World Engineering & Applied Sciences Journal 6 (1): 64-69, 2015
ISSN 2079-2204

DOI: 10.5829/idosi.weasj.2015.6.1.22179

Mining Hidden Granules from Relational Database Systems

[1]V. Ajitha and [2]P. Rajkumar

[1]CSE Department, Saveetha Engineering College, Thandalam, Chennai-602105, India
[2]Bannari Amman Institute of Technology, Erode, India

Abstract: Granular association rule mining uses granules to represent the knowledge implicitly contained in the databases. It is a two stage process where in the first stage all the frequent granules are discovered and in the second stage association rules are generated from the frequent granules. To generate all kinds of rules we developed fast forward, backward, sandwich methods. This research provides efficient methods to interpret meaningful discovered knowledge such as methods to represent the discovered granules and associations. By this way, meaningless association rules can be adjusted. Experiments are done on a publicly available dataset. The results indicates the viability of the proposed research by setting the appropriate threshold which helps in obtaining high accuracy.

Key words: Granules · Partial match · Complete match

INTRODUCTION

Association rule mining is one form of data mining that finds association among attributes of transactions. In Boolean association rule, the information is stored in a Boolean database which reveals the connection between two disjoint subsets of the same universe. Quantitative association rules are Multidimensional association rules in which numeric attributes are dynamically discretised. The relational association rule mining looks for patterns that involves multiple tables. Efficient rule mining algorithms are developed to discover knowledge from the databases. But there are some difficulties when we apply these to solve real world problems. The major challenging problem in rule generation is the assessment of derived rules based on the quality. Predictive accuracy of a decision rule is measured by applying separate training dataset and separate test dataset, which contains data instances that were not seen during training. Although this is a widely used measure to assess the quality of a rule, it does not take into account the problem of uncertainty. The larger the results, the greater redundancy exists in the patterns and rules which are not interesting for users. The other problems that are encountered are rule generation takes too much of time. Interpretability will be the issue if there is huge number of patterns. So when the dimension of the input data increases, the accuracy and efficiency of the results decreases rapidly. Thus the worth and the knowledge discovery depreciate. Finally, as the approaches uses only two measures like support and confidence, knowledge coverage becomes incomplete. However if the support and confidence value are low, then it results in large volume of results. In some cases, the entire knowledge is not always necessary to define various processes in the dataset. This motivates the need for efficient ways of representing and interpreting the discovered patterns and the rules [1].

Related Work: A fuzzy method proposed by Ma *et al*. (2010) discovered some potentially more interesting association rules. Zailani *et al* (2010) proposed a trie-based algorithm that generates the significant patterns using support and correlations. Aouad *et al* (2010) compared the proposed approach with a classical Apriori-like distributed algorithm. Many applications directly or indirectly rely on finding the frequent items. Tremblay *et al* (2010) proposed a methodology to discover patterns in related attribute values. Yin Kuo-Cheng *et al* (2010) proposed temporal association rule mining algorithm which automatically generates all the intervals without using any domain specific information. WEI Yong-Qing, *et al* (2010)[1] proposed an improved apriori algorithm is

Correspnding Author: P. Rajkumar, Bannari Amman Institute of Technology, Erode, India.

used minimum supporting degree and degree of confidence,for extracting association rules.But it has suffered from "frequent pattern sets explodes "and "rare item dilemma ".XING Xue *et al* (2010)[2]reveal knowledge hidden in the massive database and proposed an approach for Evaluation of exam paper. This paper introduces a new direction, applies interesting rules mining to evolution of completive exam and finds out some useful knowledge. But this algorithm need repeated database scan and takes more time to perform I/O operation. Wang *et al* (2011)[3] presented Apriori association rule algorithm for analysing the performance of college students. Rama subbareddy *et al* (2011) proposed an approach for mining the positive and the indirect negative associations between itemsets. Abdullah *et al* (2011) proposed a new measure to discover the significant association rules. Even though several different approaches to association rule mining are presented, starting from traditional approaches, followed by multilevel and cross-level approaches, all those focused on the proposal of different types of algorithms for Association Rule Mining with the measures support and confidence. However, the focus of recent research is on improving the efficiency of these algorithms using measures like source coverage, target coverage, source confidence and target confidence. Therefore, new algorithms have been proposed in this research work to enhance the capabilities of the existing Association Rule Mining algorithms in terms of the number of rules, checking time and basic operations[2].

Preliminaries

Generating Frequent Granules Using Apriori: The Apriori Algorithm is for mining frequent itemsets for Boolean association rules. First fin the frequent itemsets which is the set of items that have minimum support. i.e., if {AB} is a frequent itemset, both {A} and {B} should be a frequent itemset [3].

The Apriori Algorithm: Pseudo Code:

Join Step: Candidate itemset Ck is generated by joining Lk-1

Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

Pseudo-Code:

C_k: Candidate itemset of size k

L_k: frequent itemset of size k

L_1 = {frequent items};
    for (k = 1; L_k !=∅; k++) do begin
C_{k+1} = candidates that are generated from L_k;
    for each of the transaction t in database do
        increment the count of all candidates in C_{k+1} that are contained in t
    L_{k+1} = candidates in C_{k+1} with minimum support
    end
return $\cup_k L_k$;

Generating Frequent Granules Using Frequent-Pattern Growth (FP-Growth) Method: It compress a large database into a compact tree structure. It is highly condensed, but complete for frequent pattern mining and avoid costly database scans.FP-tree-based frequent pattern mining method is an efficient divide-and-conquer methodology which decomposes mining tasks into smaller ones and avoids candidate generation. To construct FP tree,first, create the root of the tree,with "null".Then scan the database D a second time.The items in each transaction are processed in L order (i.e. sorted order).A branch gets created for each transaction having their support count separated by colon.Whenever the same node is encountered in other transaction, we increment the support count of the common node or Prefix. For tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links. Thus mining of frequent patterns is transformed to that of mining the FP-Tree. Thus the performance study shows that FP-growth is an order of magnitude faster than Apriori and is also faster than tree-projection due to no candidate generation, no candidate test, uses compact data structure, eliminates repeated database scans and the basic operation is counting and FP-tree building[4].

Proposed System: The proposed system uses the following five measures to evaluate the quality of rules and represent the patterns in terms of granules. Associations can be organized and represented at multiple levels using granules. Consider the association rule

Faculty Teaches Courses: This definition is ambiguous and the following questions arises.

a. Do all female faculty teaches engineering?
b. Do all female faculty teaches all stream?
c. How many faculties are female?
d. How many courses are in Engineering?

To address these issues, we present four subtypes of rules

a. Complete match rules
   (eg)All faculty teaches all stream.
b. Left Hand side partial Match rules
   (eg)40% faculty teaches all stream.
c. Right HandSide partial Match rules
   (eg)All faculty teaches atleast 30% vocational courses.
d. Partial match rules
   (eg)40% of faculty teaches atleast 30% vocational courses.

To evaluate the quality of rules, five measures are defined.

a. Support which reflects the usefulness of the rule
b. Confidence which reflects the certainty of the rule
c. Source coverage
d. Target coverage
e. Source confidence
f. Target confidence

Therefore the complete granular association rule is

Rule 1: 40% faculty teaches atleast 30% of other discipline courses
Rule 2: 45% faculty are female and 6% of courses are in CSE.

So if we need rules covering more people and courses,rule 1 is preferred. But if we need more confidence on the rules,then prefer rule 2[5].

A many to many relation involves two universes and a relation. MMER contains 5 tuples.ES={U,A,V,B,R} Where (U,A) and (V,B) are the two information systems and R $\subseteq$ U x V.

Let U={x1,x2,x3,……xn) and V= y1,y2,y3,……yk) represent set of all objects in U and V. In the binary relation,1 indicates true and 0 indicates false.

Table 1: MMER System

A) Faculty

FID	Name	Age	Gender	Degree	Title	Dept
F1	Ajay	50	Male	Dr	Prof	CSE
F2	Baghya	32	Male	Master	Lecturer	Training
F3	Celia	55	Female	Dr	Prof	S&H
F4	Darun	46	Male	Master	AssoProf	ECE
F5	Eliza	49	Female	Master	AssoProf	MBA

(b)courses

CID	Selection	Category	Time
C1	Public	Engineering	Even
C2	Private	Arts	Odd
C3	Public	Engineering	Even

(C) teaches

FID/CID	C1	C2	C3
F1	1	1	0
F2	0	0	0
F3	0	0	1
F4	0	1	0
F5	0	0	0

If the ID is removed then the relation can be stored in numeric form. If so, we can,

a. Read the two information system directly
b. Construct the Boolean information system and read the support compressed Boolean information system.
   a. Read the third information system
   b. Convert it into Boolean one.
   c. Delete the ID of first two sets if they are not needed for internal representation.

Generation of Complete Match Rules: Complete match rules gets generated if the source confidence threshold-1<1e-6 and target confidence threshold-1<1e-6.

Overall Algorithm: Sandwich:

Input: Source coverage threshold, target coverage threshold and the base algorithm.
Output: returns all rules in a string.

Steps involved:
1. Compute source coverage
   - Compute the first set frequent granules
   - Compute the first set frequent granule extension.
2. Compute target coverage (second set granules)
   - Compute the frequent granules for the second set
   - Compute the frequent granules for the second set extension.

Check all possible rules in the first and second set granules extension length and output the valid rules in terms of checking time used, the number of basic operations performed and the time used[6].

The time complexity of this method is $O(IRU||RV||U||VI)$ where $(IRUI)$ and $(IRVI)$ are the sizes of the frequent items in the first and second set respectively.

Overall Algorithm: Fast Forward

Input: Source coverage threshold, Target coverage threshold and the base algorithm.
Output: returns all rules in a string.

Steps involved:
1. Compute source coverage
   - Compute the first set frequent granules
   - Compute the first set frequent granule extension.
2. Compute target coverage
   - Compute the frequent granules for the second set
   - Compute the frequent granules for the second set extension.

For each of the source granules obtained, construct a set of objects that are instances of the granule and store it in one dimensional positive arrays rather than storing the relation in boolean array.

The time complexity of this method is $O(IRU||RV||U|+|VI)$ where $(IRUI)$ and $(IRVI)$ are the frequent items in the first and second set respectively.

Overall Algorithm:Fast backward

Input: Source coverage threshold, Target coverage threshold and the base algorithm.
Output: Returns all rules in a string.

Steps involved:
1. Compute source coverage
   - Compute the first set frequent granules
   - Compute the first set frequent granule extension.
2. Compute target coverage (second set granules)
   - Compute the frequent granules for the second set
   - Compute the frequent granules for the second set extension.

Avoid doing computation of different rules which is having the same Right hand side. For each of the granules obtained, construct a set of objects that are instances of the granule.

Check all possible rules in the first and second set granules extension length and output the valid rules in terms of checking time used, the number of basic operations performed and the time used.

The time complexity of this method is $O(IRU||RV||UI)$ where $(IRUI)$ and $(IRVI)$ are the sizes of the frequent items in the first and second set respectively. The space complexity of the above methods are |U|*|V| as it needs to store the entire Boolean matrix in the relation [7].

Generation of Left Hand Side Partial Match Rules: Lefthand side Partial match rules gets generated if the target confidence threshold-1<1e-6.

Overall Algorithm:Sandwich

Input: Source coverage threshold,Target coverage threshold and the base algorithm.
Output: Returns rules in the form of a=>b[temporaryconfidence,1].

Steps involved:

1. Compute the first set frequent granules
2. Compute the first set frequent granule extension.
3. Compute the frequent granules for the second set
4. Compute the frequent granules for the second set extension.
5. Compute the source confidence threshold.
6. Check all possible rules in the first and second set granules extension length and output the valid rules in terms of checking time used, the number of basic operations performed and the time used.

Generation of Right Hand Side Partial Match Rules: Right hand side Partial match rules gets generated if the source confidence threshold-1<1e-6.

Overall Algorithm:Sandwich

Input: Source coverage threshold, Target coverage threshold and the base algorithm.
Output: Returns rules in the form of a=>b [1,temporaryconfidence].

Steps involved:

1. Compute the first set frequent granules
2. Compute the first set frequent granule extension.
3. Compute the frequent granules for the second set
4. Compute the frequent granules for the second set extension.
5. Compute the target confidence threshold.
5. Check all possible rules in the first and second set granules extension length and output the valid rules in terms of checking time used, the number of basic operations performed and the time used[8].

Generation of partial match rules:
Overall Algorithm:Sandwich

Input: Source coverage threshold,Target coverage threshold and the base algorithm.
Output: returns rules in the form of a=>b [1,temporarysourceconfidencestring,targetconfidence].

Steps involved:

1. Compute the first set frequent granules by specifying source coverage and base algorithm.
2. Compute the first set frequent granule extension.
3. Compute the frequent granules for the second set by specifying target coverage and base algorithm.
4. Compute the frequent granules for the second set extension.
5. Check whether the source confidence threshold is satisfied or not. If not do

Source confidence of first set frequent granule extension and second set frequent granule extension and add it to the target confidence and finally generate a temporary confidence string.

6. Check all possible rules in the first and second set granules extension length and output the valid rules in terms of checking time used, the number of basic operations performed and the time used[9].

Overall Algorithm: Fast forward

Input: Source coverage threshold,Target coverage threshold and the base algorithm.
Output: returns rules in the form of a=>b [1,temporarysourceconfidencestring, targetconfidence.

Steps involved:

1. Compute the first set frequent granules by specifying source coverage and base algorithm.
2. Compute the first set frequent granule extension.
3. Compute the frequent granules for the second set by specifying target coverage and base algorithm.
4. Compute the frequent granules for the second set extension.
5. Check all possible rules in the first and second set granules extension length and output the valid rules in terms of checking time used, the number of basic operations performed and the time used[9].

RESULTS AND DISCUSSIONS

We tested our algorithm on real world datasets. We use the version faculty and courses. The data are preprocessed. The final table contains only Boolean information.

Meaningfulness of Rules: By applying the appropriate threshold values, one can obtain many rules.

Rule 1: Faculty whose age is between 30 to 40 will come under Lecturer.
Rule 2: Faculty whose age is between 45 to 49 will come under Associate Professor.
Rule 3: Faculty whose age is between 50 to 58 will come under Professor.

All These Rules Makes Sense to us: Therefore the discussion on matching rules using four measures becomes valid. But we are unable to mine any complete match rules because they are strong.

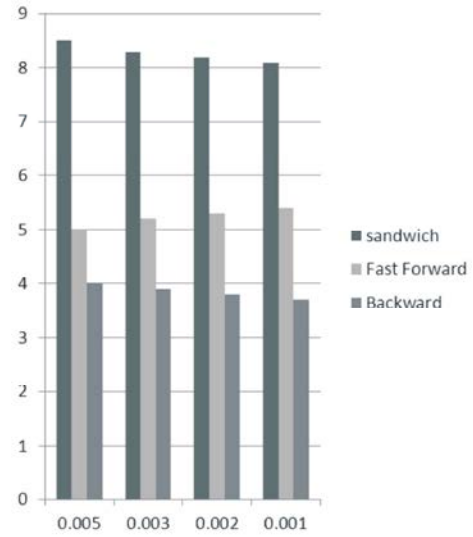

Fig. 1: Basic operation of three methods(X axis represents threshold values and Y axis represents number of basic operation that are needed).

The source coverage and the target coverage of the rules are already specified. We set the target confidence threshold to obtain different source confidences. The target confidence threshold are set to 0.1,0.2 to 0.9.A very small non zero number guarantees that atleast one object is covered by the rule. Threshold value cannot be specified as zero. If threshold value is mentioned as zero, then it means that it does not provide any meaningful rules[10].

If we increase the target confidence threshold, the source confidence gets decreased. So there exists tradeoff between the two thresholds.

The basic operations refers to the comparison, addition etc. But we will focus on runtime instead of number of basic operations since different operations take different time[11-14].

The number of basic operations is compared with all the methods. It can be naturally observed that the forward and backward methods are more efficient than sandwich method. However, with the decrease of thresholds, the number of operations increases and the backward algorithm makes the best choice. The rule checking terminates only when certain conditions are met. However the time complexities are for reference only and the runtime depends on the characteristics of data.Inshort the backward algorithm can generate many rules and is scalable whereas for small datasets and large thresholds the sandwich method is more efficient.

CONCLUSION

Most of the association rule mining algorithms suffer from the problems of too much execution time and generating too many association rules. Although conventional algorithm can identify meaningful itemsets and construct association rules, it suffers the disadvantage of generating numerous candidate itemsets that must be repeatedly contrasted with the entire database. The processing of the conventional algorithm also utilizes a large amount of memory. Thus, this approach is very significant for effective analysis and it helps the customers in purchasing their items with more comfort, which in turn increases the sales rate of the markets.

REFERENCES

1. WEI Yong-Qing, YANG Ren-hua and LIU Pei-yu, 2010. An Improved Apriori Algorithm for Association Rules of Mining, 978-1-4244-3930-0/09/$25.00 © IEEE.
2. XING Xue CHEN Yao WANG Yan-en, 2010. Study on Mining Theories of Association Rules and Its Application, International Conference on Innovative computing and communication Asia –Pacific Conference on Information Technology and Ocean Engineering 978-0-7695-3942-3/10 $26.00 IEEE.
3. Xiufend Piao, Zhan long Wang and Gang Liu, 2011. Research on mining positive and negative association rules based on dual confidence, Fifth International Conference on Internet Computing for Science and Engineering.
4. Pengfei Guo Xuezhi and Wang Yingshi Han, 2010. The Enhanced Genetic Algorithms for the Optimization Design.
5. WEI Yong-Qing, YANG Ren-hua, LIU Pei-yu, 2010. An Improved Apriori Algorithm for Association Rules of Mining.
6. Singh Rawat Sandeep and Lakshmi Rajamani, 2011. Probability Apriori based Approach to Mine Rare Association Rules".In 3rd Conference on Data Mining and Optimization (DMO), © IEEE.
7. XING Xue CHEN Yao WANG Yan-en, 2010. Study on Mining Theories of Association Rules and Its Application.International Conference on Innovative computing and communication Asia –Pacific Conference on Information Technology and Ocean Engineering 978-0-7695-3942-3/10 $26.00 IEEE.
8. CH.Sandeep Kumar, K.Shrinivas, Peddi Kishor T.Bhaskar, 2011. An Alternative Approach to Mine Association Rules, 978-1-4244-8679-3/11 $26.00 © IEEE.
9. Goethals B., W. L. Page and M. Mampaey, 2010. Mining interesting sets and rules in relational databases, in Proceedings of the 2010 ACM Symposium on Applied Computing, 997-1001.
10. Min F., H. He, Y. Qian and W. Zhu, 2011. Test-cost-sensitive attribute reduction, Information Sciences, 181: 4928-4942.
11. Min F., H. He, Y. Qian and W. Zhu, 2011. Test-cost-sensitive attribute reduction, Information Sciences, 181: 4928-4942.
12. Min F., Q. Hu and W. Zhu, 2012. Granular association rules on two universes with four measures," submitted to Information Sciences, [Online]. Available: http: //arxiv.org/abs/1209.5598
13. Min F., Q. H. Hu and W. Zhu, 2012. Granular association rules with four subtypes, in Proceedings of the 2011 IEEE International Conference on Granular Computing, 432–437.
14. Min F. and W. Zhu, 2012. Granular association rule mining through parametric rough sets, in Proceedings of the 2012 International Conference on Brain Informatics, ser. LNCS, 7670: 320–331.