CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com
SciVerse ScienceDirect
Physics Procedia 24 (2012) 1749 – 1756
Physics Procedia

2012 International Conference on Applied Physics and Industrial Engineering

Multi-Level Sequential Pattern Mining Based on Prime Encoding

Sun Lianglei, Li Yun, Yin Jiang

College of Information Engineering
Yangzhou University
Yangzhou 225009, China

Abstract

Encoding is not only to express the hierarchical relationship, but also to facilitate the identification of the relationship between different levels, which will directly affect the efficiency of the algorithm in the area of mining the multi-level sequential pattern. In this paper, we prove that one step of division operation can decide the parent-child relationship between different levels by using prime encoding and present PMSM algorithm and CROSS-PMSM algorithm which are based on prime encoding for mining multi-level sequential pattern and cross-level sequential pattern respectively. Experimental results show that the algorithm can effectively extract multi-level and cross-level sequential pattern from the sequence database.

© 2011 Published by Elsevier B.V. Selection and/or peer-review under responsibility of ICAPIE Organization Committee.
Open access under CC BY-NC-ND license.

Keywords-prime encoding; multi-level sequential pattern mining

1. Introduction

Multi-level sequential pattern mining is based on different categories and different levels, which can extract sequential pattern not only at the leaf nodes of the bottom level, but also at any other level. The theory of fuzzy set applied to the multi- level sequential pattern mining has been studied and FMSM[3] algorithm and CROSS-FMSM[3] algorithm is proposed, and the importance of encoding[[4] is emphasized. However, The general encoding approach is sequential encoding layer by layer, namely, using the parent node code as a code prefix encode for each node from left to right level by level, but, if the nodes are more, it will appear the non-unique coding, for example, the code 111 can be a the 1st of the level 3, may also be the 11th node of the level 2, and it may be a the 1st of the level 2. The prime encoding with its own specific character can effectively express the uniqueness of encoding, which is widely used in various fields, and based on prime encoding, the sequential patterns mining algorithm PRISM[5](**PRI**me-Encoding Based **S**equence **M**ining) is currently the most efficient sequential pattern mining algorithm.

1875-3892 © 2011 Published by Elsevier B.V. Selection and/or peer-review under responsibility of ICAPIE Organization Committee.
Open access under CC BY-NC-ND license.
doi:10.1016/j.phpro.2012.02.258

This paper presents PMSM(**P**rime encoding Based **M**ulti-level **S**equential Patterns **M**ining) algorithm and CROSS-PMSM algorithm which are based on prime encoding for mining multi-level sequential pattern and cross-level sequential pattern respectively.

2. Related Concepts

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of items, and the itemset is composed of a set of various items, namely, all the subsets of I. A sequence is an ordered list of itemsets, denoted as $S=<s_1,s_2,\ldots,s_m>$, where s_j is an itemset which is also called an element of the sequence, and $s_j=(v_{j1},v_{j2},\ldots,v_{jt})$, where v_{jt} is an item in the itemset s_j. When an itemset has only on item, generally speaking, the brackets are omitted, for example, itemset(v_3)has one item, which will denoted as v_3. Order exists between elements, but, no order exists between items of an element, without loss of generality, we sorted the items of an itemset with increasing order. The number of itemsets in the sequence gives its size |S|. And the length, L(S), is the total number of items in the sequence. A sequence with length L is called an L-sequence. Given a sequence database $SDB=\{S_1,S_2,\ldots,S_n\}$, which is a set of tuples <sid,S>, where S is a sequence and sid is the sequence identifier. And the size of the SDB, |SDB|, is the total number of sequences in the sequence database. A sequence s is called a frequent sequential pattern if the support of the sequence s is no less than the given minimum support threshold (support(s)$\geqslant$min_sup).

Definition 1 Taxonomy Structure tree(TS)[3]. Let TS be a taxonomy structure with l+1 levels. Level t is the root level when t=0, is an internal level when 0<t<l, and is the leaf node level when t=l. At level t (0 $\leqslant$t$\leqslant$l), each node represents a category when t $\neq$ l, or an item when t = l.

Fig. 1 is a taxonomy structure which is a tree showing the relationship of concepts from the atomic items to the most generalized categories. In the TS, at level 0 that is the uppermost category, Living Goods, contains two sub-categories, Diet and Home which are both at level 1. And at level 2, the Diet category has two sub-categories, Food and Drink, and the Home category has three sub-categories, Bedding, Bathware and Outdoor, and so on; at level 3 which is leaf node level, and in order to save space we use letters to represent items.

Table 1 Sequence Database(SDB)

Sequence ID	Sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)bc>
30	<(ef)(ab)(df)cb>
40	<egafcbc>
50	<a(ab)(cd)egh>
60	<a(abd)bc>

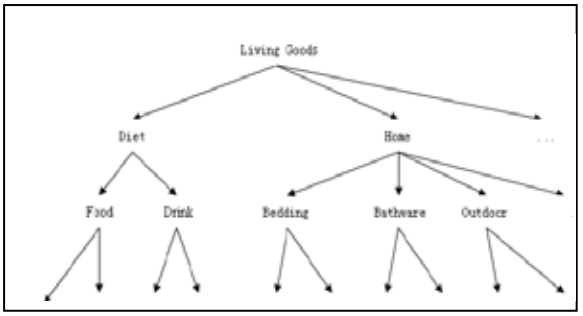

Figure 1. Taxonomy Structure Tree

3. The algorithm for multi-level sequential pattern mining based on prime encoding

3.1 prime encoding in taxonomy structure

To be able to carry out multi-level sequential pattern mining, we need to use a encoding method to encode for each node of a taxonomy structure (TS), and the encoding method shall be satisfied: (1)to express the level of the node is located (2)can be easily to come to the parent-child relationship. Therefore, we used prime encoding method for encoding TS nodes.

Definition 2 Factor Cardinality ||X||. For any square free integer X, whose prime factors are $p_1,p_2,\ldots,p_m$, then the number m is the factor cardinality of X, denoted as ||X||=m; for example, ||30||＝||2*3*5||=3； ||130||＝||2*5*13||＝3.

Let||1||=0, because 1 is has no prime factors. We can use the factor cardinality to express the level of the node is located.

Definition 3 Prime Encoding. Throughout the whole TS encoding process, we use the breadth-first traversal method, namely, for each node a, the encoding is the value of the father node encoding multiply by the prime which never allocate in upper level nodes and left brother node, denoted as encode (a).

Theorem 1 For any two nodes a and b in TS with prime encoding, the node a is the ancestor of node b if and only if encode(b) mod encode(a)=0.

Proof First of all to prove their adequacy. Because the node a is the ancestor of node b, then supposed encode (a) = A, $p_1,p_2,\ldots,p_n$ are prime factors, node b is n-generation child of node a, according to the Definition 3, we know that the encoding of the 1-generation child is A*p1, and the encoding of the 2-generation child is $A*p_1* p_2$, and then the encoding of the n-generation is $A*p_1*p_2*\ldots*p_n$, that is, encode (b)= $A*p_1*p_2*...*p_n$, apparently encode (b) mod encode (a) = 0.

Next to prove their necessity. Since encode(b) mod encode(a) = 0,we suppose that encode(b) = P*encode(a) and $P=P_1*p_2*\ldots*p_n$, then according to the Definition 3, for node b, the encoding of the 1-generation ancestor is $P_1*p_2*\ldots*p_{n-1}$*encode(a), the encoding of the 2-generation ancestor is $P_1*p_2*\ldots*p_{n-2}$*encode(a), and the encoding of the n-generation ancestor is encode(a), and because the principle of the encoding method is to assign to a prime which never allocate in upper level nodes and left brother node, so in the same level can not be repeated for any two nodes, and they cannot exist division relation, and therefore the encode(a) is just the encoding of node a, and the node a is the ancestor of the node b.

Because of the special nature of 1, it is used as the encoding of the root at the top level. And then applied the prime encoding to the Fig. 1, we get the encoded TS, shown in Fig. 2.

Lemma 1 About the multi-level sequential pattern mining, for the sequential pattern of length 1, if the support of the ancestor node a is less than the minimum support threshold at the bottom level, then all the children nodes of the node a are infrequent.

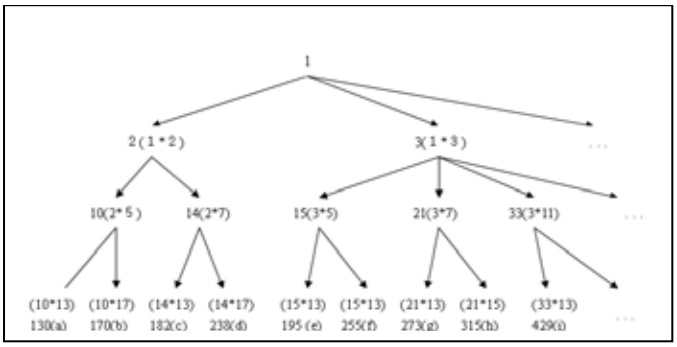

Figure 2. Encoded Taxonomy Structure

3.2 multi-level sequential pattern algorithm(PMSM)

In this paper, we propose the PMSM algorithm (Prime encoding Multi-level Sequential Patterns Mining) which is based on the well-known Apriori algorithm and can mining the patterns level by level from top to bottom with different support thresholds. Since the level 0 only has one node, we ignore it.

The PMSM algorithm is specifically described as follows:

Algorithm Procedure PMSM (*S, level,min_sup[l]*)
input：encoded sequence database S，the number of level *level*, the set of minimum support threshold of all levels *min_sup[l]*
output：all the frequent sequential pattern *FS*

```
Begin
  VISITED←Φ,FS←Φ
  For each s∈S and s∉ VISITED do
  //for each sequence count its support at all levels
    For(l=1;l<level;l++)
      C1,l = count(s)
    End for
    VISITED←s
  End for each
  //find all the frequent 1-sequential pattern
  For (l=1;l<level;l++)
    L1,l={f∈C1,l |f.count≥min_sup[l]}
    FS←L1,l
  End for
  //according to the Lemma 1, filter some unless
    sequences
  S=Filter_SDB(S,L1,level-1)
  //through two loops to find all the frequent sequential
    pattern
  For (l=1;l<level;l++)
    For (k=1;Lk-1 ,l ≠NULL;k++)
    //generate k-candidate sequence from k-1-sequence
  pattern
```

$C_{k,l}$ = Aprior_Gen_Candicate($L_{k-1,l}$)
VISITED←Φ
For each s∈*S* and s∉ VISITED do
  $C_{k,l}$ = count(s)
  VISITED←s
 End for
 $L_{k,l}$ ={f∈$C_{k,l}$ |f.count≥min_sup[*l*]}
 FS←$L_{k,l}$
 End for
End for
End

Since the algorithm discovers patterns in encoded database, the prime encoding is applied to the Table 1, and then the encoded sequence database is shown in Table 2.

Table 2 Encoded Sequence Database(SDB)

SID	Sequence
10	<130 (130 170 182) (130 182) 238 (182 255)>
20	<(130 238) 182 (170 182) (130 195) 170 182>
30	<(195 255) (130 170) (238 255) 182 170>
40	<195 273 130 255 182 170 182>
50	<130 (130 170) (182 238) 195 273 315>
60	<130 (130 170 238) 170 182>

Throughout the whole mining process, for the bottom nodes, leaf nodes, the algorithm just match the encoded database with the leaf node encoding, and for the upper level nodes, inner nodes, it just need one step of division operation to match the encoded database with the node encoding. Here, given the set of minimum support threshold of all levels: the 1st level, min_sup[1]=5; the 2ed level, min_sup[2]=4; the 3rd level, min_sup[3]=3, we take the Fig. 2 for example to describe the mining process of the PMSM algorithm in Table 2. First of all, the algorithm generates candidate sequences of length 1 from top to bottom, denoted as "<pattern>:support".

$C_{1,1}$={<2>:6,<3>:6};
$C_{1,2}$={<10>:6,<14>:6,<15>:5,<21>:2};
$C_{1,3}$={<130>: 6, <195>: 4, <170>: 6, <238>: 5, <273>: 2, <182>: 6, <315>: 1, <255>: 3}.

And then 1-sequences are obtained from the above candidate sequences of length 1:

$L_{1,1}$ ={<2>:6,<3>:6};
$L_{1,2}$={<10>:6,<14>:6,<15>:5};
$L_{1,3}$={<130>: 6, <195>: 4, <170>: 6, <238>: 5, <182>: 6, <255>: 3}.

Next, the algorithm generates candidate sequences of length n, C_n, from sequence pattern of length n-1, and generates n-sequence patterns from C_n. Until no new candidate sequences generate the algorithm ends.

3.3 cross-level sequential pattern algorithm(CROSS-PMSM)

Since users want to get useful information which is not restricted to one level, to mine the type of pattern, we propose CROSS-PMSM algorithm which is based on prime encoding for mining cross-level sequential patter. And the CROSS-PMSM uses a uniform minimum support threshold, since it is a cross process to the whole taxonomy structure.

The CROSS-PMSM algorithm is specifically described as follows:

Algorithm Procedure CROSS-PMSM (*S, level,min_sup*)
input：encoded sequence database *S*，the number of level *level*, the set of minimum support threshold of all levels *min_sup*
output：all the frequent sequential pattern *FS*

```
Begin
    VISITED←Φ,FS←Φ
    For each s∈S and s∉ VISITED do
    // for each sequence count its support at all levels
        For(l=1;l<level;l++)
            C1,l= count(s)
        End For
        VISITED←s
    End for each
    //find all the frequent 1-sequential pattern
    For (l=1;l<level;l++)
        L1,l={f∈C1,l |f.count≥min_sup}
        FS←L1,l
    End For
    S=Filter_SDB(S,L1,level-1)
    // find all the frequent sequential pattern
    For (k=1;Lk-1 ,l ≠NULL;k++)
        Ck= Aprior_Gen_Candicate( Lk-1)
        VISITED←Φ
        For each s∈S and s∉ VISITED do
            Ck = count(s)
            VISITED←s
        End For each
        Lk={f∈Ck |f.count≥min_sup}
        FS←Lk
    End for
End
```

We take the Fig. 2 for example to describe the mining process of the CROSS-PMSM algorithm in Table 3, which is same to the PMSM algorithm. Firstly, it generates all the 1-sequence pattern, $L_{1,1}$={<2>:6, <3>:6}; $L_{1,2}$= {<10>:6, <14>:6, <15>:5} and $L_{1,3}$={<130>:6, <195>:4, <170>:6, <238>:5, <182>:6, <255>:3}，and then it puts all the 1-sequence patterns to a set, L_1 ={<2>:6, <3>:6, <2>:6, <3>:6, <10>:6, <14>:6, <15>:5, <130>:6, <195>:4, <170>:6, <238>:5, <182>:6, <255>:3},and generates candidate sequences of length 2, C_2 ,based on L_1, and then it generates 2-sequence patterns from C_2, and so forth, finally, the L_n is generated and find all the frequent sequential patterns are found.

4. Experimental results and analysis

In order to verify the effectiveness of the algorithm proposed in this paper, we implement the PMSM algorithm and the CROSS-PMSM algorithm using Python2.5.2 language, and carry on the experiments in the randomly generated sequence data set on a computer with 2.8GHz CPU and 512MB main memory using the Linux operating system.

We compare the six algorithms: GSP、CROSS-GSP、FMSM、CROSS-FMSM、PMSM and CROSS-PMSM. Firstly, the runtimes of the algorithms with different support threshold applied the same

sequence dataset, and the results are shown in Fig. 3, and then tests the scalability of the algorithms with the minimum support threshold (min_sup = 2.5%) and other parameters fixed. With the changes in the length of the sequence database, performed by the six algorithms the time changes, as shown in Fig. 4.

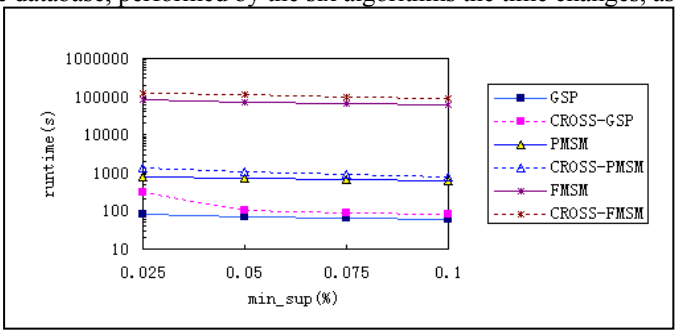

Figure 3. the relation of min_sup and runtime.

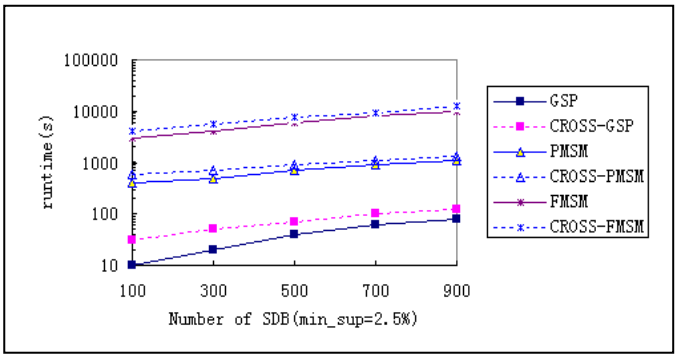

Figure 4. the relation of the length of SDB and runtime.

Fig. 3 shows that in the case of the sequence data set and other parameters are fixed, the time complexity of the six algorithms gradually reduced with the increasing of min_sup. And we can know that the algorithms proposed by this paper are lower than the GSP and CROSS-GSP algorithms, but faster than the FMSM and CROSS-FMSM algorithms. Its main reason is that, compared to the GSP and CROSS-GSP algorithms, the PMSM and CROSS-PMSM algorithms generate more candidate sequences which require more time, however, within a reasonable amount of time, and when accessing the encoded database to count the support, the PMSM and CROSS-PMSM algorithms just use one step of division operation, and then eliminate a large number of comparison operations and the support calculation of the FMSM and CROSS-FMSM algorithms , thus saving a lot of time. Compared the PMSM algorithm, the CROSS-PMSM algorithm works at cross levels which will generate more candidate sequences, therefore, the PMSM algorithm is faster than the CROSS-PMSM.

Fig. 4 shows that in the case of the minimum support threshold(min_sup=2.5%) and other parameters are fixed, the time complexity of the six algorithms gradually reduced with the increasing of min_sup, the time complexity of the six algorithms gradually increased with the increasing of the length of the sequence database, but is approximately linear. Therefore, The algorithms presented in this paper, the performance of which are between the other two pairs of algorithms, and the time-consuming of which

are reasonable, are able to meet the user's requirements, but also extract perfect multi-level and cross-level sequential pattern.

5. Conclusions

In this paper, we prove that just one step of division operation can decide the parent-child relationship between different levels by using prime encoding, and present PMSM algorithm and CROSS-PMSM algorithm which are based on prime encoding for mining multi-level sequential pattern and cross-level sequential pattern respectively. Experimental results show that the algorithms can effectively extract multi-level and cross-level sequential pattern from the sequence database.

Acknowledgment

This research was supported in part by the Chinese National Natural Science Foundation under grant No. 60673060, Natural Science Foundation of Jiangsu Province under contract BK2008206 and Natural Science Foundation of Education Department of Jiangsu Province under contract 08KJB520012.

References

[1]R.Agrawal, R.Srikant. Mining Sequential Pattern. In: Pro. of the 11st Int. Conf. on Data Engineering, Taipei, March 1995, pp.3-14.

[2]R.Srikant, R.Agrawal. Mining Sequential Patterns: Generalizations and Performance Improvements.In: Proc. of the Fifth Int. Conf. on Extending Database Technology (EDBT), 1996, Avignon, France.

[3]Y.L. Chen and T.C.K. Huang, A novel knowledge discovering model for mining fuzzy multi-level sequential patterns in sequence databases, Data & Knowledge Engineering 66 (3) (2008), pp. 349–367.

[4]Y.L. Chen and T.C.K. Huang, Developing an efficient knowledge discovering model for mining fuzzy multi-level sequential patterns in sequence databases, Data & Knowledge Engineering, Data & Knowledge Engineering, 06,2009.

[5]Karam Gouda, Mosab Hassaan, Mohammed J. Zaki, Prism: A Primal-Encoding Approach for Frequent Sequence Mining , 2007 Seventh IEEE International Conference on Data Mining, 2007 :pp.487-492.

[6]GONG Zhen-zhi, LIU Hai-dong, HU Kong-f a, DA Qing-li, Improved prime number labeling method on XML, Computer Integrated Manufacturing Systems, 2008,14(8): 1058-1664.