International Advance Journal of Engineering, Science and Management (IAJESM)
Multidisciplinary, Indexed, Double Blind, Open Access, Peer-Reviewed, Refereed-International Journal.
SJIF Impact Factor =7.938, July-December 2024, Submitted in October 2024, ISSN -2393-8048

An Efficient Theoretical Model of An Algorithm for Generating the Mining Essential Rules for Efficient Prediction

Manish Kumar Goyal, Research Scholar, Department of Computer Science Engineering, Nirwan University Jaipur, Rajasthan (India)
Dr. Amit Singla, Professor, Department of Computer Science Engineering, Nirwan University Jaipur, Rajasthan (India)

Abstract

Association policy Mining identifies intriguing relationships or correlations between data pieces. With vast volumes of data being generated and kept on a constant basis, several companies are increasingly interested in mining association rules from their databases. Because rule is one of most expressive & human readable representations of information, association rule mining is a critical task in datamining. Usually association rule mining techniques generate too many rules and it reduces the efficiency of the process. Further it complicates the pruning process also. As a result, it is required to identify a limited group of fundamental rules that allow for prediction without creating the whole set of rules. This paper proposes an efficient theoretical model of the method for creating crucial rules without producing the whole set of rules. Because the algorithm creates just the necessary rules, the rule set produced is less in size. The time necessary to generate the rules is likewise less. The efficiency of the suggested method is theoretically evaluated.

Keywords: Association Rule Mining, Algorithm, Essential and Interesting Rules Etc.

INTRODUCTION

Association rules are helpful patterns that may be derived from large datasets. An association rule states that presence of a set of items in a transaction implies inclusion of other items in same transaction. Agrawal et al. identified a difficulty with mining association rules. The primary goal of association rule mining is to identify all rules that match certain basic requirements, such as low support & minimum confidence. [1] Mining for association rules was first intended to handle market basket issues in transactional databases, but it has now been broadened to encompass challenges such as categorization. An association rule is denoted as X Y, where X and Y are sets of elements. Given a collection of interactions, every single one represents a set of items. Support and assurance are used to assess the importance of association rules.

A significant number of patterns and rules are often generated by an association rule mining approach. However, the majority of these regulations are uninteresting to users. Among the created rules, the most beneficial and intriguing rules should be chosen. This selection process may be thought of as a second level of data mining: mining among rules. However, present strategies for defining interestingness have a significant problem in that they create redundant patterns in addition to the intended patterns, i.e., same semantic information is captured by many patterns & hence some of them can be pruned. This repetition occurs because each pattern is picked individually, regardless of the other patterns chosen. Association rule mining searches for interesting connections between things in a dataset. Consider the following market basket abstraction: each client checkout item represents a transaction, and a collection of transactions represents an act in the database. An item set is a collection of things. If $X \rightarrow Y$ is an association rule b/w 2 disjoint item sets X & Y, then

- The fraction of transaction containing both Xand Y is at least s percent from a database (support) &
- At least c percent transactions containing X include Yas well (confidence).

Where s & c are minimal levels of support & confidence, respectively. Mining for association rules is a 2-step procedure.

1. Find all the common item sets. Each of these item sets will occur at least as frequently as a pre-determined minimum support count.
2. Create strong connection rules using frequent item sets: By definition, these regulations must have a minimum level of support & trust.

The first iteration of algorithm simply counts item occurrences to discover most common item sets. A following pass, say pass k, has 2 stages. First, frequent item sets Lk-1. / Discovered in (k-1) the pass are utilized to construct candidate item sets Ck, using apriori candidate generation technique outlined below. Next, the database is examined, and candidates' support in CK is tallied. We need to easily find the candidates in CK who are involved in a certain transaction t for speedy counting. The list of potential item sets is trimmed to ensure that all of its subgroups are already known to be frequent item sets. The candidate generation and trimming procedures are the most important components of the a priori algorithm (Agrawal & Srikant, 1994), as demonstrated by algorithms 1 and 2.

Algorithm: 1 [gen_ candidate_ item sets]

```
BEGIN.
$C_k = \emptyset$
for all item sets $l_1 € L_{k-1}$ do
for all item sets $l_2 € L_{k-1}$ do
If $l_1[l] = l_2[2] = l_2[2] \wedge \ldots \wedge l_1[k-1] < l_2[k-1]$
then $c = l_1[l], l_1[2] \ldots l_1[k: -1], l_2[k-1]$
$C_k = C_k \cup \{c\}$
END
```

Algorithm: 2 [prune]

```
BEGIN
for all $c € C_k$
for all (k -1) subsets d of c do
if $d € l_k$-1
then $C_k = C_k \setminus \{c\}$
END
```

Given Lk-1, set of all frequent (k-1) item sets, we must create a superset of set of all frequent k itemsets. The idea underlying a priori candid ate generation technique is that if an item set X has little support, then do all subsets of X. It conducts two types of operations: join and prune. The join component combines Lk-I with Lk-I to yield probable candidates. The prune component uses the a priori property to reject candidates with an uncommon subset. Every iteration of the a priori algorithm uses these 2 functions (candidate generation & pruning). When no candidate set remains after pruning, it moves up the lattice, beginning at level k. As a result, a priori must query a database as many times as the length of most frequently occurring item collection, plus one. The storage structure for frequent item sets is plainly crucial in keeping track of number of times an item set appears.

In a relational database, each column represents a domain, or property, & has a finite number of values. Each row is referred to as a record, which is atuple of attribute values, or simply a collection of attribute & attribute-value pairs. A database is classified as a training database if it has a certain attribute that is called the class attribute. Otherwise, it's known as a testing database. Classification is an analysis strategy that involves categorizing test database entries into known classifications. A computerized database record is sometimes known as an instance. A classification rule is a correct conclusion from the pattern to a class when seen as a set of attribute-value pairs. A rule can predict anything about a record if its previous version is a portion of that record.

Because any relational database is denser than a transaction database, finding all association rales from a relational database may be challenging owing to the enormous amount of rales. Too many rales are produced during the mining process. The presence of certain rules may render others unnecessary and so boring. The majority of the rales are neither fascinating or valuable to the user.

To overcome the problems outlined above, a limited number of intriguing rules capable of making predictions must be discovered. The resultant rule sets are necessary, fascinating, and capable of generating predictions. A collection of pruning rules is created and used to remove the unnecessary and uninteresting rules from the whole set of rules, resulting in the essential set of rules. The key rules include the rules for anticipating class connection. Classification rules may be thought of as a subset of association rules, with predefined results (classes) that can be used for classification.

An n-tuple is a record from a database with relational structure D that includes n attributes. A record is a collection of attribute-value pairs denoted by T. A pattern consists of attribute-value pairs. The support of a pattern is calculated by dividing the total quantity of records expressing A by the total number of recordings in the database. This is indicated as sup(A).

An implication is represented as A → C, where A is a pattern & C is a class. The support for implication A → C is sup (A ∪ C). Confidence in the implication is sup (A ∪ Cj) /sup(A), indicated by conf(A → C). The covered set of rule is all records that contain antecedent of rule, indicated by cov (A → C). Confidence is inappropriate for classification rule mining since its primary objective is prediction. So a statistical estimate of precision is utilized (Liu et al., 1998).

$$\text{acc}(A \to c) = \text{conf}(a \to c) - z^{N}\sqrt{\frac{\text{conf}(A \to c)(1-\text{conf}((A \to c))}{|\text{cov}(A \to c)|}}$$

where zN is a constant associated with statistical confidence intervals. For a rule r, cond(r) represents the antecedent (conditions), while cons(r) denotes the conclusion. If cond(r) ⊆ T, then rule r covers it. A rule can make a prediction on its covered record, represented by r(J') → cons(r). If cons(r) is T's class, then mle provides an accurate prediction. Otherwise, it produces an incorrect forecast. The accuracy of a prediction is equal to accuracy of underlying rule, given as acc(r(T) → c). If a rule correctly predicts a record, we call it an instance.

Assume a ruleset E with an input T. There maybe several rules in E that may produce the prediction, such as r1(T) → c1, r2(T) → cl,... If r is rule with best accuracy among all ri, and cond(ri) ⊆ T, then E's forecast is the same as r's prediction. The forecast is as accurate as rule r. If there are many rules with same greatest accuracy, we select one with most support among them. If rules have equal precision and support, the one with the shortest antecedent is chosen. If a rule set is unable to anticipate a record, it makes an arbitrary forecast with 0 accuracy. So, r2 ⊆ rl represents cond(r2) ⊂ cond(rl) & cons(r2) = cons(rl). Support may be expressed as sup(A,c) = sup(A) - sup(Ac), whereas cov(A, c) = cov(X) - cov(Xc).

LITERATUE REVIEW

A lot of effort has gone into defining the concept of "interesting" patterns. For detecting intriguing patterns, common ways employ statistical metrics. Patterns are regarded intriguing if their value with respect to a specific metric exceeds a user-specified threshold. Several similar measures, such as the support-confidence measure[1], correlation measure[2] ratio rules[3], and highly collective trends, have been suggested in the literature. [4]

Liu et al. [5] employed an independence test as the primary metric for both producing association rules and finding non-actionable rules. Various methods are given in the literature to determine the interestingness of a regulation. The rule template approach [6,7] separates only those rules that fit the template. A priori has been most significant algorithm in association rule mining, and various variation methods are discussed later. Many algorithms exploited the downwards closure property, which stipulates that all subsets of a frequent item set must be frequent [8,9,10]. Association rule mining techniques have already been used to generate categorization rules [5,6]. The categorization rules, on the other hand, are formed by producing the full association rule set.

DEFINITIONS

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

1. The rules that are pruned away with the pruning techniques are redundant rules.
2. Weak rules are those that are created as legitimate rules using any measure but whose validity is called into doubt owing to the availability of alternative causes.

3. Rules that are not redundant or weak are referred to be vital rules. Two rules are considered equal in strength if, given a small pre-defined number $l > \varepsilon > 0$, |strength (rl) - strength (r2) < ε.

4. Given 2 rules rl and r2, we say that r2 is stronger than rl if $r2 \subset rl \wedge acc(r2) \geq acc(rl)$.

Only the essential rule is powerful and precise enough to produce a forecast in the rule set.
It is obvious that non-essential rules never contribute predictions in the classifier built from the whole collection of rules. As a result, if a rule is utilized to create a precondition in the classifier, it is said to be potentially predictive. As a result, the theorem is as follows.

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

Proof: The theorem is proved in two steps. First, all essential rules are potentially predictive rules. Secondly, a rule that does not belong to a set of essential rules cannot be a potentially predictive rule.

An input record can be any combination of attribute-value pairs, but the classification formed from the set of full rules can only categorize records that are a superset of a rule's antecedent. Although various rules can be used to categorize incoming data, only powerful ones can provide predictions. The final prediction rule must be the strongest and most accurate of all the matched rules. Because an input record can include any pattern, any strong rule can do prediction. The set of significant rules comprises all of the strong rules in the whole collection of rules, as well as all of the possibly predictive rules. A rule outside of the basic set of rules may predict a record. According to the definition, the core set of rules must have a powerful rule capable of anticipating. Clearly, forecast will be made using strong rule mentioned before. As a result, a rule not included in the essential set cannot be considered potentially predictive.

The final prediction rule must be as trong rule with best accuracy among all matched rules. As a result, the key rules are powerful and accurate, and they comprise the whole collection of potentially predictive rules. As a result, the theorem is proven. An efficient technique is described in this work that creates a collection of essential rules rather than whole set of association rules. The general approach is to produce the whole collection of association rules and then prune the set of generated association rules. The derived rule sets are utilized for prediction. Because the approach provided here minimizes redundant calculation, the time required to generate necessary set of rules is decreased. The mining process's efficiency has also been increased.

ALGORITHM FOR ESSENTIAL AND INTERESTING RULES

The general process is to get the whole set of rules E set and then prune all of the weak rules in order to produce the set of predictive rules. This method, however, may take a long time and includes duplicate computation. In this part, we describe Erulegen, an efficient technique for creating an important collection of rules without producing the whole set.

Lemma 1: If sup(A,c) = sup(AB,c), then AB→c and all more specific rules are weak.

Proof: Since sup(A,c) = sup(AB,c), usings up(Ac) ≥ sup(ABc), we get conf(A → c) ≥ conf(AB→ c). Using relation |cov(A → c)| ≥ |cov(AB → c), |we get acc(A → c) ≥ acc(AB → c). So A → c > AB → C. Since sup(AC,c) = sup(ABC.c) for all C if sup(A.c) = sup(AB,c), we have AB →c > ABC→ c forall C. Consequently, AB→ c andall the more specificrules are weak.

Lemma 2: If A → C, B → C, both eitherpositive or negative ruleswith similar strength,then B → C is redundant ifB →A, but A → B is nottrue.

Proof: The first rule entails second, which means that anytime second rule is true, first rule is also true, and both laws suggest the same outcome. As a result, second ruleis considered rsedundant.

Lemma 3: If $A \rightarrow C_1$ and $A \rightarrow C_1 \wedge C_2$, then $A \rightarrow C_1$ is redundant.

Proof: $C_1 \wedge C_2$ is stronger than C_1 in logical sense. Hence rule $A \rightarrow C_1$ is redundant.

The Lemmas 1 to 3 are important for looking for essential rules since we can exclude a set of weakrules if we find one that meets aforementioned Lemmas. This procedure also narrows the search field for critical rules. The algorithm 3 Erulegen immediately creates the necessary set of rules. Using the knowledge acquired from the other patterns created, it prunes away redundant patterns. The algorithm is a level-wise technique that identifies all items having a certain property among item sets of size i and uses this information to infer supersets based on a set and its closure qualities.

```
Algorithm: 3 [Erulegen]
Input: Database D, Class attribute C
Output:  The set of essential rules Eset
BEGIN
1.   Initialize Essential rule set Eset as ϕ
2.   Initialize the tree T.
3.   Select Essential rules from T and store it in Eset, Select Erule (T).
4.   Generate new items from T and store it in new item set, N set.
5.   Compute the support of new items.
6.   Prune the items in N set.
7.   Update Eset by selecting essential rules from T.
8.   Update Nset by generating new items from T.
9.   Repeat steps5-8 until Nset ≠ ϕ
10.  Return the essential set of rules Eset
END
```

The algorithm 4 Select Erule is used to select the essential rules from relational database. The function generates (l+1) layer candidates from / layer nodes. This function combines a pair of sibling nodes & inserts their combination as anew node in next layer. The information from two nodes is used to initialize the new node. For example, a new node's identity set A is the union of its two sister nodes' identity sets, and its target set T is intersection of its 2 sibling nodes' target sets. If any of its l - sub item sets is not a frequently used item set, the node is eliminated. If any of its l - sub item sets does not receive sufficient support from any of available targets (classes), class is removed from the target set. If no potential targets remain, the new candidate is deleted.

```
Algorithm: 4 [Select Erule]
Input : Tree T
Output : The set of rules for essential rule selection, Eset
BEGIN
1. Construct child node $n_k$
a. Calculate the identity set of child node as union of identity sets of parents n, and $n_j$.
b. Calculate the target set of child node as the intersection of two target sets of parents n, and
   nj. Let it be $F(U_k)$.
2. If l sub item set of F ($U_k$) ≤ minsupport then remove the classfrom targetset.
3. If there is no possible target set, then remove the child node from target set.
4. Repeat the steps 1 to 3for every nodes $n_i$, and $n_j$.
END
```

A rule is pruned based on strong rule requirement. Algorithm 5 prune prunes the redundant rules. Closure property plays an important rule in removing weak rules. The function prune prunes weak rules & infrequent candidates in $(l+l)^{th}$ layer of candidate tree. The removal of weak rules is based on four lemmas and they specify the pruning rules. The support of pattern At is compared with support of its sub patterns. The number of such comparisons is bound by l+l. Once support of A_i,- is equal to support of any of its l sub pattern A, then leaf is removed from tree according to Lemma 1. Similarly other Lemmas are also used to prune weak rules.

```
Algorithm: 5 [Prune]
Input: Tree T
Output : New item set, Nset
BEGIN
```

1. If support(U, Ff = support(Ui, Fj) then remove the leaf from T.
2. If A → C and A^B →C then A^B → C is removedfrom the set.
3. If A → C, B→ C then
(i) if B→A then B → C is removed.
4. If A → C_1 and A→ C_1 ∧ C2 then A→ C is removed.
5. If there is no possible target set then remove no defrom target set.
6. Repeat steps 1-5for each node nt at (l+l) layer of the tree T.
END

The trimming of infrequent rule candidates is done in relation to a certain goal (class). When projected target set is empty, node is permanently removed, & no further particular rules are created. When these trimming rules are applied to the rules generated, they provide a trimmed set of rules. These pruning rules should not be used in tandem, which implies that if one rule is cut once, it should not be used to prune other rules. When a rule is pruned, it is removed from the set before the next pruning rule is applied. Changing the sequence of execution of these trimming rules may affect the fundamental set of rules created. The accuracy of the method is demonstrated using Lemma 4.

Lemma: 4 Algorithm 3 Erulegen generates the set of essential rules

Proof: The rules are generated using Erulegen, and the algorithm 5 prunes out the weak and infrequent rules. The tree structure is employed in the rule creation process, and all of the rules are listed at the tree's nodes. Weak and infrequent rules are closed upward. As a result, if a node is not eligible to generate a rule from fundamental rule set, same rule is applied to allnodes in the branch rooted by node. As a result, it is demonstrated that all trimmed rules are weak and unneeded rules, and the algorithm accurately creates the important rules.

CONCLUSION

This study is provides a theoretical model of the algorithm that creates only the most important and interesting rules without creating the whole set of association rules. A significant number of patterns and rules are often generated by an association rule mining approach. However, the majority of the regulations are uninteresting to users. The most advantageous and engaging regulations should be picked among those that have been developed. This work presents a set of pruning rules, which are used to develop the algorithm for creating the basic set of rules. The pruning method eliminates duplicated and weak rules from the dataset. The collection of important rules is equivalent to the set of predictive rules. The algorithm's efficiency is evaluated theoretically. The approach avoids duplicate calculation, which reduces the time required to generate the fundamental set of rules. The core rule set is less than the total collection of rules.

REFERENCES

1. Agrawal, R., Imielinski, T., and Swami, A. (1993) “Mining association rules between sets of items in massive databases”, Proceedings of the SIGMOD international conference on Management of Data, pp 207-216, May 1993.
2. Serge, B., Motwani., R and Silver stein, C., (1997) Beyond market basket: Generalizing association rules. In SIGMOD Conference, 1997
3. Korn, F., Labrinidis, A. Kotidis, Y., and Faloutsos, C., (1998) “Ratio rules: A new paradigm for fast, quantifiable data mining”. In VLDB conference, 1998.
4. Agrawal, C.C., and Philip.Y.U., (1998) “A new framework for item set generation”. In PODS symposium, 1998
5. Liu, B., Hsu, W., and Ma, Y., (1999) “Pruning and summarizing the discovered Associations”. In proceedings of the fifth international Conference on Knowledge discovery and Data mining, 1999.
6. Aha, D. W., Kibler, D., and Albert. K. (1991) “Instance - based learning algorithms” Machine Learning, 6(l);37-66, Jan 1991.

7. Ali, K., Manganaris, S., and Srikant, R. (1997) "Partial classification using association rules". In proceedings of the third international conference on knowledge discovery and data mining KDD-97, 115, Menlo Park, CA, 1997.
8. Houtsma, M., and Swami, A., (1995), "Set-oriented mining for association Rules in relational databases". In Proceedings of the eleventh international Conference on Data engineering, p25-34, Los Alamitos, CA, USA, 1995.
9. Mannila, H., Toivonen, H., and Verkamo, I., (1994) "Efficient algorithms for discovering association rules. In AAAI Workshop of Knowledge Discovery in databases, pp 181-192. AAAI Press, July 1994.
10. Park, J. S., Chen, M.,and Yu, P. S.,(1995)"An effective hash based algorithm for ining association rules". In ACM SIGMOD International conference on Management of Data, 1995.
11. Meloncon, L., & Warner, E, (2017), "Data visualizations: A literature review and opportunities for technical and professional communication". 2017 IEEE International Professional Communication Conference (ProComm).
12. AlZoubi, Omar, Fossati, Davide, Di Eugenio, Barbara and Green, Nick and Cjen, Lin. (2013). "Predicting Students' Performance and Problem Solvig Behavior from iList Log Data". Proceedings of the 21st International Conference on Computers in Education, ICCE 2013. pp. 1-6.
13. S. Visalakshi and V. Radha, (2014) "A literature review of feature selection techniques and applications: Review of feature selection in data mining," 2014 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, 2014, pp. 1-6.