CHAPTER 4

T-ID based RTPM: RELATIONAL TUPLE ID BASED TREE PATTERN MINING ALGORITHM

4.1 OVERVIEW

Various real-world database organized in relational datasets comprising of multiple tables and associations. The other forms of information such as in bioinformatics, computational biology, HTML and XML documents need a perceptive regarding the arrangement of the object. Still, maximum existing methodologies to machine learning naturally adopt that the information preserved in a single table, and make use of propositional language for determining analytical models. Therefore, there is a requisite for data mining algorithms to determine a-priori unidentified relationships from multi-relational data. There are numerous existing data mining techniques presented, which search for pattern in single table, but in real world, information is stored in structured layout known as relational database. Such a data set contains multiple relation interconnected together theoretically through an entity relationship links.
An accumulative amount of data mining applications encompasses the exploration of composite and organized form of data and necessitates the use of expressive pattern languages.

Recent improvements in extraordinary amount of information achievement, digital storage, and communications skills have prepared it conceivable to collect huge volumes of data in several technical and commercial areas. Ample of this information exist in relational databases. Although, the data warehouse is not a relational database, it is frequently appropriate to sight varied information sources as if they are group of relations [127] for mining and establishing data from multiple sources. Therefore, the job of educating from relational data has activated to obtain important responsiveness in the research. Multi-relational data mining framework depends on searching of fascinating patterns in the relational database, where multi-relational patterns observed as fragments of sub patterns met in the arrangement of the objects of importance [26].

Data mining is a procedure used to recognize hider, unpredicted pattern, or relationships in great magnitudes of data. Traditionally, the idea of discovering suitable patterns in database given various names, comprising data mining, knowledge abstraction, information detection, data collecting, and data pattern processing. Mathematicians, data experts, and the Management Information Systems (MIS) groups have
typically make use of the word data mining. KDD been propagated in the artificial intelligence and machine learning domains.

In this chapter, mining of tree patterns are processed same as chapter 3 for the medical database using Tuple ID Propagation Technique. The proposed methodology known as relational tuple id based tree pattern mining. This methodology proposed to increase the performance and efficiency of the algorithm compared to the other algorithm. At first, the preprocessing applied to the multi relational database given in section 4. After preprocessing the database, a tree constructed using Tuple ID Propagation technique and calculating the information gain for the tuple (attribute). The tree pattern-mining algorithm applied on the constructed tree to extract the relevant multi relational patterns from the data set.

The rest of the chapter organized in the following way. Multi Relational Classification (MRC) is one of the field in the mining algorithms that deal with tuple id propagation discussed in section 4.2. The section 4.3 discusses about the relational learning of the multi relational database. Section 4.4 briefly discusses the Tuple ID Propagation technique given in the proposed methodology. Finally, section 4.5 gives the brief discussion of the proposed relational tuple id propagation tree pattern mining algorithm methodology.
4.2 MULTI RELATIONAL CLASSIFICATION (MRC)

Multi-relational classification (MRC) is one of the most quickly increasing subfields of multi relational data mining which concepts a classification model that employs information collected in numerous relations. Multi Relational Classification (MRC) is a significant job in data mining and machine learning, which is deliberated comprehensively and has a varied range of applications. Several classification problem occurred and there is a need to solve. There are different types of classification algorithms like tree-based, rule-based etc., are broadly used. Multi-relational data mining studies the interesting patterns straightly from several interconnected tables with the provision of primary key /foreign keys. Multi-relational classification is the process of constructing a classifier grounded on information deposited in multiple relations. Therefore, to categorize object in one relation, the information from another relation is to calculate. Multi relational classification used to forecast behavior and unidentified patterns mechanically.

In common, there are two methods available for MRC. i) Exploit conventional data mining methods, known as propositionalisation, which modify multiple relational data into a single flat data relative by means of physical joins and aggregations. Subsequently, some vital information taken by the relations is lost and produces a tremendously fat table with vast number of extra attributes and abundant missing values [155]. ii) The
surviving classification algorithms can be revived to handle the data in the multiple tables [154, 21, 72, 119].

To challenge the multi-relational classification issue in relational databases, there are two foremost tasks: one is effectiveness and scalability, and the other is the exactness of classification. While constructing a classifier for a database with various relations, the search area is commonly huge, and it is unreasonable to accomplish comprehensive exploration. Conversely, the semantic links commonly turn into very feeble after moving through a sequence of links. Thus, a multi-relational classifier requires handling both effectiveness and exactness issue. So far, there is again deficiency of precise, effective, and scalable multi-relational classification methods to handle enormous data sets with compound schematics.

4.3 RELATIONAL LEARNING

Conventionally, research in data mining has concentrated primarily on attribute-value learning where every instance or occurrences described by a static group of attributes for which values prescribed. The elements in this case observed as a table or a relation where every row resembles to an instance and every column to an attribute. Most prevailing data sets in diverse fields not deposited as a single relation, but as numerous relations due to non-redundancy and storage and access competence. Primarily, attribute-value learning cannot signify this type of circumstantial
knowledge or catch associations among various records from more than one table. Even though in particular, it is likely to restructure a single relation from multiple tables; such methodology is apprehensive with countless complications in nature.

Luc De Raedt in [133] summaries of how an exceptional case of relational learning can be altered into attribute-value learning. However, in the simple cases, if conventional or propositional learning algorithms needed to use then the relational data base that contains multiple tables needed to reconstruct into a single table. There are two common traditions of renovating a multi relational database into a single relation database.

1. Produce a general relation that encompasses the combination of all the multi relational tables to form a single one. Still, there are some prospective problems to this method:

   - The resultant worldwide relation can be tremendously huge and unfeasible to handle.
   - Improved data redundancy resulting in a significant upsurge in the magnitude of the data set
   - The data replication causing from flattening the data might familiarize statistical angle [136]
   - Attribute value learning comes to be more incompetent compared to first order learning on some group of problem [133].
Order Bayesian Networks, SLP: Stochastic Logic Program, MRDM: Multi-Relational Data Mining MRDT: Multi-Relational Decision Tree, FOIL: First Order Inductive Logic”

Fig. 4.1: Relationships between propositional and relational learning algorithms

2. Alter a multiple relation data set into a single relation by generating novel attributes in a dominant relation that encapsulates or combines information from other tables. This methodology shares alike disadvantage with the previous technique along with the extra one. Though the novel attributes do not augment additional data regarding the samples in the database, it might not be easy to discovery suitable ones. The particular stage might need a countless volume of domain understanding and it is fragment of the knowledge discovery procedure. One of the method to deal with this is to allow the attribute value beginner itself to arise using worthy attributes or to expand the hypothesis region by permitting assessments including multiple attributes [91].

4.4 TUPLE ID PROPAGATION TECHNIQUE

In a data set for multi-relational classification, there is single target relation $R_t$ and tuples in it are named as target tuples. Every target tuple connected with a class label. The remaining relations are non-target relations and might encompass appropriate statistics to supports classification. To figure a decent multi relational classifier, once requires
discovering decent predicates in every non-target relation R that assist to
differentiate positive and negative target tuples. Numerous ILP methods
are not effective since they calculate numerous predicates independently,
and consider one or more join functions to estimate all predicates. One of
the way to decrease the calculation cost is to initially combine the target
relation with remaining relations and calculate the predicates on the
outcome relation. The countless predicates assessed instantaneously by
perusing the output relation only once. Nevertheless, recurrently discover
good predicates to figure strategies, one desires to combine various
relations in diverse ways that is lavish in both time and space.

4.4.1. Basic Definitions

A database comprises of a group of relations, where target relation
is one of them, with class labels accompanying its tuples [121]. The
remaining relations are non-target relations. Every relation might have
single primary key and numerous foreign keys, each indicating to the
primary key of some other relation. The subsequent types of joins
considered:

- “Join between a primary key k and certain foreign key indicating to
  k.”
- “Join between two foreign keys k1 and k2, which indicate to the
  similar primary key k.”
The other probable joins ignored since they do not signify robust semantic associations among the objects in the data set. The goodness of predicates defined first. A predicate is a limitation on certain attribute in certain relation. Assuming that primary key of the target relation is an attribute of integers that signifies the ID of every target tuple. Instead of accomplishment physical join, the IDs and class labels of target tuples broadcasted to the Account relation. The procedure officially given as follows.

“Definition 1 (Tuple ID propagation): Suppose two relations $R_1$ and $R_2$ combined by attributes $R_1: A$ and $R_2: A$. Every tuple $t$ in $R_1$ related with a group of IDs in the target relation, symbolized by idset ($t$). For every tuple $u$ in $R_2$, idset ($u$) = $\bigcup_{t \in R_1, t.A = u.A} \text{idset}(t)$ is set.”

The below given theorem and its corollary demonstrate the exactness of tuple ID propagation.

“Theorem 1: Suppose two relations $R_1$ and $R_2$ combined by attribute $R_1: A$ and $R_2: A$, and $R_1$ is the target relation, with primary key $R_1: \text{id}$. All the tuples in $R_1$ satisfy the existing rule. The existing rule incorporates a predicate $R_1$ ($R_1: \text{id}, R_1: A \ldots$), which allows the combination of $R_1$ with $R_2$. Using tuple ID propagation from $R_1$ to $R_2$, for every tuple $u$ in $R_2$, idset ($u$) signifies all target tuples combinable with $u$, with the help of the join path stated in the existing rule.”
“Corollary 1: Suppose two relations R1 and R2 combined by attribute R1: A and R2: A, R1 is the target relation, and all the tuples in R1 satisfy the existing rule by eradicate the remaining ones. If R1's IDs propagated to R2, then the foil gain of each predicate in R2 calculated with the help of the propagated IDs on R2.”

Individually with the propagated IDs, one can determine the precise group of target tuples that please an instruction. In addition to propagating IDs from the target relation to the relations straightly combinable with it, IDs propagated transitively by propagating the IDs from one non-target relation to another, agreeing to the subsequent theorem.

“Theorem 2: Suppose two non-target relations R2 and R3 can be combined by attribute R2: A and R3: A, and all the tuples in R2 satisfy the existing rule by eradicate the remaining ones. For each tuple v in R2, idset (v) signifies the target tuples combinable with v (using the join path specified by the current rule). By propagating IDs from R2 to R3 through the join R2: A = R3: A, for every tuple u in R3, idset (u) denotes target tuples that can be joined with u with the help of join path in the existing rule and the join R2: A = R3: A.”

Consider an example multi-relational database that shown in Table 4.1. The target table is student and its primary key consists of set of integers, which denotes the ID of each target tuples. The physical joins
between these tables shown in Table 4.2 and the virtual join using tuple id propagation shown in Table 4.3, which is more powerful than physical join.

Tuple ID propagation is a technique for fundamentally connecting non-target associations with the target one. This propagation is stretchy and active process, and is less expensive compared to physical join in both time and space [114]. Using tuple ID propagation among non-target relations, the foil gain depending on the propagated IDs calculated. This propagation is a method to accomplish virtual join. Instead of materially combining the associations, the tables virtually combined by assigning the tuple IDs of the target relation to the tuples of a non-target relation with the help of certain join path. Similarly, the semantic associations among the target relation and non-target relations originated, and the foil gain of predicates calculated as if physical join is accomplished.

Table 4.1: Example Multi Relational Database

<table>
<thead>
<tr>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sid</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>Cno</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
</tbody>
</table>

Table 4.2: The join of student and course

<table>
<thead>
<tr>
<th>Sid</th>
<th>Cno</th>
<th>Grade</th>
<th>Cname</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>A</td>
<td>Java</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>B</td>
<td>Perl</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>C</td>
<td>PHP</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>A</td>
<td>Perl</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>C</td>
<td>PHP</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 4.3: Result of Tuple Id Propagation

<table>
<thead>
<tr>
<th>Cno</th>
<th>Cname</th>
<th>Credits</th>
<th>IDs</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Java</td>
<td>3</td>
<td>1</td>
<td>1+, 0-</td>
</tr>
<tr>
<td>12</td>
<td>Perl</td>
<td>2</td>
<td>2,4</td>
<td>1+, 1-</td>
</tr>
<tr>
<td>13</td>
<td>PHP</td>
<td>4</td>
<td>3,5</td>
<td>1+, 1-</td>
</tr>
</tbody>
</table>
Tuple ID propagation is an operative and expandable procedure since little bit of information is stimulated among target and non-target tables that desires only small capacity of additional storage region [122]. It is a way to transfer information between diverse relations by virtually joining them. The Propagation technique reveals to explore in the relational database and perceived it is less cost compared to physical joins in both time and space. It is the procedure for accomplishment virtual joins among relations, which is little extensive compared to physical joins. When there is, need to look for a good predicate then propagate Tuple IDs used amongst any two relations that delivers fewer calculation and storage cost related to generating join necessities [122].

IDs and their related class labels naturally broadcasted from one to other relations. In this way, predicates in diverse relations computed with minor recurrent computations. The essential region is also less since the IDs do not occupy much added room for the data. ID propagation, though treasured, imposed with semantic constrictions. There are two circumstances where the propagation could be counter-productive: (1) propagate via large fan-outs, and (2) propagate via long weak links. The former issue occurs if there are numerous tuples created with the help of propagation. Supposing after the IDs broadcasted to a relation R, it is to observe each tuple in R combined to various target tuples and every target tuple combined to various tuples in R. The semantic relation between R and the target relation is generally feeble since the relation is
indiscriminate. For example, propagation among people by means of birth-country relations might not be creative. The latter issue occurs if the propagation passes along extensive feeble relations, e.g., relating a student with his car dealer's pet might not be creative either. From the view point of both efficiency and accuracy, there are two issues found where propagates could be counterproductive. They are: (1) have very large fan-out, (2) not on active relations.

*The Drawback is:*

Propagates with large fan-outs

- Propagate with long weak relations

The two most challenges these are as below:

*Challenge-I: Finding the Useful Relations*

Solution for Problem-I: It is essential to discover the procedures that can calculate the effectiveness of relations across tables and then apply the most suitable relations to attain the better data-mining task.

*Challenge-II: Transferring information proficiently*

Solution for Problem-II: There is a need to cultivate the approaches with as little inter database communication cost as possible.
Limitations:

- Improving scalability by straightly using database operation to attain Tuple ID propagation
- Occasionally too many IDs propagated to every tuple in a relation, which makes it firm to limit the time and space complexity of the algorithm.

4.5 RELATIONAL TUPLE ID PROPAGATION TREE PATTERN MINING ALGORITHM

Data mining is the procedure of examining information from diverse perceptions, concluding it into beneficial data, and discovering diverse patterns (e.g. classification, regression, and clustering). Several issues are complex to resolve systematically in a reasonable time. Hence, scholars are trying to find exploration techniques or heuristics to attain a good adequate or acceptable solution in a rational time [83, 6]. This chapter proposed a methodology known as Relational Tuple ID Propagation Tree Pattern Mining Algorithm to mine the multi relational database. Same as chapter 3, this chapter also make the experimental analysis on the medical database for mining the relational patterns but using the tuple ID propagation technique. Fig 3.3 represents the patient’s hospital multi relational database. The proposed methodology broadly divided into three stages for mining the medical database efficiently. They are:

1. Preprocessing of entire hospital medical database
2. Converting the entire preprocessed data into relational tree structure

3. Mining the important relational patterns from the constructed multi-relational tree

![Data Preprocessing Flow Chart](image)

**Fig 4.2 Data Preprocessing Flow Cart**

### 4.5.1 Data Preprocessing of Multi Relational database

Since the data is often collected for unspecified applications. Real-world data is often imperfect, unpredictable, and/or lacking in certain
performances or tendencies, and is likely to comprise numerous errors. Data preprocessing is a verified method of resolving such issues. Preprocessing needed to make data more suitable for data mining. The preprocessing of raw database helps to improve the performance and efficiency of algorithm in further implementation. This preprocessed removes the unwanted, incomplete and inconsistent noise and data from the database, include the missing values in the database. Database preprocessing is very necessary to prepare database to use in knowledge extraction by data mining. In Relational Tuple ID Propagation, the hospital multi relational database as shown in Fig 3.3 use for preprocessing. This is first stage in the proposed methodology. Fig 4.2 represents the flow chart for data preprocessing. The procedure for preprocessing the database given as follows:

1. The preprocessing applied on the target relations of the medical multi relational database only.
2. The number of same class labels or class categories is determined in the target relation.
3. Then a symbolic name assigned to each class category of the target relation.
4. The symbolic names that are assign filled inside the class labels column in the multi relational database respectively.
5. Finally, a preprocessed database obtained that is use for construction of relational tree.
4.5.2 Construction of Multi Relational Tree Structure

In the proposed methodology, a preprocessed database obtained from the multi relational database. This processed database used to construct the relational tree structure. The proposed methodology uses Tuple ID Propagation technique and Information Gain to construct the tree. Information gain defined as “the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on attribute, A)”. This technique briefly discussed in section 4.3 and its advantages and limitations. This is the second and most important stage in the proposed methodology. Algorithm 4.1 illustrates the procedure for construction of relation tree structure from the database.

Algorithm 4.1: Construction Relation Tree

1. Consider the preprocessed multi relational database as input.
2. The Tuple Id propagation technique performed on the database.
3. Count the number of target tuples present in the target relation.
4. If no. of target tuples < MIN_SUP then Return. (MIN_SUP=Minimum No. of tuples required for classification)
5. Evaluate all attributes (A) of the target relation and calculate the Information Gain of each attribute.

Information gain of class label calculated by
Information gain of attribute calculated by

\[ \text{Info}(D) = -\sum_{i=1}^{m} p_i \log_2(p_i), \]

Information Gain calculated by

\[ \text{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \text{Info}(D_j). \]

6. If Information Gain (A) < MIN_INFO_GAIN then return.

7. The attribute with highest INFO_GAIN becomes root node of the tree.

8. Then divide the target relation according to root node.

9. Repeat until all attributes evaluated and finally set the class label with higher frequency.

10. Thus, a relational tree structure obtained with the attributes as the nodes of the tree.

**4.5.3 Mining of Important Relational Patterns**

This is the third stage in the proposed relational Tuple Id Propagation methodology. In this stage, the important patterns mined or extracted from the constructed relational structure using Tuple Id Propagation technique. The attributes or the nodes constructed tree
known as patterns user for mining from the relational tree. The multi-relational tree contains $M-(1+n(f))$ number of length and every length has $T$ number of nodes with each having count of their individual. Here $M$ signifies the total number of attributed present in the tree and $n(f)$ represents number of foreign keys present in the database table. The system visits single every node of the each length patterns. If count value of that particular node greater than user-defined threshold then that relation said to be a significant amalgamation. The cnt_val of node that does not support to the threshold then such a node not considered as significant node. Algorithm 4.2 represents the pseudo code for mining of significant relations from the tree.

**Algorithm 4.2: Mining of Multi Relational Patterns**

Begin

For each length pattern

If (cnt-val (subnode (i)) > Th)

Add that subnode into set of important relations \{imp relations ()\}

Go to next subnode i+1

Else

Terminate the (subnode (i))
End if

End for

End