International Journal of Computer Theory and Engineering, Vol. 2, No. 5, October, 2010
1793-8201

Mining Imperfectly Sporadic Rules with Two Thresholds

Cu Thu Thuy and Do Van Thanh

***Abstract*—A sporadic rule is an association rule which has low support but high confidence. In general, sporadic rules are of rare occurrence but high value in many cases. Of the two types of perfectly and imperfectly sporadic rules, imperfectly sporadic rules are more difficult to mine since they consist of individual items with high support whereas the support of combinations of these items is low. The problem of mining imperfectly sporadic rules has not been completely solved till now. Thus, the paper describes an absolute answer to the question by proposing a problem of mining imperfectly sporadic rules with two thresholds and developing a MCISI (mining closed imperfectly sporadic itemsets) algorithm to find imperfectly sporadic itemsets with two thresholds. The development of MCISI algorithm is based on a closed itemset lattice, therefore efficiency of the algorithm can be improved through reduction of search space and removal of redundant imperfectly sporadic rules with two thresholds. We also point out that mining imperfectly sporadic rules could be considered as a special case of mining imperfectly sporadic rules with two thresholds, and imperfectly sporadic rules with two thresholds are of rare occurrence comparing with imperfectly sporadic rules.**

***Index Terms*—Rare Association Rule; Imperfectly Sporadic Rule; Imperfectly Sporadic Rule with Two Thresholds.**

I. Introduction

In recent years, mining association rules with low support and high confidence (one called rare association rules) has received much attention [3-8,10]. These rules rarely occur but in many cases they are very valuable. Koh, Rountree, and O'Keefe [3-5] have proposed the problem of mining these rules called sporadic ones. They are concretely defined as follows:

An association rule $A \to B$ is called a perfectly sporadic rule for maxSup and minConf if:

$$\text{confidence}(A \to B) \geq m\text{inConf, and}$$
$$\forall x \in A \cup B,\ \text{support}(x) < m\text{axSup}$$

An association rule $A \to B$ is called an imperfectly sporadic rule for maxSup and minConf if:

$$\text{confidence}(A \to B) \geq m\text{inConf, and}$$
$$\text{support}(\text{A} \cup \text{B}) < \text{maxSup, and}$$
$$\exists x \in A \cup B,\ \text{support}(x) \geq \max \text{Sup}$$

As we know, the focal point of mining association rules in general and of mining sporadic rules in particular is to find out frequent itemsets. Mining perfectly sporadic rules has been solved completely in [3], where the algorithm for finding frequent itemsets of these rules was developed by basing on the Apriori algorithm (called Apriori-Inverse one).

Until now the problem of mining imperfectly sporadic rules has not been solved completely. To discover imperfectly sporadic rules, in [4,5] the authors have divided the imperfectly sporadic rules into four types including: (1) rules have both frequent and infrequent itemsets in its antecedent and consequent; (2) rules have only frequent itemsets in both its antecedent and consequent; (3) rule have only frequent itemsets in its consequents and infrequent itemsets in its antecedents; and (4) rules have only infrequent itemsets in its consequents and frequent itemsets in its antecedents. The authors then proposed the MIISR algorithm to discover rules of the type 3 above. Mining imperfectly sporadic rules of the types 1 and 2 is open while meaning of imperfectly sporadic rules of the type 4 is very poor.

The goal of our study is to develop an efficient algorithm for mining imperfectly sporadic rules proposed in [3-5]. Specifically, we will focus to find imperfectly sporadic itemsets for association rules $A \to B$ as follows:

$$\text{confidence}(A \to B) \geq m\text{inConf, and}$$
$$\text{minSup} \leq \text{support}(\text{A} \cup \text{B}) < \text{maxSup, and}$$
$$\exists x \in A \cup B,\ \text{support}(x) \geq \max \text{Sup}$$

where minSup, maxSup (minSup < maxSup), minConf are below minimum support, above minimum support and minimum confidence of these rules respectively. All of the three thresholds are user-defined values.

In this paper, such rules are called imperfectly sporadic rules with *two thresholds*. The itemsets used to generate these rules are also called imperfectly sporadic ones with two thresholds. The proposed problem is different from the one of mining imperfectly sporadic rules in [4-5]. Here the minSup is added and is considered a below minimum support of sporadic rules. In fact there are some reasons for adding the minSup. *First,* we remark that support of every association rules is always positive and always bigger than or equal to $1/|O|$, where $|O|$ is number of transactions of a database. So if minsup = $1/|O|$ then the mining imperfectly sporadic rules with two thresholds is the one imperfectly sporadic rules in [3-5] and we can say that the mining imperfectly sporadic rules can be considered as a specific case of the above mining of imperfectly sporadic rules with two thresholds. *Second,* it supports to develop a new algorithm for finding out imperfectly sporadic itemsets in other approach in term of

Manuscript received June 14, 2010.

Cu Thu Thuy is a teacher at Economic Information System - Academy of Finance, Ha Noi, Viet Nam (e-mail: cuthuthuy@hvtc.edu.vn).

Do Van Thanh is a researcher and scientific manager. He has been working for National Center for Scio-Economic Information and Forecast, Ha Noi, Viet Nam (e-mail: Thanhdv_db@mpi. gov.vn).

comparing with the one of the defined algorithm in [4-5].

Concretely in this paper, the algorithm for finding imperfectly sporadic itemsets with two thresholds was developed under the approach of the CHARM algorithm [11], which is one of the most effective algorithms for finding frequent itemsets from transaction databases. Under this approach finding imperfectly sporadic itemsets with two thresholds is implemented on closed itemset lattice. Hence the search space for imperfectly sporadic itemsets is reduced and many redundant imperfectly sporadic rules can be removed.

The structure of the paper is as follows: Following the Introduction Section, Section 2 will discuss some important properties of imperfectly sporadic itemsets with two thresholds. This will serve as the base to propose an algorithm to find imperfectly sporadic itemsets with two thresholds in the next section. Section 3 will propose the algorithm and prove its soundness and completeness. This section also points out that the mining of imperfectly sporadic rules in [4-5] can be considered as a special case of the mining of imperfectly sporadic rules with two thresholds. Section 4 will introduce some experiments to evaluate performances of the proposed algorithm. Last, Section 5 will present some conclusions.

II. Preliminaries

Suppose that $\mathbf{D} \subseteq (\mathbf{O}, \mathbf{I})$, where $\mathbf{I} = \{i_1, i_2, ..., i_n\}$ is the universe of items, and $\mathbf{O} = \{t_1, t_2, ..., t_m\}$ is the universe of transactions, called a transaction database. Let $X \subseteq \mathbf{I}$, sup(X) be the support of X, that is a number (or percentage) of transactions in **D** containing X. An association rule is a conditional relation among itemsets $X \to Y$, where $X \subseteq \mathbf{I}$, $Y \subseteq \mathbf{I}$, $X \cap Y = \varnothing$. X is referred to as an antecedent of the rule and Y as a consequent. The confidence of an association rule conf($X \to Y$) is a number (or percentage) of transactions in **D** containing X, given that they also contain Y [1,2]. Let minSup, maxSup, minConf be below minimum support, above minimum support, and minimum confidence respectively, these values are user-determined with a range of (0, 1].

Definition 1. X is called an imperfectly sporadic itemset with two thresholds if:

$$\text{minSup} \le \text{sup}(X) < \text{maxSup}, \text{ and}$$
$$\exists x \in X, \text{sup}(x) \ge \text{maxSup}$$

X is called a maximal imperfectly sporadic itemset with two thresholds if it is not a subitemset of any imperfectly sporadic itemset with two thresholds.

The Definition 2 and Definition 3 are developed directly from the related definitions in [9].

Definition 2. (Data mining context) A data mining context is a triple $\hat{D} = (\mathbf{O}, \mathbf{I}, \mathbf{R})$, where $\mathbf{R} \subseteq \mathbf{O}\text{x}\mathbf{I}$ is a binary relation. Each couple $(t,i) \in \mathbf{R}$ denote the fact that the object $t \in \mathbf{O}$ is related to the item $i \in \mathbf{I}$.

Definition 3. (Galois connection) Let $\hat{D} = (\mathbf{O}, \mathbf{I}, \mathbf{R})$ be a data mining context. For $O \subseteq \mathbf{O}$ and $I \subseteq \mathbf{I}$, we define:

$$f: 2^{\mathbf{O}} \to 2^{\mathbf{I}}$$
$$f(O) = \{i \in I \mid \forall t \in O, (t,i) \in \mathbf{R}\}$$
$$g: 2^{\mathbf{I}} \to 2^{\mathbf{O}}$$
$$g(I) = \{t \in O \mid \forall i \in I, (t,i) \in \mathbf{R}\}$$

f(O) is a set of items associated with all transactions of O, and g(I) is a set of transactions related with all items of I. The couple of applications (f,g) is a Galois connection between the power set of **O** and the power set of **I**.

The operator $h = f_o g$ in $2^{\mathbf{I}}$, and $h' = g_o f$ in $2^{\mathbf{O}}$ are Galois closure operators.

Definition 4. Let X be an imperfectly sporadic itemset with two thresholds, X is called a closed imperfectly sporadic itemset with two thresholds if it is a closed itemset, i.e. h(X) = X, where h is the Galois connection.

Remark 1.

- According to the Definition 1, an imperfectly sporadic itemset with two thresholds is an infrequent itemset for the above minimum support maxSup but is a frequent itemset for the below minimum support minSup.
- Imperfectly sporadic itemsets with two thresholds do not have the Apriori property, i.e. subset of an imperfectly sporadic itemset with two thresholds may not be an imperfectly sporadic itemset with two thresholds.

Property 1. *The support of an imperfectly sporadic itemset with two thresholds is equal to the support of a smallest closed itemset containing it.*

In other words, if X is an imperfectly sporadic itemset with two thresholds then sup(X) = sup(h(X)).

This Property is obvious because "*The support of an itemset is equal to the support of a smallest closed itemset containing it"* was already proved in [9].

Property 2. *The set of maximal imperfectly sporadic itemsets with two thresholds is equal to the set of maximal closed imperfectly sporadic itemsets with two thresholds. Proof.* It is sufficient to prove that if X is a maximal imperfectly sporadic itemset with two thresholds then X is a closed itemset, i.e. X = h(X).

Firstly, we need to prove X is a maximal frequent itemset for minSup.

According to the Definition 1, it is obvious that X is a frequent itemset for minSup. Assume that X is not a maximal frequent itemset for minSup, then $\exists X'$ will be a frequent itemset for minSup so that $X \subset X'$. According to the Apriori property, $\text{sup}(X') \le \text{sup}(X) < \text{maxSup}$. Since X is an imperfectly sporadic itemset with two thresholds, so $\exists x \in X \subset X'$ so that $\text{sup}(x) \ge \text{maxSup}$. Hence, X' is an imperfectly sporadic itemset with two thresholds containing X. This contradicts the definition of X.

On the other hand, according to the property of the Galois connections [9] $X \subseteq h(X)$ and since sup(h(X)) = sup(X) ≥ minSup, so h(X) is a frequent itemset for minSup. Because X is a maximal frequent itemset for minSup we obtain X = h(X), this means X is a closed itemset ■

Remark 2. Assume that X is an imperfectly sporadic itemset with two thresholds, if X is a maximal frequent itemset for minSup then X is a maximal imperfectly sporadic itemset with two thresholds.

This remark is implied directly from the proof of the Property 2.

Property 3. *There is no difference between the association rules generated from imperfectly sporadic itemsets with two thresholds and ones generated from maximal imperfectly*

sporadic itemsets with two thresholds.

Proof. We only need to prove that each imperfectly sporadic rule with two thresholds can always be generated from a maximal imperfectly sporadic itemset with two thresholds.

Let $A \to B$ be such rule, then A∪B is an imperfectly sporadic itemset with two thresholds and $A \to B$ is an association rule for the minimum support minSup and the minimum confidence minConf. According to [1-2], $A \to B$ is generated from a maximal frequent itemset for minSup.

Without losing generality, we can assume that A ∪ B is a maximal frequent itemset for minSup, and we will prove that A∪B is a maximal imperfectly sporadic itemset with two thresholds.

If it is not the case then ∃C ⊃ A∪B, C is a maximal imperfectly sporadic itemset with two thresholds, hence it implies that minSup ≤ sup(C) < sup(A∪B) < maxSup and C is a maximal frequent itemset for minSup containing A∪B. This contradicts the above assumption about A∪B ■

The properties 2 and 3 above are bases for developing an algorithm in the section below.

III. The MCISI Algorithm

A. MCISI overview

According to the Definition 1, search space for mining imperfectly sporadic itemsets with two thresholds includes itemsets, which are created from the set of frequent items for maxSup in combination with themselves and with the set of infrequent items for maxSup but frequent for minSup. For mining imperfectly sporadic itemsets with two thresholds, we use four properties about itemset-tidset in CHARM algorithm [11] as follows:

Assume that I_1 x $g(I_1)$ is a node on a branch of a searching tree (g is the Galois connection), and it is to combine with another node I_2 x $g(I_2)$ on another branch of the same level in the tree then there are four following cases:

If $g(I_1) = g(I_2)$ then $g(I_1 \cup I_2) = g(I_1) \cap g(I_2)$. This property implies that we can replace every occurrence of I_1 by $I_1 \cup I_2$ and $g(I_1)$ is replaced by $g(I_1 \cup I_2)$. I_2 will be removed from father consideration.

If $g(I_1) \subset g(I_2)$ then $g(I_1 \cup I_2) = g(I_1) \cap g(I_2) = g(I_1) \neq g(I_2)$. Thus we can replace every occurrence of I_1 by $I_1 \cup I_2$, but since $g(I_1) \neq g(I_2)$ we can not remove I_2.

If $g(I_1) \supset g(I_2)$ then $g(I_1 \cup I_2) = g(I_1) \cap g(I_2) = g(I_2) \neq g(I_1)$. Thus we can replace every occurrence of I_2 by $I_1 \cup I_2$, and I_1 will be kept.

If $g(I_1) \neq g(I_2)$ then $g(I_1 \cup I_2) = g(I_1) \cap g(I_2) \neq g(I_2) \neq g(I_1)$. In this case, no itemset can be eliminated; both I_1 and I_2 lead to different closed itemsets.

MCISI algorithm will find closed imperfectly sporadic itemsets with two thresholds using the following process:

- It is started by creating two sets of items from the database: (1) The set of frequent items for maxSup; (2) The set of infrequent items for maxSup but frequent for minSup. All the items in these two sets are sorted (in order of support or order of lexicography).

- To combine each item in (1) with other items on the right hand of the item in (1) and all the items in (2) to create a search space. This space is the set of itemsets containing two items which have at least a frequent item for maxSup.

- Based on the search space, the MCISI algorithm will find closed imperfectly sporadic itemsets with two thresholds under the approach of the CHARM algorithm [11]. It performs a search over a novel Itemset-Tidset search space by removing all itemsets which are not imperfectly sporadic itemsets with two thresholds and are not closed, and by applying one of the four cases mentioned before for each (Itemset x Tidset).

B. MCISI algorithm

MCISI ALGORITHM (**D** ⊆ **I** x **O**, minSup, maxSup):

1) **FI** = {I_jx $g(I_j)$: $I_j \in$ **I** $\wedge \left| g(I_j) \right| \geq$ maxSup}
   //**FI** is the set of frequent items for maxSup
   //All the items in **FI** are sorted.
2) **IFI** = {K_j x $g(K_j)$: $K_j \in$ **I** $\wedge \left| g(K_j) \right| <$ maxSup $\wedge \left| g(K_j) \right| \geq$ minSup}
   //**IFI** is the set of items which are infrequent for maxSup and frequent for minSup
   //All the items in **IFI** are sorted.
3) For each I_j x $g(I_j)$ in **FI**
4) Nodes={P_j x $g(P_j)$: $P_j = I_j \cup M_j$, $g(P_j) = g(I_j) \cap g(M_j)$, ($M_j \in$ **FI**\{I_1,...,I_j} or $M_j \in$ **IFI**) $\wedge \left| g(Pj) \right| \geq$ minSup}
   //Combine I_j with the other items on the right hand of the item in **FI** and all the items in **IFI**.
5) MCISI-EXTEND(Nodes,**C**)
6) **CS** = **CS** ∪ **C**

MCISI-EXTEND(Nodes, **C**):

7) For each X_i x $g(X_i)$ in Nodes
8) NewN = ∅ and X = X_i
9) For each X_j x $g(X_j)$ in Nodes
10) X = X ∪ X_j and Y = $g(X_i) \cap g(X_j)$
11) CHARM-PROPERTY(Nodes, NewN)
12) If NewN ≠ ∅ then MCISI-EXTEND(NewN)
13) If sup(X) < maxSup
14) **C** = **C** ∪ X // if X is not subsumed

Here g is the Galois connection defined above. CHARM-PROPERTY function is built in [11].

C. The soundness and completeness of the MCISI algorithm

Proposition 1. MCISI algorithm is sound and complete.

The soundness of the MCISI algorithm.

The MCISI algorithm enumerates all closed imperfectly sporadic itemsets with two thresholds. The MCISI comprises three stages:

In the first stage (line 1, 2), two sets of items are created: **FI** is a set of frequent items for maxSup; **IFI** is a set of items which are infrequent for maxSup and frequent for minSup. All the items in **FI** and **IFI** are ordered.

In the second stage (line 3-5), each item in **FI** will be combined with the other items on the right hand of this item in **FI** and all the items in **IFI** to create a search space called Nodes. MCISI-EXTEND(Nodes,**C**) function is called in the new search space. This function will find closed frequent itemsets for minSup in Nodes by applying the approach of the

CHARM algorithm in [11]. The last itemsets in each branch of the search tree is the maximal closed itemsets. After that, this function will check the condition for removing all itemsets having support equal to or greater than maxSup (line 13). MCISI-EXTEND returns the set of itemsets called **C** containing closed itemsets whose support is equal to or greater than minSup and smaller than maxSup. Each itemset in **C** has at least a frequent item for maxSup. According to Definition 1, these itemsets are closed imperfectly sporadic itemsets with two thresholds.

Finally, line 6 combines all the sets found in the second stage. This is a set of all closed imperfectly sporadic itemsets with two thresholds.

The completeness of the MCISI algorithm

We need to show that every imperfectly sporadic rule with two thresholds is generated from a sporadic itemset found by this algorithm.

Obviously, according to the Property 3, an imperfectly sporadic rule with two thresholds is generated from a maximal imperfectly sporadic itemset with two thresholds and according to the Property 2, this itemset is a maximal closed imperfectly sporadic itemset with two thresholds and the MCISI algorithm find out such itemsets ■

D. Mining imperfectly sporadic rules

It is obvious that the support of an association rule from any transaction databases is always bigger than or equal to $1/|O|$, where $|O|$ is the number of transactions of the database. Therefore, given that minsup = $1/|O|$, the mining imperfectly sporadic rules with two thresholds will be the mining imperfectly sporadic rules. In other words, the MCISI algorithm for mining closed imperfectly sporadic itemsets with two thresholds can be applied for mining closed imperfectly sporadic itemsets in [4-5]. Hence the mining imperfectly sporadic rules has been absolutely solved.

Proposition 2. *Set of imperfectly sporadic rules with two thresholds is contained in the set of imperfectly sporadic rules.*

The proof of this proposition is quite simple therefore not mentioned here ■

So imperfectly sporadic rules with two thresholds are "rarer" in term of comparing with imperfectly sporadic rules.

IV. Experimental Results

In order to evaluate the performance of the MCISI algorithm such as times for finding out closed imperfectly sporadic itemsets with two thresholds from transaction databases we chose several synthetic and real databases. Real databases are available from [13]. All experiments were performed on a Lenovo-IBM Co dual 2.0ghz with 2GB of memory, running Windows Vista. The MCISI algorithm was coded in C++.

A. Experiment on synthetic database

The purpose of this experiment is to evaluate the performance of the MCISI algorithm over a large range of data characteristics. We generated synthetic databases based on the principle proposed by Agrawal R., and Srikant R. [1,2].

The synthetic databases simulate transactions in the retailing environment with some defined parameters. To generate the synthetic databases, we takes the following parameters: |D| is the number of transactions, |T| is the average size of the transactions, |L| is the number of frequent itemset, and I is the number of items. The first step, the size of the next transaction is picked from a Poisson distribution with the mean set to the average size of the transactions. Then we fill the transactions with items. Each transaction is assigned a series of potentially frequent itemsets. The complete detail can be found in [1,2]. Table 1 shows the characteristics of the synthetic databases.

TABLE 1. The characteristics of the Synthetic Databases

No	Database	# of Items (I)	# of Transactions (D)	The average size of the transaction (T)
1	T05I1000D10K	1 000	10 000	5
2	T10I1000D10K	1 000	10 000	10
3	T15I1000D10K	1 000	10 000	15
4	T20I1000D10K	1 000	10 000	20
5	T25I1000D10K	1 000	10 000	25
6	T30I1000D10K	1 000	10 000	30

Table 2 shows the performance of the MCISI algorithm for mining closed imperfectly sporadic itemsets with two thresholds in the synthetic databases with minSup and maxSup are appropriately chosen.

Experimental results show that with the same number of transactions and the same number of items in the synthetic databases, running time of the MCISI algorithm depends on an average size of transactions. The running time of MCISI will increase if average size of transactions in the synthetic databases increase. Experimental results also show the MCISI algorithm can be applied for large databases with acceptable time.

TABLE 2. Running time of the MCISI Algorithm in the Synthetic Databases

No	Database	minSup	maxSup	# of sporadic itemsets with two thresholds	Time (sec)
1	T05I1000D10K	0.005	0.05	0	0.122
2	T10I1000D10K	0.005	0.05	5	1.652
3	T15I1000D10K	0.005	0.05	211	14.396
4	T20I1000D10K	0.005	0.05	1841	52.020
5	T25I1000D10K	0.005	0.05	6715	142.087
6	T30I1000D10K	0.005	0.05	15593	315.711

TABLE 3. RUNNING TIME OF THE MCISI ALGORITHM FOR FIVE REAL DATABASES

Database	# of Items	# of Records	minSup	maxSup	# of sporadic itemsets with two thresholds	Time (sec)
Soybean	76	47	0.1	0.5	2987	0.452
Zoo	43	101	0.1	0.5	3125	0.515
Bridge	220	108	0.1	0.5	398	0.062
Teaching AE	104	151	0.1	0.5	5	0.027
Mushroom	118	8124	0.1	0.5	6365	279

TABLE 4. RUNNING TIME OF THE MCISI ALGORITHM FOR MINING IMPERFECTLY SPORADIC ITEMSETS IN REAL DATABASES

Database	# of Items	# of Records	minSup	maxSup	# of sporadic itemsets	Time (sec)
Soybean	76	47	1/47	0.5	8853	15.273
Zoo	43	101	1/101	0.5	5253	9.126
Bridge	220	108	1/108	0.5	1253	2.605
Teaching AE	104	151	1/151	0.5	7	0.34

B. Experiment on real database

We chose five real databases from the UCI Machine Learning Repository [13]. All the databases were converted to transaction databases. Table 3 also shows the characteristics of the databases and the result of the MCISI algorithm. Experimental results show that running time of the MCISI algorithm not only depends on number of items, numbers of transactions, and an average size of transactions but also depends on the number of closed imperfectly sporadic itemsets with two thresholds found from real databases.

We knew that when minsup= $1/|O|$, where $|O|$ is number of transactions of a database, the MCISI algorithm will find closed imperfectly sporadic itemsets from the database for imperfectly sporadic rules in [4-5]. Table 4 shows the performance of the MCISI algorithm for finding closed imperfectly sporadic itemsets with minSup chosen under this way.

Table 5, 6 show running time of the MCISI in the Mushroom database which has more itemsets and transactions in term of comparing with the four another real databases.

For the Table 5, minSup = 0.1 and is fixed, maxSup changes from 0.2 to 0.5.

TABLE 5. RUNNING TIME OF MCISI ALGORITHM IN THE MUSHROOM DATABASE WHEN CHANGING MAXSUP

minSup	maxSup	# of sporadic itemsets with two thresholds	Time (sec)
0.1	0.5	6365	279
0.1	0.4	6174	220
0.1	0.3	5717	181
0.1	0.2	4773	163

For the Table 6, maxSup = 0.5 and is fixed, minSup changes from 0.1 to 0.4.

TABLE 6. RUNNING TIME OF THE MCISI ALGORITHM IN THE MUSHROOM DATABASE WHEN CHANGING MINSUP

minSup	maxSup	# of sporadic itemsets with two thresholds	Time (sec)
0.1	0.5	6365	279
0.2	0.5	1367	138
0.3	0.5	440	61
0.4	0.5	106	27

Experimental results show that with the same database, running time of the MCISI algorithm depends on values of the two thresholds. When the minSup is fixed, maxSup changes from 0.2 to 0.5 (Table 5) the running time of MCISI increases, and the maxSup is fixed, minSup changes from 0.1 to 0.4 (Table 6) the running time of MCISI decreases.

For visualizing correlations between the minSup, maxSup, found closed imperfectly sporadic itemsets with two thresholds and running time of the MCISI algorithm for finding out these itemsets, the figures in Table 5, 6 are presented in following graphs:

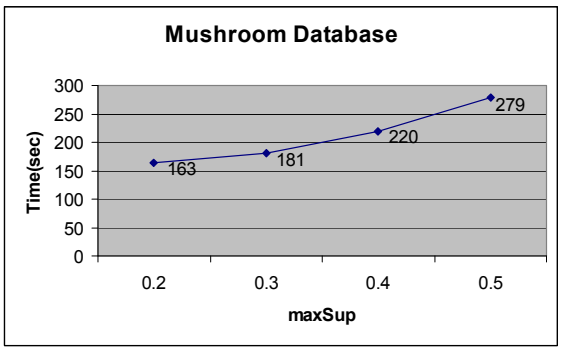

Figure 1. Running time of the MCISI algorithm in the Mushroom database with minSup = 0.1, maxSup changes from 0.2 to 0.5.

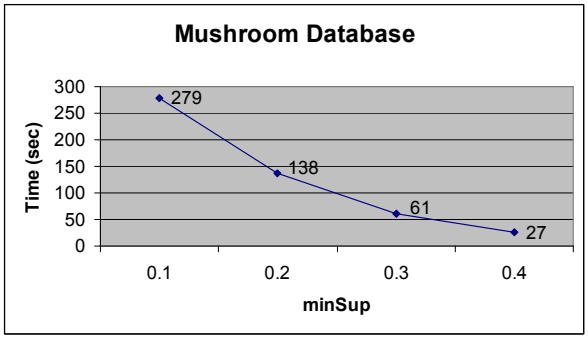

Figure 2. Running time of the MCISI algorithm in the Mushroom database with maxSup = 0.5, minSup changes from 0.1 to 0.4.

V. CONCLUSIONS

We have solved the problem of mining imperfectly sporadic rules in [3-5] by proposing the problem of mining imperfectly sporadic rules with two thresholds. The second threshold is added and considered a below minimum support

of imperfectly sporadic rules to make the problem more general.

Basing on the proved event that sets of sporadic rules generated from maximal imperfectly sporadic itemsets with two thresholds and ones generated from maximal closed imperfectly sporadic itemsets with two thresholds are the same, the MCISI algorithm was proposed to find closed imperfectly sporadic itemsets with two thresholds. This will enable the finding of many imperfectly sporadic rules which can not be done by other algorithms such as MIISR algorithm.

The MCISI algorithm will be the algorithm for finding imperfectly sporadic itemsets in [4-5] when the minSup is appropriately chosen. Therefore the problem of mining imperfectly sporadic rules has been completely solved.

We knew that algorithms developed under an approach based on a closed itemset lattice such as the CHAM, CLOSE algorithms [9,11,12], are more efficient than those developed under other approaches because of the former the search space for frequent itemsets in general is reduced [9, 11] and many redundant rules can be removed [11-12]. Thus the MCISI algorithm for finding closed imperfectly sporadic itemsets also promises to be more efficient than algorithms developed under other approaches for finding imperfectly sporadic itemsets.

Like frequent itemset mining, generating rare association rules from all rare itemsets will produce a very large set of association rules. Hence, a question for our future research is that we need to find ways to generate useful imperfectly sporadic rules with two thresholds from imperfectly sporadic itemsets with two thresholds.

REFERENCES

[1] Agrawal R., and Srikant R.: Fast Algorithms for Mining Association Rules. Proc. Very Large Database International Conference, Santiago, pp. 487–498, 1994.
[2] Agrawal R., Mannila H., Srikant R., Toivonen H., and Inkeri Verkamo A.: Fast Discovery of Association Rules. Advances in Knowledge Discovery and DataMining. The MIT Press, pp.307-328,1996.
[3] Koh Y. S., and Rountree N.: Finding Sporadic Rules Using Apriori-Inverse. PAKDD 2005, LNAI 3518, pp 97-106, 2005.
[4] Koh Y. S., and Rountree N.: Finding Interesting Imperfectly Sporadic Rules. PAKDD 2006, pp 473-482, 2006.
[5] Koh Y. S., Rountree N., and O'Keefe R. A.: Mining Interesting Imperfectly Sporadic Rules. Knowledge and Information System; 14(2), pp179-196, 2008.
[6] Koh Y. S., and Rountree N.: Rare Association Rule Mining via Transaction Clustering. The Seventh Australasian Data Mining Conference (AusDM 2008).
[7] Kiran R. U., and Reddy P. K.: An Improved Multiple Minimum Support Based Approach to Mine Rare Association Rules. http://www.iiit.net/techreports/2009_24.pdf
[8] Ling Zhou, and Stephen Yau: Association Rule and Quantitative Association Rule Mining among Infrequent Items. MDM'07, August 12, 2007, San Jose, California, USA.
[9] Pasquier N., Bastide Y., Taouil R., and Lakhal L.: Efficient Mining of Association Rules Using Closed Itemset Latics. Information Systems, Vol 24, No. 1, pp. 20-46, 1999.
[10] Szathmary L., Napoli A., and Valtchev P.: Towards Rare Itemset Mining. In Proceedings of the 19th IEEE international Conference on Tools with Artificial Intelligence - Vol.1 (ICTAI 2007) - Volume 01 (October 29 - 31, 2007). ICTAI. (pp. 305-312). Washington, DC: IEEE ComputerSociety
[11] Zaki M. J., and Hsiao C.: CHARM: An Efficient Algorithm for Closed Association Rule Mining. In Proceedings, SIAM-02 International Conference on Data Mining, 2002.
[12] Zaki M. J.: Mining Non-Redundant Association Rules. Data Mining and Knowledge Discovery, 9, 223–248, 2004.
[13] UCI-Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets.html