International journal of Advance Engineering and Research Development (IJAERD)
Volume 1 Issue 1, February 2014, ISSN: 2348 - 4470

Discover Multi-Label Classification using Association Rule Mining

Kanu Patel[1], Niki Kapadia[2], Mehul Parikh[3]

[1]*Assist. Prof, I.T Depart, BVM Engineering College, V.V.Nagar, kanu.patel@bvmengineering.ac.in*

[2]*Assist. Professor, Computer Department, BIT,Babaria nikikapadia4@gmail.com*

[3]*Assist. Prof., Computer Department, GEC, Modasa, mehulcparikh@gmail.com*

Abstract: Association rule mining and classification are two major task of data mining. They are attracted wide attention in both research and application area recently. I propose a method for classification rules from multi-label dataset using association rule analysis. Multi label dataset contains multiple class label attribute for predict target variable. We classify that attribute using different approaches like naviye-baies, decision tree, Back propagation, Neural based classification and association rule based classification. Finding association rule from dataset we have to apply various algorithms like Apriori, Fp-Growth, etc. I proposed Fp-Growth algorithm for finding association rule from dataset because of Fp-Growth is an improved algorithm of Apriori and Fp-Growth is more efficient than Apriori. The number of associations present in even moderate sized databases can be, however, very large – usually too large to be applied directly for classification purposes. Therefore, any classification learner using association rules has to perform three major steps: Mining a set of potentially accurate rules, evaluating and pruning rules, and classifying future instances using the found rule set. Implementation of improved Fp-Growth algorithm gives accurate and classify rule. This approach is more effective, accurate and efficient than other tradition algorithms.

Keywords: Rule mining; Association rule, Mulans; Classification; Fp-Growth ;ImprovedFp-Growth;

I. INTRODUCTION

The classification problem is to build a model, which, based on external observations, assigns an instance to one or more labels. A set of examples is given as the training set, from which the model is built. A typical assumption in classification is that labels are mutually exclusive, so that an instance can be mapped to only one label. However, due to ambiguity or multiplicity, it is quite natural that most of the applications violate this assumption, allowing instances to be mapped to multiple labels simultaneously. For example, a movie being mapped to action or adventure, or a song being classified as rock or ballad, could all lead to violations of the single-label assumption? Multi-label classification consists in learning a model from instances that may be associated with multiple labels, that is, labels are not assumed to be mutually exclusive. Most of the proposed approaches.[1] for multi-label classification employ heuristics, such as learning independent classifiers for each label, and employing ranking and thresholding schemes for classification. Although simple, these heuristics do not deal with important issues such as small disjuncts and correlated labels.

In essence, small disjuncts are rules covering a small number of examples, and thus they are often neglected. The problem is that, although a single small disjuncts covers only few examples, many of them, collectively, may cover a substantial fraction of all examples, and

simply eliminating them may degrade classification accuracy [2]. Small disjuncts pose significant problems in single-label classification, and in multi label classification these problems are worsened, because the search space for disjuncts increases due to the possibly large number of label combinations. Also, it is often the case that there are strong dependencies among labels, and such dependencies, when properly explored, may provide improved accuracy in multi-label classification. we propose an approach which deals with small disjuncts while exploring dependencies among labels. To address the problem with small disjuncts, we adopt a lazy associative classification approach. Instead of building a single set of class association rules (CARs) that is good on average for all predictions, the proposed lazy approach delays the inductive process until a test instance is given for classification, therefore taking advantage of better qualitative evidence coming from the test instance, and generating CARs on a demand-driven basis. Small disjoints are better covered, due to the highly specific bias associated with this approach. We address the label correlation issue by defining multi-label class association rules (MCARs), a variation of CARs that allows the presence of multiple labels in the antecedent of the rule. The search space for MCARs is huge and to avoid an exhaustive enumeration. Which would be necessary to find the best label combination, we employ a novel heuristic called progressive label focusing, which makes feasible the exploration of associations among labels.

.

II. PRELIMINARY

In this section, we explain the concept of association rule mining and classification.

2.1 Association rules Mining

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness.[1] Based on the concept of strong rules, Rakesh Agrawal et al.[2] introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule $\{\text{onions}, \text{potatoes}\} \Rightarrow \{\text{burger}\}$ found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection, Continuous production and bioinformatics. As opposed to sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.

Many algorithms for generating association rules were presented over time.

Some well known algorithms are Apriori, Eclat and FP-Growth, but they only do half the job, since they are algorithms for mining frequent item sets. Another step needs to be done after to generate rules from frequent item sets found in a database.

2.1.1 Apriori algorithm

Apriori[6] is the best-known algorithm to mine association rules. It uses a breadth-first search strategy to count the support of item sets and uses a candidate generation function which exploits the downward closure property of support.

2.1.2 Eclat algorithm

Eclat is a depth-first search algorithm using set intersection.

2.1.3 FP-growth algorithm

FP stands for frequent pattern. In the first pass, the algorithm counts occurrence of items (attribute-value pairs) in the dataset, and stores them to 'header table'. In the second pass, it builds the FP-tree structure by inserting instances. Items in each instance have to be sorted by descending order of their frequency in the dataset, so that the tree can be processed quickly. Items in each instance that do not meet minimum coverage threshold are discarded. If many instances share most frequent items, FP-tree provides high compression close to tree root.

Recursive processing of this compressed version of main dataset grows large item sets directly, instead of generating candidate items and testing them against the entire database. Growth starts from the bottom of the header table (having longest branches), by finding all instances matching given condition. New tree is created, with counts projected from the original tree corresponding to the set of instances that are conditional on the attribute, with each node getting some of its children counts. Recursive growth ends when no individual items conditional on the attribute meet minimum support threshold, and processing continues on the remaining header items of the original FP-tree. Once the recursive process has completed, all large item sets with minimum coverage have been found, and association rule creation begins.

2.2 Classification.

Classification refers to an algorithmic procedure for assigning a given piece of input data into one of a given number of categories. An example would be assigning a given email into "spam" or "non-spam" classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, that maps input data to a category.

2.2.1Multi-label classification

In machine learning, multi-label classification is a variant of the classification problem where multiple target labels must be assigned to each instance. Multi-label classification should not be confused with multiclass classification, which is the problem is categorizing instances into more than two classes.Several problem transformation methods exist for multi-label classification; a common one is the binary relevance (BR) where one binary classifier is trained per label. Various other transformations exist: The Label Combinations (LC) transformation, creates one binary classifier for every possible label combination. Other transformation methods include RAkEL[5] and Chain Classifiers(CC)[6]. Various problem transformation methods have been developed such as Ml-kNN[7], a variant of the k-nearest neighbors lazy classifiers.

III. THE PROPOSED ALGORITHM

In this Section, We have to apply methods or algorithms on datasets for generating rules. So first we have to preprocessing the datasets after getting final data we have to apply FP-growth algorithms for finding association rule and then after we have to prune that rules so ultimately we got classify rules of that datasets. This section combines the methods for class association rule mining, pruning and classification in different ways and evaluates their performances. The results are not only used to compare the performance of the different classification approaches but also to evaluate the underlying mining processes. The main focus of the experiments is on the rule mining algorithms. Therefore classification using association rules provides a mechanism by which to compare the different mining approaches. In this chapter we compare Fp-growth and Improved Fp-growth. Therefore, we primarily focus on their interestingness measures, because this is the main difference between the two mining algorithms. The different interestingness measures induce a different rule ranking.

3.1 Datasets

In order to compare the different approaches we use standard benchmark datasets from the UCI Machine Learning Repository our selection of datasets and their properties. Their size ranges from a few tens of instances to one thousand instances and they are composed of varying numbers of numeric and nominal attributes. The class attribute is always nominal. Some of them contain missing values. In this paper we use Contact-lenses, soybean,CAL500 and weather nominal datasets for finding rules.

3.2 The proposed algorithm

In this paper first we have to select datasets from the UCI data repository or MULAN. The data sets contain multiple class label attribute so we can use it .FP-Growth algorithm apply on that dataset. now we find occurrence of all item sets and arrange in descending order.. Based on that, Construct FP-tree for storing data in tree format. After that generates Association rules.

Algorithm: Create FP-tree

Input : A database DB and a minimum support threshold ξ

Output : FP-tree

```
Procedure CreateFP-tree
        scan the DB once to collect the frequent items and their support then sort in
        support ascending and create the header table.
FP-tree is null
for each transaction ti in DB
        select and sort the frequent items in ti according to the order in the header
        table
        callInsertTree(FP-tree, ti)
end for

Procedure InsertTree( root, tran)
        for each item ki in transaction
        if root has a child N that N.item_name = ki
                increment N 's count by 1 & root = N
```

else

create the new node ki is the child of root link the header table to node

end if

end for

After apply FP-Growth algorithm we get some rule. This rules are not classified now we pruned that rules using any classifier but In this paper we introduce one new classified Improved Fp-Growth algorithm. In this classifier apply the concept like this. Right side rules contains always class label attribute and Prune that rules so we can eliminating redundant rules.

IV. EXPERIMENTS

In this paper, Experimental result is shown that the Improved FP Growth algorithm is more accurate in terms of rules as well as time over here we compare both algorithm for weather-nominal and CAL500 datasets. Improved Fp-growth algorithm is faster than FP growth and is generate accurate rule or we can say more efficient. Below table & diagram is gives the comparison.

Dataset	Based on Rules		Based on time(sec)	
	Fp-growth	Improved Fp-growth	Fp-growth	Improved Fp-growth
Weather.nominal	5784	16	13	0.07
CAL500	12764	28	22	1.6

Table 1: Comparison of rule mining and classification on two datasets

Comparison based on number of rule and time taking for execution between fp-growth & Improved Fp-growth. In diagram Red is for CAL500 and blue is for weather-nominal. Y axis denoted no of rule and time respectively in diagram

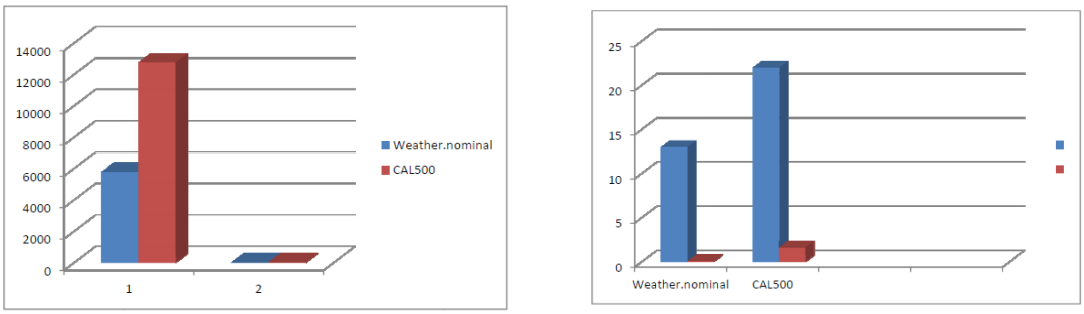

Fig. 1 In graph representation of comparison

V. CONCLUSION

In this paper we conclude that It produces classifiers that contain rules with multiple labels.It's an efficient method for discovering rules that requires only two scan over the training data. We have to use Fp-Growth algorithm for finding association rule so Fp-growth requires only two time scan the datasets so we can reduce our time using this algorithm. Using multi-label classification we overcome of all problems arise in single-label classification In addition, the propose technique is able to extract rules with up to multiple labels from the datasets, which results in a higher classification accuracy for test instances. Using this method we generate generalize rule and reduce number of association rule. Which

prunes redundant rules, and ensures only effective ones are used for classification. so we can optimize the memory space and generate accurate rule.

VI. REFERENCES

[1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[3] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[4] K. Elissa, "Title of paper if known," unpublished.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[5] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[6] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989

