Mining Least Relational Patterns from Multi Relational Tables

Siti Hairulnita Selamat[1], Mustafa Mat Deris[2], Rabiei Mamat [1], and Zuriana Abu Bakar[1]

[1] Department of Computer Science,
University College of Science and Technology,
21030 Kuala Terengganu, Malaysia
{rab, zuriana}@kustem.edu.my
[2] Faculty of Information Technology and Multimedia,
College University Technology Tun Hussein Onn,
86400 Parit Raja, Batu Pahat, Johor, Malaysia
mmustafa@kuittho.edu.my

Abstract. Existing mining association rules in relational tables only focus on discovering the relationship among large data items in a database. However, association rule for significant rare items that appear infrequently in a database but are highly related with other items is yet to be discovered. In this paper, we propose an algorithm called Extraction Least Pattern (ELP) algorithm that using a couple of predefined minimum support thresholds. Results from the implementation reveal that the algorithm is capable of mining rare item in multi relational tables.

1 Introduction

Nowadays, the quantity of data is expanding rapidly. Most of the data are stored in a relational table in order to support a variety of administrative management where it provides valuable input for organizational decision-making [11]. It will be extracted from the tables through various techniques to obtain valuable information and knowledge from those vast amounts of data. Currently, mining association rules in relational tables only focus on discovering the relationship among large itemsets in the tables that satisfy the support and confidence set by the users [4]. Nevertheless, the existing association rule discovery techniques do not consider the occurrence frequency pattern of data, and discover the association rules using the same support on the whole data, so the discovered rules with regard to rare data may be redundant and as a result, unnecessary rules may be generated.

In this paper, a new algorithm called Extraction Least Pattern (ELP) algorithm to extract least relational patterns of data items from multi relational tables is proposed. Least data items are referred to the data items that its frequency in the relational tables does not satisfy the minimum support but are highly associated with the specific [16]. This enables us to identify significant rare data associated with specific data in a way that rare data occur simultaneously with specified data more frequently that the

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 59–66, 2005.

average co-occurrence frequency in the relational tables. A range of predefined minimum support thresholds are used to discover the least data. By using a couple of minimum support thresholds, it captured more meaningful data to discover interesting patterns. This paper is organized as follows. In section 2, the related wok will be discussed. In section 3, we discuss the background of this project. ELP (Extraction Least Pattern) algorithm and its detail experiments are present in section 4. Finally, we conclude this paper in section 5.

2 Related Work

The concept of relational patterns and the utilized elements of Apriori [10] algorithm to extract the relational patterns from multiple relational tables have been proposed in [1], which used bottom-up approach. Another previous works focus on advanced association rules problems that involved relational tables have been briefly discussed in [2], [3], [8], [20] and [21]. However, the model used in these studies only focus on providing an approach to generate the large datasets that satisfied predefined minimum support threshold. In other cases, there are also researches discovering on significant rare data in the table that have been studied extensively in [14], [15] and [16] but, unfortunately all these models only discovering the rare data in a single table instead of multi relational tables.

3 Background

Association rule mining is one of the processes of discovering hidden patterns in data. It also known as finding association, correlation or causal structures among sets of data items or objects in transaction tables, relational tables or other information repositories. Thereupon, one of its mining algorithms, Apriori algorithm is an influential algorithm used to mine all frequent data items in a table that satisfy the user predefined minimum support and minimum confidence constraints. A frequent data item is the data whose support is greater than user predefined minimum support threshold.

Relational data mining is one of data mining techniques for relational tables. Most existing traditional data mining approaches which are look for patterns in a single table are called propositional patterns. In contrast, relational data mining approaches that are seek for patterns among multiple tables are called relational patterns. That is, relational pattern involve multiple relations that represent the information as a set of relations. This is because, a relational table consists of a set of multiple tables and a set of associations (i.e. constraints) between pairs of tables describing how records in one table relate to records in another table. An association between multiple tables describes the relationships between records in these tables. In relational model, the association between these relational tables is defined through primary and foreign key attributes. If relation R_j includes, among its attributes, relation R_{j+1}'s primary key, then a tuple t_1 in R_j and a tuple t_2 in R_{j+1} refer to one another if t_1[Foreign_Key] = t_2[Primary_Key] [1].

4 Approach

The bottom-up approach used in this paper to extract the least patterns beginning from the leaf relation R_i up to relation R_{i-n}, where the leaf relation R_i is n levels downward in the path. In addition, based on hierarchical concept, the leaf relation R_i is a leaf tuple. Thus, the relations composing the path are considered as R_{i-n}, R_{j-n+1}, …, R_i. Our approach is only considered the least data items that occur infrequently but appear simultaneously with specific data items in high proportion. In brief, the least items are data items that rarely occur in a table. Hence, the least data items can only be found in the data if the predefined minimum support threshold has to be set very low. However, this situation would cause too many rules generated, which most of them are not important. If a higher minimum support threshold is used, we might miss out on generating important association rules. This problem is known as the rare item problem. Despite these drawbacks, our approach introduces usage of a range of two predefined minimum support thresholds that may overcome these problems. The extracted least data items must be satisfied the range of predefined minimum support thresholds, that is data items must be contained in between first and second user-predefined minimum support threshold. Four phases are involved in ELP algorithm in mining least data items on multiple relational tables. Those are *Extract least data items*, *Extract sibling patterns*, *Extract join patterns*, and *Extract least relational patterns*.

Phase 1: Extract Least Data Items

Basically, in normalized relational tables design, it would be to have three tables that is first table for contact, second table for groups, and the third table called 'joiner' table depicting what groups a contact belongs to. In this phase, we select the related attributes from the 'joiner' table, and construct a table called JoinTable as shown in Table 2. For example, we extract relational patterns from the sample hospital database as shown in Table 1, where the sample database has two main tables, i.e., Department Table, and Procedures Table. Assume that the first minimum support parameter, *fminsup* is 25%, and the second minimum support parameter, *sminsup* is 10%. Those least data items are only extracted if they are satisfying a range of two predefined minimum support thresholds. Using bottom-up approach starting from the leaf level tuple L_j^{i-n}, each extracted least data item is mapped to a unique key and stored in a set that is split up into a few tables according to the field name. Eventually, from this example, two tables as shown in Table 3 and Table 4 will be generated.

Algorithm 1. Algorithm applied to JoinTable in order to extract least data items matched

```
for each f_j ∈ D  do        // each field, f in table, D
    for each i_n ∈ I     // I = i_1,i_2,…,i_n (A set of data items)
      if (sminsup≤ i_n.support < fminsup) then
            i_n ∈ I
              k=k+1     // increase unique key,k
    end
  L=LUL_j       // L Least data items
end
```

Table 1. Two tables from Hospital Database

R_2 : Department Table

ID	Department	LOS
100	ER	1 day
100	internal	2-3 days
200	pediatric	2-3 days
300	ER	3-6 hour
400	ER	1 day
400	pediatric	1 day
400	surgery	7-10 hour
500	surgery	3-6 hour
600	ICU	1-2 hour

R_3 : Procedures Table

ID	Department	Procedures	Cost($)
100	ER	BC	2-5K
100	ER	ECG	1-2K
200	pediatric	BC	1-2K
200	pediatric	X-ray	5-7K
300	ER	ECG	1-2K
400	pediatric	BC	1-2K
400	pediatric	ECG	5-7K
400	surgery	operation	7-9K
500	surgery	operation	2-5K
600	ICU	fixation	1-2K

Table 2. JoinTable table

ID	Department	Procedures
100	ER	BC
100	ER	ECG
200	pediatric	BC
200	pediatric	X-ray
300	ER	ECG
400	pediatric	BC
400	pediatric	ECG
400	surgery	operation
500	surgery	operation
600	ICU	fixation

Table 3. Procedures table

ID	Procedures	Key
200	X-ray	1
400	operation	2
500	operation	2
600	fixation	3

Table 4. Department table

ID	Department	Key
400	surgery	4
500	surgery	4
600	ICU	5

Phase 2: Extract Sibling Patterns

At this phase, either Descendant or Transformed, both tables are constructed in order to extract any sibling pattern that exists in '*sibling_collection*' field. These tables transform relation R_j that each least data items of tuple in R_j is replaced with set of unique keys depending on the parent tuple (i.e., ID) that representing all least data items contained in that tuple's sub-tree. For instance, the Descendant table as shown in Table 5 consists of joining of the least data items in Table 3, and Table 4 but the least data items from Table 4 have been replaced with matched unique keys. As a consequence, in the Transformed table, contained all patterns in Least table, Join table and Sibling table, as these patterns constitute all possible least relational patterns with respect to relation $R_{i\text{-}n}$, contained in

sub-trees of tuples in R_j. Each extracted least $(n \geq 2)$-Sibling pattern is mapped to a unique key and stored in set S_j^{i-n}. Algorithm 2 is the algorithm used for extracting any sibling pattern that exists in generated table. For example, based from result of our implementation, there is no sibling pattern extracted from Table 5 but there is one sibling pattern extracted from Table 7.

Table 5. Descendant table

ID	Data_Item	Sibling_Collection
200	pediatric	1
400	surgery	2
500	surgery	2
600	ICU	3

No Sibling Pattern

Algorithm 2. Algorithm that used to generate sibling pattern

```
for each l_j ∈ L      do
     for each t_m ∈ T do    // T data items in sibling collection
          if (PID= t_m.ID) and (t_m.duplicate = True)  then
               t_m ∈ S_j
               k=k+1     // increase unique key,k
     end
     S=SUS_j
end
```

Phase 3: Extract Join Patterns

If the tuple is a leaf relation, its tuples have no join pattern and thus, this phase is skipped. These join patterns generated by joining all descendant patterns in Descendant table with all the least data items in least table using Algorithm 3. Each generated least join pattern is then mapped to a unique key and stored in a set J_j^{i-n} as shows in Table 6. An extracted join pattern is represented with an ordered list of two members $< l, ds >$, where $l \in L_j^{i-n}$ and $ds \in DS_{j+1}^{i-n}$, which these data items are contained in tuple *t*'s sub tree and following criteria are satisfying:

1. *t* contains *l*
2. there exist $< PID, \{P_{PID}\} > \in DS_{j+1}^{i-n}$, such that PID = *t*.ID, and $ds \subseteq \{P_{PID}\}$, where *t*.ID is tuple *t*'s ID.

Table 6. JoinPattern table

Join_Pattern	Key
4 , 2	6
5 , 3	7

Algorithm 3. Algorithm that used to generate join pattern

```
for each l_j∈L do
    for each t_m∈T do
        if (PID= t_m.ID) then
            <l_j, t_m> ∈J_j
            k=k+1    // increase unique key
    end
    J=JUJ_j
 end
```

Table 7. Transformed table

ID	Sibling_Collection
400	{4 , 2 , 6}
500	{4 , 2 , 6}
600	{5 , 3 , 7}

Sibling Pattern Extracted

Table 8. S2Pattern table

Sibling_Pattern	Key
{4 , 2 , 6}	8

Phase 4: Extract Least Relational Patterns

This final phase construct the sets that contained all least patterns in the Department table, Procedures table, SiblingPattern table and JoinPattern table. In other form of results generated from phase one to four are: $\hbar_j L_j^{i-n}, \hbar_j S_j^{i-n}, \hbar_j J_j^{i-n}$, where j starting from the leaf relation's index i to i-n, as shown in Table 9. These set contains unique keys representing all least relational patterns that have been removed all redundant patterns. In addition, these patterns are encapsulated in their respected parent tuples (i.e., ID tuple).

Table 9. LeastPattern table

ID	Set_Patterns
400	8 , 4 , 2 , 6
500	8 , 4 , 2 , 6
600	5 , 3 , 7

Based on the implementation results, the extracted least relational patterns contains unique keys that represented each of its nodes having a number pointer to its parent data items, which indicate that the data items are related in each other. Specifically,

the least relational patterns captured relationships between the tuples across multi relational tables from which co-occurrence of attributes were extracted. Although these least relational patterns are rarely occur in a database, it is special interesting cases to be discovered. Therefore, the least data items should not totally ignore to avoid potentially valuable information loss. These extracted least relational patterns can be used to improve and support variety of organizational decision-making tasks such as hospitalization administrative databases. For instance, least relational patterns may be used to support hospitalization's decision making by identifying their patient behavior. More precisely, this implementation may used to discover unexpected data into an interesting pattern.

5 Conclusion

In this paper, we presented an ELP algorithm and discussed the approach used to discover the least relational patterns from multi relational tables. The ELP algorithm generated all least data items that satisfied a couple of predefined minimum support thresholds. Specifically, we used a couple of predefined minimum support threshold to extract least patterns be more meaningful and avoid valuable 'nuggets' of information from loss. The implementation results indicate that the introduced algorithm is capable of mining rare items for multi relational tables.

References

[1] M. Saar-Tsechansky, N. Pliskin, G. Rabinowitz, A. Porath, "Mining Relational Patterns from Multiple Relational Tables". Decision Support Systems, v.27, n.1-2, 177-195.

[2] R. Agrawal, R. Srikant, "Mining Qualitative Association Rules in Large Relational Tables". SIGMOD'96, Montreal, Canada, June 1996.

[3] X. Shang, K. Sattler, I. Geist, "Efficient Frequent Pattern Mining in Relational Databases". Workshop GI Working Group of the Knowledge Discovery (AKKD) in the context of the LWA 2004.

[4] R. Agrawal, R. Srikant, "Mining Association Rules Between Sets of Items in Large Databases". Proceedings of ACM SIGMOD, 207-216.

[5] R. Agrawal, R. Srikant, "Mining Sequential Patterns". In Proc. Of the 11th International Conference Data Engineering 1995.

[6] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules". Proceedings of the VLDB Conference, 487-499.

[7] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, "From data mining to knowledge discovery: an overview", in: U.M. Fayyad, G. Piastetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996.

[8] A Swami, M. Houtsma, "Set-oriented Data Mining in Relational Databases". In International Conference Management of Data Engineering, Taipei, Taiwan, March 1995.

[9] R. Agrawal, R. Srikant, "Mining Generalized Association Rules". In Proc. Of the 21st International Conference on VLDB, Zurich, Switzerland, September 1995.

[10] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo, "Fast Discovery of Association Rules", in: U.M. Fayyad, G. Piastetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press.

[11] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth: "From Data Mining to Knowledge Discovery in Databases". In: Fayyad et al: Advances in Knowledge Discovery and Data Mining. AAAI Press/The MIT Press, Menlo Park 1996.

[12] M.H. Dunham, "Data Mining: Introductory and Advanced Topics", New Jersey, Prentice Hall, 2003.

[13] B. Liu, W. Hsu, Y. Ma, "Mining Association Rules with Multiple Minimum Supports", Proceedings of the 5th ACM SICKDD International Conference on Knowledge Discovery and Data Mining, San Deigo, California, United States, 337-341, Aug 1999.

[14] H. Yun, D. Ha, B. Hwang, K.H. Ryu, "Mining Association Rules On Significant Rare Data Using Relative Support", The Journal of Systems and Software, Elsevier, v.67. n.3. 181-191, Sept 2003.

[15] N.F. Nabila, M.M. Deris, M. Y. Saman, A. Mamat, "Association Rules On Significant Rare Data Using Second Support", (forthcoming).

[16] P.S.M. Tsai, C.-M. Chen, "Mining Interesting Association Rules From Customer Databases and Transaction Databases", Information Systems, Elsevier, v.29. n.8. 685-696, Dec 2004.

[17] S. Sarawagi, S. Thomas, R. Agrawal, "Integrating Association Rule Mining with Relational Database Systems: Alternatives and Impilcations", SIGMOD Record (ACM Special Interest Group on Management of Data), v.27. n.2. 343-355.