

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Granule Based Inter-transaction Association Rule Mining

Wanzhong Yang, Yuefeng Li and Yue Xu
*Queensland University of Technology
 Australia*

1. Introduction

Association rule mining extends data mining in a number of ways. Han et al. (Han & Kamber, 2006) summarized the features of association mining as single (Agraw et al., 1994) or multi-dimensional (Lee et al., 2006), multiple level (Han et al., 1999), quantitative (Ruckert et al., 2004), sequential pattern (Tzvetkov, P. et al., 2003), constraint based (Pei et al., 2001) etc. However, in line with requirements of knowledge discovered in the real applications, people also divide the association rule mining into intratransaction and intertransaction.

Intratransaction was the focus for most traditional approaches which include two stages of frequent itemset mining and rule generation. In frequent itemset mining, Apriori-like (Agraw et al., 1993) and FPT (Han et al., 2000) are two important methods applied in association rule mining. In a transaction database, intratransaction discusses the association of items in the same transaction. For example, if intratransaction association mining is applied in a market basket analysis, the high profit products could be associated with the low profit products in the same transaction.

Beyond intratransaction association mining, Lu et al. (2000) proposed intertransaction association mining which is the most widely used in some industry areas. Intertransaction association mining is used to discover patterns among different transactions. In the example of McDonalds, Burger King and KFC, the business association is along locations and also the time dimension as well.

Obviously, knowledge discovery in intertransaction association mining is more complicated than intratransaction association mining. To solve intertransaction association mining, Lu et al. (2000) proposed the EH-Apriori and E-Apriori algorithms. However, both algorithms take a significant amount of time to generate large extended itemsets.

Tung et al. (2003) proposed the FITI (First Intratransaction Then Intertransaction) algorithm which is adaptive to intertransaction association mining. It focuses on the two stages of mining frequent intertransaction itemsets and rule generation. The experimental results show this method is better for real world applications when compared to the high running time of EH-Apriori.

However, finding frequent intertransaction itemsets is still a time consuming process in intertransaction association mining. The FITI algorithm can show the efficiency when the transaction length is very short. If there are many items in each transaction and the threshold is low, the discovery process in frequent intertransaction itemsets could take a very long time. This algorithm considers the intervals of the intertransaction, but does not

include the average size of transactions. The extended database produces many unneeded combinations of items. The number of extended itemsets is much larger than the number of the set of items. It is difficult to apply this technique in high dimensional data.

It is a challenging issue to find the association among high dimensional data in industry. To breakthrough the classical methods in mining association rules, Li et al. (2006) proposed multi-tier granule mining for intratransaction association rules. Granule mining is a new initiative that attempts to improve the quality of discovered knowledge in multidimensional databases. The basic idea of granule mining came from decision tables presented by Pawlak (2002), where the attributes are divided by users into two groups: condition attributes and decision attributes; and a granule is a predicate that describes the common feature of a group of objects (transactions). In intratransaction association mining, granule mining has been proved to reduce the meaningless association rules significantly.

In this paper, we propose the method of granule based intertransaction association rule mining. It takes advantage of granule mining's capacity to transfer intertransaction association into a multi-tier structure and simplify the process of the long pattern in intertransaction association.

The remainder of the paper is structured as follows. We begin by introducing classical intertransaction association rule mining in Section 2. In Section 3, we introduce the concept of granule mining. In Section 4, we present granule based intertransaction association rule mining and use the multi-tier structure to deal with the process. In Section 5, we propose the concept of *precision* to evaluate the effectiveness of association rules and introduce the experimental results. The related work is in Section 6 and last Section contains the conclusion.

2. Intertransaction association mining

2.1 Intertransaction association mining

Let $T = \{t_1, t_2, \dots, t_n\}$ be a transaction database, and each transaction is a set of items. Tung et al. (2003) used the sliding window and extended-items to describe the intertransaction. Each sliding window W can be viewed as a continuous ω (a fixed interval called *maxspan*, or *sliding_window_length*) sub-windows such that each sub-window contains only one transaction. Let e_i be an item, its occurrences in different transactions in a sliding window can be extended from $e_i(0)$ to $e_i(\omega)$, where $0, \dots, \omega$ are positions of transactions in the window. The transactions in a sliding window W can be merged into a mega transaction (or extended transaction) by putting all of W 's extended items in a collection. Hence, an inter itemset refers to a set of extended-items, and an inter association rule can be represented as $X \rightarrow Y$, where X and Y are both a set of extended-items and $X \cap Y = \emptyset$.

The definition of the support and confidence in inter association mining follows up the intra association mining. Let N be the number of megatransactions and, X and Y both be a set of extended-items and $X \cap Y = \emptyset$. Let T_{xy} be the set of megatransactions that contains X and also Y , and T_x be the set of megatransactions that contains X . We have

$$\text{sup}(X \rightarrow Y) = |T_{xy}| / N, \text{ conf}(X \rightarrow Y) = |T_{xy}| / |T_x|.$$

2.2 FITI algorithm

FITI algorithm is the state of the art in intertransaction association rule mining, it consists of three stages. For example 1 (in Table 1), we choose the same example used by Tung et al. But we assign the real meaning for each variable in ASX share market for bank and insurance. In

this example, there are only four transactions. Let a, b, c represent bank shares and d, e, f, g, h, i be insurance shares.

Stage I of FITI algorithm offers a data structure FILT to store frequent intra transaction items. Using FILT, stage II of FITI represents transaction database by FIT tables. Stage I and stage II is data preparation where FITI represents the database by an embedded data structure. We also can view the data preparation as the generation of an extended database. Stage III of FITI is data processing which generates frequent intertransaction itemsets. In this stage FITI controls the implementation using the algorithms.

d	E
100	a, b, c
104	a, b, c, d, e
105	a, f, g
109	e, h

Table 1. A database with four transactions

In stage I, the data structure, called Frequent-itemsets Linked Table (FILT) is used to store frequent intratransaction itemsets. This stage is based on the property that a frequent intertransaction itemset must be a frequent intratransaction itemset. To generate frequent intratransaction itemset, the Apriori algorithm is applied for the item selection. The basic structure of FILT for Table 1 is lookup links, where the frequent itemsets are sorted from one-itemset to n -itemset and stored in Table 2. This table can be extended to links in different directions. In table 2 a, b, c not only can be generator of $\{a, b\}$, $\{a, c\}$ and $\{b, c\}$, but also can be the subset of $\{a, b\}$, $\{a, c\}$ and $\{b, c\}$.

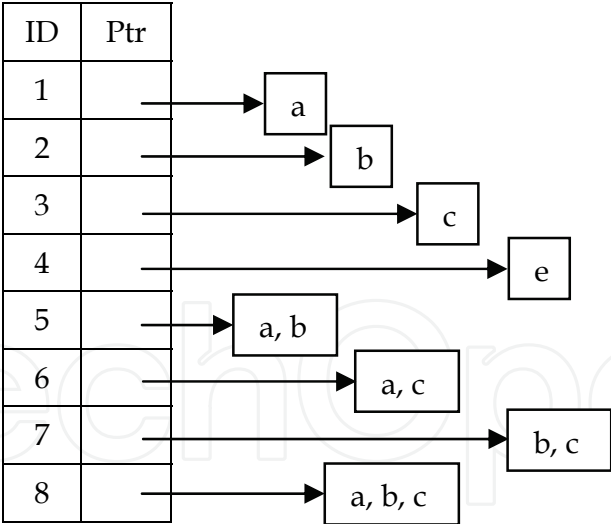


Table 2. Lookup links

Stage II of FITI is database transformation, where the database is represented by a set of *encoded Frequent-Itemset Tables* (**FIT** tables). A feature of FIT tables is that they represent each transaction by the ID of frequent itemsets in FILT. In particular, the FIT tables (in Table 3) can represent each transaction by the IDs from one-itemsets to n -itemsets separately.

Phase I and Phase II can be viewed as data preparation for frequent intertransaction itemsets mining. This process is mining frequent intratransaction itemsets and represents them by using encoded IDs sorted from one itemsets to n -itemsets.

F_1		F_2		F_3	
di	$IDset_i$	di	$IDset_i$	di	$IDset_i$
100	1, 2, 3	100		100	
104	1, 2, 3, 4	104	5, 6, 7	104	8
105	1	105	5, 6, 7	105	8
109	4	109		109	

Table 3. FIT tables

Stage III of FITI is data processing where FITI uses the algorithm to control intertransaction association and output a set of frequent intertransaction itemsets. The algorithm is based on the Apriori principle.

The input layer of the algorithm is the ID encoded intertransaction itemset I , which is corresponding to each sliding window. For the itemsets in each sub window, only one top ID from the FIT tables can be selected into I if it is available. Otherwise, it is zero.

During the data processing, the algorithm divides the generation of frequent intertransaction itemset I into two cases of $k = 2$ and $k > 2$ in order to control the implementation. When $k= 2$, FITI generates frequent intertransaction 2-itemsets, L_2 . FITI makes use of the hashing approach in previous research and refines the hash formula for the bucket number. When $k >2$, FITI uses the loop to generate the candidates of frequent intertransaction itemsets.

```
While ( $L_{k-1} \neq \emptyset$ )
{
    Generate candidate intertransaction itemsets,  $C_k$ ;
    Scan transformed database to update the count for  $C_k$ ;
    Let  $L_k = \{c \in C_k \mid support(c) \geq mins\ up\}$ ;
     $k++$ ;
}
```

The algorithm inside the loop is based on the classic Apriori principle which generates the itemsets based on the joins. However FITI separates the joins into intertransaction join and cross transaction join.

d	E
100	a
101	e, h
102	a, b
103	b, d
104	a
105	f, i
106	a, c
107	b, g

Table 4. A part of transaction database

W1	a(0) , b(1) f(1), a(2) d(2), g(3)
W2	a(0) h(0), b(1), i(2), e(3)

Table 5. Sliding windows

I	1, 0, 5, 2
J	1, 0, 6, 2

Table 6. Intertransaction joins

We use example 2 (in Table 4) to describe the intertransaction join in FITI. Table 4 is a part of the transaction database where the FIT tables are the same as Table 3. The sliding window size is four. So there are two sliding windows *W1* and *W2* for Table 4. The intertransaction itemsets in *W1* and *W2* are listed in Table 5. Their encoded IDs in the itemset of *I* and *J* are mapped from FIT tables listed in Table 6. According to the rule of intertransaction join, because 5 and 6 are in the same column of *I* and *J* in Table 6, and other items in *I* and *J* are same, the inter- transaction itemset is {1, 0, 8, 2}.

<i>d</i>	<i>E</i>
100	a
101	b f
102	a, d
103	g
104	a, h
105	b,
106	i
107	e

Table 7. A part of transaction database

W1	a(0) , e(1) h(1), a(2) b(2), b(3)d(3)
W2	a(0), f(1) i(1), a(2) c(2), b(3) g(3)

Table 8. Sliding windows

I	1, 2, 1, 0
J	1, 2, 0, 4

Table 9. Cross transaction join

We use example 3 (in Table 7) to describe cross transaction join in FITI. Table 7 is a part of the transaction database where the FIT tables are the same as Table 3. The sliding window size is four. So there are two sliding windows *W1* and *W2*. The cross transaction itemsets in *W1* and *W2* are listed in Table 8. Their encoded IDs in the itemset of *I* and *J* are mapped from FIT tables listed in Table 9. According to the rules of cross transaction join, because {1, 0} and {0, 4} are in the last two columns of *I* and *J* in Table 8, and the values in the other columns are same. The intertransaction itemset is {1, 2, 1, 4}.
The next step in the algorithm is to calculate the count of each itemset and sort the itemset by the frequency. Following the Apriori algorithm, the process prunes off the itemsets which are not frequent and generates all frequent itemsets.

2.3 The restriction of FITI algorithm

The advantage of the FITI algorithm is is that it provides a complete set of frequent itemsets. However, the weakness is obvious as well. The FITI algorithm generates many redundant

itemsets. In particular in industry, those itemsets are useless. We use an example from the share market to illustrate this. Also, the large amount of calculation is time consuming.

FITI is based on the Apriori algorithm, which is used twice in mining inter- transaction association rules, including both generation of frequent intratransaction itemsets and generation of frequent intertransaction itemsets.

The advantage of Apriori is the output of a complete set of itemsets. The weakness is the complexity of the calculation. Therefore, intertransaction association is very complex. In particular when applying in high dimensional database, Apriori not only means a large amount of calculation for item joins, but also loss of the common feature in industry.

In the example in Table 1, each transaction has different length, where the number of items could be various. But in the share market, most important shares all have values on each transaction. We view all transactions as same length.

The FITI algorithm consists of three phases. The evaluation of the efficiency and effectiveness includes the three stages. We use the same example of bank and insurance in ASX. The example in Table 10 is a transaction database in ASX. There are over 250 working days in ASX each year and we view each working day as one transaction. Let a, b, c be the major banking shares and d, e, f, g, h, i represent the major insurance shares.

ID	a b c	d e f g h i
100	1 0 1	1 0 0 0 1 1
101	-1 0 1	1 0 1 -1 0 -1
102	0 -1 0	-1 0 1 0 0 -1
103	1 1 0	0 1 -1 0 -1 0

Table 10. Transaction database

In phase 1, FITI stores the intratransaction itemsets in the data structure FILT. In the ASX share market, each share has three statuses, booming, steady, dropping, where let 1, 0, -1 represent them separately. Let $a1, a2, a3$ represent the three status for a . With the same idea, we convert the transaction database into Table 11.

ID	a b c	d e f g h i
100	a1 b2 c1	d1 e2 f2 g2 h1 i1
101	a3 b2 c1	d1 e2 f2 g3 h2 i3
102	a2 b3 c2	d3 e2 f1 g2 h2 i3
103	a1 b1 c2	d2 e1 f3 g2 h3 i2

Table 11. Transformed database.

In FILT, if the transactions database is for one year, the max number of itemsets could be $3^9 = 19683$, where the length of FILT is hundreds times of the length of ASX transaction table. If let N be the length of transaction table, the length of FILT could be $O(N^2)$. Apparently many of itemsets in FILT are redundant for rule generation, which means a large amount of calculations are ineffective. This data structure expands the complexity of the algorithm and causes an expensive cost in industry.

The phase II of FITI is database transformation, which transforms the database into a set of FIT tables. FIT tables consist of one-item sets, two-item sets N -item sets FIT tables. Let N be the number of the FIT tables. If the transaction table is for one year, FIT tables could be from F_1 to F_{27} . If the status of each share becomes multiple choices in industry, the number

of FIT tables could be N times of the width of transaction database. Therefore, the input of the algorithm is a very large table set.

Moreover, the complexity extends into Phase III. The generation of candidate frequent intertransaction itemsets is based on the regulation of transaction joins. As an additional cost, we have to design many regulations for various joins. However, in Apriori algorithm we need to prune off the non-frequent itemsets. It causes a large amount of time.

To generate a complete set of itemsets, the length of FILT in the phase I and the width of FIT tables in phase II of FITI algorithm are N times of length and width of the transaction table separately. Therefore, the data structure become $O(N^2)$ times of the transaction database. We have to generate a large amount of meaningless itemsets. If we apply FITI algorithm in the high dimensional data set, the complexity is over reasonable estimation and the cost are so expensive. Apparently, it is not reasonable for the application in industry.

3. Granule mining

3.1 Decision tables and granules

In the multidimensional database, Pawlak proposed the decision tables in rough set theory to represent the association rules from the hidden patterns (Pawlak, Z., 2002) (Pawlak, Z., 2003). A feature of decision tables is related to user constraints, which divide the attributes of a database into condition attributes and decision attributes, respectively. We call the tuple (T, V^T, C, D) a *decision table* of (T, V^T) if $C \cap D = \emptyset$ and $C \cup D \subseteq V^T$, T is a set of transactions, and V^T is the set of attributes (items). The condition attributes C represent the premise (antecedent) of association rules, while the decision attributes D can be interpreted as the post-condition (consequent) of association rules.

In a decision table, there is a function for every attribute $a \in V^T$ such that $a: T \rightarrow V_a$, where V_a is the set of all values of a . We call V_a the domain of a . C (or D) determines a binary relation $I(C)$ (or $I(D)$) on T such that $(t_1, t_2) \in I(C)$ if and only if $a(t_1) = a(t_2)$ for every $a \in C$, where $a(t)$ denotes the value of attribute a for object $t \in T$. It is easy to prove that $I(C)$ is an equivalence relation, and the family of all equivalence classes of $I(C)$, that is a partition determined by C , is denoted by T/C .

The classes in T/C (or T/D) are referred to C -granules (or D -granules). The class which contains t is called C -granule induced by t , and is denoted by $C(t)$.

Table 12 simulates a part of the daily transactions for product sales in a shop, which is a multidimensional database. There are 200 transactions for 7 different products in the

Granule	Department	Commodity	Profit (%)	N _g
1	F & V	Accessories	Over 70	47
2	Bakery	General Merch	30-40	12
3	Bakery	Glassware	20-30	48
4	Soft Drinks	Dinners Frozen	Over 70	12
5	F & V	Accessories	30-40	21
6	Bakery	General Merch	20-30	40
7	Soft Drinks	Dinners Frozen	20-30	20

Table 12. A decision table

database. The possible attributes are *department*, *commodity*, *cost*, *price*, *profit*. The users choose only three attributes and let $C = \{department, commodity\}$ and $D = \{profit\}$. We compress the database into a decision table, where each product is viewed as a granule and N_g is the number of transactions that belong to the granule. Using Table 12 we can classify the condition granules (*C-granules*) as $T/C = \{\{1,5\}, \{2,6\}, \{3, \{4,7\}\}$ and decision granules (*D-granule*) as $T/D = \{\{1,4\}, \{2,5\}, \{3,6,7\}\}$, respectively.

Time	Products
t ₁ : 02/12/2003	a ₁ , a ₅ , a ₆ , a ₇
t ₂ : 02/01/2004	a ₁ , a ₅ , a ₆ , a ₇
t ₃ : 02/02/2004	a ₁ , a ₂ , a ₄ , a ₅ , a ₇
t ₄ : 02/03/2004	a ₁ , a ₂ , a ₃ , a ₅ , a ₆
t ₅ : 02/04/2004	a ₁ , a ₂ , a ₄ , a ₅ , a ₆
t ₆ : 02/05/2004	a ₁ , a ₂ , a ₃ , a ₅ , a ₆
t ₇ : 02/06/2004	a ₁ , a ₄ , a ₅ , a ₇

Table 13. A time slice transaction table

Granule	a ₁	a ₂	a ₃	a ₄	a ₅
cg ₁	1	0	0	0	1
cg ₂	1	1	0	1	1
cg ₃	1	1	1	0	1
cg ₄	1	0	0	1	1

Granule	a ₆	a ₇
dg ₁	1	1
dg ₂	0	1
dg ₃	1	0

(a) C-granules

(b) D-granules

Granule	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	N _g
g ₁	1	0	0	0	1	1	1	2
g ₂	1	1	0	1	1	0	1	1
g ₃	1	1	0	1	1	1	0	1
g ₄	1	1	1	0	1	1	0	0
g ₅	1	0	0	1	1	0	1	2

(c) Decision Table

Table 14. Granules

We also can view the transactions in Table 13, where the transactions come from the different time slices during 7 months and all products are frequent. Let $V^T = \{a_1, a_2, \dots, a_7\}$, $T = \{t_1, t_2, \dots, t_7\}$. According to the profit, *bananas Cavendish* (a_1), *coca cola 2LT* (a_2), *1.25 LT* (a_3), *chicken pieces* (a_4) and *potatoes brushed* (a_5) are all high profit products; *bread white* (a_6) and *sandwich* (a_7) are both low profit products. We set up the user constraint with the profit and classify the products into two groups. Let a_1, a_2, a_3, a_4, a_5 be condition attributes and a_6, a_7 decision attributes. Table 14 (a) is the *C-granules*; Table 14 (b) is the *D-granules*; Table 14 (c) is

the decision table, where $T/C \cup D = \{g_1, g_2, g_3, g_4, g_5\}$ and N_g is the number of objects in the same granule.

In the representation of association rule mining, we view a decision table as a multidimensional database. Every granule in the decision table can be mapped into a decision rule (Pawlak, 2002). The condition attribute can be viewed as the premise of association rules; the decision attributes can be viewed as the post-conditions. The presence or absence of items is viewed as the same position. Therefore, we can obtain 5 decision rules in Table 14 (c), and the first one can be read as the following decision rule:

$$(a_1=1) \wedge (a_2=0) \wedge (a_3=0) \wedge (a_4=0) \wedge (a_5=1) \rightarrow (a_6=1) \wedge (a_7=1)$$

or in short $C(g_1) \rightarrow D(g_1)$ (or $C(t_1) \rightarrow D(t_1)$), where \wedge means “and”.

From the above examples, we can now interpret association rules based on granules rather than patterns. In particular, we can view the association rules based on different granularities of multidimensional databases according to what users want.

3.2 Data mining and granule mining

Decision tables only provide a straightforward way to represent association rules. They only cover some kinds of larger patterns, but avoid many of the frequent patterns.

As the representation of association rule mining, we need to understand the difference between the patterns used in the decision tables and the association rules. To interpret this puzzle, we present the concept of decision patterns. Therefore, we need to define a series of concepts for illustrating the decision patterns.

Definition 1. A set of items X is referred to as an *itemset* if $X \subseteq V^T$. Let X be an itemset, where $[X]$ denotes the *covering set* of X , which includes all objects t such that $X \subseteq t$, i.e., $[X] = \{t \mid t \in T, X \subseteq t\}$.

Given an *itemset* X , its occurrence frequency is the number of objects that contain the *itemset*, that is $|[X]|$; and its support is $|[X]| / |T|$. An itemset X is called a *frequent pattern* if its support $\geq min_sup$ is a minimum support.

Definition 2. Given a set of objects Y , its itemset which satisfies

$$itemset(Y) = \{a \mid a \in V^T, t \in Y \Rightarrow a \in t\}.$$

Given a frequent pattern X , its *Closure*

$$Closure(X) = itemset([X]).$$

From the above definitions, we have the following theorem (Zaki, 2004).

Theorem 1. Let X and Y be frequent patterns. We have

$Closure(X) \supseteq X$ for all frequent patterns X ;	(1)
$X \subseteq Y \Rightarrow Closure(X) \subseteq Closure(Y)$.	(2)

Definition 3. A frequent pattern X is *closed* if and only if $X = Closure(X)$.

Given a *C-granule* $cg = C(t)$, its *covering set* $[cg] = \{t' \mid t' \in T, (t', t) \in I(C)\}$. Let cg be a *C-granule* and dg be a *D-granule*, we define $[cg \wedge dg] = [cg] \cap [dg]$. For example, in Table 14 $g_1 = \{(a_1=1) \wedge (a_2=0) \wedge (a_3=0) \wedge (a_4=0) \wedge (a_5=1) \wedge (a_6=1) \wedge (a_7=1)\} = C(g_1) \wedge D(g_1) = cg_1 \wedge dg_1$; therefore

$[g_1] = [cg_1 \wedge dg_1] = [cg_1] \cap [dg_3] = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\} \cap \{t_1, t_2\} = \{t_1, t_2\}.$

Table 15 illustrates the covering sets of granules, where (a) includes the covering sets of *C-granules*, (b) includes the covering sets of *D-granules*, and (c) includes the covering sets of *C∪D-granules*.

Theorem 2. Let (T, V^T, C, D) be a *decision table*. We have

$[C(t)] \supseteq [C\cup D(t)]$, for all $t \in T$.	(1)
The derived decision pattern of every granule $g \in T/C\cup D$ is a closed pattern.	(2)

Granule	a ₁	a ₂	a ₃	a ₄	a ₅	covering set
cg ₁	1	0	0	0	1	{t ₁ , t ₂ , t ₃ , t ₄ , t ₅ , t ₆ , t ₇ }
cg ₂	1	1	0	1	1	{ t ₃ , t ₅ }
cg ₃	1	1	1	0	1	{ t ₄ , t ₆ }
cg ₄	1	0	0	1	1	{ t ₇ }

(a) *C-granules*

Granule	a ₆	a ₇	covering set
dg ₁	1	1	{ t ₁ , t ₂ }
dg ₂	0	1	{t ₃ , t ₇ }
dg ₃	1	0	{t ₄ , t ₅ , t ₆ }

(b) *D-granules*

Granule	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	N _g	Covering set
g ₁	1	0	0	0	1	1	1	2	{ t ₁ , t ₂ }
g ₂	1	1	0	1	1	0	1	1	{ t ₃ }
g ₃	1	1	0	1	1	1	0	1	{ t ₅ }
g ₄	1	1	1	0	1	1	0	0	{t ₄ , t ₆ }
g ₅	1	0	0	1	1	0	1	2	{ t ₇ }

(c) *Decision Table*

Table 15. Covering set of *C∪D-granules*

Proof: (1) is obvious in accordance with the definition of closure.
For (2), Let X be the derived pattern of g , that is, $X=\{a_i \in C\cup D \mid a_i(g) = 1\}$. From the definition of the granules, we know there is an object $t_0 \in [g]$ such that $X = \{a_i \in C\cup D \mid a_i(t_0) = 1\}$, that is $t_0 \in [X]$.
Given an item $a \in itemset([X])$, according to Definition 2 we have $a \in t$ for all $t \in [X]$, that is, $a \in t_0$ and also $a \in X$. Therefore, $Closure(X) = itemset([X]) \subseteq X$.
We also have $X \subseteq Closure(X)$ from Theorem 1, and hence we have $X = Closure(X)$.

4. Granule based intertransaction association rule mining

Formally a transaction database can be described as an information table $(\mathcal{D}, V^{\mathcal{D}})$, where \mathcal{D} is a set of objects in which each object is a sequences of items, and $V^{\mathcal{D}} = \{a_1, a_2, \dots, a_n\}$ is a set of selected items (or called attributes) for all objects in \mathcal{D} .

Decision tables are efficient for dealing with multiple dimensional databases in line with user constraints. Formally, users may use some attributes of a database; and they can divide these attributes into two target groups: condition attributes and decision attributes, respectively. We call the tuple $(\mathcal{D}, V^{\mathcal{D}}, C, D)$ a decision table of $(\mathcal{D}, V^{\mathcal{D}})$ if $C \cap D = \emptyset$ and $C \cup D \subseteq V^{\mathcal{D}}$. The classes in \mathcal{D} / C (or \mathcal{D} / D) are referred to C -granules (or D -granule).

For example, in the share market, a transaction contains different shares at the same day. To reduce the risk of investments, share-market experts usually consider a group of shares rather one or two shares based on the current performance of another group of shares. To help such investments, we can group shares into different industry categories. For instance, we may choose two industries: bank and insurance.

The mining process has three sub stages.

1. Transform the transaction database into the form of a decision table;
2. Generate C -granules and D -granules based user selected two industry categories;
3. Generate inter association rules between C -granules and D -granules.

The original transaction database records the data of ASX share transactions along the date dimension. The data includes attributes like high, low, open and close, which represent the price status in a day. To keep up the monotonic property, we assume the transactions are continuous and all records are complete filled. The empty records are instead of null value.

Since the mining object is transferred from the item to the group, a sliding window not only considers an interval (*sliding_window_length*), but also the number of attributes (we call *sliding_window_width*).

When transforming the transaction database to the decision table $(\mathcal{D}, V^{\mathcal{D}}, C, D)$, let the banking shares be condition attributes C and the insurance shares be decision attributes D .

We can use the normal way for dealing with C -granules. We use the technique of sliding windows to generate D -granules, where *sliding_window_width* = $|D|$. Let \mathcal{D} be all the transactions and V_a refers to the profit gain of all shares in each transaction. V_a includes three statuses: increased, neutral and loss, represented by 1, 0 and -1.

In Figure 1 there are three bank shares a, b, c as condition attributes that represent *Westpac* bank, *ANZ* bank and *National* bank separately. Let a_i, b_i, c_i be the profit gain of bank shares on day i . The decision attributes d, e, f, g, h represent insurance shares *PMN*, *IAG*, *AMP*, *QBE*, *AXA*, where d_i, e_i, f_i, g_i, h_i refer to the profit gain of insurance shares on day i . The sliding windows only contains decision attributes, and the *sliding_window_width* = 5 and *sliding_window_length* = 3. The interval of the transactions decides the block of transactions in the sliding window, which would be used to generate D -granules for a same C -granule.

To describe the inter associations between condition granules and decision granules, we can extend the normal decision table into an extended decision table such that each condition granule is linked to all possible sub-windows in sliding windows. For example, Table 16 illustrates an extended decision table when we let *sliding_window_length* = 2.

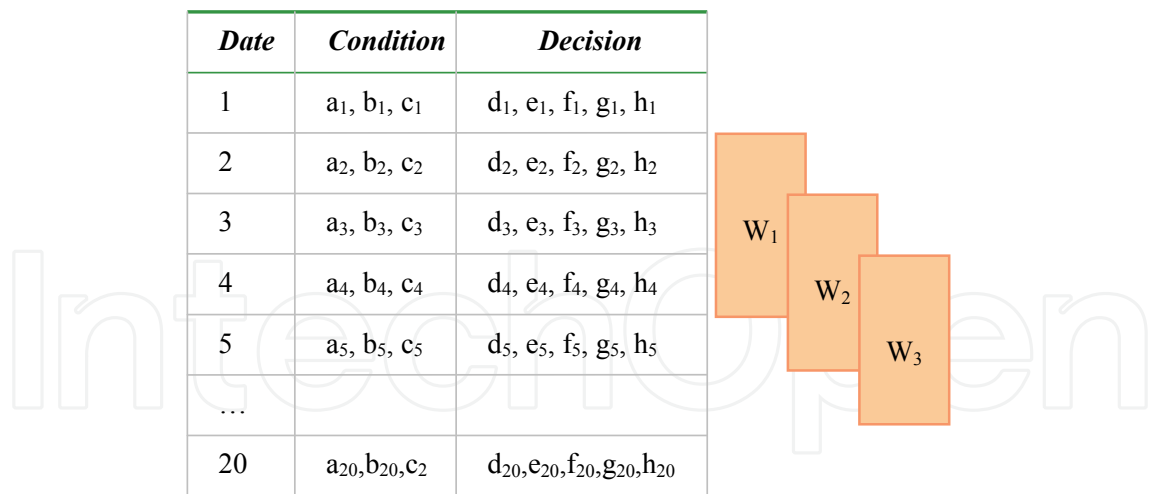


Figure 1. A decision table with sliding windows

ID	Condition	Decision
1	a_1, b_1, c_1	d_2, e_2, f_2, g_2, h_2
2	a_1, b_1, c_1	d_3, e_3, f_3, g_3, h_3
3	a_2, b_2, c_2	d_3, e_3, f_3, g_3, h_3
...		
39	a_{20}, b_{20}, c_{20}	$d_{21}, e_{21}, f_{21}, g_{21}, h_{21}$
40	a_{20}, b_{20}, c_{20}	$d_{22}, e_{22}, f_{22}, g_{22}, h_{22}$

Table 16. An extended decision table with maxspan = 2

The data compression is along the vertical direction in the extended decision table. Let \mathcal{D}/C be the set of C -granules that refer to all classes of the profit situations for three bank shares. Let \mathcal{D}/D be the set of D -granules that refer to all classes of the profit situations for five insurance shares. The inter association rule mining can be represented by mining granules now.

It is hard to clearly understand the intertransaction associations between condition granules and decision granules because of many duplicates. For this purpose we would like to represent the extended decision table as a 2-tier structure. The first tier contains all condition granules, the second tier contains decision granules and the intertransaction associations are the links.

For the above example, people concern the gain of the group of shares, not only single share. Therefore, we can use a simple SUM measure to denote the gain information of a group of shares, where $SUM > 0$ means positive gain, $SUM < 0$ means negative gain and $SUM = 0$ means no-gain.

Figure 2 depicts an example of a 2-tier structure, where we have seven condition granules that describe the possible changes of three bank shares; and have only three decision granules that describe the possible gains of buying five insurance shares after 1 or 2 days based on the changes of the three bank shares.

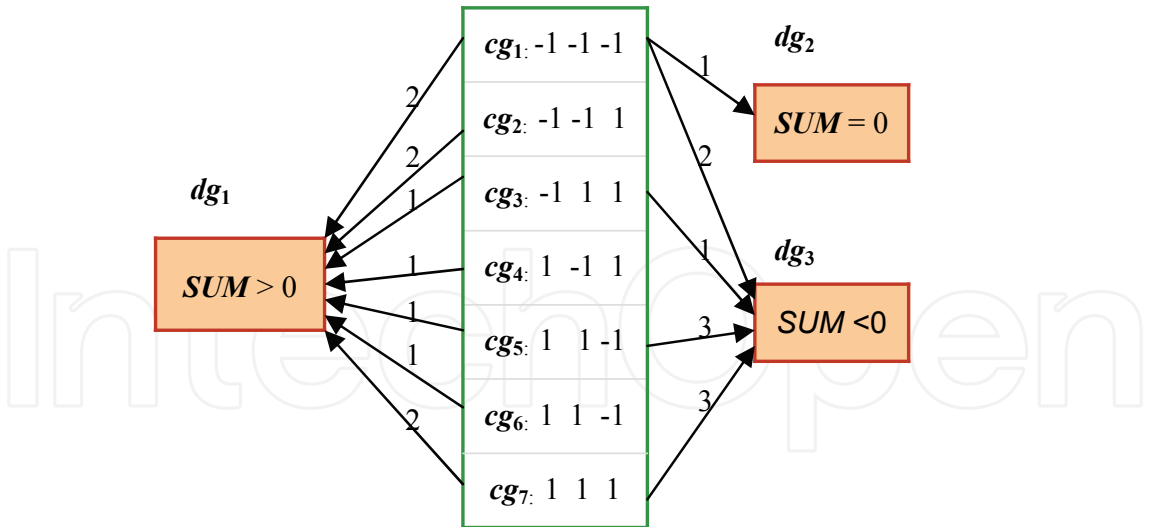


Figure 2. The association of C granules and D granules

Formally, a set of items X is referred to as an itemset if $X \subseteq V^{\mathcal{D}}$. Let X be a itemset, we use $[X]$ to denote the covering set of X , including all objects d such that $X \subseteq d$, i.e., $[X] = \{d \mid d \in D, X \subseteq d\}$.

Let $\mathcal{D}/C = \{cg_1, cg_2, \dots, cg_m\}$ and $\mathcal{D}/D = \{dg_1, dg_2, dg_3\}$. The decision rules in Figure 2 can be illustrated as follows:

$$cg_x \rightarrow dg_z$$
$$conf = |[cg_x \wedge dg_z]| / |cg_x|$$
$$support = |[cg_x \wedge dg_z]| / N$$

In Figure 2, there are twelve associations. If we set up the $min_sup = 2$, we have the following six inter association rules:

$$cg_1 \rightarrow dg_1 (conf = 2/5) \quad cg_1 \rightarrow dg_3 (conf = 2/5)$$
$$cg_2 \rightarrow dg_1 (conf = 2/2) \quad cg_5 \rightarrow dg_3 (conf = 3/4)$$
$$cg_7 \rightarrow dg_1 (conf = 2/5) \quad cg_7 \rightarrow dg_3 (conf = 3/5)$$

5. Experiments

5.1 Basic experiments

In the ASX share market, there are 26 industries and almost 2000 companies. We take the ASX data of four industries from January 2005 to January 2007. We divide the data into two sections: a training set and a testing set. The first section contains over 260,000 transactions in 2005. The second section includes over 340,000 transactions in the other.

We choose two pairs of industries for the experiments: bank vs. insurance and food beverage & tobacco vs. retailing. In each pair, according to the yearly share volumes, we select the top three shares of one industry as condition granules and the top five products of another industry as decision granules.

Table 17 describes some samples for the first pair of industries in 2005 and the interval is one day. There are 15 condition granules. In the second pair of industries, there are 23 condition granules. The constraint-based decision granules are decided base on $SUM > 0$, $SUM = 0$ and $SUM < 0$. We choose three intervals for inter association mining. The intervals are one day, two days and three days.

ID	B1	B2	B3	SUM>0	SUM=0	SUM<0
1	-1	-1	-1	27	6	27
2	-1	-1	0	5	0	0
3	-1	-1	1	12	1	11
...						
15	1	1	1	33	5	29

Table 17. Bank vs. insurance in 2005

5.2 Precision

When applying the inter association rule in the real data, we propose *Precision* as the criterion to evaluate the effectiveness of inter association rules.

In share market, investors should be interested in the prosperous shares where $SUM \geq 0$. Let $cg_x \rightarrow dg_z$ be an inter association rule discovered in training phase and $SUM(dg_z) > 0$, a positive gain.

Let S_{fst} be the number of transactions in the testing set that match cg_x . Let S'_{snd} be the number of dg_z with $SUM(dg_z) \geq 0$ that match cg_x , and S_{snd} be the number of dg_z with $SUM(dg_z) > 0$ that match cg_x .

We define P_N as *Non_ Negative_ Precision* where

$$P_N(cg_x \rightarrow dg_z) = (S'_{snd} / S_{fst}) * 100\% .$$

We also define P_P as *Positive_ Precision* where

$$P_P(cg_x \rightarrow dg_z) = (S_{snd} / S_{fst}) * 100\% .$$

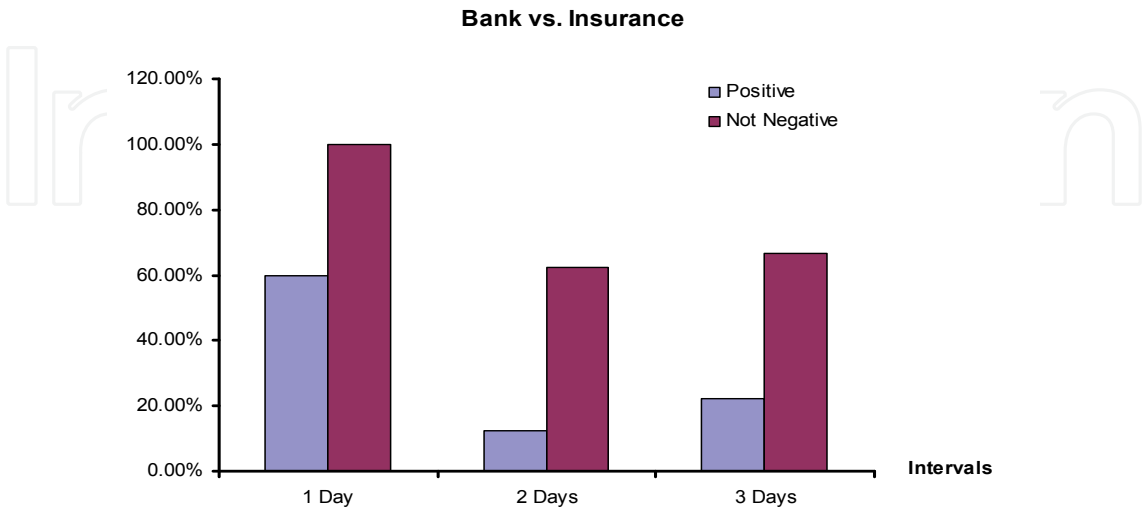


Figure 3. Precision for bank vs. insurance

In Figure 3 the pair is bank and insurance. All *Non_Negative_Precisions* are between 60% and 100%. All *Positive_Precisions* are greater than 10%. When the interval is one day, the positive percentage reaches 60%.

5.3 Efficiency

Compared to the FITI algorithm, granule-based inter association mining makes long pattern mining possible and easier. In the FITI algorithm, the max frequent patterns of eight items in Figure 1 listed in Figure 4 $N_P = 28 = 256$. It expands the scope of the user requirement and generates many extra items. In the basic experiments, each pair includes eight different frequent items. In both pairs of industries, the minimum numbers of association rules are 15 and 23 separately; the maximum numbers of association rules are 45 and 69 separately. Our method obviously reduces the time and looks more efficient and applicable in the above example.

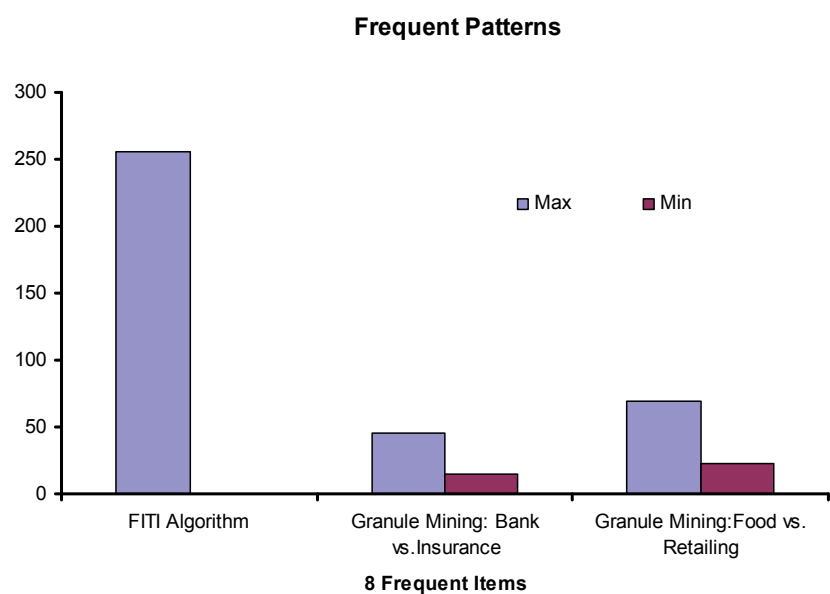


Figure 4. Frequent Patterns

In addition, the granule based approach has advantage in common feature instead of FITI based on item association for decision rule generation. It avoids of generation of the extended database and the joins in Apriori-like algorithms. As knowledge representation, granule based approach simplifies the complexity of the algorithms and keeps the major benefit of intertransaction association mining, which is easy to understand and use. In particular, it is more practical and meaningful in industry. However, the weakness is the loss of the completeness of item association.

For the future research, we need to develop granule based intertransaction association rule mining in two aspects. At first we will keep studying how to apply granule based intertransaction association rule in high dimensional data. Secondly we need to consider how to use multi-tier structure to improve the quality of rules during the rule discovery. Also the data scope of the experiments will expand to various share products running for long term. Moreover, the design idea can extend to different fields in industry.

6. Related work

Granule based intertransaction association rule mining is involved in intratransaction association rules and granule mining. The related work is for both issues.

Most current researchers endeavor to use the existing efficient algorithms for mining intratransaction association rules. Apriori like algorithm (Agraw et al., 1993) and FP-tree algorithm (Han et al., 2000) are two foundation methods in this field. To apply association rule mining in industry, the association mining scope expands from single dimensional association to multidimensional association, even extending to the multilevel.

However, to satisfy complex requirements in industry, intertransaction association rule mining looks attractive. Lu et al. (2000) first presented the concept of intertransaction association rule mining and contributed E-Apriori and EH-Apriori algorithms. The performance of these algorithms suffered when dealing with real data in industry. To speed up the above process, Feng et al. (2002) presented a template model that includes several optimization techniques, i.e., joining, converging.

Moreover, Tung et al. (2003) proposed the FITI algorithm to overcome the shortcoming in previous methods. Also FITI turns to be a brilliant milestone in intertransaction association rule mining. FITI algorithm offers the data structure FILT to store frequent intratransaction itemsets and transfer database into FIT tables as input for data processing, where FITI algorithm generates frequent intertransaction itemsets by joins. Mining both intra and inter frequent itemsets are all based on Apriori algorithm. The nature of FITI algorithm seems an extension of Apriori like idea in interaction association. Because of the complexity of the interaction association, the disadvantage of Apriori causes many extra itemsets during the joins. It is difficult to cope with the long patterns in the intertransaction.

To improve the efficiency of intertransaction association, few methods recently are proposed. In intertransaction frequent closed itemsets algorithm (IFCIA) (Dong et al. 2007), the basic design follows up FITI algorithm. The contribution is the closed itemsets, which are applied in mining process in order to avoid of the extended database. But it is still in the frame of Apriori like scope. MMIT is the interaction based on matrix mining (Zhang et al., 2007). In the algorithm design, MMIT is different from FIT in two aspects. First, MMIT directly moves into mining and sorting of intertransaction itemsets at the first step. It avoids of intratransaction itemsets mining and Apriori like idea. Secondly, MMIT uses matrix for mining frequent intertransaction itemsets. However, the experiments seem not enough to prove this method.

Granule mining originally is from rough set theory (Pawlak, 1982). The rough set theory can be used to describe the knowledge in information tables (Guan et al. 2003; Li et al. 2003). Further, rough sets based decision tables presented by Pawlak (2002) can be used to represent some sorts of association rules. Li and Zhong (2003) presented a structure to disconnect the condition granules and decision granules in order to improve the efficiency of generating association rules from decision tables.

To cope with a multiple dimensional transaction database with current algorithms, rough set theory becomes more abstractive in association rule mining. Pawlak (2002) proposed the decision table and divided the transaction table into the condition attribute and the decision attribute. Li et al. (2003) presented a new algorithm to modify Pawlak's method and improved the efficiency. Both algorithms can make use of the advantage of rough sets, which can find minimal sets of data and generate minimal sets of association rules (Pawlak, 1996). However, they are only suitable for a low dimensional system. It does not offer a

solution for the decomposition of a high dimensional database. We also need to reduce dimensionality for a large database.

Li et al. (2006) presented a multi-tier structure for granule mining to represent multidimensional intratransaction association rules. It breakthroughs traditional methods in association rule mining. One feature of this approach is focusing on the association among the granules. The multi-tier structure can improve the quality of association rule mining and reduce attributes for a large database. Also this method can be applied in data processing in data warehouse (Yang et al., 2008).

7. Conclusion

In this chapter, we present granule based inter association mining to reduce the complexity of intertransaction association rule mining. To compare with other methods, our method can reduce the width of sliding windows. It uses granules to replace extended item sets. Thus, we do not need to consider too many combinations of extended items. We also propose the concept of precision in order to evaluate the effectiveness of intertransaction association rule mining. The experiments show that the proposed method is promising.

8. References

- Agraw, R., Imielinski, T. & Swami, A. (1993). Mining association rules between sets of items in large database, *Proceedings of ACM-SIGMOD*, pp. 207-216, Montreal, Canada, 1993
- Agraw, R., Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases, *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 487-499, 1994
- Dong, J. & Han, M. (2007). IFCIA: An Efficient Algorithm for Mining Intertransaction Frequent Closed Itemsets, *Proceedings of fourth international conference on fuzzy systems and knowledge discovery*, pp. 678-682, Haikou, China, 2007
- Feng, L., Yu, J. X., Lu, H. & Han, J. (2002). A template model for multidimensional inter-transactional association rules, *The International Journal on Very Large Data Bases*, 11(2), (2002) pp. 153 -175
- Han, J. & Fu, Y. (1999). Mining multiple-level association rules in large databases, *IEEE Transaction on Knowledge and Data Engineering*, Vol. 11, No. 5, pp. 798-805, 1999
- Han, J. & Kamber, M. (2006). *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers
- Han, J., Pei, J. & Yin, Y. (2000). Mining frequent patterns without candidate generation, *Proceedings of the 2000 ACM SIGMOD international conference on Management of data table of contents*, pp. 1-12, Texas, United States, 2000
- Lee, A. J. T., Lin, W. & Wang, C. (2006). Mining association rules with multi-dimensional constraints, *The Journal of Systems and Software*, pp. 79-92, 2006
- Li, Y. & Zhong, N. (2003). Interpretations of association rules by granular computing, *Proceedings of 3rd IEEE International Conference on Data Mining*, pp. 593-596, USA, 2003
- Li, Y., Yang, W. & Xu, Y. (2006). Multi-Tier Granule Mining for Representations of Multidimensional Association Rules, *Proceedings of 6th IEEE International Conference on Data Mining*, pp. 953-958, Hong Kong, 2006.

- Lu, H., Han, J. & Feng, L. (2000). Beyond intratransaction association analysis: mining multidimensional intertransaction association rules, *ACM Transactions on Information Systems*, 18(4), (2000) pp.423 – 454
- Pawlak, Z. (1982) Rough Sets, *International Journal of Computer and Information Science*, Vol.11, No.5, (1982), pp. 341-356
- Pawlak, Z. (1996). Rough sets and data analysis, *Proceedings of IEEE AFSS*, pp. 1-6, Kenting, Taiwan, 1996
- Pawlak, Z. (2002). In pursuit of patterns in data reasoning from data, the rough set way, *Proceedings of 3rd International Conference on Rough Sets and Current Trends in Computing*, pp. 1-12, USA, 2002
- Pawlak, Z. (2003). Flow graphs and decision algorithms, *Proceedings of 9th International Conference on Rough Set, Fuzzy Sets, Data Mining and Granular Computing*, pp. 1-10, Chongqing, China, 2003
- Pei, J., Han, J. & Lakshmanan, L.V.S. (2001). Mining frequent itemsets with convertible constraints, *Proceedings of 17th International Conference on Data Engineering*, pp. 433-442, Heidelberg, Germany, 2001
- Ruckert, U., Richter, L. & Kramer, S. (2004). Quantitative association rules based on half-spaces: an optimization approach, *Proceedings of fourth IEEE International Conference on Data Mining*, pp. 507 – 510, Brighton, UK, 2004
- Tung, A.K.H., Lu, H., Han, J. & Feng, L. (2003). Efficient mining of intertransaction association rules, *IEEE Transactions on Knowledge and Data Engineering*, Vol.15, No.1, (2003), pp.43-56
- Tzvetkov, P., Yan, X. & Han, J. (2003). TSP: mining top-K closed sequential patterns, *Proceedings of 3rd IEEE International Conference on Data Mining*, pp. 347-354, Urbana, IL, USA, 2003
- Yang, W., Li, Y., Wu, J. & Xu, Y., Granule Mining Oriented Data Warehousing Model for Representations of Multidimensional Association Rules, *International Journal of Intelligent Information and Database Systems*, Vol.2, No.1, (2008), pp. 125-145 2008.
- Yang, W., Li, Y. & Xu, Y. (2007). Granule Based Intertransaction Association Rule Mining, *Proceedings of 19th IEEE International Conference on Tools with Artificial Intelligence*, Vol.1, pp. 337-340, Patras, Greece, 2007
- Zhang, Z., Wang, H. & Huang, G. (2007). A New Algorithm Based on Matrix for Mining Inter-Transaction Association Rules, *International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 6717-6720, Shanghai, China, 2007



Tools in Artificial Intelligence

Edited by Paula Fritzsche

ISBN 978-953-7619-03-9

Hard cover, 488 pages

Publisher InTech

Published online 01, August, 2008

Published in print edition August, 2008

This book offers in 27 chapters a collection of all the technical aspects of specifying, developing, and evaluating the theoretical underpinnings and applied mechanisms of AI tools. Topics covered include neural networks, fuzzy controls, decision trees, rule-based systems, data mining, genetic algorithm and agent systems, among many others. The goal of this book is to show some potential applications and give a partial picture of the current state-of-the-art of AI. Also, it is useful to inspire some future research ideas by identifying potential research directions. It is dedicated to students, researchers and practitioners in this area or in related fields.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Wanzhong Yang, Yuefeng Li and Yue Xu (2008). Granule Based Inter-transaction Association Rule Mining, Tools in Artificial Intelligence, Paula Fritzsche (Ed.), ISBN: 978-953-7619-03-9, InTech, Available from: http://www.intechopen.com/books/tools_in_artificial_intelligence/granule_based_inter-transaction_association_rule_mining

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen