Asian Journal of Information Technology 6 (11): 1192-1195, 2007
ISSN: 1682-3915

Mining Essential and Interesting Rules for Efficient Prediction

K. Shyamala and S.P. Rajagopalan
Department of Computer Science, Dr. Ambedkar College, Chennai 600 039, India
Professor Emeritus, School of Computer Science and Engineering, Dr. MGR University,
Chennai 600 095, India

Abstract: Since the introduction of association rules, many algorithms have been developed to perform the computationally very intensive task of association rule mining. During recent years, there has been the tendency in research to concentrate on developing algorithms for specialized tasks, for example, mining optimized rules or incrementally updating rule sets. The classic problem of association rules deals with efficient generation of association rules with respect to minimum support and minimum confidence. But the performance problem concerning this task is still not adequately solved. In this study, a theoretical model of algorithm is presented which generates set of essential rules directly without generating the entire set of association rules. A set of pruning rules are formed and they are applied in the design of the algorithm for generating the essential set of rules. The set of essential rules are the set of predictive class association rules. The efficiency of the proposed algorithm is analyzed theoretically. The application of this algorithm avoids redundant computation and also the time required for generating the essential set of rules is subsequently reduced.

Key words: Data mining, association rules, essential rules, pruning

INTRODUCTION

Data mining (Han and Kamber, 2001; Hand *et al*., 2001) is a multi-disciplinary research field and many research fields have their contributions such as database, machine learning, statistics and artificial intelligence. It includes several sub-fields such as rule generation, classification and clustering, probabilistic modeling and visualization.

The rule is one of the most expressive and human readable representations for knowledge and hence association rule mining is one of the central tasks in data mining. The problem of mining association rules was introduced in Agrawal *et al*. (1993). Given a set of transactions, where each transaction is a set of items, an association rule is an expression $X \rightarrow Y$ where X and Y are set of items. The significance of association rules is measured via support and confidence. The primary goal of association rule mining (Agrawal *et al*., 1993) is to find all rules satisfying minimum support and minimum confidence requirements.

Interesting rules must be picked from the set of generated rules. This might be quite costly because the generated rule sets normally are quite large. For example, more than 1,00,000 rules are not uncommon and in contrast the percentage of useful rules is typically only a very small fraction. Presence of some rules may make others redundant and therefore interesting. These principles are formalized in the form of pruning rules. The pruning rules are used in the design of the proposed algorithm for generating the essential set of rules.

In this study, a theoretical model of algorithm is presented which generates set of essential rules directly without generating the entire set of association rules. A set of pruning rules are formed and they are applied in the design of the algorithm for generating the essential set of rules. The set of essential rules are the set of predictive class association rules. The efficiency of the proposed algorithm is analyzed theoretically. The application of this algorithm avoids redundant computation and also the time required for generating the essential set of rules is subsequently reduced. The size of the essential rule set is also smaller than the entire set of rules.

There are major problems with association rule generation. The first problem is the generation of too many rules and the second problem is that not all of the generated rules are important. There has been various research effort aimed at mitigating both problems. The rule quantity problem can be handled by pruning or summarizing the discovered rules. Toivonen *et al*. (1995) proposed the idea of using structural rule covers to remove redundant rules and clustering as a means for grouping together related rule covers. Liu *et al*. (1999) used the standard χ^2 test to prune insignificant rules and

Corresponding Author: K. Shyamala, Department of Computer Science, Dr. Ambedkar College, Chennai 600 039, India

introduced the concept of direction setting rules to summarize the patterns. Other researchers such as Srikant *et al.* (1997) and Ng *et al.* (1998) have used the constraints provided by a user to limit the number of rules that are generated.

Liu *et al.* (1999) have used a chi-square test of independence as a principal measure for both generating the association rules and identifying non-actionable rules. In literature different measures are proposed to discover the interestingness of a rule. Rule templates (Aha *et al.*, 1991; Ali *et al.*, 1997) is a technique that separates only those rules that match the template. Neighborhood based interestingness (Menlo, 1997) defines interestingness within a set of rules in terms of their density and relative confidences.

Apriori has been the most important algorithm in association rule mining and some variant algorithms (Holsheimer *et al.*, 1995; Mannila *et al.*, 1994) are presented subsequently. Many algorithms (Holsheimer *et al.*, 1995; Houtsma and Swami, 1995; Park *et al.*, 1995) used the downwards closure property which states that all subsets of a frequent itemset must be frequent. Techniques for association rule mining have already been applied to generation of classification rules (Ali *et al.*, 1997; Liu *et al.*, 1998). However, the classification rules are generated by generating the entire association rule set.

Given a relational database D with n attributes, a record of D is a n-tuple. Record is considered as a set of attribute-value pairs, denoted by T. A pattern is a set of attribute-value pairs. The support of a pattern A is the ratio of the number of records containing A to the number of records in the database, denoted by sup(A).

An implication is denoted by $A \rightarrow C$, where A is a pattern and C is a class. The support of the implication $A \rightarrow C$ is sup $(A \cup C)$. The confidence of the implication is $sup(A \cup C)/sup(A)$, denoted by $conf(A \rightarrow C)$. The covered set of the rule is the set of all records containing the antecedent of the rule, denoted by $cov(A \rightarrow C)$.

As the main goal of classification rule mining is prediction, confidence is not suitable for this purpose. So statistical estimate of accuracy is used.

$$acc(A \rightarrow c) = conf(A \rightarrow c) - z^N$$

$$\sqrt{\frac{conf(A \rightarrow c)(1 - conf(A \rightarrow c))}{|cov(A \rightarrow c)|}}$$

Where, z^N is a constant related with a statistical confidence interval.

The closure property: Using the support-confidence test, the problem is usually divided into two parts. First finding supported item sets and then discovering rules in those that have large confidence. Almost all research has focused on the first of these tasks. One reason is that finding support is usually the more expensive step, but another reason is that rule discovery does not lend itself as well top clever algorithms.

This is because, confidence possess no closure property. Support, on the other hand is downward closed. If a set of items has support, then all its subsets also have support. The advantage of this closure property is been taken care in devising an algorithm in this study level wise algorithms (Hand *et al.*, 2001). Find all items with a given property among item sets of size i and use this knowledge to explore item sets of size i+1.

Upward closure properties are used in pruning weak rules in construction of designing the algorithm for generating essential rules.

Definitions: Before the construction of the algorithm and pruning rules, the following definitions are required.

Definition 1: The rules that are pruned away with the pruning techniques are redundant rules.

Definition 2: Weak rules are the rules generated as valid rules using any measures, but due to the presence of alternative causes, their validity may be questionable.

Definition 3: Rules that are neither redundant nor weak are called essential rules. We say that two rules are of similar strength if for a small pre-defined value $1>\varepsilon>0$, $|strength (r1) - strength (r2)| < \varepsilon$.

Definition 4: Given two rules r1 and r2, we say that r2 is stronger than r1 if $r2 \subset r1 \wedge acc(r2) \geq acc(r1)$.

It is clear that only the essential rule is strong and accurate and can make a prediction in rule set. It is clear that rules which are not essential never provide predictions in the classifier built from the entire set of rules. So it is understood that a rule is said to be potentially predictive if it is used to make a precondition in the classifier.

Theorem 1: The essential rules are the set of all potentially predictive rules.

Proof: The rule that eventually makes the prediction must be a strong rule with the highest accuracy among all matched rules as since the essential rules are strong and accurate, the essential rules are the set of all potentially predictive rules.

An efficient algorithm is presented which generates set of essential rules rather than entire set of association rules. General practice is to generate the entire set of association rules and applying pruning to the set of association rules generated. The resulting sets of rules are used for prediction.

Since the algorithm presented here avoids redundant computation. The time required for generating the essential set of rules is subsequently reduced. The efficiency of the mining process has also been improved.

Algorithm for essential rules: The General procedure is to obtain the entire set of rules E and then prune all the weak rules to generate the set of predictive rules. However, this process may take long time and it involves redundant computation.

In this study, we present an efficient algorithm to generate an essential set of rules without generating the entire set of rules.

Lemma 1: If $A \rightarrow C$ and $A \wedge B \rightarrow C$ and either both rules are positive or negative with similar strength, then $A \wedge B \rightarrow C$ is redundant.

Proof: This follows from first order logic.

Lemma 2: If sup(A,C) = sup(AB,C), then $AB \rightarrow C$ and all more specific rules are weak.

Proof: Since sup(A,C) = sup(AB,C), using sup(AC) $\geq$ sup(ABC), we get conf($A \rightarrow C$) $\geq$ conf($AB \rightarrow C$). Using relation $|cov(A \rightarrow c)| \geq |cov(AB \rightarrow C)|$, we get acc($A \rightarrow C$) $\geq$ acc($AB \rightarrow C$). So $A \rightarrow C > AB > C$. Since sup(AB,C) = sup(ABC,C) for all Z if sup(A,C) = sup(AB,C), we have $AB \rightarrow C > ABC \rightarrow C$ for all Z. Consequently, $AB \rightarrow C$ and all more specific rules are weak.

Lemma 3: If $A \rightarrow C$, $B \rightarrow C$, both either positive or negative rules with similar strength, then $B \rightarrow C$ is redundant if $B \xrightarrow{\subset} A$, but $A \xrightarrow{\subset} B$ is not true.

Proof: The first rule subsumes the second that is whenever the second rule is true, the first rule is also true and both rules imply the same effect. So the second rule is classified as redundant.

Lemma 4: If $A \rightarrow C1$ and $A \rightarrow C1 \wedge C2$, then $A \rightarrow C1$ is redundant.

Proof: $C1 \wedge C2$ is stronger than C1 in logical sense. Hence the rule $A \rightarrow C1$ is redundant.

The Lemmas 1-4 are useful for searching essential rules, since we can remove a set of weak rules as soon as we find one that satisfies the above Lemmas. This process also reduces the search space for essential rules.

The algorithm given below generates the essential set of rules directly. The algorithm is a level wise algorithm which finds all items with a given property among item sets of size i and use this knowledge to a set and its closure properties to make inferences about its supersets.

```
Algorithm : Erulegen
Input     : Database D, Class attribute C
Output    : The set of essential rules Eset

Eset   = ϕ
Initialize tree T
Eset   = SelectErule(T)
Nset   = New candidates generated from T
While (Nset ≠ϕ) do
{
   Support = Support(Nset)
   Nset = prune(Nset)
   Eset = SelectErule(T)
   Nset = New candidates generated from T
}
return Eset
```

The function SelectErule is used to select the essential rules from relational database. The function generate the (l+1) layer candidates from the l layer nodes. The function combines a pair of sibling nodes and insert their combination as a new node in the next layer. If any of its l - sub item set is not a frequent item set, then the node is removed. If any of its l - sub patterns cannot get enough support with any of the possible targets (classes) and then the class is removed from the target set. The new candidate is removed if no possible target is left.

The function prune prunes weak rules and infrequent candidates in the (l+1)th layer of candidate tree. The removal of weak rules is based on the four lemmas and they specify the pruning rules. These pruning rules applied together on the rules generated give a pruned set of rules. Application of these pruning rules should not be overlapping in the sense that if one rule is pruned once, it should not be considered to prune other rules. Once a rule is pruned, it is removed from the set, before applying the subsequent pruning rules. Changing the order of application of these pruning rules may change the essential set of rules generated.

CONCLUSION

An efficient theoretical model of the algorithm is proposed in this study for generating essential set of rules without generating entire set of rules. Pruning rules are formed and applied to prune the weak rules. The algorithm avoids much redundant computation required in generating the entire set of rules. The time required for generating the rules also is less. The efficiency of the proposed algorithm is analyzed theoretically. Our next objective is to confirm the analysis by testing the algorithm on certain real world databases.

REFERENCES

Agrawal, R., T. Imielinski and A. Swami, 1993. Mining associations between sets of items in massive databases. In: Proceedings of the ACM SIGMOD Int. Conf. Manage. Data, pp: 207-216.

Ali, K., S. Manganaris and R. Srikant, 1997. Partial classification using association rules. In: Proceedings of the 3rd International Conference on Discovery and Data Mining (KDD), pp: 115.

Ali, K., S. Manganaris and R. Srikant, 1997. Partial classification using association rules. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD), Menlo Park, CA, pp: 115.

Han, J. and M. Kamber, 2001. Data mining: Concepts and Techniques. Morgan Kaufmann.

Hand, D.J., H. Mannila and P. Smyth, 2001. Principles of Data Mining. MIT Press.

Holsheimer, M., M. Kersten, 1995. Mannila and Toivonen. A perspective on databases and data mining. In 1st Int. Conf. Knowledge Discovery and Data Mining, pp: 10.

Houtsma, M. and A. Swami, 1995. Set-oriented mining for association rules in Relational databases. In: Proceedings of the 11th International Conference on Data Engineering, Los Alamitos, CA, USA., pp: 25-34.

Liu, B., W. Hsu and Y. Ma, 1998. Integrating classification and association rule mining. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD), pp: 27-3

Liu, B., W. Hsu and Y. Ma, 1999. Pruning and summarizing the discovered associations. In: Proceedings of the fifth international Conference on Knowledge discovery and Data mining.

Mannila, H., H. Toivonen and I. Verkamo, 1994. Efficient algorithms for discovering association rules. In AAAI Wkshp. Knowledge Discovery in Databases, AAAI Press, pp: 181-192.

Menlo Park, C.A., 1997. AAAI Press.

Ng, R., L. Lakshmanan, J. Han and A. Pang, 1998. Exploratory mining and pruning optimizations of constrained association rules. In: Proceedings of 1998 ACM-SIGMOD International Conference on Management of Data. Aha, D.W., D. Kibler and M.K. Albert, 1991. Instance-based learning algorithms. Machine Learning, 6: 37-66.

Park, J.S., M. Chen and P.S. Yu, 1995. An effective hash based algorithm for mining association rules. In ACM SIGMOD International Conference Management Data.

Srikant, R., R. Vu and Q. Agrawal, 1997. Mining association rules with item constraints. In: Proceedings of the 3rd International Conference on Knowledge on Knowledge Discovery and Data mining.

Toivonen, H., M. Klemettinen, P. Ronkainen, K. Hatonen and H. Mannila, 1995. Pruning and grouping discovered association rules. In ECML-95 Workshop on Statistics, Machine learning and Knowledge Discovery in Databases.