ISSN-2249 5460

INTERNATIONAL
JOURNAL OF
MATHEMATICAL SCIENCES,
TECHNOLOGY AND HUMANITIES

www.internationalejournals.com

Available online at www.internationaleJournals.com

InternationaleJournals

International Journal of Mathematical Sciences, Technology and Humanities 1 (2018) Vol. 8, Issue 1, pp: 1 – 11

Classification Based on Association Rule Mining

Dr. K. V. Sobha Rani
Department of Computer Applications, P R Govt. Degree College,
Kakinada, A.P.,India

Dr. K. Sandhya Rani
Department of Computers, Sri Padmavathi Mahila Visvavidyalayam,
Tirupati, A.P., India

Abstract: *Knowledge Discovery in databases or data mining is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. In this paper, Associative Classification for mushroom dataset is considered. CMAR method is used for associative classification. The results are analyzed and compared.*

Keywords: *Data Mining, Association Rules, Classification, CMAR.*

1. INTRODUCTION

The growth in the size and number of existing databases far exceeds human abilities to analyze such data, thus creating both a need and an opportunity for extracting knowledge from databases. Recently, data mining has been ranked as one of the most promising research topics for the 1990s by both database and machine learning researchers. Researchers identify two fundamental goals of data mining: prediction and description. Prediction makes use of existing variables in the database in order to predict unknown or future values of interest, and description focuses on finding patterns describing the data and the subsequent presentation for user interpretation. The relative emphasis of both prediction and description differ with respect to the underlying application and the technique. There are several data mining techniques fulfilling these objectives. Some of these are associations, classifications, sequential patterns, clustering and visualization. Association rule mining is a form of data mining to discover interesting relationships among attributes in large databases. Classification involves finding rules that partition the data into disjoint groups. The input for the classification is the training data set, whose class labels are already known. Classification analyzes the training dataset and constructs a model based on the class label, and aims to assign a class label to the future unlabelled records. Given a set of data sequences, the problem of sequence discovery is to discover subsequences that are frequent, in the sense that the percentage of data sequences containing them exceeds a user-specified minimum support. Clustering is the process of grouping the data into clusters, so that objects within a cluster have high similarity in comparison to one another but are very dissimilar to objects in other clusters.

Association rule mining is one of the most important and well researched techniques of data mining. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. The prototypical example is determining what things go together in a shopping cart at the supermarket, the task at the heart of market basket analysis. Retail chains can use affinity grouping to plan the arrangement of items on store shelves or in a catalog so that items often purchased together will be seen together.

2. Significance of Classification

Classification, one of the most common data mining tasks, seems to be a human imperative. In order to understand and communicate about the world, we are constantly classifying, categorizing and grading. We divide living things into phyla, species, and general; matter into elements; dogs into breeds; people into races; steaks and maple syrup into USDA grades. There are handful techniques for classification [HK01]. Classification by decision tree was well researched and plenty of algorithms were already designed. A comprehensive survey on decision tree induction in [Mur98] and Bayesian classification in [DH73] were discussed. Bayesian classification is based on Bayes' theorem. Nearest neighbor classifiers were introduced in 1951 by Fix and Hodges [FH51]. A rule-based classifier uses a set of IF-THEN rules for classification. Rules can be extracted from a decision tree [Qui87, Qui93]. Rules may also be generated directly from training data using sequential covering algorithms and associative classification algorithms.

Associative classification uses association mining techniques that search for frequently occurring patterns in large databases. The patterns may generate rules, which can be analyzed for use in classification. Many algorithms have been proposed that adopt association rule mining to the task of classification. The CBA (Classification-Based Association) algorithm for associative classification was proposed by Liu, Hsu, and Ma [LHM98]. A new classification approach, CPAR (Classification Based on Predictive Association Rules), which combines the advantages of both associative classification and traditional rule-based classification was discussed in [YH03].

Classification of large datasets has been proposed to be handled in several ways. Sampling has been proposed in [Cat91]. Partitioning the data and creating classifiers for each was proposed in [CS93]. Linear regression for classification in [HTF01] and extracting rules from neural networks in [TS93, LSL95] were discussed.

Classification tasks include

- Classification of credit application as low, medium, or high risk
- Choosing content to be displayed on a web page
- Determining which phone numbers correspond to fax machines
- Spotting fraudulent insurance claims
- Assigning industry codes and job designation on the basis of free-text job descriptions

3. CMAR METHOD

An associative classification method, called CMAR performs Classification based on Multiple Association Rules. CMAR consists of two phases: rule generation and classification.

In the first phase, rule generation, CMAR computes the complete set of rules in the form of R : P → c, where P is a pattern in the training dataset, and c is a class label such that sup(R) and con(R) pass the given support and confidence thresholds, respectively. Furthermore, CMAR prunes some rules and only selects a subset of high quality rules for classification.

In the second phase, classification, for a given data object obj, CMAR extracts a subset of rules matching the object and predicts the class label of the object by analyzing this subset of rules.

3.1 Mining Class-Association Rules

To find rules for classification, CMAR first mines the training dataset to find the complete set of rules passing certain support and confidence thresholds. To make mining highly scalable and efficient, CMAR adopts a variant of FP-growth method [HPY00]. FP-growth is a frequent pattern mining algorithm which is faster than conventional Apriori-like methods, especially in the situations where there exist large datasets, low support threshold, and/or long patterns.

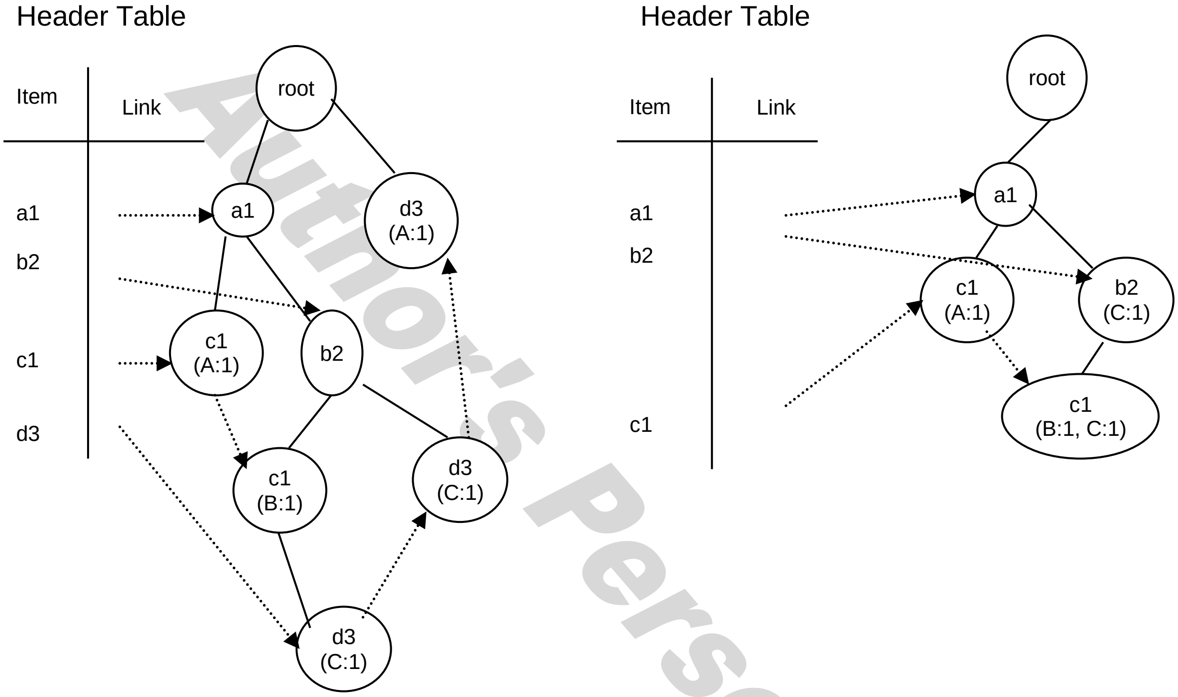
Consider the training dataset T as shown in Table 5.1. Let the support threshold is 2 and confidence threshold is 50%. CMAR mines class-association rules as follows:

Table 5.1 A training dataset

Row-id	**A**	**B**	**C**	**D**	**Class label**
1	a_1	b_1	c_1	d_1	A
2	a_1	b_2	c_1	d_2	B
3	a_2	b_3	c_2	d_3	A
4	a_1	b_2	c_3	d_3	C
5	a_1	b_2	c_1	d_3	C

First, CMAR scans the training dataset T once and finds the set of attribute values happening at least twice in T. The set is F= {a_1, b_2, c_1, d_3} and is called frequent item set. All other attribute values, which fail the support threshold, cannot play any role in the class-association rules, and thus can be pruned. Then, CMAR sorts attribute values in F in support descending order, i.e., F-list = a_1 - b_2 - c_1 - d_3. Then, CMAR scans the training dataset again to construct an FP-tree, as shown in figure 5.1(a). FP-tree is a prefix tree with respect to F-list. For each tuple in the training dataset, attributes values appearing in F-list are extracted and sorted according to F-list. For the first tuple, (a_1, c_1) are extracted and inserted in the tree as the left-most branch in the tree. The class label is attached to the last node in the path.

Tuples in the training dataset share prefixes. The second tuple carries attribute values (a_1, b_2, c_1) in F-list and shares a common prefix a_1 with the first tuple. So, it also shares the a_1 sub-path with the left-most branch. All nodes with same attribute value are linked together as a queue started from the header table.



(a) FP-tree (b) FP-tree after merging nodes of d3

Figure 5.1 FP-tree

Third, based on F-list, the set of class-association rules can be divided into 4 subsets without overlap: (a) the ones having d_3; (b) the ones having c_1 but no d_3; (c) the ones having b_2 but no d_3 nor c_1; and (d) the ones having only a_1. CMAR finds these subsets one by one.

Fourth, to find the subset of rules having d_3, CMAR traverses nodes having attribute value d_3 and look "upward" to collect a d_3-projected database, which contains three tuples: (a_1, b_2, c_1, d_3) : C, (a_1, b_2, d_3) : C and d_3 : A. It contains all the tuples having d_3. The problem of finding all frequent patterns having d_3 in the whole training set can be reduced to mine frequent patterns in d_3-projected database. Recursively, in d_3-projected database, a_1 and b_2 are the frequent attribute values, i.e., they pass support threshold. The projected database can be mined recursively by constructing FP-trees and projected databases.

After search for rules having d_3, all nodes of d_3 are merged into their parent nodes, respectively. That is, the class label information registered in a d_3 node is registered in its parent node. The FP-tree is shrunk as shown in figure 5.1(b). The remaining subsets of rules can be mined similarly.

3.2 Storing Rules in CR-tree

Once a rule is generated, it is stored in a CR-tree, which is prefix tree structure. The general idea of CR-tree is as follows:

Consider the table 5.2. It reveals four rules from mining training dataset.

Table 5.2 Rules found in a training dataset

Rule-id	Rule	Support	Confidence
1	abc → A	80	80%
2	abcd → A	63	90%
3	abe → B	36	60%
4	bcd → C	210	70%

A CR-tree is built for the set of rules, as shown in figure 5.2. A CR-tree has a root node. All 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.

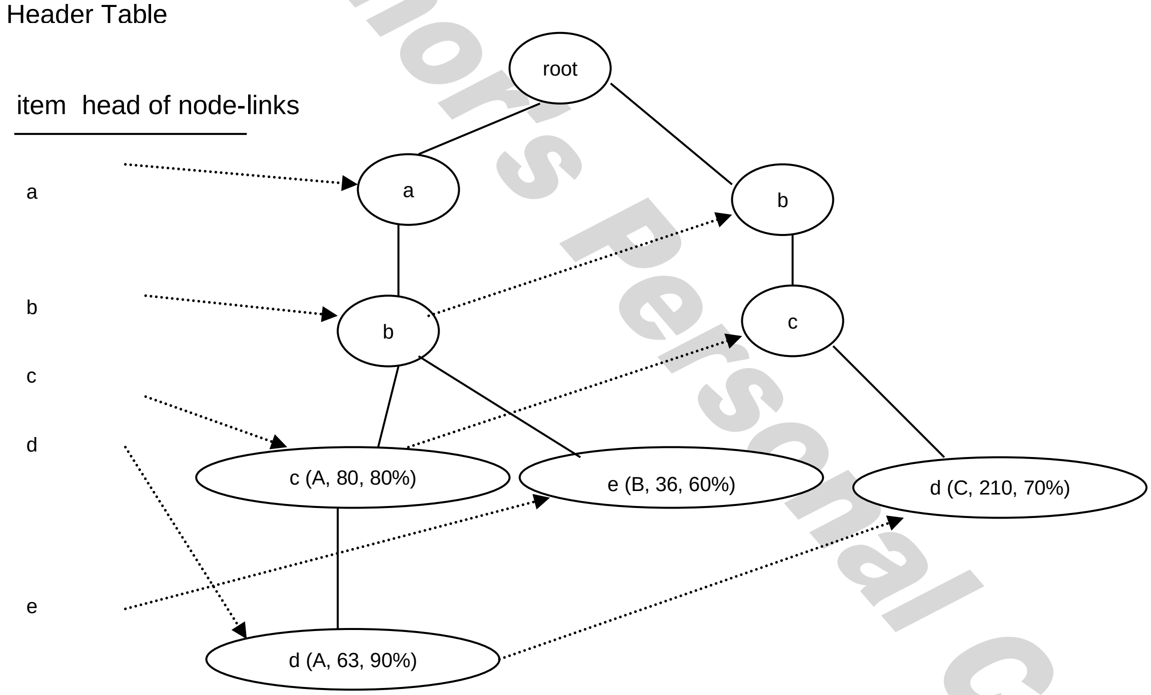


Figure 5.2 A CR-tree

To store the original rule set, 13 cells are needed for the left hand sides of the rules. Using CR-tree, only 9 nodes are needed. Once a CR-tree is built, rule retrieval becomes efficient. That facilitates the pruning of rules and using of rules for classification dramatically.

3.3 Pruning Rules

The number of rules generated by class-association rule mining can be huge. To make the classification effective and also efficient, the rules can be pruned to delete redundant and noisy information.

According to the facility of rules on classification, a global order of rules is composed. Given two rules R_1 and R_2, R_1 is said having higher rank than R_2, denoted as $R_1 > R_2$, if and only if (a) conf(R_1) > conf(R_2); (b) conf(R_1) = conf(R_2) but sup(R_1) > sup(R_2); or (c) conf(R_1) = conf(R_2), sup(R_1) = sup(R_2) but R_1 has fewer attribute values in its left hand side than R_2 does. In addition, a rule R_1: $P \rightarrow c$ is said a general rule in respect of rule R_2: $P' \rightarrow c'$, if and only if P is a subset of P'.

CMAR employs the following methods for rule pruning:

Firstly, the general and high-confidence rule is used to prune more specific and lower confidence ones. This pruning is pursued when the rule is inserted into the CR-tree.

Secondly, only positively correlated rules are selected. This pruning happens when a rule is found. Since the distribution of class labels with respect to frequent patterns is kept track during the rule mining, the χ^2 testing is done almost for free.

Thirdly, pruning of rules is based on database coverage. This pruning is pursued when the rule mining process finishes. It is the last pruning of rules. CMAR selects a subset of high quality rules for classification. This is achieved by pruning rules based on database coverage. CMAR uses a coverage threshold to select database coverage. CMAR uses the following procedure for selecting rules based on database coverage.

- Sorts rules in the rank descending order;
- For each data object in the training dataset, sets its cover-count to 0;
- While both the training dataset and rule set are not empty, for each rule R in rank descending order, finds all data objects matching rule R. If R can correctly classify at least one object then select R and increase the cover-count of those objects matching R by 1. A data object is removed if its cover-count passes coverage threshold δ;

3.4 Classification

After a set of rules is selected for classification, CMAR is ready to classify new objects. Given a new data object, CMAR collects the subset of rules matching the new object from the set of rules for classification. If all the rules matching the new object have the same class label, CMAR just simply assigns that label to the new object. If the rules are not consistent in class labels, CMAR divides the rules into groups according to the class labels. All rules in a group share the same class label and each group has a distinct class. CMAR compares the effects of the groups and yields to the strongest group. CMAR uses a weighted χ^2 measure to find the "strongest" group of rules, based on the statistical correlation of rues within a group. It then assigns the new object the class label of the strongest group. In this way it considers multiple rules, rather than a single rule with highest confidence, when predicting the class label of a new object.

4. Associative Classification Using CMAR

CMAR Method which is discussed in the previous section is implemented on the mushroom dataset. The dataset consists of 23 attributes and 8,124 records for mushrooms. During the data preparation phase of data mining, it is important to handle the missing values in the mushroom dataset. One of the preprocessing techniques, data cleaning is applied on the mushroom dataset before generation of association rules.

4.1 Generated Ruleset for Classification

By using CMAR method, a rule set is generated for classifying the mushroom data. Some of the rules in rule set are as follows:

Rules for poisonous state

Rule 1 for poisonous (1,584, 1.0)
if ring-number = one and spore-print-color = chocolate
then poisonous

Rule 2 for poisonous (1,584, 1.0)
if veil-color = white and spore-print-color = chocolate
then poisonous

Rule 3 for poisonous (1,584, 1.0)
if stalk-root = bulbous and spore-print-color = chocolate
then poisonous

Rule 4 for poisonous (1,584, 1.0)
if gill-size = broad and spore-print-color = chocolate
then poisonous

Rule 5 for poisonous (1,584, 1.0)
if gill-spacing = close and spore-print-color = chocolate
then poisonous

Rule 6 for poisonous (1,584, 1.0)
if gill-attachment = free and spore-print-color = chocolate then poisonous

Rule 7 for poisonous (1,584, 1.0)
if odor = foul
then poisonous

Rule 8 for poisonous (1,332, 1.0)
if bruises = no and stalk-surface-above-ring = silky
then poisonous

Rule 9 for poisonous (1,408, 0.989)

if bruises = no and gill-spacing = close and stalk-root = bulbous
then poisonous

Rule 10 for poisonous (1,296, 1.0)
if bruises = no and stalk-root = bulbous and ring-type = large
then poisonous

Rule 11 for poisonous (1,296, 1.0)
if bruises = no and stalk-root = bulbous and stalk-surface-below-ring = silky
then poisonous

Rules for edible state

Rule 1 for edible (2,648, 0.997)
if odor = none and veil-color = white and ring-number = one
then edible

Rule 2 for edible (2,496, 1.0)
if odor = none and stalk-shape = tapering
then edible

Rule 3 for edible (2,496, 1.0)
if odor = none and gill-size = broad and ring-number = one
then edible

Rule 4 for edible (2,496, 1.0)
if odor = none and gill-size = broad and stalk-shape = tapering
then edible

For poisonous state, rule set consists of 41 rules and for edible state has only 4 rules. The rules are self explanatory.

4.2 PERFORMANCE ANALYSIS OF RULE SET

In the performance analysis of classification rule set, the predictive accuracy of the rule set is estimated based on test set. The accuracy of a rule set on a given test set is the percentage of test set tuples that are correctly classified by the classifier (rule set). The associated class label of each test tuple is compared with the learned classifier’s class prediction for that tuple. If the accuracy of the classifier is considered acceptable, the classifier can be used to classify future data tuples for which the class label is not known. The generated rule set for classification is applied on the mushroom test set to estimate its accuracy

Confusion Matrix:

For classification problems, a confusion matrix is a very useful tool for understanding results. A confusion matrix shows the counts of the actual versus predicted class values. It shows not only how well the model predicts, but also presents the details needed to see exactly where things have gone wrong. The confusion matrix for the mushroom data is given in table 5.4. The rows show the actual classes, and the columns show the predicted classes. Therefore, the diagonal shows all the correct predictions. The confusion matrix reveals that 2640 instances are correctly classified as edibles where as 848 instances are misclassified as poisonous even though they are edibles. 2148 instances are correctly classified as poisonous where as 8 instances are misclassified as edibles though they are poisonous.

Table 5.4 Confusion matrix for mushroom data Prediction

Actual

Class	edible	poisonous
Edible	2640	848
Poisonous	8	2148

Table 5.5 Performance Analysis Table for the classifier (rule set)

Prediction	No. of Instances	Percentage
Correct	4788	84.83%
Wrong	856	15.17
Total	5644	

`From the table5.5, the predictive accuracy of the classification rule set is 84.83% and considered acceptable. Once the accuracy is tested and fully satisfied with the performance of rule set, it can be used to predict the class labels for new mushroom data.

For the given mushroom tuple,

X = (veil-type=partial, spore-print-color=chocolate, odor=foul, stalk-root=bulbous)
CMAR method assigns the class label as poisonous.

For the given mushroom tuple,

Y=(bruises=no, gill-spacing=close, stalk-root=bulbous, cap-shape=bell, cap-surface=scaly)

CMAR method assigns the class label as poisonous.

For the given mushroom tuple,

Z=(gill-attachment=free, spore-print-color=chocolate, population=solitary, cap-color=green)

CMAR method assigns the class label as poisonous.

For the given mushroom tuple,

U = (odor=none, veil-color=white, ring number =one, bruises=yes)

CMAR method assigns the class label as edible.

For the given mushroom tuple,

V = (odor=none, stalk-shape=tapering, habitat=woods)

CMAR method assigns the class label as edible.

For any given tuple, by using the classification rule set one can find the state of the mushroom whether edible or poisonous.

5. CONCLUSIONS

In this paper, Integrating classification and association rule mining is discussed. Classification is the process of finding a model that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model may be represented in various forms such as classification rules, decision trees etc., Generated association rules should not be used directly for prediction without further analysis or domain knowledge. This is due to the many different possible conclusions for the rules. For prediction, the association rules can be transformed into a classification ruleset. The generated classification ruleset for the mushroom dataset is used to predict the edibility or poisonous of a mushroom.

REFERENCES:

[Cat91] J. Catlett, "Megainduction: Machine Learning on Very Large Databases", Ph.D. Thesis, University of Sydney, 1991.

[CS93] P. K. Chan and S. J. Stolfo, "Experiments on Multistrategy Learning by Metalearning", Proc. 2nd International Conference on Information and Knowledge Management, pages 314-323, 1993.

[DH73] R. Duda and Hart, "Pattern Classification and Scene Analysis", Wiley & Sons, Inc. 1973.

[HK01] J. Han and M. Kamber, "Data Mining Concepts and Techniques", Elsevier 2001.

[HPY00] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation", Proc. 2000 ACM-SIGMOD International Conference on Management of Data (SIGMOD'00), pages 1-12, Dallas, TX, May 2000.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer-Verlag, 2001.

[FH51] E. Fix and J. L. Hodges Jr, "Discriminatory Analysis, Non-Parametric Discrimination: Consistency Properties", Technical Report 21-49-004(4), USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[LHM98] B. Liu, W. Hsu, and Y. Ma, "Integrating Classification and Association Rule Mining", Proc. 1998 International Conference on Knowledge Discovery and Data Mining (KDD'98), pages 80-86, New York, NY, August 1998.

[Mur98] S. K. Murthy, "Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey", Data Mining and Knowledge Discovery, 2:345-389, 1998.

[Qui87] J. R. Quinlan, "Simplifying Decision Trees", Int. J. Man-Machine Studies, 27:221-234, 1987.

[Qui93] J. R. Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann, 1993.

[LHM98] B. Liu, W. Hsu, and Y. Ma, "Integrating Classification and Association Rule Mining", Proc. 1998 International Conference on Knowledge Discovery and Data Mining (KDD'98), pages 80-86, New York, NY, August 1998.

[LSL95] H. Lu, R. Setiono, and H. Liu, "Neurorule: A Connectionist Approach to Data Mining", Proc. 1995 International Conference on Very Large Data Bases (VLDB'95), pages 478-489, Zurich, Switzerland, September 1995.

[TS93] G. G. Towell and J. W. Shavlik, "Extracting Refined Rules from Knowledge-Based Neural Networks", Machine Learning, 13:71-101, October 1993.