October 2011

Performance Comparison of Rule Based Classification Algorithms

Prafulla Gupta

Deptt. of Electronics and Computer, Indian Institute of Technology, Roorkee, India,
prafullatech@gmail.com

Durga Toshniwal

Deptt. of Electronics and Computer, Indian Institute of Technology, Roorkee, India, durgafec@iitr.ernet.in

Follow this and additional works at: <https://www.interscience.in/ijcsi>

Part of the [Computer Engineering Commons](#), [Information Security Commons](#), and the [Systems and Communications Commons](#)

Recommended Citation

Gupta, Prafulla and Toshniwal, Durga (2011) "Performance Comparison of Rule Based Classification Algorithms," *International Journal of Computer Science and Informatics*: Vol. 1 : Iss. 2 , Article 12.

DOI: 10.47893/IJCSI.2011.1025

Available at: <https://www.interscience.in/ijcsi/vol1/iss2/12>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer Science and Informatics by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

Performance Comparison of Rule Based Classification Algorithms

Prafulla Gupta & Durga Toshniwal

Deptt. of Electronics and Computer, Indian Institute of Technology, Roorkee, India

E-mail: prafullatech@gmail.com, durgafec@iitr.ernet.in

Abstract - Classification based on predictive association rules (CPAR) is a kind of association classification methods which combines the advantages of both associative classification and traditional rule-based classification. For rule generation, CPAR is more efficient than traditional rule-based classification because much repeated calculation is avoided and multiple literals can be selected to generate multiple rules simultaneously. CPAR inherits the basic ideas of FOIL (First Order Inductive Learner) algorithm and PRM (Predictive Rule Mining) algorithm in rule generation. It integrates the features of associative classification in predictive rule analysis. In comparison of FOIL, PRM algorithm usually generates more rules. PRM uses concept of lowering weights rather than removing tuple if tuple is satisfied by the rule. The distinction between CPAR and PRM is that instead of choosing only the attribute that displays the best gain on each iteration CPAR may choose a number of attributes if those attributes have gain close to best gain.

Keywords - Data mining, Classification, Association rules, classification using predictive association rule(CPAR), Predictive Rule Mining(PRM), First Order Inductive Learner(FOIL).

I. INTRODUCTION

Classification is a technique of data mining in which a model or classifier is constructed to predict the class or categorical labels of given data. Data classification is two step process. In first step classifier is created by analyzing training data. This step is known as learning step of classification. Then in second step using classifier made in first step, prediction of given data tuples are done.

In classification using association rule (CPAR), learning step uses predictive association rules to build a classifier. Association rules which can be used to predict a class are known as predictive association rules. In predictive rule ($A \rightarrow B$), 'B' can only be class label. 'A' is frequent pattern which is showing behavior or interesting relationship in data. To build predictive association rules different techniques are adopted such as FOIL(First Order Inductive Learner), PRM(Predictive Rule Mining) etc. CPAR takes the basic idea of FOIL and PRM in rule generation step but CPAR is more efficient than FOIL or PRM in terms of calculation and classification accuracy. In CPAR repeated calculation is avoided by selecting multiple literals to generate multiple rules simultaneously. After making classifiers using predictive rules, this classifier is applied on real data to predict the class of that.

II. ALGORITHMS DESCRIPTION

The main objective of this paper is to implement CPAR, PRM and FOIL algorithms and to judge their relative performances. FOIL is basic algorithm which is

then improved by PRM and this PRM algorithm is more improved as CPAR. Basic difference in these strategies is in rule generation process. FOIL generates rules which are not redundant but to achieve this, it loses some important rules. So PRM extracted these rules also but with cost of some redundancy. Some rule may be extracted more than once. To remove these rules pruning of rule will also be done in PRM. CPAR also uses similar concept of PRM as to generate more rule with some redundant rules, but it can test more than one attribute at a time to judge whether this attribute can also give some useful rule or not. So more rules and less computation is needed in CPAR in comparison to the PRM algorithm.

To implement these algorithms, whole work can be divided in three phases

1. Generation of rule.
2. Estimate Accuracy of rules.
3. Classification and Result analysis.

1. Generation of rule

Generation of rule is different in all three algorithms. Rest two modules are same for each algorithm. Results are shown as confusion matrix of class labels.

Let T be a set of tuples. Each tuple t in T follows the scheme $(A_1, A_2, \dots, A_k)$, where $A_1, A_2, \dots, A_k$ are k attributes. Each continuous attribute is first transformed into a categorical attribute. Two definitions[2] of rules and literals is given as following.

Literal: A literal p is an attribute-value pair, taking the form of (A_i, v) , in which A_i is an attribute and v a

value. A tuple t satisfies a literal $p = (A_i; v)$ if and only if $t_i = v$, where t_i is the value of the i th attribute of t .

Rule: A rule r , which takes the form of $p_1 \wedge p_2 \wedge \dots \wedge p_n \rightarrow c$, consists of a conjunction of literals $p_1, p_2, \dots, p_n$, associated with a class label c . A tuple t satisfies rule r 's body if and only if it satisfies every literal in the rule. If t satisfies r 's body, r predicts that t is of class c . If a rule contains zero literal, its body is satisfied by any tuple.

In CPAR, PRM and FOIL algorithm not all association rules are generated rather only specific rules related to classification are extracted with certain parameters of accuracy and convergence.

Rule generation algorithm is as firstly a general rule is evaluated and then to make its accuracy better some specific rule is searched with this rule. This approach is also called General-to-specific rule search approach[1].

A. FOIL (First Order Inductive Learner) Rule Generation algorithm

FOIL (First Order Inductive Learner), proposed by Ross Quinlan in 1993 [3], is a greedy algorithm that learns rules to distinguish positive examples from negative ones. FOIL repeatedly searches for the current best rule and removes all the positive examples covered by the rule until all the positive examples in the data set are covered. The algorithm FOIL is presented below. For multi-class problems, FOIL is applied on each class: for each class, its examples are used as positive examples and those of other classes as negative ones. The rules for all classes are merged together to form the result rule set.

When selecting literals, Foil Gain is used to measure the information gained from adding this literal to the current rule. Suppose there are $|P|$ positive examples and $|N|$ negative examples satisfying the current rule r 's body. After literal p is added to r , there are $|P'|$ positive and $|N'|$ negative examples satisfying the new rule's body. Then the Foil gain of p is defined as,

$$\text{Gain}(p) = |P'| * [\log\{|P'| / (|P'| + |N'|)\}] - \log\{|P| / (|P| + |N|)\}] \quad (1)$$

FOIL Algorithm

- Input: Training set $D = (P \text{ Union } N)$. (P and N are the sets of all positive and negative examples, respectively.)
- Output: A set of rules for predicting class labels for examples.
- Procedure: FOIL

```
rule set R <- ∅
While |P| > 0
N' <- N, P' <- P
Rule r <- empty rule
While |N'| > 0 and r.length < max rule length
```

```
find the literal p that brings most gain
according to |P'| and |N'|
append p to r
remove from P' all examples not satisfying r
remove from N' all examples not satisfying r
end
R <- R Union {r}
remove from P all examples satisfying r's body
end
return R
```

B. PRM (Predictive Rule Mining) algorithm

Predictive Rule Mining (PRM), an algorithm which modifies FOIL to achieve higher accuracy and efficiency [3]. One reason that FOIL does not achieve as high accuracy is that it generates a very small number of rules. In PRM, after an example is correctly covered by a rule, instead of removing it, its weight is decreased by multiplying a factor. This weighted version of FOIL produces more rules and each positive example is usually covered more than once. The most time consuming part of FOIL is evaluating every literal when searching for the one with the highest gain.

Algorithm: Predictive Rule Mining (PRM)

- Input: Training set $D = (P \text{ Union } N)$. (P and N are the sets of all positive and negative examples, respectively.)
- Output: A set of rules for predicting class labels for examples.
- Procedure: Predictive Rule Mining

```
Set the weight of every example to 1
rule set R <- ∅
Total Weight <- TotalWeight(P)
While TotalWeight(P) > λ(totalWeight)
N' <- N, P' <- P
Rule r <- empty rule
While true
Find best literal p
Append p to r
For each example t in P', N' not satisfying r's body
remove t from P' or N'
end
R <- R Union {r}
for each example t in P satisfying r's body
t.weight <- α(t.weight)
end
Return R
```

C. CPAR (Classification Based on predictive association rule) rule generation algorithm

When selecting literals during the rule building process, PRM selects only the best literal and ignores all the others. There are often a few literals with a similar gain. Thus there are usually many rules with similar accuracy based on the remaining dataset. The best rule among them may not be the best rule based on the whole dataset. However, PRM selects only one of them, which may lead to missing some important rules. CPAR stands in the middle between exhaustive and greedy algorithms and combines the advantages of both[5]. CPAR builds rules by adding literals one by one, which is similar to PRM. However, instead of ignoring all literals except the best one, CPAR keeps all close-to-the-best literals during the rule building process. By doing so, CPAR can select more than one literal at the same time and build several rules simultaneously. For detailed description of the rule generation algorithm of CPAR[3], Suppose at a certain step in the process of building a rule, after finding the best literal p , another literal q that has similar gain as p (e.g., differ by at most 1%) is found. Besides continuing building the rule by appending p to r , q is also appended to the current rule r to create a new rule r_0 , which is pushed into the queue. Each time when a new rule is to be built, the queue is first checked. If it is not empty, a rule is extracted from it and is taken as the current rule. This forms the depth-first-search in rule generation. Fig 1 shows an example of how CPAR generates rules. After the first literal ($A1 = 2$) is selected, two literals ($A2 = 1$) and ($A3 = 1$) are found to have similar gain, which is higher than other literals. Literal ($A2 = 1$) is first selected and a rule is generated along this direction. After that, the rule ($A1 = 2, A3 = 1$) is taken as the current rule. Again two literals with similar gain ($A4 = 2$) and ($A2 = 1$) are selected and a rule is generated along each of the two directions.

Figure 1. Rule generation example in CPAR

2. Rules Evaluation

Before making any prediction, every rule needs to be evaluated to determine its prediction power. For a rule $r = p_1 \wedge p_2 \wedge \dots \wedge p_n \rightarrow c$, we define its expected accuracy as the probability that an example satisfying r 's body belongs to class c . Laplace expected error estimate

is used to estimate the accuracy of rules, which is defined as follows.

The expected accuracy of a rule is given by [5]:

$$\text{Laplace Accuracy} = (n_c + 1) / (n_{\text{tot}} + k) \quad (2)$$

Where k is the number of classes, n_{tot} is the total number of examples satisfying the rule's body, among which n_c examples belong to c , the predicted class of the rule.

3. Classification

Given a rule set containing rules for each class, we use the best k rules of each class for prediction, with the following procedure: (1) select all the rules whose bodies are satisfied by the example; (2) from the rules selected in step (1), select the best k rules for each class; and (3) compare the average expected accuracy of the best k rules of each class and choose the class with the highest expected accuracy as the predicted class.

III. DATASET DESCRIPTION

Data Set which is taken to perform classification algorithm is Car Evaluation Dataset. It is taken from UCI-KDD Machine learning repository[4]. This dataset contains 6 normal attributes and 1 class attribute.

Normal attributes are features of car, which are having values as

buying	v-high, high, med, low
maintenance	v-high, high, med, low
doors	2, 3, 4, 5-more
persons	2, 4, more
lug_boot	small, med, big
safety	low, med, high

Cars are evaluated using these attributes. Classes for evaluation are unacc (unacceptable), acc (acceptable), good, vgood (very good) .

A. Statistics of Dataset

- Number of Attribute: 6 normal attributes
1 Class attribute
3 extra attribute to be used by algorithms.
- Number of classes: 4
- Class labels: unacc, acc, good, vgood
- Number of Tuple (For Training):1728
- Number of Tuple (For Testing): 1525
- Class-wise distribution of data (For Training Set)

TABLE I. CLASS-WISE DISTRIBUTION IN TRAINING SET

CLASS	OCCURANCE
Unacc	1210
Acc	384
Good	69
VGood	65

- Class-wise distribution of data(For Test Set)

TABLE II.CLASS-WISE DISTRIBUTION IN TEST SET

CLASS	OCCURANCE
Unacc	1108
Acc	309
Good	63
VGood	45

IV. SYSTEM DESIGN

Each algorithm is implemented in following five modules

- I. Rule extraction
- II. Pruning of rules
- III. Accuracy estimation
- IV. Classification
- V. Analysis

Rule extraction part in each algorithm is different and rests of four modules are same for all algorithms.

A. Rule Extraction Module

In this module, training data is taken as input in form of ms-access database and as output rules are generated and saved in database. Description for each algorithm is as follow :

- In FOIL, firstly class labels are given to function one by one ,which generates rules for each class. This function selects best literal according to gain index for making rule of given class label, then this literal is judged to find whether this literal is sufficient for rules antecedent or not. If literal has no negative tuple then it is taken as rule antecedent and all tuples satisfying this rule is checked as „no“, which shows that rules according to these tuples are made. Otherwise one more literal is chose so that both literal can make rule antecedent. This

process is repeated until some threshold gain is not achieved by best literal chosen. Rules which are generated are saved in separate database for further processing.

- In PRM , initial steps are same as FOIL, but in place of checking tuples „no“, weight corresponding to these tuples are reduced by factor α . No gain threshold is set in this algorithm, so for each class this module runs till total positive weight of class is not less then total_weight_threshold[6].
- In CPAR, more than one literal is selected in single pass for making rule antecedent. These literals are chosen with help of foil gain index. Literal with best gain and literals with gain more than θ times maximum gain are selected to judge whether these literals are sufficient alone for rule antecedent or not. Remaining same step is performed as in FOIL and PRM.

B. Pruning of Rules

In this module, duplicate rules are deleted from database. As PRM and CPAR can generate duplicate rule so this module is important for them. For FOIL this module is of no use.

C. Accuracy Measurement

In this module, Laplace accuracy of each rule is calculated and saved in database. Formula for calculating accuracy is given in section 2.4.

D. Classification

In classification module, best K rules are selected from each class according to accuracy measure. Then these rules are applied for each test data record. Rules which are fired for certain record are selected and average accuracy of those rules is calculated according to each class. Class with maximum average accuracy is selected as classified rule.

E. Analysis

In this module, confusion matrix is calculated using classification results.

V. RESULT ANALYSIS

Classification results are shown as a confusion matrix. True positive(TP), false positive (FP), precision, recall are also calculated for each class[7].Overall accuracy is also computed.

In characteristics table TP represents True Positive, FP represents False Positive, TPR represents True Positive Rate, FPR stands for False Positive Rate, RCL

stands for Recall PRS stands for Precision, FMR stands for F-measure.

A. FOIL algorithm:

- Number of rules generated: 31
- Confusion Matrix

TABLE III. CONFUSION MATRIX BY FOIL ALGORITHM

Classified	Unacc	Acc	Good	VGood	Blank
Unacc	1101	0	0	0	7
Acc	6	32	20	20	231
Good	0	4	50	7	2
VGood	0	0	0	43	2

- Classification Characteristics

TABLE IV. CHARACTERISTICS OF CLASSIFICATION BY FOIL ALGORITHM

Class	TP	FP	TPR	FPR	RCL	PRS	FMR
Unacc	1101	6	0.993	0.014	0.993	0.994	0.993
Acc	32	4	0.103	0.003	0.103	0.888	0.184
Good	50	20	0.793	0.013	0.793	0.714	0.751
VGood	43	27	0.955	0.018	0.955	0.614	0.747

- Overall Accuracy of algorithm=80.39%

B. PRM algorithm

- Number of rules generated: 128
- Confusion Matrix

TABLE V. CONFUSION MATRIX BY PRM ALGORITHM

Classified	Unacc	Acc	Good	VGood	Blank
Unacc	1096	1	2	0	9
Acc	2	175	8	10	114
Good	0	3	54	6	0
VGood	0	0	2	43	0

- Classification Characteristics

TABLE VI. CHARACTERISTICS OF CLASSIFICATION BY PRM ALGORITHM

Class	TP	FP	TPR	FPR	RCL	PRS	FMR
Unacc	1096	2	0.989	0.004	0.989	0.998	0.993
Acc	175	4	0.566	0.003	0.566	0.977	0.716
Good	54	12	0.857	0.008	0.857	0.818	0.837
VGood	43	16	0.955	0.010	0.955	0.621	0.752

- Overall Accuracy of algorithm=89.70%

C. CPAR algorithm

- Number of rules generated: 132
- Confusion Matrix

TABLE VII. CONFUSION MATRIX BY CPAR ALGORITHM

Classified	Unacc	Acc	Good	VGood	Blank
Unacc	1083	0	2	0	23
Acc	0	200	15	6	88
Good	0	0	60	3	0
VGood	0	0	4	41	0

- Classification Characteristics

TABLE VIII. CHARACTERISTICS OF CLASSIFICATION BY CPAR ALGORITHM

Class	TP	FP	TPR	FPR	RCL	PRS	FMR
Unacc	1083	0	0.977	0.000	0.977	1.000	0.988
Acc	200	0	0.647	0.000	0.647	1.000	0.785
Good	60	21	0.952	0.014	0.952	0.740	0.832
VGood	41	9	0.911	0.006	0.911	0.820	0.863

- Overall Accuracy of algorithm=90.75%

Above Results Show that CPAR is better Algorithm in terms of rule generation and classification Accuracy.

VI. CONCLUSION AND FUTURE WORK

In this paper, Classification is performed with use of Predictive association rule mining, which is widely used in data mining processes. CPAR (classification based on predictive association rules) is extended version of PRM(predictive rule mining) algorithm which is improvement of FOIL (first order inductive learner) algorithm. Whereas FOIL deletes those records immediately which are satisfied by current rule, CPAR and PRM retains them to explore more features within them. These records are used with lesser weights in further processing. CPAR and PRM both extracts duplicate rules from data, so additional pruning step is added in it to delete duplicate records. Number of rules extracted by CPAR and PRM are greater than FOIL, so better classification is achieved by them. CPAR is more efficient than PRM because much repeated calculation is avoided and multiple literals can be selected to generate multiple rules simultaneously.

The future work includes making much efficient rules so that all the records can be classified in one of the classes. Some specific pre-processing also required to improve classification accuracy. There is also need of post processing step to classify those records which not satisfied by any rules.

REFERENCES

- [1] J. Han and M. Kamber, Data mining: Concepts and Techniques, 2nd ed., Morgan Kaufmann, 2009.

- [2] X.Yin and J.Han, "CPAR: Classification Based on Predictive Association Rules", Proceedings of SIAM International conference on Data Mining, 2003, pp.331-335.
- [3] J. R.Quinlan and R.M.Cameron-Jones, "FOIL: A midterm report", Proceedings of European Conference of Machine Learning, 1993, pp. 3-20.
- [4] "UCI machine learning repository-car evaluation dataset" , Available Online: "[http:// archive. ics. uci.edu/ml/datasets/Car+Evaluation](http://archive.ics.uci.edu/ml/datasets/Car+Evaluation)".
- [5] "Classification based on predictive association rule" , Available Online: "[http:// www .csc. liv. ac.uk/~frans/KDD/Software/FOIL_PRM_CPAR/c par.html](http://www.csc.liv.ac.uk/~frans/KDD/Software/FOIL_PRM_CPAR/cpar.html)".
- [6] Z.Hao, X.Wang, L.Yao, "Improved Classification Based on Predictive Association Rules", Proceedings of IEEE International Conference on System, Man, Cybermatics, 2009, pp.1165-1170.
- [7] "Precision and Recall" ,Available Online: http://en.wikipedia.org/wiki/Precision_and_recall

□□□