An Efficient Approach for Effectual Mining of Relational Patterns from Multi-Relational Database

Vimalkumar Dhanasekar[1] and Tamilarasi Angamuthu[2]
[1]Department of Master of Computer Applications, Anna University of Technology, Coimbatore
[2]Department of Master of Computer Applications, Kongu Engineering College, India

Abstract: *Data mining is an extremely challenging and hopeful research topic due to its well-built application potential and the broad accessibility of the massive quantities of data in databases. Still, the rising significance of data mining in practical real world necessitates ever more complicated solutions while data includes of a huge amount of records which may be stored in various tables of a relational database. One of the possible solutions is multi-relational pattern mining, which is a form of data mining operating on data stored in multiple tables. Multi-relational pattern mining is an emerging research area and it has been received considerable attention among the researchers due to its various applications. In the proposed work, we have developed an efficient approach for effectual mining of relational patterns from multi-relational database. Initially, the multi-relational database is represented using a tree-based data structure without changing their relations. A tree pattern mining algorithm is devised and applied on the constructed tree-based data structure for extracting the frequent relational patterns. The experimentation is carried out on customer order database and the comparative results demonstrate that the proposed approach is effective and efficient in mining of relational patterns.*

Keywords: *Data mining, multi-relational data mining, relational pattern, tree pattern mining, multi-relational database, customer order database.*

Received September 10, 2010; accepted January 9, 2012; published online March 1, 2012

1. Introduction

Large amount of information belonging to diverse enterprises can easily be accessible from the databases because of widespread computerization and affordable storage facilities. The main target of this huge data collection is the utilization of this information to achieve competitive remunerations, by identifying the previously hidden patterns in data that can direct the process of decision making [23]. Data mining emerge as a promising solution for discovering the knowledge hidden in databases. Data Mining is formally defined as "the non-trivial mining of implicit, previously unknown and potentially useful information from data in databases" [21]. Data mining is used in both private and public sectors for their multiple purposes [27]. Descriptive and Predictive are the two general classes of data mining techniques. Pattern recognition is the main intention of descriptive data mining e.g., product configurations produced in mass customization applications [3]. Clustering, Association rule mining, and sequential pattern mining are some of the descriptive data mining methods [18]. The intention of predictive data mining is to construct frameworks for resolving or predicting an outcome, e.g., a stock level [3]. The tasks of predictive data mining are classification, regression and deviation detection.

Data mining algorithms searches the patterns in data. Most of the mining approaches are propositional and search for patterns in a single data table. Moreover, a huge body of work is available for methods handling data from a single relation. The single relation approaches are often extended effectively to handle multi-relational data. Mining data includes complex or structured objects and the normalized representation of such objects in a relational database needs multiple tables [10, 31]. Relational databases are one of the wealthiest sources of knowledge in the world and also it is the most reputed warehouse for structured data [8, 29, 30]. A relational database contains a set of labeled tables, often called as relations that individually act as a single table. This relation describes how certain columns in one table can be used to search information in related columns in another table, thus relating sets of records in the two tables [12]. Relational database requires effective and efficient technique for mining the pattern based on the data stored in multiple tables. In this process, important characteristics are extracted from datasets stored in multiple tables with one-to-many relationships format [2].

Relational Data Mining (RDM) approaches search for patterns which contain multiple tables (relations) from a relational database [18]. To draw attention to this fact, RDM is usually referred to as Multi-Relational Data Mining (MRDM). MRDM is the multi-disciplinary field, which discovers knowledge from relational databases containing multiple tables [28]. The aim of MRDM is to discover patterns and models spanning all the tables and links which either describe or predict the target entity or attribute, using a given database that contains multiple tables related through foreign key joins, a target table (that normally represents a certain real-world entity type) and preferable target attribute (e.g., a class label attribute) [7, 11]. An extension to the simple transactional data model is called Multi-relational data mining. The major data mining tasks, including association analysis, classification, clustering, learning probabilistic models and regression, has been considered in the current MRDM approaches [9, 16].

While searching for patterns in multi-relational data, it is noted that patterns are enclosed with multiple relations. These patterns are usually affirmed in a more expressive language than the patterns defined on a single data table [10]. This specifies that, relational pattern engross multiple relations signifies the information as a set of relations. This is due to the reason that, a relational table comprises of a set of multiple tables and a set of associations (i.e., constraints) among pairs of tables that describes how records in one table relate to records in another table [25]. The relationship of certain pattern hidden inside data collections are discovered by relational patterns across multiple databases. The mining of relational patterns across multiple databases is an interactive process, where one of its techniques is described as follows. When a user provides a query to the system it identifies all patterns satisfying the query, in an effectual way. If the pattern exists within the relational database (pattern tables), it is impossible for a user to query the collection of all possible patterns using SQL [6]. Moreover, several researches are available in the literature for multi-relational pattern mining in recent times.

With great interest in this research area, we have proposed an efficient multi-relational pattern mining approach. In the proposed approach, the data in the multiple tables are represented using the tree-based structure. The constructed tree is named as Multi-Relational tree (R_M-tree), where the nodes denote the tuples of the relational database. The R_M-tree is then used to mine the relational patterns and these patterns are said to be frequent, only if the number of tuples that contain the relational pattern are higher than the minimum support threshold level. For mining frequent relational pattern, we have developed an efficient tree pattern mining algorithm that uses the positional data for traversing and searching the nodes of the R_M-tree.

The basic outline of the paper is organized as follows. A brief review of the researches related to the proposed approach is discussed in section 2. The proposed approach for relational pattern mining is described in section 3. The experimental results of the proposed approach are presented in section 4. Finally, the conclusion is given in section 5.

2. Review of Related Researches

A handful of research works available in the literature deals about the mining of multi-relational data. Several literary works related to multi-relational pattern mining and multi-relational association rule mining exist in the literature. A few of the most recent literature works in this topic are reviewed in this section.

Prolog databases and Datalog queries are the two streams of preceding work that deal with the discovery of association rules over multiple relations. The MRI Iceberg-cubes mining technique have established a new approach. But it does not consider the cyclic join paths; hence, Liang *et al.* [17] have proposed an algorithm called Extended-MRI-cube, which is based on the MRI-Cube algorithm, in order to manage the cyclic join path situation. Their experiments have revealed that the algorithm was more applicable and efficient than the existing methods.

MRDM is concerned with data having heterogeneous and semantically rich relationships between various entity types. Seid and Mehrotra [24] have introduced a multi-relational iceberg-cubes (MRI-Cubes) as a scalable method for calculating data cubes (aggregations) effectively over multiple database relations, and this method specifically served as mechanisms to calculate frequent multi-relational patterns "item sets".

Makino and Inuzuka [20] have proposed an ILP (inductive logic programming) based approach which is more efficient and treats patterns on various tables. IPL approach based patterns miners have produced significant patterns and these patterns are widely applicable but still, computationally expensive. The benefit of MAPIX is it creates patterns by combining atomic properties extracted from samples. Contrasting to other algorithms, it gained efficiency through confining patterns into combinations of the atomic properties.

Based on concepts and methods of relational database, Zhang [32] has proposed a new general algorithm called MMRFP for multi-level multi-relational frequent pattern discovery. Particularly, they have defined the search space on the basis of conjunctive query containment. It is a well-known concept in relational database theory, which successfully discover multi-level multi-relational frequent pattern as well as diminish the semantically reoccurring patterns regarding the concept hierarchies' background knowledge. Theoretical analyses and experimental results have revealed the high understandability, accessibility, efficiency and scalability of the proposed algorithms.

Inuzuka *et al.* [14] have extended the bottom-up relational miner MAPIX. It produces the patterns containing a large part of instances in the target relation match by inputting a relational database with multiple relational tables including a target relation. The patterns are specified using logical formulae. Even though the widely accepted system WARMR produces and analyses possible patterns, it still has certain restrictions in its efficiency. MAPIX used a bottom-up approach and achieved efficiency at the cost of variety of patterns. The merits of bottom-up approach have been maintained by the proposed algorithm EQUIVPIX (an equivalent-class-based miner using property items extracted from examples). The merits are time-efficiency and prohibition of duplicate patterns. In addition it amplifies pattern variation. Equivalent classes of properties extracted from samples as well as two combination operators belonging to them have been established by equivpix.

Selamat *et al.* [25] have proposed an Extraction Least Pattern (ELP) algorithm which uses a pair of predefined minimum support thresholds. The implementation results have proved that their algorithm is efficient in mining rare items in multi relational tables.

Relational patterns across multiple databases can expose unique pattern relationships hidden inside data collections. Zhu and Wu [33] have proposed a systematic framework called Discovering Relational patterns Across Multiple databases (DRAMA). Particularly, they have attempted to discover patterns from various databases with patterns' relationships satisfying the user specified constraints, by providing a series of data collections. Their proposed method tries to construct a Hybrid Frequent Pattern tree (HFP-tree) from multiple databases, and extract patterns from the HFP-tree by incorporating users' constraints into the pattern mining process. Discovering the relationship between huge data items in a database is focused only by current mining association rules in relational tables. Association rule for important atypical items that occur occasionally in a database are decidedly related with other items yet to be discovered.

Calders *et al.* [6] have proposed a novel approach which extends the DBMS itself, not the query language, and it incorporates the mining algorithms into the database query optimizer. Towards this end, they have established a virtual mining views that can be queried as if they were conventional relational tables (or views). A mining algorithm gets triggered to appear all tuples needed to answer the query, when each time the database system accesses one of these virtual mining views. They have revealed how this can proficiently be performed for the popular association rule and frequent set mining problems.

JimEnez *et al.* [15] have presented a new technique for mining relational patterns from multi-relational databases. The representation of a multi-relational database as a group of trees has been used as a basis for their method. Their fundamental idea is to construct a tree representation for all tuples in the target data by pursuing the present foreign keys that link tables in the multi-relational database. The trees representing each tuple in the target relation have used primary keys as intermediate nodes in the key-based representation scheme. Frequent patterns in the trees representing the multi-relational database have been identified and differences that occur due to discovery of induced or embedded patterns in the key-based or object-based representation scheme have been analyzed. These frequent patterns have been used for extracting the association rules from the multi-relational clustering techniques. The practicability of the approach has been shown from their experiments conducted on a real database.

3. Proposed Approach for Mining of Relational Patterns from Multi-Relational Database

MRDM [13, 22, 26] is a multi-disciplinary field which discovers the knowledge from relational databases containing multiple tables [5]. Mining of relational patterns from the relational databases is one of the data mining techniques, which have been received great attraction among the researchers, because of its real world applications. While comparing with the mining patterns from the single table, the approaches that are used for mining relational patterns are somewhat difficult. Basically, the two alternative approaches of multi-relational data mining methods are structural and propositional [4]. In the first approach, by utilizing the mining method, the entire hypothesis space is directly explored. Whereas in the second approach, also known as propositionalisation, relational learning problems are converted into attribute-value representations, which is acquiescent for conventional data mining methods. In the proposed approach, the relational data that exist in the multiple tables are first transformed into a tree-based structure. Then, the frequent relational patterns are extracted from the constructed R_M-tree using the proposed tree mining algorithm. Our proposed approach has two important steps:

- Tree-based representation of multi-relational database.
- Mining of relational patterns.

3.1. Tree-Based Representation of Multi-Relational Database

The initial step of the proposed approach is to transform the multi-relational database into tree-based structure. Before seeing the representation model, we have first described the basic structure of the multi-relational database and then, we have presented the

R_M-tree constructed from the multi-relational database.

3.1.1. Multi-Relational Database: An Overview

A relational database contains several relations that are in the form of 2-D tables of rows and columns having related tuples. The rows or tuples of each table is called as records and the columns i.e., fields in the record is called as attributes. The structural representation of the multi-relational database is described as: The multi-relational database has a set of tables $T=\{T_1,T_2,\ldots,T_n\}$ and these tables are then related through the matching of primary and foreign keys. There is a set of descriptive attributes $A(T)$ for each table T and these attributes are corresponds to columns and rows of the table belonging to the tuples. Generally, attributes in different tables are independent and the relations between the tables are exactly defined using the foreign keys.

- *Definition 1:* A primary key is a field or combination of fields which uniquely find a record in a table, and thus an individual record can be mined without any confusion.
- *Definition 2:* A foreign key is a referential constraint between two tables in the context of relational databases. The foreign key identifies a column or a group of columns in one (referencing) table that links to a column or group of columns in another (referenced) table. The columns in the referencing table should be the primary key or other candidate key in the referenced table.
- *Definition 3:* A target table may refer to a table that acts as a back-up table to the source table.
- *Definition 4:* A source table is a table which has the recent data used by an external data source.

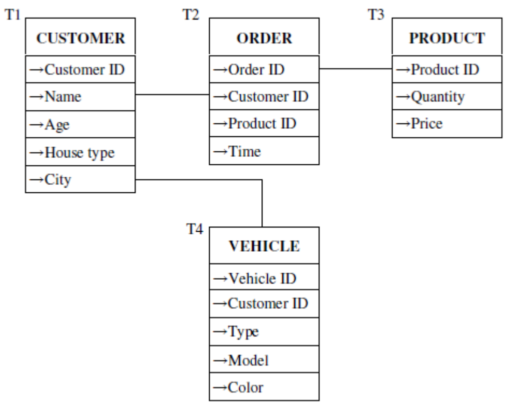

Figure 1. Multi-relational database: an example.

- *Example 1:* In the multi-relational database, the structured data is represented with multiple tables. The information about a specific topic is distributed over these tables. Among these tables, one particular table is considered as target table which is linked with the other table using the foreign key relations. In the given example i.e., Figure 1 shows four different tables that are customer (T_1), order (T_2), product (T_3) and vehicle (T_4) which represents the structured data. Here, 'order' (T_2) is a target table, which has four field records: OrderID, ProductID, CustomerID and Time of buying where, ProductID and CustomerID are two foreign key relations. The order table contains one record for each new order, and its key is OrderID. The customer table has the information about the customer and its field records are customerID, Name, Age, Housetype and City. The relationship between the four tables presented in the example is represented as: $T_2 \rightarrow T_1$; $T_2 \rightarrow T_3$ and $T_4 \rightarrow T_1$, where $T_i \rightarrow T_j$ represent that the foreign key of table T_i is the primary key of T_j.

3.1.2. Construction of R_M-Tree from the Multi-Relational Database

Generally, a tree is a data structure which builds a hierarchical tree structure with a group of linked nodes. In the proposed approach, the tree-based data structure has been used for representing the multi-relational database. The records and their relations present in the multiple tables are represented using a set of nodes in the tree structure. For constructing the tree-based data structure, first we find the target table from the multiple tables in the relational database. The target table mainly focuses on processing and also it retrieves any information regarding each object that is stored in other tables. Once the target table has been chosen, the construction of R_M-tree is described as follows:

1. The tuples, corresponding to the primary key of the target table is created as a set of nodes i.e., next to the root node.
2. For each constructed node, we add the fields (attributes) corresponding to the target table as a child node.
3. If the target table contains any foreign key, then the tables corresponding to the foreign key are retrieved.
4. The fields related to the retrieved tables are included as a leaf node of their corresponding parent node.

The tree-based representation is very useful for mining the patterns because we can apply the tree pattern mining algorithm to the constructed R_M-tree. Moreover, the easiness of tree structure is more beneficial for an efficient implementation of the tree pattern mining algorithms.

- *Example 2:* The tree-based structure for the example given in section 3.1.1 is shown in Figure 2. In the target table T_2 the primary key is OrderID, which contains a set of tuples in the relational database and each tuples are formed as separate nodes (next to the root node) in the tree-based structure. The field records linked to target table such as orderID, customerID, productID and time are added as a child node of constructed node. The shaded boxes have the information about the sub-nodes. Next, we consider the foreign keys of the target table, i.e., in our example customerID and productID are the foreign keys. The tables, identified with the foreign keys are again added as sub nodes of the related parent node. The table T_4 contains field records such as vehicleID, type, model and color which are used only for the customers having own vehicle. Therefore, when we construct a R_M-tree, the only customers having own vehicle should contain the sub-node of the relevant parent node and others did not have any sub-node for their corresponding parent node. So, we denote this sub-node as dotted lines to handle this kind of unordered format. And also, we develop the efficient tree pattern mining algorithm by keeping this unordered tree in mind.

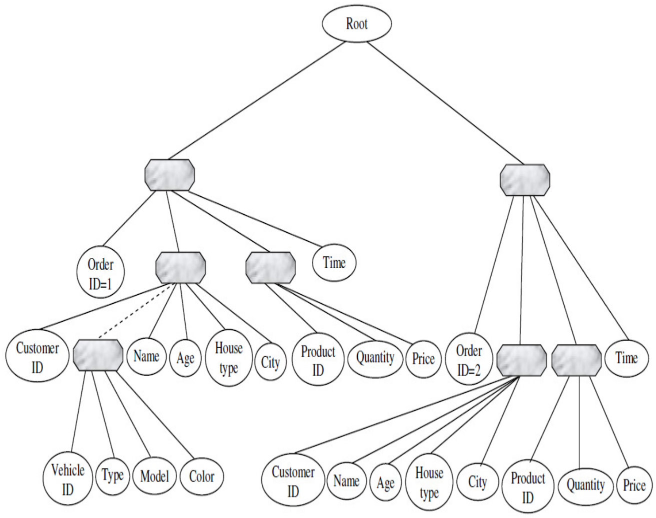

Figure 2. Tree-based representation of the multi-relational database.

3.2. Mining of Relational Patterns

The second step of the proposed approach is to mine the relational patterns from the constructed R_M-tree. An overview about the relational patterns is briefly described in this section. And, we have presented the proposed tree pattern mining algorithm for mining of such relational patterns.

3.2.1. Relational Patterns

The mining of relational pattern is a descriptive mining task which aims to detect associations between target objects and some target-relevant objects. The target objects are represented as tuples of the target table, which cope with the main subject of the description, whereas the target-relevant objects are represented as tuples in source table, that are relevant for the task in dispenses and is associated to that of the former by means of foreign key constraints [4]. The conjunction $X_1 \wedge X_2 \ldots \wedge X_n$ is called relational pattern. Relational pattern is same as the itemsets generated from the Apriori algorithm [1] in which single table is employed to mine the itemsets. An itemset is said to be frequent only if the support of equivalent itemsets is greater than the minimum threshold level.

- *Definition 5:* The *support* of a relational pattern X in a set S, is the number of the tuples in S which contain X versus the total number of tuples in S.
- *Definition 6:* A relational pattern X is frequent in set S if the support of X is above the minimum support threshold, min_sup.

3.2.2. Proposed Tree Pattern Mining Algorithm

From the manually constructed R_M-tree, the relational patterns are mined. For mining such frequent relational patterns, the constructed R_M-tree is given to the proposed tree pattern mining algorithm. The following step shows the procedure for mining relational patterns from the constructed R_M-tree in the proposed approach.

- Building the positional data of R_M-tree.
- Identifying all possible distinctive nodes.
- Discovering the frequent relational patterns for length, L=1.
- Generating join nodes using frequent relational pattern of length, L=1.
- Discovering all frequent relational patterns.

- *Step1. Building the Positional Data of R_M-tree:* The central structure of the R_M-tree shown in Figure 3. Is used for creating the positional data P. The shaded boxes have their own sub-nodes information. The central structure of the R_M-tree is built by finding all possible leaf nodes (Nodes without children) for single orderID in the constructed R_M-tree. The positional data P stores the positional information of each leaf nodes. The string in the positional data p_k (positional information) representing the node of the R_M-tree is formed by adding the label of the tree nodes. The positional data P is then utilized for further processing such as traversing or searching a node in the R_M-tree.

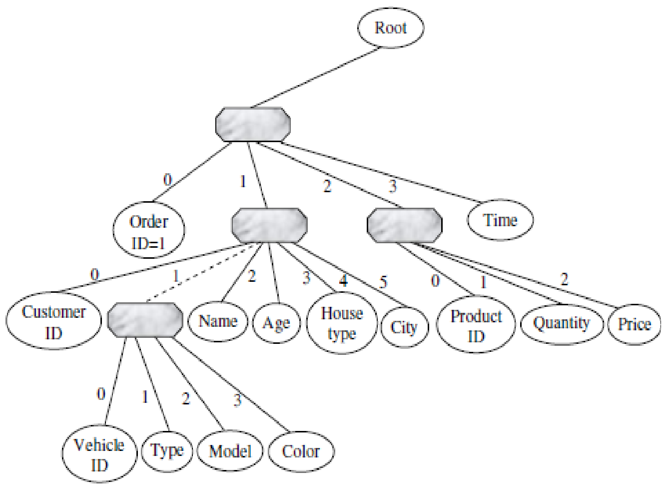

Figure 3. Central structure of the R_M-tree with positional information.

$$P = \{ p_1 \; p_2 \; \cdots \; p_n \}; \quad 1 \le k \le n \tag{1}$$

The number of elements in the positional data P is equivalent to the number of leaf nodes in the central structure of the R_M-tree (n). The positional data P of the constructed R_M-tree is given in Table 1.

Table 1. Positional data P of the constructed R_M-tree.

Positional Data (P)	Positional Information of the Node
p_1	[0]
p_2	[3]
p_3	[1 0]
p_4	[1 2]
p_5	[1 3]
p_6	[1 4]
p_7	[1 5]
p_8	[2 0]
p_9	[2 1]
p_{10}	[2 2]
p_{11}	[1 1 0]
p_{12}	[1 1 1]
p_{13}	[1 1 2]
p_{14}	[1 1 3]

- *Step 2. Identifying all Possible Distinctive Nodes:* For identifying all possible individual nodes, we have used the positional data P. A set of nodes (C_1) has been discovered for every positional information P_k. The C_1 represents the distinct nodes corresponding to every position P_k in the i^{th} order ID of constructed R_M-tree. These nodes are called as candidate nodes for the length, L=1.

$$C_1 = \{ p_k \mid x(p_k) \;\; ; \forall \, p_k \in P \} \tag{2}$$

$$x(p_k) = N_{p_k}^{(i)} \quad ; 1 \le i \le m; \; 1 \le k \le n \tag{3}$$

where,
$m \rightarrow$ Number of nodes (next to the root node) in the constructed R_M-tree.
$N_{p_k}^{(i)}$ $\rightarrow$ Node corresponding to the position P_k in the i^{th} *orderID.*

- *Step 3. Discovering the Frequent Relational Patterns for Length, L=1:* Using step1, we obtained a set of candidate nodes (C_1) for length, L=1 and the support of every candidate node for length L=1 is calculated by finding the number of occurrences of the candidates in the constructed R_M-tree. The candidate node is said to be frequent node, if the support of the candidate node is higher than the user specified threshold, and it is named as 1-length frequent relational pattern, represented in a set F_1.
- *Step 4. Generating Join Nodes Using Frequent Relational Pattern of Length, L=1:* The join nodes are generated by using the frequent relational pattern identified in the previous step. The candidate nodes for length L=2 (C_2) is generated by combining the relational patterns of length L=1 in the first position p_1 with the other position p_k.

$$C_2 = \{ \; p_{kl} \mid x(p_{kl}) \;\; ; \forall p_{kl} \in P \} \tag{4}$$

$$x(p_{kl}) = N_{p_k}^{(i)} \wedge N_{p_l}^{(j)} \; ; 1 \le i \;\&\; j \le m \; ; 1 \le k \;\&\; l \le n \; ; k \ne l \tag{5}$$

where, $N_{p_k}^{(i)}$ *and* $N_{p_l}^{(j)} \ge min\_sup$.

- *Step 5. Discovering all Frequent Relational Patterns:* Subsequently, for every candidate nodes C_2 the support is calculated and the frequent relational pattern for length L=2 (F_2) is discovered based on the minimum support. Similarly, the frequent relational patterns for length L=3 are generated. This procedure is performed repeatedly until we discover all possible frequent relational patterns that vary in terms of length from 1 to n. The pseudo code of the proposed tree pattern mining algorithm is given below:

```
Input:  RM-tree, Positional Data (P), min_sup
Output:  A complete set of frequent relational pattern
Assumptions:
m → Number of nodes (next to the root node) in the
constructed RM-tree
min_sup → Minimum support Threshold
rel_pat → Frequent relational patterns
Pseudo code:
Begin
    for each p ∈ P
       For each node m
            d=distinct data. RM- tree
            if ( support (d) ≥ min_sup)
                 c1[p]<<d
            endif
        endfor
      endfor
    for each p ∈ P
       for (j=1 ; j<size of c1[p] ; j++)
                 node=c1[p,j]
                   do_miner(node,p)
            endfor
    endfor
end
sub routine:do_miner( node_tree , p*)
Begin
    P=p*+1
    for each p ∈ P
       for ( j=1 ; J<size of c1[p] ;  j++)
              node_tree= node_tree  ∧ c1[p,j]
              if(support(node_tree) ≥ min_sup)
                    rel_pat<<node_tree
                      do_miner[node_tree,p]
endif
        endfor
      endfor
end
```

4. Experimentation and Results

The experimental results of the proposed approach for mining the relational pattern are presented in this section. The proposed approach has been implemented in java (jdk 1.6) and the experimentation is performed on a 3.0GHz Pentium PC machine with 2GB main memory.

4.1. Dataset

We have used the synthetic dataset, where the data source is customer order data containing 10,000 orders. The synthetic dataset used for experimentation consists of following relations:

1. The order relation-containing one tuple for every new order and the order table consists of four different fields such as orderID, customerID, productID and time.
2. The customer relation-containing five different fields (customerID, name, age, house type and city) describing the details of the customer. This table contains 500 records where, tuples represents the customers who order the products.
3. The product relation-consisting three different fields (productID, quantity and price). This table consists of 1500 records that describe the details of product.
4. The vehicle relation- consisting of five different fields (vehicleID, customerID, type, model, color). This relational table includes 250 records and every record indicates the details of vehicle.

4.2. Comparative Analysis

The database described in section 4.1 is given as an input to the proposed approach. Our ultimate aim is to mine the relational patterns from the relational database. Based on the proposed approach, we construct the R_M-tree for the input relational database using the proposed approach.

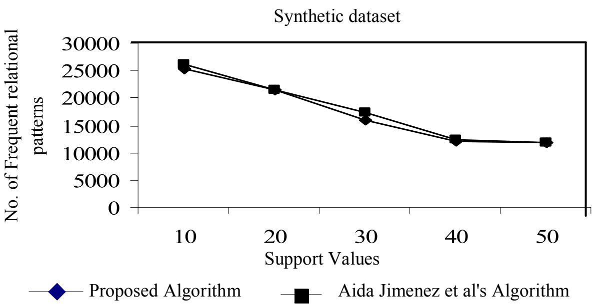

Figure 4. Number of frequent relational patterns for different supports.

The constructed tree contains a set of nodes which are equivalent to the number of tuples in all the relational tables. Then, the relation patterns are mined from the relational database using the proposed approach. The effectiveness of the proposed algorithm was analyzed in terms of number of frequent relational patterns and time required to complete the mining process. Furthermore, we have made a comparative analysis in frequent itemset mining approach for multi-relational database proposed by JimEnez *et al*. [15]. For experimentation, we have discovered the frequent relational patterns for different support threshold. The number of frequent relational patterns generated for different support threshold level is plotted in Figure 4. The run time performance of both the results is given in Figure 5. It shows that the proposed approach performs efficiently to mine the frequent relational patterns from the tree structure, when the support is increased.

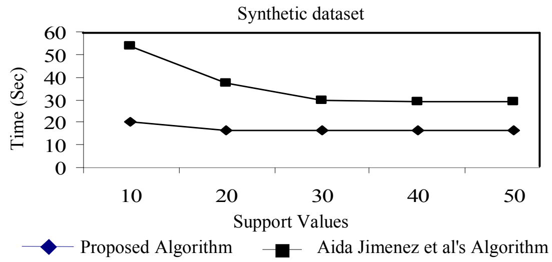

Figure 5. Run time performance of the proposed approach.

5. Conclusions

In this paper, we have presented an efficient approach for mining of relational patterns from the multi-relational database. The proposed approach is composed of two modules namely: 1). tree-based representation, 2). Mining of relational patterns. In the first model, we have constructed the R_M-tree where, the multi-relational database is represented using the tree-based structure. Secondly, we have developed an efficient tree pattern mining algorithm for mining the frequent relational patterns from the constructed R_M-tree. The tree pattern mining algorithm is developed by taking the advantage of the simplicity of tree-based structure. We have used the synthetic datasets for experimentation and the result ensures that the devised approach effectively discovers the relational patterns in the multi-relational database.

References

[1] Agrawal R., Imielinski T., and Swami A., "Mining Association Rules Between Sets of Items in Large Databases," *in Proceedings of the International Conference on Management of Data, ACM SIGMOD*, Washington, pp. 207-216, 1993.

[2] Alfred R. and Kazakov D., "Aggregating Multiple Instances in Relational Database Using Semi-Supervised Genetic Algorithm-Based Clustering Technique," *in Proceedings of the 11th East-European Conference on Advances in Databases and Information Systems*, Bulgaria, pp. 136-147, 2007.

[3] Andrew K. and Matthew S., "Data Mining in Design of Products and Production Systems," *Annual Reviews in Control*, vol. 31, no. 1, pp. 147-156, 2007.

[4] Appice A., Ceci M., and Malerba D., "Mining Relational Association Rules for Propositional Classification," *in Proceedings the 9th Conference on Advances in Artificial Intelligence*, Berlin, vol. 3673, pp. 522-534, 2005.

[5] Blockeel H. and Dzeroski S., "Multi-Relational Data Mining," *in Proceedings of ACM SIGKDD Conference*, Chicago, pp. 126-128, 2005.

[6] Calders T., Goethals B., and Prado A., "Integrating Pattern Mining in Relational Databases," *in Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Berlin, pp. 454-461, 2006.

[7] Ceci M. and Appice A., "Spatial Associative Classification: Propositional vs Structural approach," *Journal of Intelligent Information Systems*, vol. 27, no. 3, pp. 191-213, 2006.

[8] Dzeroski S., "Multi-Relational Data Mining: an Introduction," *ACM SIGKDD Explorations Newsletter, COLUMN: Multi Relational Data Mining*, vol. 5, no. 1, pp. 1-16, 2003.

[9] Dzeroski S. and Raedt L., "Multi-Relational Data Mining," *in Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, USA, vol. 5, pp. 200-202, 2003.

[10] Dzeroski S. and Raedt L., "Multi-Relational Data Mining: The Current Frontiers," *ACM SIGKDD Explorations Newsletter*, vol. 5, no. 1, pp. 100-101, 2003.

[11] Dzeroski S. and Raedt L., "Multi-Relational Data Mining: a Workshop Report," *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 2, pp.122-124, 2002.

[12] Flank A., "Multi-Relational Association Rule Mining," available at: http://www8.cs.umu.se/education/examina/Rapporter/AntonFlank.pdf, last visited 2004.

[13] Guo J., Bian W., and Li J., "Multi Relational Association Rule Mining with Guidance of User," *in Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge Discovery*, Haikou, vol. 2, pp. 704-709, 2007.

[14] Inuzuka N., Motoyama J., Urazawa S., and Nakano T., "Relational Pattern Mining Based on Equivalent Classes of Properties Extracted from Samples," *in proceedings of the 12th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, Berlin, vol. 5012, pp. 582-591, 2008.

[15] JimEnez A., Berzal F., and Cubero J., "Frequent Itemset Mining in Multi-relational Databases," *in Proceedings of the 18th International Symposium on Foundations of Intelligent Systems*, Berlin, vol. 5722, pp. 15-24, 2009.

[16] Kavurucu Y., Senkul P., and Toroslu I., "ILP-Based Concept Discovery in Multi-Relational Data Mining," *An International Journal of Expert Systems with Applications*, vol. 36, no. 9, pp. 11418-11428, 2009.

[17] Liang B., Hong X., Zhang L., and Li S., "Extended MRI-Cube Algorithm for Mining Multi-Relational Patterns," *in Proceedings of the 9th International Conference for Young Computer Scientists*, China, pp. 1132-1136, 2008.

[18] Mahesh M., Rana J., and Jain R., "Use of Domain Knowledge for Fast Mining of Association Rules," *in Proceedings of the International Multi-Conference of Engineers and Computer Scientists*, Hong Kong, pp. 978-988 2009.

[19] Maimon O. and Rokach L., *Data Mining and Knowledge Discovery Handbook*, Springer, New York, 2005.

[20] Makino T. and Inuzuka N., "Implementing Pattern Mining Using Extended Attribute Expression on Relational DB," *in Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, Phuket, pp. 502-505, 2010.

[21] Osmar R., "Introduction to Data Mining," *Principles of Knowledge Discovery in Databases,* University of Alberta, Canada, 1999.

[22] Pizzi L., Ribeiro M., and Vieira M., "Analysis of Hepatitis Dataset using Multi-Relational Association Rules," *in Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases Discovery Challenge*, Slovenia, pp. 161-167, 2005.

[23] Satheesh A., Mishra D., and Patel R., "Classification Rule Mining for Object Oriented Databases: A Brief Review," *in Proceedings of the 1st International Conference on Computational Intelligence, Communication Systems and Networks*, Indore, pp. 259-263, 2009.

[24] Seid D. and Mehrotra S., "Efficient Relationship Pattern Mining Using Multi-Relational Iceberg-Cubes," *in Proceedings of IEEE International Conference on Data Mining*, UK, pp. 515-518, 2004.

[25] Selamat S., Deris M., Mamat R., and Bakar Z., "Mining Least Relational Patterns from Multi Relational Tables," *in Proceedings of Advanced Data Mining and Applications, Lecture Notes in Computer Science*, Berlin, vol. 3584, pp. 59-66, 2005.

[26] Teredesai A., Ahmad M., Kanodia J., and Roger S., "Comma: A Framework for Integrated Multimedia Mining Using Multi-Relational Associations," *Knowledge and Information Systems*, vol. 10, no. 2, pp. 135-162, 2006.

[27] Wainright M., Carol V., Daniel W., Jeffrey A., and William C., *Managing Information Technology*, Pearson Prentice-Hall, 2005.

[28] Wrobel S., "Inductive Logic Programming for Knowledge Discovery in Databases," *in Proceedings of Relational Data Mining*, Berlin, pp. 74-101, 2001.

[29] Yafi E., Al-Hegami A., Alam A., and Biswas R., "YAMI: Incremental Mining of Interesting Association Patterns," *The International Arab Journal of Information Technology*, vol. 9, no. 2, pp. 504-510, 2012.

[30] Yin X. and Han J., "Exploring the Power of Heuristics and Links in Multi-relational Data Mining," *in Proceedings of 17th International Symposium*, Toronto, pp. 17-27, 2008.

[31] Zhang W., "Multi-Relational Data Mining Based on Higher-Order Inductive Logic Programming," *in Proceedings of WRI Global Congress on Intelligent Systems*, Xiamen, vol. 2, pp. 453-458, 2009.

[32] Zhang W., "Mining Multi-Level Multi-Relational Frequent Patterns Based on Conjunctive Query Containment," *in Proceedings of WRI Global Congress on Intelligent Systems*, Xiamen, vol. 2, pp. 436-440, 2009.

[33] Zhu X. and Wu X., "Discovering Relational Patterns Across Multiple Databases," *in proceedings of the IEEE 23rd International Conference on Data Engineering*, Turkey, pp. 726-735, 2007.

Vimal Dhanasekar received MCA degree at KSR College of Technology, Periyar University from the Department of Master of Computer Applications, India, in 2002. He received his M.Phil Computer Science degree at Kongu arts & Science College, Bharathiar University in the Year 2007. He has presented papers in National and International Conferences. His area of interest includes data mining, network, software engineering, mobile computing and image processing. He is currently pursuing the PhD degree working closely with Prof. A.Tamilarasi Simultaneously. He is working as an associate professor and head in the Department of MCA in SAN International Information, Anna University of Technology, Coimbatore.

Tamilarasi Angamuthu post graduated from Bharathiar University, India in 1986. She obtained her PhD from University of Madras, Chennai in 1994. She was awarded JRF by UGC in the year 1986. She has also published more than 40 research papers in the reputed national/international Journals also, she is the author of 10 books. Her area of interest includes semigroup theory, software computing, data mining. Presently, she is working as a head of the Department & Professor Department of MCA, Kongu Engineering College, India