Study on Algorithm of Multidimensional Sets Sequential Patterns Mining Based on Identification of Position

Yan Liu[1, a], Xiaowei Qin[1,b] and Yan Zou[1,c]

[1] College of Computer and Information Engineering, Chifeng University, China

[a]cfxyly5962@163.com, [b]qxw3409@163.com, [c] zouyankb@163.com

Keywords: multidimensional sets; support; sequential patterns mining

Abstract. Introduce the concepts related to multidimensional set sequence database, and in order to mine a special multidimensional set sequence database, in which all the sequences have the same length, items inside each dimension are ordered, and dimensions are unordered, we propose an algorithm of multidimensional sets sequential patterns mining. This algorithm is on the basis of the multidimensional sets sequential patterns mining based on bitmap, first uses the representation of multidimensional sets sequential database based on identification of position to mine the single frequent itemsets, and then uses the bitmap representation to mine the multidimensional frequent sequences. In each mining, the algorithm only records frequent itemsets of single dimensional element and corresponding frequent sequences, which can reduce memory overhead, improve the counting efficiency of support and mining speed.

Introduction

Data mining has been widely used in various fields, and the mined data patterns have been extended from single data sets insde dimesions to multiple data sets among dimensions, it is so called multidimensioanl mining[1][2]. The system construction of college students' innovation ability training and the research on related factor indicate that[3] the cultivation of creativity is related with many factors, such as education system, education process, family environment, personal hobbies, enterprise culture, government advocacy and social environment. Each element contains several attributes, such as education process includes the accumulation of knowledge, the cultivation of creative thinking, innovation practice, innovation achievement display etc. Each attribute contains some different items, such as innovation practice includes experimental innovation activities, innovation activities of science and technology, social practice, innovation practice base etc. Each item also has different values. These elements compose a multidimensional transaction database. How to mine the elements closely related to college students' innovation ability development from this database, and to use in the education process of students' innovation ability in order to improve the students' innovation ability, is not only related to personal development of students, but also give an important scientific support to the cultivation of application and creative ability of students. Based on this requirement, we have studied on multidimensional sets of sequential patterns mining.

Concepts Related to Multidimensional Sets[4][5]

Single dimensional set: The set of items denoted by $I = \{ i_1 , i_2 , \ldots, i_m \}(m \geq 0)$, in which i_j ($1 \leq j \leq m$) is called item, I sreprents single dimensional set on attribute set E whose value domain is V, and $i_j \in V$. When $m = 0$, I is empty set; When $m = 1$, I is called the single dimensional set with single item; When $m \geq 1$, I is called real single dimensional set.

The length of single dimensional set: The number of items included in single dimensional set.

Multidimensional set: Suppose $A = \{E_1 , \ldots, E_{dim}\}$ $(dim \geq 1)$ is any attribute set, is the single dimensional set on any attribute set, dim is the number of dimension, the value domain of each attribute is $V_1, \ldots ,V_{dim}$ respectively, the multidimensional set $t = \{s_1, \ldots, s_{dim}\}$ $(dim \geq 1)$, in which s_i $(1 \leq i \leq dim)$ is the single dimensional set on attribute E_{dim}, i.e. $s_i \subseteq V_i$, then t is called multidimensional set or *dim*-dimensional set on attribute set *A*. If s_i is composed of the single dimensional set with single item, then we call it multidimensional set with single item, else we call it real multidimensional set. The number of single dimensioanl sets contained in multidimensional set is called the length of multidimensional set.

Contain: Suppose *t* and *s* are two multidimensional sets on attribute set *A*, for any attribute $E_{dim} \in A$ ($dim \geq 1$), t_i is the multidimesional set of *t* on attribute E_{dim}, s_i is the multidimesional set of *s* on attribute E_{dim}, the clusion relation $s_i \subseteq t_i$ are both ture. So we call multidimensional set *t* contains multidimensional *s* denoted by $t \geq s$.

Multidimensional sets sequential patterns: Suppose *D* is a *dim*-dimensional set on $A = \{E_1, \ldots, E_{dim}\}$ ($dim \geq 1$), *B* and *C* are attribute sets on *A*, $B \subset A$ and $C \subset A$ and $B \cap C = \varnothing$. Suppose *r* is a real multidimensional set on attribute set *B*, *t* is a real multidimensional set on attribute set *C*. Given a support threshold ξ, if the number of multidimensional set *s* that contains both *r* and *t* in *D* is larger than ξ, then call $r \Rightarrow t$ is multidimensional sets association rules based on attribute relations *B* and *C*.

Multidimensional Sets Database

According to the above definition of multidimensional sets, we give the multidimensional sets databse to be mined as shown in Table 1. Each itemset represents a single dimensional set on each attribute, each sequence represents multidimensional set on attribute sets. To meet the need of the analysis of elements related to the innovation ability of college students, we call the attributes in this database elements E_{dim} that represent the elements related to the innovation ability of college students. Each sequence with the same dimension number *dim* is composed of single dimensional sets. Items are ordered inside each dimension, but not ordered among dimesions. Items in different dimensions represent different meaning, this is different from definiton of traditional transaction database.

Transform the database in Table 1 into the form in Table 2, there are 3 sequences that represent 3 multidimensional sets, the length of each sequence is 3 that represents three itemsets, each itemset represents one single dimensional set on element E_{dim}. In each multidimensional set, the order of itemsets can not be changed, e.g. (*b c a*) and (*a b c*) are two different itemsets in this paper and represent different meaning. The item *a* of the single dimensional set on E_1 is different from the item *a* in E_2 and E_3, this is just to simplify the representation and no effect on mining result.

Table 1 Multidimensional sets database

Sid	**Sequence**
1	< (*b c a*) (*e a*) (*b a c*) >
2	< (*a b d*) (*e c a*) (*a b*) >
3	< (*b c d a*) (*d e b*) (*a c b*) >

Table 2 Multidimensional sets sequence database

Sid	***Itemsets of Sequence on E_{dim}***		
	dim=1	***dim=2***	***dim=3***
1	(*b c a*)	(*e a*)	(*b a c*)
2	(*a b d*)	(*e c a*)	(*a b*)
3	(*b c d a*)	(*d e b*)	(*a c b*)

The Algorithm of Multidimensional Sets Sequential Patterns Mining Based on Identification of Position

The idea of the algorithm. On the basis of existing sequential patterns mining algorithm [6], to meet the need of new data model, we propose an algorithm of multidimensional sequential patterns mining based on positional identification. This algorithm uses database based on identification of position to mine frequent itemsets that can identify the order among items accurately, and uses bitmap representation to mine frequent sequences that improves counting efficiency. In each mining, the algorithm only records frequent itemsets of one dimensional element and corresponding association rules, which can reduce memory overhead. The idea of the algorithm is described as follows:

(1) Generate the sequence database based on the identification of position *Pos_D*.

(2) On element E_{dim}, mine the frequent itmesets FI_{k_dim} first, and then mine the corresponding frequent sequence FS_{dim}.

(3) Mine frequent itemsets FI_{k_dim}. Generate candidate itemsets CI_{k_dim} through sequential patterns tree, and then count the support using database based on position identification to generate the frequent sequence FI_{k_dim} on E_{dim}, and record the frequent itemsets in table Pos_FI_{dim}.

(4) Mine frequent sequence FS_{dim}. Generate candidate sequences CS_{dim} through sequential patterns tree constructed by the frequent sequence $FS_{dim\text{-}1}$ of the dimension before current dimension element and frequent itemsets FI_{k_dim} of current dimension element, and count the support by bitmap representation to generate the multidimensional frequent sequence FS_{dim}, and then record the frequent sequences in table Bit_FS_{dim}.

Description of algorithm. Input: Multidemisional sets sequence dababase based on identification of position *Pos_D*, *min_support*

Output: Multidemisional sets frequent sequential pattern FS_{dim}

① for each dimension element E_{dim}

② FI_{1_dim} = generate (*Pos_D*) ; // FI_{1_dim} is frequent 1-itemsets on the *dim*th dimesion element E_{dim}

③ for ($k = 2$; $FI_{k\text{-}1\_dim} \neq \varnothing$; k + +) // generate frequent k-itemsets FI_{k_dim} on E_{dim}

④ CI_{k_dim} = generate_*CI* ($FI_{k\text{-}1\_dim}$) ; // CI_{k_dim} is candidate k-itemsets on E_{dim}

⑤ FI_{k_dim} = generate_frequent_*k*itemsets (CI_{k_dim} , Pos_FI_{dim}) ; //Pos_FI_{dim} is frequent table on E_{dim}

⑥ if (dim = = 1) // generate frequent 1-sequences FS_1

⑦ $FS_1 = FI_{k\_dim}$; // FS_1 is composed of frequent k-itemsets on the 1st demision element E_1

⑧ else if ($FS_{dim\text{-}1} \neq \varnothing$ && $FI_{k\_dim} \neq \varnothing$)//generate multidemisioanl frequent sequential patterns FS_{dim}

⑨ CS_{dim} = generate_*CS* ($FS_{dim\text{-}1}$, FI_{k_dim}) ; // CS_{dim} is the candidate *dim*-sequences

⑩ FS_{dim} = generate_frequent_sequences(CS_{dim}, Pos_FI_{dim}, Bit_FS_{dim});

⑪ return FS_{dim} ; // FS_{dim} is frequent *dim*-sequences

The Description of Algorithm Flow

This paper takes multidimensional sets sequence database in Table 2 as example to describe the algorithm flow, and supposes the minimum support count is *min_support* = 2.

Generate multidimensional sequence database based on identification of position *Pos_D*. Save the sequence database of Table 2 into *Pos_D* shown in Table 3. Record the occurrence position of each item on each dimension element in *Pos_D*. For example, the 1st element E_1 of sequence 1 is the itemset (*b c a*), in which the occurrence position of item *a* is 3, so noted as *Pos*(*a*) = 3, the occurrence position of item *b* is 1, so noted as *Pos*(*b*) = 1, the occurrence position of item *c* is 2, so noted as *Pos*(*c*) = 2, while item *d* doesn't occur, so noted as *Pos*(*d*) = 0.

Generate frequent *k*-itemsets FI_{k_dim} on each dimension element E_{dim}.

Generate frequent *k*-itemsets FI_{k_1} on the 1st dimension element E_1.

Generate frequent 1-itemsets FI_{1_1} on E_1. (1) Take $dim = 1$, $Sid = 1$, scan the item *a* in database *Pos_D* longitudinally. If the value of *Pos(a)* is not 0, add 1 to support(*a*). (2) Add 1 to *Sid*, continue to perform the above operation until *Sid* = 3. If support(*a*) ≥ *min_support*, then (*a*) $\in FI_{1\_1}$, and record in Pos_FI_{dim} as shown in Table 4; else (*a*) $\notin FI_{1\_1}$. (3) Continue to scan next item *b* unitl to *d*, all of the frequent 1-itemsets FI_{1_1} on 1st dimension element E_1 are generated, which are (*a*), (*b*), (*c*), (*d*) as shown in Table 4.

Table 3 Multidimensional sets sequence database based on identification of position *Pos_D*

Sid	***Pos* (E_{dim})**											
	***dim* = 1**				***dim* = 2**				***dim* = 3**			
	a	***b***	***c***	***d***	***a***	***b***	***c***	***d***	***e***	***a***	***b***	***c***
1	3	1	2	0	2	0	0	0	1	2	1	3
2	1	2	0	3	3	0	2	0	1	1	2	0
3	4	1	2	3	0	3	0	1	2	1	3	2

Generate candidate 2-itemsets CI_{2_1} on E_1. Construct the sequential patterns tree[8], *root* is the root node. Its subnodes formed the first layer nodeds are the first items of CI_{2_1} composed of all the frequent 1-itemsets on 1st dimension element. The subnodes of first layer nodes composed of their sibling nodes, formed the second layer nodes, and are the second items of CI_{2_1}. For example, in Fig. 1, the father node of (*a*) is *root*, the subnodes of *root* are (*a*), (*b*), (*c*), (*d*), so the subnodes of (*a*) are its sibling nodes (*b*), (*c*), (*d*). And the candidate 2-itemsets CI_{2_1} are (*a b*), (*a c*), (*a d*), (*b a*), (*b c*), (*b d*), (*c a*), (*c b*), (*c d*), (*d a*), (*d b*), (*d c*).

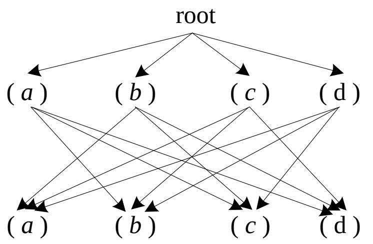

Fig.1 Candidate 2_itemsets CI_{2_1} on E_1

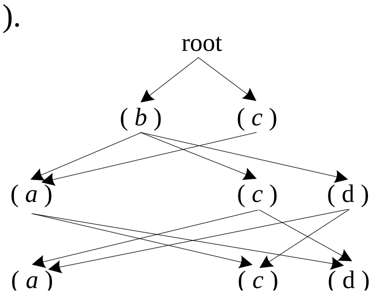

Fig.2 Candidate 3_itemsets CI_{3_1} on E_1

Generate frequent 2-itemsets FI_{2_1} on E_1. (1) Take one candidate 2-itemset CI_{2_1}, such as (*b a*). Take *dim* = 1, *Sid* = 1, scan (*b*) and (*a*) of FI_{1_1} in Table 4. If $Pos(a) \neq 0$, $Pos(b) \neq 0$, $Pos(a) > Pos(b)$ are all true, which indicate that (*a*) and (*b*) both occur, and (*a*) is located behind (*b*), so add 1 to support(*ab*), and record *Pos*(*b a*) = *Pos*(*a*); else record *Pos*(*b a*) = 0. (2) Increase *Sid* until *Sid* = 3. If support(*a b*) ≥ *min_support*, then (*b a*) ∈ FI_{2_1}, record (*a b*) in table Pos_FI_{dim}, as shown in Table 4; else (*b a*) ∉ FI_{2_1}. (3) Continue to take next candidate 2-itemset CI_{2_1} until to the end, and then generate all the frequent 2-itemsets FI_{2_1} on E_1, which are (*b a*), (*b c*), (*b d*) and (*c a*).

Table 4 Frequent itemsets table on the *dim*th element E_{dim}—Pos_FI_{dim}

Sid	FI_{k_dim} (*dim* = 1)								
	FI_{1_1} (*k* = 1)				FI_{2_1} (*k* = 2)				FI_{3_1}(*k* = 3)
	(*a*)	(*b*)	(*c*)	(*d*)	(*b a*)	(*b c*)	(*b d*)	(*c a*)	(*b c a*)
1	3	1	2	0	3	2	0	3	3
2	1	2	0	3	0	0	3	0	0
3	4	1	2	3	4	2	3	4	4

Sid	FI_{k_dim} (*dim* = 2)			FI_{k_dim} (*dim* = 3)				
	FI_{1_2} (*k* = 1)		FI_{2_2} (*k* = 2)	FI_{1_3} (*k* = 1)			FI_{2_3} (*k* = 2)	
	(*a*)	(*e*)	(*e a*)	(*a*)	(*b*)	(*c*)	(*a b*)	(*a c*)
1	2	1	2	2	1	3	0	3
2	3	1	3	1	2	0	2	0
3	0	2	0	1	3	2	3	2

Note: Pos_FI_{dim} only records frequent *k*-itemsets on one dimension element at one time, here lists records for three times.

Generate candidate 3-itemsets CI_{3_1} on E_1. In sequential patterns tree, the first layer nodes are the first items of frequent 2-itemsets FI_{2_1}, the second layer nodes are the second items of frequent 2-itemsets FI_{2_1}, the third layer nodes are composed of the sibling nodes of the second layer nodes. For example, in Fig. 2, the father node of (*a*) is (*b*) whose subnodes are (*a*), (*c*) and (*d*), so the subnodes of (*a*) are its sibling nodes (*c*) and (*d*). And the candidate 3-itemsets CI_{3_1} are (*b a c*), (*b a d*), (*b c a*), (*b c d*), (*b d a*), (*b d* c), (*c a d*).

Generate frequent 3-itemsets FI_{3_1} on E_1. (1) Take one candidate 3-itemset CI_{3_1}, e.g. (*b c a*). Take *dim* = 1, *Sid* = 1, scan (*b c*) and (*a*) in Table 4. If $Pos(bc) \neq 0$, $Pos(a) \neq 0$, $Pos(a) > Pos(bc)$ are all true, which indicate (*a*) and (*b c*) both occur, and (*a*) is located behind (*b c*), then add 1 to support(*bca*), and record *Pos*(*bca*) = *Pos*(*a*); else record *Pos*(*bca*) = 0. (2) Increase *Sid* until *Sid* = 3. If support(*bca*) ≥ *min_support*, then (*b c a*) ∈ FI_{3_1}, record (*b c a*) in table Pos_FI_{dim} as shown in Table 4; else (*b c a*) ∉ FI_{3_1}. (3) Continue to take next candidate 3-itemset CI_{3_1} until to the end, then generate all the frequent 3-itemsets CI_{3_1} on E_1, which is (*b c a*).

Generate candidate *k*-itemsets CI_{k_1} on E_1. Construct sequential patterns tree, the (*k*-1)th items of frequent (*k*-1)-itemsets FI_{k-1_1} compose the (*k*-1)th layer nodes, whose sibling nodes compose the subnodes of it. Use generate_*C* (FI_{k-1_dim}) to generate candidate *k*-itemsets CI_{k_1}.

Generate frequent k-itemsets FI_{k_1} on E_1. (1) Take one candidate *k*-itemset CI_{k_1}, take *dim* = 1, *Sid* = 1, scan the top *k*-1 item and *k*th item of CI_{k_1} in Table 4. If both values are not 0, and the value of *k*th item is greater than top *k*-1 item, then add 1 to support(CI_{k_1}), and record the value of $Pos(CI_{k\_1})$ the same as the value of *k*th item; else record the value of $Pos(CI_{k\_1})$ as 0. (2) Increase *Sid* until *Sid* = 3. If support(CI_{k_1}) ≥ *min_support*, then $CI_{k\_1} \in FI_{k\_1}$, record CI_{k_1} in table Pos_FI_{dim}, as shown in table 4; else $CI_{k\_1} \notin FI_{k\_1}$. (3) Continue to take next candidate *k*-itemset CI_{k_1}, until to the end, then generate all the frequent *k*-itemsets FI_{k_1} on E_1.

Generate frequent *k*-itemsets FI_{k_dim} on E_{dim}. Add 1 to *dim*, the process of algorithm is the same as section 2.1, until to the last dimension element. Record the frequent *k*-itemsets FI_{k_dim} on the *dim*th dimension element in table FI_{k_dim}.

Generate multidimensional frequent sequences FS_{dim}.

Generate frequent 1-sequences FS_1. If *dim* = 1, frequent 1-sequences FS_1 is composed of the frequent *k*-itemsets FI_{k_1} on E_1, then save FI_{k_1} of Table 4 into frequent sequence table Bit_FS_{dim} in the form of bitmap, as shown in Table 5. The value is not 0 in Table 4 will be recorded as 1 in Table5 which represents occurrence, and the value is 0 will be still recorded as 0 which represents not occurrence. For example, in Table 4, if *Sid* = 1, *Pos*(*a*) = 3, then record *Bit*(*a*) = 1 in Table 5. And then , clear table Pos_FI_{dim} in order to record the frequent *k*-itemsets on next dimension element.

Generate candidate 2-sequences CS_2. If *dim* = 2, construct sequential patterns tree, the root node is *root*. Its subnodes formed the first layer nodes are the first itemsets of CS_2 composed of all the frequent 1-sequences FS_1. The subnodes of first layer nodes formed the second layer nodes are the second itemsets of CS_2, which are composed of all the frequent *k*-itemsets FI_{k_2} on the 2nd dimension element E_2 in Table 4. As shown in Fig. 3.

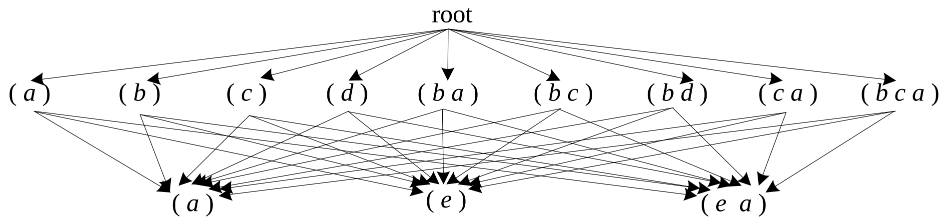

Fig.3 Candidate 2_sequence CS_2

Generate frequent 2-sequence FS_2. (1) Record FS_1 of table Bit_FS_{dim} into table $Temp\_FS_{dim}$, as shown in Table 6. (2) Take one candidate 2-sequence CS_2, e.g. (*c a*) (*e*). Take *Sid* = 1, scan (*c a*) in Table 6, and scan (*e*) with *dim* = 2 in Table 4. If both *Bit*(*ca*) and *Pos*(*a*) are not 0, which indicates (*c a*) and (*e*) all occur, then add 1 to support((*ca*) (*e*)), and record *Bit*((*ca*)(*e*)) = 1; else record *Bit*((*ca*)(*e*)) = 0. (3) Increase *Sid* until *Sid* = 3. If support((*ca*)(*e*)) ≥ *min_support*, then (*c a*) (*e*) ∈ FS_2, record (*c a*) (*e*) in table Bit_FS_{dim}, as shown in Table5; else (*c a*) (*e*) ∉ FS_2. (4) Continue to take next candidate 2-sequence CS_2, until to the end, and generate all the frequent 2-sequences FS_2, they are (*a*) (*a*), (*a*) (*e*), (*a*) (*a e*), (*b*) (*a*), (*b*) (*e*), (*b*) (*a e*), (*c*) (*e*), (*d*) (*e*), (*b a*) (*e*), (*b c*) (*e*), (*b d*) (*e*), (*c a*) (*e*), (*b c a*) (*e*). As shown in Table5.

And then, clear table Pos_FI_{dim} to record the frequent *k*-itemsets on next dimension element.

Table 5 Frequent sequences table Bit_FS_{dim}

Sid	***Bit* (FS_{dim})**											
	FS_1 (*dim* = 1)									**FS_2 (*dim* = 2)**		
	(*a*)	**(*b*)**	**(*c*)**	**(*d*)**	**(*b a*)**	**(*b c*)**	**(*b d*)**	**(*c a*)**	**(*b c a*)**	**(*a*) (*a*)**	**(*a*) (*e*)**	**(*a*) (*a e*)**
1	1	1	1	0	1	1	0	1	1	1	1	1
2	1	1	0	1	0	0	1	0	0	1	1	1
3	1	1	1	1	1	1	1	1	1	0	0	0

Sid	***Bit* (FS_{dim})**									
	FS_2 (*dim* = 2)									
	(*b*) (*a*)	**(*b*) (*e*)**	**(*b*) (*a e*)**	**(*c*) (*e*)**	**(*d*) (*e*)**	**(*b a*) (*e*)**	**(*b c*) (*e*)**	**(*b d*) (*e*)**	**(*c a*) (*e*)**	**(*b c a*) (*e*)**
1	1	1	1	1	0	1	1	0	1	1
2	1	1	1	0	1	0	0	1	0	0
3	0	1	0	1	1	1	1	1	1	1

Note: Table Bit_FS_{dim} only record frequent *dim*-sequences corresponding to current dimension, here lists two records.

Generate candidate sequence CS_{dim}. Add 1 to *dim*, construct sequential patterns tree, the root node is *root*, the first layer nodes are composed of all frequent (*dim*-1)-sequences FS_{dim-1}, the second layer nodes are composed of all frequent *k*-itemsets on *dim*th dimension element, and use generate_*CS* (FS_{k_dim-1} , FI_{k_dim}) to generate candidate sequence CS_{dim}.

Generate frequent sequence FS_{dim}. (1) Record FS_{dim-1} of table Bit_FS_{dim} into table $Temp\_FS_{dim}$. (2) Take one candidate sequence CS_{dim}. Let *Sid* = 1, scan the top *dim*-1 itemset of CS_{dim} in table $Temp\_FS_{dim}$, and scan the *dim*th itemset of CS_{dim} in table Pos_FI_{dim}, if both are not 0, indicates that both occur, then add 1 to support(CS_{dim}), and record $Bit(CS_{dim}) = 1$; else record $Bit(CS_{dim}) = 0$. (3) Increase $Bit(CS_{dim}) = 0$ until to the end. If support(CS_{dim}) $\geq$ *min_support*, then $CS_{dim} \in FS_{dim}$, and record CS_{dim} in table Bit_FS_{dim}; else $CS_{dim} \notin FS_2$. (4) Continue to take next candidate sequence CS_{dim} until to the end, and then generate all the frequent sequenes FS_{dim}.

Table 6 Temporary frequent sequences table $Temp\_FS_{\mathrm{dim}}$

Sid	FS_1								
	(*a*)	(*b*)	(*c*)	(*d*)	(*b a*)	(*b c*)	(*b d*)	(*c a*)	(*b c a*)
1	1	1	1	0	1	1	0	1	1
2	1	1	0	1	0	0	1	0	0
3	1	1	1	1	1	1	1	1	1

Conclusions

The algorithm of multidimensional set sequential patterns mining can mine the sequence database of teaching effect, in which the items are in order while itemsets are not. Through the mining, we can find the relationships among elements. And in the practical application, according to the learning habits of student, as well as the relationship between the grades of courses related to practice teaching and innovation achievements, we can mine the learning state in which they can get good effect of cultivation of innovation ability. This can help to adjust teaching method, fully develop the students' potential, and improve teaching quality. The algorithm can be further optimized in the candidate set and candidate sequence, i.e. effective pruning in the generation of CI_{k_dim} and CS_{dim} to improve the efficiency, and should add different weighs in items, which is a study emphasis in further research.

Acknowledgements

Project supported by the Inner Mongolia Autonomous Region Natural Science Foundation (2011MS0914).

References

[1] R. Agrawal,R. Srikant. Mining Sequential Patterns. Proc. 11st Int. Conf. Data Engineering. (TaiPei, 1995), pp. 3-14

[2] R.Agrawal, R.Srikant. Mining sequential pattern: Generalizations and performance improvements. Proc. 5th Int. Conf. Extending Database Technology. Vol. 1057(1996), pp.3-17

[3] Li Cunjin. The Realization Way and Mechanism of Training Students' Creative Thinking Ability. Innovation and Entrepreneur Education, Vol.4(2013) , pp.1-5

[4] Zhong Yong, Qi n Xiaoli n, Bao Lei. An Association rule mining Algorithmof multidimensional sets. Journal of Computer Research and development, Vol.43(2006), pp. 2117-2123

[5] Song Shijie, Hu Haping. A Sequential Pattern Mining Algorithm Based on Large Itemset Reuse. Journal Computer Research Development, Vol. 43(2006), pp.68-74

[6] YanLiu, Xiaowei Qin, YanZou. Application of Multivariable Sequential Patterns Mining in Teaching Effect Evaluation. IEEE Computer Society (2013), pp.369-372