An Efficient and Effective Method for Sequential Rule Mining

Vinay Raj Pandey, Shivesh Tiwari, Arun Kumar Shukla, Ashutosh Shukla

Abstract --- Tremendous amount of data being collected is increasing speedily by computerized applications around the world. Hidden in the vast data, the valuable information is attracting researchers of multiple disciplines to study effective approaches to derive useful knowledge from within. This thesis aims to investigate efficient algorithm for mining including association rules and sequential patterns. Mining sequential patterns with time constraints, such as time gaps and sliding time-window, may reinforce the accuracy of mining results. However, the capabilities to mine the time-constrained patterns were previously available only within Apriori framework. Recent studies indicate that pattern-growth methodology could speed up sequence mining. Current algorithms use a generate-candidate-and-test approach that may generate a large amount of candidates for dense datasets. Many candidates do not appear in the database. Therefore we are introducing a more efficient algorithm for sequential rule mining. The time & space consumption of proposed algorithm will be lesser in comparison to previous algorithms.

Keywords--- Sequential rule Mining, Confidence, Support

I. INTRODUCTION

Data mining is the process of extracting interesting (nontrivial, implicit, previously unknown and potentially useful) information or patterns from large information repositories such as: relational database, data warehouses, XML repository, etc. Also data mining is known as one of the core processes of Knowledge Discovery in Database (KDD). Of all the mining functions in the knowledge discovering process, frequent pattern mining is to find out the frequently occurred patterns. The measure of frequent patterns is a user specified threshold that indicates the minimum occurring frequency of the pattern. We may categorize recent studies in frequent pattern mining into the discovery of association rules and the discovery of sequential patterns. Association discovery finds closely correlated sets so that the presence of some elements in a frequent set will imply the presence of the remaining elements (in the same set). Sequential pattern discovery finds temporal associations so that not only closely correlated sets but also their relationships in time are uncovered.

Fig. 1.1: The process of knowledge discovery in databases

In a Sequence Database, each sequence is an ordered list of itemsets. An itemset is an unordered set of items (symbols), considered to occur simultaneously. Sequential Pattern Mining is probably the most popular set of techniques for discovering temporal patterns in sequence Databases. SPM finds subsequences that are common to more than minsup sequences. SPM is limited for making predictions. For example, consider the pattern \{x\},\{y\}. It is possible that y appears frequently after an x but that there are also many cases where x is not followed by y. For prediction, we need a measurement of the confidence that if x occurs, y will occur afterward A sequential rule typically has the form X→Y. A sequential rule X→Y has two properties:

I. Support: the number of sequences where X occurs before Y, divided by the number of sequences.

II. Confidence the number of sequences where X occurs before Y, divided by the number of sequences where X occurs.

Sequential Rule Mining finds all valid rules, rules with a support and confidence not less than user-defined thresholds minSup and minConf.

For Example: An example of Sequential Rule Mining is as follows:

Consider minSup= 0.5 and minConf= 0.5:

Fig. 1.2: A sequence database
An Efficient and Effective Method for Sequential Rule Mining

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Support</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>{a, b, c} \rightarrow {e}</td>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>r2</td>
<td>{a} \rightarrow {e, f}</td>
<td>0.5</td>
<td>0.65</td>
</tr>
<tr>
<td>r3</td>
<td>{a, b} \rightarrow {e, f}</td>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>r4</td>
<td>{b} \rightarrow {e, f}</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>r5</td>
<td>{a} \rightarrow {e, f}</td>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>r6</td>
<td>{c} \rightarrow {f}</td>
<td>0.5</td>
<td>0.65</td>
</tr>
</tbody>
</table>

Fig. 1.3: Some Rules Found

II. BACKGROUND AND RELATED WORK

Association rule mining (Agrawal et al., 1993) is a popular knowledge discovery technique for discovering associations between items from a transaction database. Formally, a transaction database \( D \) is defined as a set of transactions \( T=\{t_1, t_2, \ldots t_n\} \) and a set of items \( I=\{i_1, i_2, \ldots i_m\} \), where \( t_1, t_2, \ldots t_n \subseteq I \). The support of an itemset \( X \subseteq I \) for a database is denoted as \( \text{sup}(X) \) and is calculated as the number of transactions that contains \( X \). The problem of mining association rules from a transaction database is to find all association rules \( X \rightarrow Y \), such that \( X, Y \subseteq I \), \( X \cap Y = \emptyset \), and that the rules respect some minimal interestingness criteria. The two interestingness criteria initially proposed (Agrawal et al., 1993) are that mined rules have a support greater or equal to a user-defined threshold \( \text{minsup} \) and a confidence greater or equal to a user-defined threshold \( \text{minconf} \). The support of a rule \( X \rightarrow Y \) is defined as \( \text{sup}(X \cup Y) / |T| \). The confidence of a rule is defined as \( \text{conf}(X \rightarrow Y) = \text{sup}(X \cup Y) / \text{sup}(X) \). Since \( |T| \geq \text{sup}(X) \) for any \( X \subseteq I \), the relation \( \text{conf}(r) \geq \text{sup}(r) \) hold for any association rule \( r \). Association rules are mined from transaction databases. A generalization of a transaction database that contains time information about the occurrence of items is a sequence database (Agrawal & Srikant, 1995). A sequence database \( SD \) is defined as a set of sequences \( S=\{s_1, s_2, \ldots s_n\} \) and a set of items \( I=\{i_1, i_2, \ldots i_m\} \), where each sequence \( s_x \) is an ordered list of transactions \( s_x=\{X_1, X_2, \ldots X_n\} \) such that \( X_1, X_2, \ldots X_n \subseteq I \).

III. PROPOSED SOLUTION

We will propose a novel algorithm for mining sequential rules common to several sequences. Unlike other algorithms, new algorithm uses a pattern-growth approach for discovering sequential rules such that it can be much more efficient and scalable. The proposed algorithm will outperform CMRules and CMDeo in terms of execution time and memory usage.

Algorithm:
The algorithm that we propose uses an approach that is different from CMDeo and CMRules. Instead of using a generate-candidate-and-test approach, it relies on a Pattern-Growth approach similar to the one used in the PrefixSpan [7] algorithm for sequential pattern mining. Our algorithm first find rules between two items and then recursively grow them by scanning the database for single items that could expand their left or right parts (these processes are called left and right expansions). Like PrefixSpan, Our algorithm also includes some ideas to prevent scanning the whole database every time. The idea of proposed algorithm is to grow rule by starting with rules of size 1*1 and to recursively add one item at a time to the left or right side of a rule (left/right expansions) to find larger rules.

Input:
1: A source database \( D \).
2: MST (Minimum Support Threshold).
3: MCT (Minimum Confidence Threshold).

Output:
A set of sequential rules

IV. CONCLUSION

In this paper, we presented a novel algorithm for mining sequential rules common to several sequences. Unlike previous algorithms, it does not use a generate-candidate-and-test approach. Instead, it uses a pattern-growth approach for discovering valid rules such that it can be much more efficient and scalable. It first finds rules between two items and then recursively grows them by scanning the database for single items that could expand their left or right parts. We have evaluated the performance of our algorithm by comparing it with the CMDeo and CMRules algorithms. Results show that our algorithm clearly outperforms CMRules and CMDeo and has a better scalability.

REFERENCES
Vinay Raj Pandey is currently pursuing Master of Technology (M.Tech) in Computer Science and Engineering from Sam Higginbothom Institute of Agriculture, Technology & Sciences, Deemed University from Allahabad (U.P.) . He has done his Master of Computer Application (MCA) from Ewing Christian Institute of Management & Technology (ECIMT), Allahabad (U.P) in the year 2010.

Shivesh Tiwari is currently pursuing Master of Technology (M.Tech) in Computer Science and Engineering from Swami Vivekanand University, Sagar (M.P) . He has done his Masters in Cyber Law & Information Security (MSCLIS) from Indian Institute of Information Technology (IIIT), Allahabad (U.P) in the year 2008. He has done his Bachelor of Technology (B.Tech) in Computer Science & Engineering from United College of Engineering & Research, Allahabad in the year 2004.

Arun Kumar Shukla is currently working as Assistant Professor in the Computer Science and IT Department in SHIATS, Allahabad. He has done his Bachelor of Engineering from Jawaharlal Institute of Technology from Khargone (M.P.) in the year 2006 and Mater of Engineering from I.E.T. –D.A.V.V., Indore in the year 2011.

Ashutosh Shukla is currently pursuing Ph.D in Computer Science and Engineering from Birla Institute of Technology Mesra Ranchi, Jharkhand. He has done his Bachelor of Technology from United College of Engineering & Research from Allahabad (U.P.) in the year 2006 and Mater of Technology from Sam Higginbottom Institute of Agriculture, Technology & Sciences, Deemed University from Allahabad (U.P.) in the year 2013.