ISSN 22773061

A Survey of Mining Association Rules Using Constraints

.Varsha Mashoria , Dr. Anju Singh
Barkatullah University, Institute of Technology ,Hoshangabad Road, Bhopal
varsha.uitbu@gmail.com
Barkatullah University, Institute of Technology ,Hoshangabad Road, Bhopal
asingh0123@rediffmail.com

ABSTRACT

As we all know that association rule is used to find out the rules that are associated with the items present in the database that satisfy user specified support and confidence. There are many algorithms for mining association rules. For improving efficiency and effectiveness of mining task. Constraints based mining enable users to concentrate on mining interested association rules instead of the complete set of association rule.”The constraints can be defined as the condition that a pattern has to satisfy ” . This paper provides or gives the major advancement in the approaches for association rule mining using different constraints.

Index Terms

Association rule mining, Data mining, Constraints, Patterns.

Council for Innovative Research

Peer Review Research Publishing System

Journal: **INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY**

Vol 7, No 3
editor@cirworld.com
www.cirworld.com, member.cirworld.com

INTRODUCTION

Data mining, also known as Knowledge Discovery in Databases (KDD), is the field of discovering novel and potentially useful information from large amounts of data. Data mining has been applied to a great number of fields, including bioinformatics, retail sales, and counter-terrorism. In recent years, there has been increasing interest in the use of data mining to investigate scientific questions within educational research, an area of inquiry termed educational data mining. The field of data mining has attracted the attention of researchers in different areas [5][17]. This is due to the fact that the volume of data stored in databases continues to become larger and larger. Hence, the manual analysis of such databases is in general infeasible, and data mining methods are often necessary to extract knowledge from data in a (partially-) automated fashion. The need for data mining is clear in biology, where the amount of data available in biological databases (such as protein databases) keeps increasing very fast. Mining association rules is a very important problem in the data mining field. It consists of identifying the frequent itemset and then forming conditional implication rules among them this information is useful in improving the quality of many business decision- making processes, such as customer purchasing behaviour analysis, cross-marketing and catalogue design.

BACKGROUND

(A) Association Rule

The objective of association rule mining is to find out the relationships among set of items in a database. A typical application of association rule mining is market basket analysis. An association rule is an implication of the form A B, where A and B are frequent item sets in a transaction database and A∩B=Ǿ. The rule A B can be interpreted as "if item set A occurs in a transaction, then item set B will also likely occur in the same transaction". By such type of information, market personnel can place item sets A and B within close immediacy, which may encourage the sale of these items together and develop discount strategies based on such association/correlation found in the data. Therefore, association rule mining has received a lot of attention. With the development of data mining techniques, quite a few researchers have worked on alternative patterns. In many (but not all) situations, we only care about association rules or inspiration involving sets of items that appear frequently in baskets. For example, we cannot run a good marketing strategy involving items that no one buys anyway. Thus, much data mining starts with the expectation that we only care about sets of items with high support; i.e., they appear together in many baskets. We then find association rules or causalities only involving a high-support set of items must appear in at least a certain percent of the baskets, called the support threshold. Support should not be confused with confidence. While confidence is a measure of the rule's strength, support corresponds to statistical significance. Besides statistical significance, another motivation for support constraints comes from the fact that we are usually interested only in rules with support above some minimum threshold for business reasons. If the support is not large enough, it means that the rule is not worth consideration or that it is simply less preferred [3]. We should only consider **rules** derived from item sets with **high support**, and that also have high confidence."A rule with low confidence is not meaningful."

(B) Support

The rule X ⇒ Y holds with support s if s% of transactions in D contains X ∪ Y. Rules that have a s greater than a user-specified support is said to have minimum support. (Every association rule has a support and a confidence. "The support is the percentage of transactions that demonstrate the rule.").An itemset is called frequent if its support is equal or greater than an agreed upon minimal value – the support threshold.

(C) Confidence

The rule X ⇒ Y holds with confidence c if c% of the transactions in D that contain X also contain Y. Rules that have a c greater than a user-specified confidence is said to have minimum confidence.(The confidence is the conditional probability that, given X present in a transition , Y will also be present. Confidence measure, by definition: Confidence(X=>Y) equals support(X,Y) / support(X)) As we all know that many algorithms for generating association rules[6] were presented over time. To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best known constraints are minimum thresholds on support and confidence. Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Association rule generation is usually split up into two separate steps: First, minimum support is applied to find all frequent item sets in a database. Second, these frequent item sets and the minimum confidence constraint are used to form rules. While the second step is straightforward, the first step needs more attention. Finding all frequent item sets in a database is difficult since it involves searching all possible item sets (item combinations). The set of possible item sets is the power set over and has size (excluding the empty set which is not a valid item set). Although the size of the power set grows exponentially in the number of items in , efficient search is possible using the downward-closure property of support[2][4] (also called antimonotonicity[8]) which guarantees that for a frequent item set, all its subsets are also frequent and thus for an infrequent item set, all its supersets must also be infrequent. Exploiting this property, efficient algorithms can find all frequent item sets. Some well known algorithms are Apriori, Eclat and FP-Growth, but they only do half the job, since they are algorithms for mining frequent itemsets. Another step needs to be done after to generate rules from frequent item sets found in a database.

CONSTRAINTS

Mining is vastly undetermined. To make the task more precise, we first have to specify the type of patterns considered such as frequent patterns, a clustering, a predictive model or other regularities in the data. But the discovered patterns may not be contemporary or actionable in fields where domain expertise already exists or users have strong expectations. We then have to specify what conditions the patterns have to satisfy in order to consider them as solutions to the data mining task at hand. The conditions that a pattern has to satisfy can be gracefully specified as constraints [9], stated explicitly and under direct control of the user/data miner. Over the last decade, mining with constraints has emerged as a distinct and important research area in data mining. Constraints play an important role in data mining as the use of constraints enables more efficient data mining and focuses the search for

patterns on patterns likely to be of interest to the end user. The ability to express and exploit constraints allows the data miner to inject knowledge into the process of data mining and knowledge discovery. Several sub-communities have explored the use of constraints in data mining. These include the communities concerned with the topics of clustering with constraints, finding frequent patterns under constraints, and inductive databases/queries. Clustering with constraints typically includes instance-level constraints, specifying which instances should or should not be put within the same cluster. Typical constraints in finding frequent patterns, besides frequency, include closeness or maximality. We can conceive running a mining algorithm with several constraints as running a query on a database

that stores patterns (in addition to data). Such a database is called an inductive database, and such queries are called inductive queries. Inductive databases are therefore closely related to constraint based mining. Constraint-based mining of frequent patterns, predictive models and clusterings has been considered in this research area. Tutorials by each sub-community have been presented at leading data mining conferences A major goal of this workshop is to bring together the researchers from the above research areas, namely clustering with constraints, finding frequent patterns under constraints, and inductive databases/queries. We believe it will be important to profit from each field's expertise to further the aim of practical data mining with constraints. We divide constraints into five categories:

• **Knowledge constraints** specify the type of knowledge to be mined, such as concept description, association, classification, prediction, clustering, or anomaly. This constraint, unlike other constraints, is usually specified at the beginning of a query because different types of knowledge can require different constraints at later stages.

• **Data constraints** specify the set of data relevant to the mining task. We often specify such constraints in a form similar to that of an SQL query and process them in query processing.

• **Dimension/level constraints** confine the dimensions or levels of data to be examined in a database or a data warehouse. Such constraints follow the model of a multidimensional database and demonstrate the spirit of multidimensional mining.

• **Rule constraints** specify concrete constraints on the rules to be mined.

• **Interestingness constraints** specify what ranges of a measure associated with discovered patterns are useful or interesting from a statistical point of view.

VARIOUS APPROACHES FOR MINING ASSOCIATION RULE USING DIFFERENT CONSTRAINTS

(A) Constraint-Based Rule Mining in Large, Dense Databases[16]

Auther Robert J,.Bayardo Jr, ,Rakesh Agrawal and Dimitrios Gumopulos in this paper describe a algorithm which directly exploits all user-specified constraints including minimum support, minimum confidence, and a new constraint that ensures every mined rule offers a predictive advantage over any of its simplifications. there algorithm maintains efficiency even at low supports on data that is dense (e.g. relational data). The algorithm can exploit such minimums on predictive ability during mining for vastly improved efficiency. Even given strong minimums on support and predictive ability, the rules satisfying these constraints in a dense dataset are often too numerous to be mined efficiently or comprehended by the end user. To remedy this problem, there algorithm exploits another constraint that eliminates rules that are uninteresting because they contain conditions that do not (strongly) contribute to the predictive ability of the rule. There algorithm allows the user to specify a *minimum improvement* constraint. The idea is to mine only those rules whose confidence is at least *minsup* greater than the confidence of *any* of its proper subrules, where a proper sub-rule is a simplification of the rule formed by removing one or more conditions from its antecedent. Any positive setting of minsup would prevent the undesirable rules. More generally, the minimum improvement constraint remedies the rule explosion problem resulting from the fact that in dense data-sets, the confidence of many rules can often be marginally improved upon in an overwhelming number of ways by adding additional conditions. However, instead of using a single pruning function for optimization, they use several for constraint enforcement. Also, because the itemset frequency information required for exploiting pruning functions is expensive to obtain from a large data-set, they frame there pruning functions so that they can accommodate restricted availability of such information.

(B) Mining Association Rules with Multiple Minimum Supports Using Maximum Constraints.[15]

In this paper, the auther Yeong-Chyi Lee ,Tzung-Pei Hong, and Wen-Yang Lin with maximum constraint provide another point of view about defining the minimum supports of itemsets when items have different minimum supports.Then they propose a simple algorithm based on the Apriori approach to find the large-itemsets and association rules under this constraint. The numbers of association rules and large itemsets obtained by the proposed mining algorithm using the maximum constraint are also less than those using the minimum constraint.So in this algorithm under the maximum constraint items may have different minimum supports and the maximum constraint is adopted in finding large itemsets.Under the constraint, the characteristic of level-by-level processing is kept, such that the original Apriori algorithm can be easily extended to find the large itemsets. first they finds all the large 1-itemsets L1 for the given transactions by comparing the support of each item with its predefined minimum support. After that, candidate 2- itemsets C2 can be formed from L1. Note that the supports of all the large 1-itemsets comprising each candidate 2-itemset must be larger than or equal to the maximum of the minimum supports of them. This feature provides a good pruning effect before the database is scanned for finding large 2-itemsets. The proposed algorithm then finds all the large 2-itemsets L2 for the given transactions by comparing the support of each candidate 2-itemset with the maximum of the minimum supports of the items contained in it. The same procedure is repeated until all large itemsets have been found.

(C) Association Rules Mining with Multiple Constraints[7]

In this paper, the auther with multiple constraints, proposed algorithm simultaneously copes with two different kinds of constraints for mining association rule, the proposed algorithm deal with two constraints, the two constraints are anti-monotone and monotone. they use the FP-Growth algorithm as the basic approach to mine frequent itemsets since it is more efficient compare with many other algorithms such as Apriori-like algorithm. Given a DB as well, is as two constraints a anti-monotone constraint, and is a monotone constraint. .The algorithm consists of three phases, first, the frequent 1-itemset are generated, second, they exploit the properties of the given constraints to prune search space or save constraint checking in the conditional databases. Third, for each itemset possible to satisfy the constraint, they generate its conditional database and perform the three phases in the conditional database recursively.

(D) Mining association rules with multidimensional constraints [11]

In this paper the author Anthony J.T. Lee *, Wanchuen Lin, Chun-sheng Wang we enhance the item representation by using a number of attributes to describe the item_s properties. they call them dimensional attributes (dimensions for short), because these attributes in fact form a multidimensional data space. Items with multiple dimensions are called multi-dimensional items. They call constraints against multiple dimensional attributes multi-dimensional constraints. In this paper, they choose the FP-growth method as the basic approach in their model and develop the algorithms to mine frequent itemsets with multi-dimensional constraints. this algorithms First, collect frequent items and prune infrequent items. According to the Apriori property, if a set is not frequent, all of its supersets will be infrequent as well. Thus, if a single item b is not frequent, any itemsets containing b is impossible to be frequent. after this step they exploit the properties of the given constraints to prune search space or save constraint checking in the conditional databases. and lastly, for each itemset possible to satisfy the constraint, they generate its conditional database, construct its corresponding FP-tree and perform the three phases in the FP-tree recursively. this algorithms can exploit the properties of constraints to prune search space or save constraint checking. for following the step they first classify multi-dimensional constraints into two cases according to the number of sub-constraints included: <1> a single constraint against multiple dimensions, such as max(S.cost) 6 min (S.price), where S is an itemset and each item in S contains two attributes cost and price, max(S.cost) denotes the maximum cost of all items in S and min(S.price) denotes the minimum price of all items in S; <2> a conjunction and/or disjunction of multiple sub-constraints, such as (S.cost 6 v1) ^ (S.price 6 v2), where v1 and v2 are constant values, and S.cost 6 v1 denotes every cost of the items in S is less than or equal to v1.The main idea of this algorithm is that the overall mining process can be partitioned into three phases: <1> frequency checking phase, <2> constraint checking phase, we exploit the properties of constraints to prune search space or save constraint checking in further conditional databases. Phase 2 can be divided into two sub-phases: phase 2.1 checks the potential largest frequent itemset[13], and phase 2.2 checks each individual frequent item. If the potential largest frequent itemset satisfies U, the steps in phase 2.2 and phase 3 can be skipped and <3> conditional FPtree[14] construction phase. we will know if the itemsets co-occuring with {a} [a are possible to satisfy the constraint. If yes,they generate the {a} [aconditional database, construct its FP-tree Tj{a}[a, and recursively perform the three phases in Tj{a}[a.

(E) Mining Association Rules with Item Constraints[13], An Efficient Method for Mining Association Rules with Item Constraints.[12]

In this paper the author present three integrated algorithms for mining association rules with item constraints and also discuss their tradeoffs. With the help of the Apriori algorithm they find all frequent itemsets [1]. After this they apply there algorithm which cover basic three steps or phases. In the first phase they find all frequent itemsets whose support is greater than minimum support which satisfy the Boolean expression B..In this phase they use 2 type of operation: candidate generation and counting support. The techniques for counting candidates support remain. the author introduce item constraints that help the Apriori candidate generation ,so that with its help the procedure will no longer generate all the potentially frequent itemsets as candidates. For this they consider three different approach, in which the first two approaches are the , \Multiple Joins" and \Reorder”. Second phase consist of Generation of rules from those frequent itemsets, they also need to find the support of all subsets of frequent itemsets that do not satisfy B. Phase 3 consist of the Generated rules from the frequent itemsets found in Phase 1, by using the frequent itemsets found in Phases 1 and 2 to compute confidences, as in the Apriori algorithm. Author in this paper instead of first generating a set of selected items S from B, finding all frequent itemsets that contain one or more items from S and then applying B to filter the frequent itemsets, they can directly use B in the candidate generation procedure. They first make a pass over the data to find the set of the frequent items F. Lb 1 is now the set of those frequent 1-itemsets that satisfy B. Generating the set of selected items, S is more expensive since for elements in B that include an ancestor or descendant function, they also need to find the support of the ancestors or descendants. Checking whether an itemset satisfies B is also more expensive since they may need to traverse the hierarchy to find whether one item is an ancestor of another. Cumulate does not count any candidates with both an item and its ancestor since the support of such an itemset would be the same as the support of the itemset without the ancestor. Cumulate only checks for such candidates during the second pass (candidates of size

2). For subsequent passes, the apriori candidate generation procedure ensures that no candidate that the other fast algorithm[10],EstMerge, is similar to Cumulate, but also uses sampling to decrease the number of candidates that are counted contains both an item and its ancestor will be generated. In tradeoffs Reorder and MultipleJoins will have similar performance since they count exactly the same set of candidates. Reorder can be a little faster during the prune step of the candidate generation, since checking whether an k-subset contains a selected item takes O(1) time for Reorder versus O(k) time for MultipleJoins. However, if most itemsets are small, this difference in time will not be significant. Execution times are typically dominated by the time to count support of candidates rather than candidate generation. Hence the slight differences in performance between Reorder and MultipleJoins are not enough to justify choosing one over the other purely on performance grounds. The choice is to be made on whichever one is easier to implement. Direct has quite different properties than Reorder and MultipleJoins. They illustrate the tradeoffs between Reorder/MultipleJoins and Direct with an example. they use \Reorder" to characterize both Reorder and MultipleJoins in the rest of this comparison. Another algorithm[12] with same constraint also do the same work that is they also mine the association rule with this constraint but the worse part in the existing mining algorithms is that it will repeat reading the whole database several passes for a subsequent query even it involves the same specified items as the previous query but changes only the minimum confidence and support due to which the author in this research, present a novel mining algorithm that can efficiently discover the association rules between the user-specified items or categories with the help of feature extraction approach and in this only one scan of the database is needed for each query; from which the disk access overhead can be reduced substantially and the query be responded quickly .They also investigate another problem related to mining generalized association rules. They call it as “mining categorized association rules” for mining categorized association rules efficiently. The main features of the method are For each interested item or category, a compressed feature vector and feature record are built to represent the occurrence patterns of the items belonged to this category. The feature vector and feature record are built only once while reading the database first time. Then, the associations between the interested items or categories are constructed by using the feature record information and performing simple logical operations on the feature vectors without reading the large database again. Hence, all the disk access overhead, calculation time for mapping the items to belonged categories, and the calculation time for finding the associations between the categories can be reduced substantially. Besides, this they also describe a methodology to provide users with useful information regarding the database to be mined, like the estimated number of potential rules under various settings of the constraints. Thus, users can make suitable constraint settings more easily and less mining processes are needed. So with this method their proposed can reduce both the disk access time and computation

CONCLUSION

This paper present a summarized survey report regarding association rule mining with the help of different constraints , this paper shows that with the use of certain constraint we can generate better rules with many advantages .There are number of constraint presented with the help of all those constraint we can generate better rule with different techniques. the constraints factor reduce the unnecessary rule set in process of rule generation. Association rule mining also face problem of space and time ,constraint also solve this problem to an extent. From all these constraint based rule we get an idea for developing new strategies, which can give more better result from the above one.

REFERENCES

[1] Agrawal, R.and Shafer, J. "Parallel mining of association rules". IEEE Transactions on Knowledge and Data Engineering 8(6).(1996)

[2] Agrawal, R.; Imieliński, T.; Swami, A."Mining association rules between sets of items in large databases".Proceedings of the (1993) ACM SIGMOD international conference on Management of data - SIGMOD '93. pp. 207.

[3] Rakesh Agrawa; Tomasz Imielinski; Arun Swami. "Mining Association Rules between Sets of Items in Large Databases". IBM Almaden Research Center 650 Harry Road, San Jose, CA 95120.

[4] Tan, Pang-Ning; Michael, Steinbach; Kumar, Vipin (2005). "Chapter 6. Association Analysis: Basic Concepts and Algorithms". Introduction to Data Mining. Addison-Wesley. ISBN 0-321-32136-7.

[5] A. A. Freitas ." Data Mining and Knowledge Discovery with Evolutionary Algorithms". Springer-Verlag .(2002).

[6] Qiankun and Sourav S. Bhowmick ."Association Rule Mining: A Survey". Nanyang Technological University, Singapore.

[7] Li Guang-yuana,ba, Cao Dan-yanga, Guo Jianweia . "Association Rules Mining with Multiple Constraints" a)School of Computer and Communication Engineering, University of Science&Technology Beijing, b)Shool of Computer and Information Engineering, Guangxi Teachers Education University, Nanning, China. Procedia Engineering . www.elsevier.com/locate/procedia.(2011)

[8] Pei, Jian; Han, Jiawei; and Lakshmanan, Laks V. S."Mining frequent itemsets with convertible constraints". in Proceedings of the 17th International Conference on Data Engineering, April 2–6, 2001, Heidelberg, Germany.(2001)

[9] Lo¨ıc Cerf_ J´er´emy Besson† C´eline Robardet‡ Jean-Fran¸cois Boulicaut§.Data-Peeler:"Constraint-Based Closed Pattern Mining in n-ary Relations".

[10] Srikant, R., and Agrawal, R. "Mining Generalized Association Rules". In Proc. of the 21st Int'l Conference on Very Large Databases.(1995).Toivonen, H. "Sampling large databases for association rules". In Proc. of the 22nd Int'l Conference on Very Large Databases, (1996).

[11] Anthony J.T. Lee ; Wan-chuen Lin; Chun-sheng Wang. "Mining association rules with multidimensional constraints". Department of Information Management .(2005).

[12] Shin-Mu Vincent Tseng . "An Efficient Method for Mining Association Rules with Item Constraints". Computer Science Division, EECS Department University of California, Berkeley.

[13] Srikant, R., Vu, Q., Agrawal, R. "Mining association rules with item constraints". In: Proceedings of ACM International Conference on Knowledge Discovery and Data Mining, pp. 67–73.(1997).

[14] Han, J., Pei, J., Yin, Y." Mining frequent patterns without candidate generation". In: Proceedings of ACM-SIGMOD, pp. 1– 12.(2000).

[15] Yeong-Chyi Lee ,Tzung-Pei Hong ,Wen-Yang Lin "Mining Association Rules with Multiple Minimum Supports Using Maximum Constraints".a)Institute of Information Engineering, I-Shou University, Kaohsiung, 840, Taiwan, R.O.C. bDepartment of Electrical Engineering, National University of Kaohsiung.

[16] Roberto J. Bayardo Jr. "Constraint-Based Rule Mining in Large, Dense Databases". IBM Almaden Research Center Rakesh Agrawa Dimitrios Gunopulos Appears in Proc. of the 15th Int'l Conf.on Data Engineering, 188-197.(1999).

[17] I. H. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques.Morgan Kaufmann, 2nd edition.(2005).

Ms. Varsha Mashoria born in Bhopal in 18 auguest 1984,done B.E in Computer Science Engineering from University Institute of Technology , Barkatullah University, Bhopal, Madhya Pradesh, India in 2008. My research interest is in Data mining, Database Management System and in Networking.

She is working as lecturer in University Institute of Technology , Barkatullah University, Bhopal, Madhya Pradesh, India in Computer Science Department from 28/july/2008 to 15/may/2013. She is perusing M.Tech in Computer Science from University Institute of Technology, Barkatullah University, Bhopal.

Dr. Anju Singh completed B.E. in computer science engineering, 2003. M.Tech in Information Technology, 2008. PhD in 2013. guided 10 M.Tech dissertation. Currently working in Barkatullah university Institute of Technology, Bhopal, . Publications:- 11 International Journal and 11 International Conference