ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 11, November 2013

Ensemble of Classifiers Based on Association Rule Mining

Divya Ramani, Dept. of Computer Engineering, LDRP, KSV, Gandhinagar, Gujarat, 9426786960.
Harshita Kanani, Assistant Professor, Dept. of Computer Engineering, LDRP, KSV, Gandhinagar, Gujarat, 9924505607.
Chirag Pandya, Assistant Professor, Dept. of Computer Engineering, LDRP, KSV, Gandhinagar, Gujarat, 9687225780.

Abstract— **In Data Mining, Classification is the process of finding and applying a model to describe and distinguish data classes, concepts and values. The model that is built is called a Classifier or Predictor depending upon whether the model finds the unknown data class or data value. Single classifier may not be very much accurate; Ensemble systems use an "ensemble" or group of classifiers to improve the accuracy.**

Associative classification is an approach in data mining that utilizes the association rule discovery techniques to build classification systems, also known as associative classifiers. Ensemble methods have been called the most powerful development in data mining. They combine multiple classification models into one, usually more accurate one. Here in this thesis an efficient approach for classification using association rule ensemble is proposed. This presents an associative classification algorithm BCAR, which remove the frequent items that cannot generate frequent rules directly by adding the count of class labels. Main purpose is to ensemble association rule classifier without loss of performance & accuracy of the resultant classifier.

***Index Terms*—Ensemble System, Association, Classification.**

I. INTRODUCTION

Data mining is a part of a process called KDD-knowledge discovery in databases. This process consists basically of steps that are performed before carrying out data mining, such as data selection, data cleaning, pre-processing, and data transformation.[1]

- **Association rule mining**

Association rule is a data mining technique which discovers the strong associations or correlation relationships among given data.

The discovery of association rules has been known to be useful in selective marketing, decision analysis, and business management. An important application area of mining association rules is the market basket analysis, which studies the buying behaviors of customers by searching for sets of items that are frequently purchased together. With the increasing use of the record-based databases whose data is being continuously added, recent important applications have called for the need of incremental mining. In dynamic transaction databases, new transactions are appended and obsolete transactions are discarded as time advances.

Support for an itemset X in a transactional database D is defined as count(X) / |D|. [1]

For an association rule X ∪ Y, we can calculate

Support (X ∪ Y) = support (XY) = support(X union Y).

Confidence (X ∪ Y) = support (XY) / support(X).

Support (S) and Confidence (C) can also be related to joint probabilities and conditional probabilities as follows.

Support (X ∪ Y) = P (XY).

Confidence (X ∪ Y) = P(Y/X).

- **Ensemble System**

In data mining, a model generated by machine learning can be regarded as an expert. An obvious approach to making decisions more reliable is to combine the output of different models.

Ensemble methods have been called the most influential development in Data Mining and Machine Learning in the past decade. They combine multiple models into one usually more accurate than the best of its components. Ensembles can provide a critical boost to industrial challenges from investment timing to drug discovery, and fraud detection to recommendation systems where predictive accuracy is more vital than model interpretability.

Several machine learning techniques do this by learning an **ensemble of models** and using them in combination: prominent among these are schemes called *bagging*, *boosting*, and *stacking*. They can all, more often than not, increase predictive performance over a single model. And they are general techniques that can be applied to numeric prediction problems and to classification tasks.

Ensemble based algorithms, such as bagging, boosting, AdaBoost, stacked generalization, and hierarchical mixture of experts; as well as commonly used combination rules, including algebraic combination of outputs, voting based techniques, behavior knowledge space, and decision templates. [8]

Reasons for Using Ensemble Based Systems

There are several theoretical and practical reasons why we may prefer an ensemble system:

Statistical Reasons: Combining the outputs of several classifiers by averaging may reduce the risk of an unfortunate selection of a poorly performing classifier. The averaging may or may not beat the performance of the best classifier in the ensemble, but it certainly reduces the overall risk of making a particularly poor selection.

Large Volumes of Data: The amount of data to be analyzed can be too large to be effectively handled by a single classifier. Training a classifier with such a vast amount of data is usually not practical; partitioning the data into smaller subsets, training different classifiers with different partitions of data, and combining their outputs using an intelligent combination rule often proves to be a more efficient approach.

Too Little Data: In the absence of adequate training data, resampling techniques can be used for drawing overlapping random subsets of the available data, each of which can be used to train a different classifier, creating the ensemble.

Divide and Conquer: The decision boundary that separates data from different classes may be too complex, or lie outside the space of functions that can be implemented by the chosen classifier model. Two class problem with a complex decision boundary.

A linear classifier, one that is capable of learning linear boundaries, cannot learn this complex non-linear boundary. However, appropriate combination of an ensemble of such linear classifiers can learn this (or any other, for that matter) non-linear boundary.

In a sense, the classification system follows a divide-and-conquer approach by dividing the data space into smaller and easier to learn partitions, where each classifier learns only one of the simpler partitions. The underlying complex decision boundary can then be approximated by an appropriate combination of different classifiers.

Data Fusion: If we have several sets of data obtained from various sources, where the nature of features are different, a single classifier can not be used to learn the information contained in all of the data. Applications in which data from different sources are combined to make a more informed decision are referred to as data fusion applications, and ensemble based approaches have successfully been used for such applications.[8]

Two Key Components of an Ensemble System

All ensemble systems consist of two key components. First, a strategy is needed to build an ensemble that is as diverse as possible. Some of the more popular ones, such as bagging, boosting, AdaBoost, stacked generalization, and mixture of experts.

A second strategy is needed to combine the output of individual classifiers that make up the ensemble in such a way that the correct decisions are amplified and incorrect ones are cancelled out. [8]

Creating an Ensemble

1. Bagging
2. Boosting
3. Stack generalization

Bagging

Bagging, short for bootstrap aggregating, Most intuitive and simplest to implement, with a surprisingly good performance.

Diversity in bagging is obtained by using bootstrapped replicas of the training data: different training data subsets are randomly drawn with replacement from the entire training data. Each training data subset is used to train a different classifier of the same type. Individual classifiers are then combined by taking a majority vote of their decisions. For any given instance, the class chosen by most classifiers is the ensemble decision.

Bagging is particularly appealing when available data is of limited size.

Neural networks and decision trees are good candidates for this purpose, as their instability can be controlled by the selection of their free parameters. [8]

Boosting

"Boosting" is a general method for improving the performance of any learning algorithm.

Boosting can be used to significantly reduce the error of any "weak" learning algorithm that consistently generates classifiers which need only be a little bit better than random guessing.

Boosting works by repeatedly running a given weak learning algorithm on various distributions over the training data, and then combining the classifiers produced by the weak learner into a single composite classifier.

Compared boosting to "bagging," a method which works in the same general fashion (i.e., by repeatedly rerunning a given weak learning algorithm, and combining the computed classifiers), but which constructs each distribution in a simpler manner. These includes (1) an algorithm that searches for very simple prediction rules which test on a single attribute (2) an algorithm that searches for a single good decision rule that tests on a conjunction of attribute tests.

The main conclusion of our experiments is that boosting performs significantly and uniformly better than bagging when the weak learning algorithm generates fairly simple classifiers.

An algorithm that generates classifiers that can merely do better than random guessing, can be turned into a strong learner that generates a classifier that can correctly classify all but an arbitrarily small fraction of the instance.

Boosting, the algorithm is now considered as one of the most important developments in the recent history of machine learning.

Boosting also creates an ensemble of classifiers by resampling the data, which are then combined by majority voting. [8]

In boosting, resampling is strategically geared to provide the most informative training data for each consecutive classifier. In essence, boosting creates three weak classifiers: the first classifier C 1 is trained with a random sub- set of the

available training data. The training data subset for the second classifier C 2 is chosen as the most informative subset, given C 1. That is, C 2 is trained on a training data only half of which is correctly classified by C 1, and the other half is misclassified. The third classifier C 3 is trained with instances on which C 1 and C 2 disagree. The three classifiers are combined through a three-way majority vote.

Adaboost

AdaBoost is a more general version of the original boosting algorithm.

AdaBoost.M1 and AdaBoost.R are more commonly used, as they are capable of handling multiclass and regression problems.

AdaBoost generates a set of hypotheses, and combines them through weighted majority voting of the classes predicted by the individual hypotheses. The hypotheses are generated by training a weak classifier, using instances drawn from an iteratively updated distribution of the training data. This distribution update ensures that instances misclassified by the previous classifier are more likely to be included in the training data of the next classifier. Hence, consecutive classifiers' training data are geared towards increasingly hard-to-classify instances.

AdaBoost is ready for classifying unlabeled test instances. Unlike bagging or boosting, AdaBoost uses a rather undemocratic voting scheme, called the weighted majority voting. [8]

II. EXISTING METHODOLOGIES

Various Approaches Available for Association Rule Based Classification

As mentioned, association-rule based classification helps to discover classification rules more efficiently, we are in this paper also going to implement the association methods to discover classification rules. So let us take some light on the currently available approaches or methods concerning with association rule methods in depth.

CBA (Classification Based Association rules) Algorithm

Classification Based on Association rules (CBA) algorithm is an integration of two important data mining techniques: Classification rule mining and association rule mining. The strength of CBA is its ability to use the most accurate rules for classification. However, the existing techniques based on exhaustive search face a challenge in the case of huge amount data due to its computation complexity. CBA deals with centralized databases. In today's Internet environment, the databases may be scattered over different locations and heterogeneous. We will combine CBA and distributed techniques to develop a distributed CBA algorithm to mine distributed and heterogeneous databases. [7]

CAEP(Classification by Aggregating Emerging Pattern)

It is suitable for many applications ,even those with large volumes of high dimensional data. It does not depend on dimensional reduction on data.It is usually equally accurate on all classes even if their populations are unbalanced. Experiments shows that CAEP outperforms both CBA and C4.5. [10]

CMAR (Classification Based on Multiple Association Rule)

An associative classification method, CMAR, i.e., **C**lassification based on **M**ultiple **A**ssociation **R**ules. The method extends an efficient frequent pattern mining method, FP-growth, constructs a class distribution-associated FP-tree, and mines large database efficiently. Moreover, it applies a CR-tree structure to store and retrieve mined association rules efficiently, and prunes rules effectively based on confidence, correlation and database coverage. The classification is performed based on a weighted analysis using multiple strong association rules. [5]

CARGBA (Classification based on Association Rule Generated in a Bidirectional Approach)

CARGBA generates the rules in two steps.

1] It generates a set of high confidence rules of smaller length with support pruning. Then augments this set with some high confidence rules of higher length with support below minimum support. The purpose is not knowledge extraction but to obtain better accuracy.

2] Rules are generated as specific as possible. They have higher length and therefore lower support and thus they easily capture the specific characteristics about the data set. So if there is a classification pattern that exists over very few instances or there are exceptions to the general rule, then it will be covered by the specific rules. Since these instances are small in number, specific rules are produced without any support pruning. This result is a better mixture of class association rules. All the rules generated by CARGBA rule generator will not be used in the classification. So, the second part builds a classifier with the essential rules and is called CARGBA Classifier Builder. [10]

CPAR (Classification based on Predictive Association Rules)

CPAR adopts a greedy algorithm to generate rules directly from training data.

To avoid over fitting it uses expected accuracy to evaluate each rule and uses the best k rules in prediction. CPAR inherits the basic idea of FOIL in rule generation also integrates the features of associative classification in predictive rule analysis.

CPAR generates a smaller set of rules with higher quality and lower redundancy. So CPAR is much more time efficient in both rule generation and prediction. It also achieves as high accuracy as associative classification.

To avoid generating redundant rules it generates each rule

by comparing with the set of "already-generated" rules When predicting the class label it uses the best k rules. It uses dynamic programming to get better results. In rule generation instead of selecting only the best literal all the close-to-the-best literals are selected. [10]

CARPT (classification algorithm based on trie-tree of associative rule)

In this method first scene whole data base and convert it into two-dimensional array, in which the horizontal position said the item number and types of properties, the vertical position said the transaction number.

According to the definition and the construction method of Trie-tree. The next work is to export those association rules that meet the given minimum confidence and take category labels for rule consequent from Trie-tree.

Property 1 of Trie-tree: If a sub-tree takes a non- frequent bucket for root node, then all the buckets of the sub- tree are not frequent.

Property 2 In order <I1, I2,…, In>, there cannot be frequent itemset which contains two or more items take in for a prefix. When p>q, frequent item Ip cannot take Iq for a prefix.

Using property 1 and property 2, remove the frequent item that cannot generate frequent rules directly when transform the database into vertical bitmap of two-dimensional array to improve the achievement of Trie-tree and reduce the number of its nodes. [3]

III. PROPOSED METHOD

BCAR: Boosting on Multiple Classifiers Based on Association Rule Mining

BCAR, i.e. Boosting on multiple classifiers based on association rule mining. The method extends an **ef**ficient frequent pattern mining method, FP-growth, and mines large database efficiently. It applies a CR-tree structure to store and retrieve mined association rules efficiently. Select the subsets of rules and then does classification. This completes the first iteration then for the next iteration for selecting subset of rules change threshold as well as support and confidence and makes a new subset and classification is performed again after taking different subset. The classification is performed based on a weighted **x2** analysis using multiple strong association rules. At last Boosting will be apply on this multiple classifier and collect votes.

Develop a BCAR for accurate and efficient classification and make the following contributions.

First, instead of relying on a single rule for classification, BCAR determines the class label by a set of rules. Second, to improve both accuracy and efficiency, CR-tree is used to compactly store and efficiently retrieve a large number of rules for classification. Third, to speed up the mining of complete set of rules, BCAR use FP-growth method.

Mining Class Association Rules Passing Support and Confidence Thresholds

BCAR first mines the training data set to find the complete set of rules passing certain support and confidence thresholds. To make mining highly scalable and efficient, BCAR adopts a variant of FP-growth method.

Storing Rules in CR-tree

Once a rule is generated, it is stored in a CR-tree. A CR-tree is built for the set of rules. All the attribute values appearing at the left hand side of rules are sorted according to their frequency, i.e., the most frequently appearing attribute value goes first. [5]

The CR-tree structure has some advantages as follows

CR-tree is a compact structure. It explores potential sharing among rules and thus can save a lot of space on storing rules

CR-tree itself is an index for rules. Once a CR-tree is built, rule retrieval becomes efficient. [5]

ALGORITHM

1. Start
2. Load a training data from the database that fits in the memory.
3. Apply FP-Growth to find the frequent itemsets with the minimum threshold value.
4. Store frequent itemsets in X, where Suppose X is set of the frequent item set generated by FP-Growth algorithm.
5. Store All rule in CR-TREE,
6. Select value of support and threshold.
7. Select set of rules from CR-Tree , based on Support and threshold.
8. Classification based on X2.
9. If the desired number of classification model is not prepaid , then go to Step 6.
10. Vote for classification using boosting.
11. Call BCAR to create classifier.
12. Stop

Description of an Algorithm

First start it and Load the training data from the database. In the third step for finding frequent itemsets with the minimum threshold apply FP-Growth method. In the fourth step store this generated itemsets which is generated by FP-Growth algorithm. Generate association rules and store all rules into the CR-tree. In the sixth step select value of support and threshold and then select set of rules from CR-tree based on selected support and threshold. In the eighth step it does classification based on selected subset. Until the desired number of classification model not prepared repeat step six to nine. In the tenth step vote for classification using boosting and for classification call BCAR to combine multiple classifiers and create more accurate one.

IV. CONCLUSION

In this paper, we examined two major challenges in ensemble of classifiers based on association rule mining. 1] efficiency of handling huge number of mined association rules & 2] effectively predict new class labels with high classification accuracy. We proposed classification method BCAR i.e. boosting on multiple classifiers based on association rule mining. The method has several features 1] it leads to better overall classification accuracy. 2] it applies CR-tree structure to store & retrieve mined association rules efficiently. 3] it highly effective at classification of various kinds of databases.

V. ACKNOWLEDGMENT

We would like to thanks my research guide Mrs. Harshita Kanani for many discussions and for providing a guidelines throughout my research work.

REFERENCES

[1] jiawei han, micheline kamber "Data mining – concept and techniques"
[2] Agrawal R, Imielinski T, and Swami "A., Mining association rules between set of items in large databases" In Proceedings of ACM SIGMOD, pages 207-216, May 1993
[3] Yang Junrui, Xu Lisha, He Hongde "A Classification Algorithm Based on Association Rule Mining" College of Computer Science and Technology Xi'an University of Science and Technology Xi'an, China-2012
[4] Yingqin Gu, Hongyan Liu, Jun He, Bo Hu and Xiaoyong Du "MrCAR: A Multi-relational Classification Algorithm based on Association Rules" Key Labs of Data Engineering and Knowledge Engineering, MOE, China Information School, Renmin University of China, Beijing, 100872, China School of Economics and Management, Tsinghua University, Beijing,100084
[5] Wenmin Li Jiawei, Han Jian Pei_ "CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules" School of Computing Science, Simon Fraser University Burnaby, B.C., Canada
[6] Fadi Thabtah, Peter Cowling, Yonghong Peng "MCAR: Multi-class Classification based on Association Rule" Modelling Optimization Scheduling And Intelligent Control Research Centre University of Bradford, Department of Computing, University of Bradford, BD7 IDP, UK
[7] Bing Liu, Wynne Hsu, Yiming Ma "Integrating classification and association rule mining" Department of Information Systems and Computer Science National University of Singapore Lower Kent Ridge Road, Singapore 119260-1998
[8] Robi polikar "Ensemble based system in decision making" IEEE circuits and systems magazine-2006
[9] Yoav Freund, Robert E. Schapire "Experiments with a New Boosting Algorithm" AT&T Laboratories 600 Mountain Avenue Murray Hill, NJ 07974-0636 Machine Learning: Proceedings of the Thirteenth International Conference, 1996.
[10] Sohil Gambhir, Prof. Nikhil Gondliya "A Survey of Associative Classification Algorithms" International Journal of Engineering Research & Technology (IJERT)]