Discovery and Data Mining, 2009, pages 677-686.
[11] Xufei Zheng, Yanhui Zhou, Yonghui Fang. The Dual Negative Selection Algorithm Based on Pattern Recognition Receptor Theory and Its Application in Two-class Data Classification. Journal of computers, Vol 8, No 8, 2013, p:1951-1959.
[12] Haichao Luo, Xiaomin Li, Shujing Zheng, Mei Li, Lili Song. Study on Synthesis Evaluation of Intensive Land Use and Growth Pattern Transformation of Towns. Journal of computers, Vol 7, No 8, 2012, p:1959-1966.
[13] Mishra S, Mishra D, Satapathy S K. Fuzzy Frequent Pattern Mining from Gene Expression Data using Dynamic Multi-Swarm Particle Swarm Optimization. Procedia Technology, 2012, 4: 797-801.
[14] Király A, Abonyi J, Laiho A, et al. Biclustering of High-throughput Gene Expression Data with Bicluster Miner. Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on. IEEE, 2012: 131-138.
[15] Yun U, Lee G, Kim S J. Analyzing Efficient Algorithms of Frequent Pattern Mining. IT Convergence and Security 2012. Springer Netherlands, 2013: 937-945.
[16] Zaki MJ, Hsiao CJ. CHARM: An efficient algorithm for closed itemset mining. In: Grossman R, et al, eds. Proc. of the 2nd SIAM Int'l. Conf. on Data Mining. Arlington: SIAM, 2002, 12-28.
[17] Meng Han, Zhihai Wang, Jidong Yuan. Closed Sequential Pattern Mining in High Dimensional Sequences. Journal of Software, Vol 8, No 6, 2013, p:1368-1373.
[18] Qunhui Wu, Shilong Ma, Hao Wang. Extracting Feature Sequences in Software Vulnerabilities Based on Closed Sequential Pattern Mining. Journal of Software, Vol 8, No 8, 2013, p:1809-1817.
[19] Diana W. Verzi. Subset selection processes to identify significant input variables for nonlinear target functions. Journal of Interdisciplinary Mathematics, Volume 14, Issue 3, June 2011, p: 303-320.
[20] Miao Wang, Xuequn Shang, Qian Zhao, Zhanhuai Li. Strong association rules mining without using frequent items for microarray analysis. Proceedings of the 3rd International Conference on Bioinformatics and Biomedical Engineering, iCBBE2009.

Lihua Zhang is a doctoral student at the school of computer science and engineering at the northwestern polytechnical university, Xi'an China. She completed her master degree from northwestern polytechnical university in 2008. Her current research interests are PHM, avionics, data mining and safety. Since 2013, she has been studying at science and technology on avionics integration laboratory.

Miao Wang is an engineer at science and technology on avionics integration laboratory. He completed his doctor degree and master degree from northwestern polytechnical university in 2013 and 2018, respectively. He is a member of China computer federation. His research interests mainly include data mining, PHM, avionics and safety.

Zhengjun Zhai is a professor at the school of computer science and engineering at the northwestern polytechnical university, Xi'an China. He is vice chairman of NPU youth association for science and technology, distinguished expert of aerospace electrical & electronics and weapon system Standardization technology committee, distinguished expert of AAMRI and premium member of china computer federation. His research interests include experiment and testing systems Integration, remote maintenance and fault diagnosis and virtual visualization.

Guoqing Wang is a professor and a supervisor of Ph.D. student in Northwestern Polytechnical University. He was born in 1956 and received his M.S. and Ph.D. degrees in computer science and technology from the Northwestern Polytechnical University in 1984 and 1991 respectively. He is the institute director of China aeronautical radio electronics research institute, and the director of science and technology on avionics integration laboratory. He has long been engaged in the related technical research of avionic system integration, distributed parallel processing, high reliable fault-tolerant system, network and bus system etc. He serves as the vice director of national serve environment computer academy.