Nehru E- Journal
A Journal of Nehru Arts and Science College (Autonomous)
Journal Home Page: http://nehrucolleges.net/jo.html
ISSN: 2349-9052

Research Article
Vol 5 issue 1(2017)
(Jan - July 2017)

Performance Analysis of Ripper Algorithm using Weka Tool

L.Gnanaprasanambikai
Assistant Professor, Department of Information Technology/Computer Technology
Nehru Arts and Science College, Coimbatore-641105, Tamil Nadu

Keywords:

Classification,
Weka,
Ripper, UCI
Datasets

Abstract

Data Mining is understandable generation of models and patterns from a database. Classification is one of the tasks of Datamining. Various Classification Algorithms derive different models of data classification. Rule Generation is one derivation model. Number of Algorithms is there to derive Rules. Ripper Algorithm is commonly used for two Class Problem. Ripper Algorithm suitability for generating rules is analyzed with Weka 3.6 tool with two UCI Repositary Datasets.

INTRODUCTION

1. Introduction

Classification is one of the Data mining task. The purpose of Classification is to classify instances of dataset into different classes based on some constraints[1]. The aim of classification task is to derive a model that distinguishes data classes. The Derived Model is based on the data analysis of training dataset. The derived model can be presented in number of ways like IF-then rules, decision trees, and neural networks. There are several classification algorithms used for classification. In this paper, analyses the Performance of Ripper Algorithm in WeKa3.6 using two UCI Machine learning Repositary datasets.

2. Weka 3.6

Weka is a data mining tool developed by University of Waikato in New Zealand that implements data mining algorithms. It is a collection of algorithms of different tasks of data mining. The algorithms use dataset for data preprocessing, classification, clustering, and association rules. In this paper, the Weka is used for task of classification [2]. It is a open source software (free). The algorithms can be used by the datasets directly or called from our own java code. Weka originally written in C and rewritten completely in Java and is compatible with almost all computing platforms. It is user friendly software with GUI platform for easy quick set up or installation. It allows new user to identify hidden information from database and file systems with simple to use options and visual interfaces [3].

3. RIPPER Algorithm

RIPPER is expanded as Repeated Incremental Pruning to Produce Error Reduction. RIPPER Algorithm is commonly used for two class problem. The Algorithm was first designed by Cohen in 1995. RIPPER is especially more efficient on large noisy datasets[4] .There are two kinds of loop namely Inner Loop and Outer Loop in the algorithm. It is easy to understand, usually better than Decision trees, as it generates, directly if-then rules. It is representable in first order logic.

In two class problem, it chooses one class as positive and the other class as negative class. It learns rules for positive class and make the other class as the negative class. The positive class is a class with smaller number of instances in the dataset. The negative class is a class with larger number of instances in the dataset [5].

RIPPER Algorithm is sequential covering Algorithm that generates the rules with the current set of positive class instances and grows up with remaining positive class instances.

In this paper, the rule generation using RIPPER Algorithm is analyzed by two UCI Machine Learning Dataset Repository namely Banknote authentication dataset and Blood Transfusion Service Center Dataset. WEKA 3.6 Data mining tool for our task. Ripper Algorithm is called as Jrip Algorithm in Weka.

RIPPER Algorithm

```
Step1 : Start from an empty rule: {} => class

Step 2 : Add conjuncts that maximizes FOIL's information gain measure:

    ➢ R0: {} => class (initial rule)

    ➢ R1: {A} => class (rule after adding conjunct)

    ➢ Gain(R0, R1) = t [ log (p1/(p1+n1)) – log (p0/(p0 + n0)) ]

    ➢ where t: number of positive instances covered by both R0 and R1

        p0: number of positive instances covered by R0
        n0: number of negative instances covered by R0
        p1: number of positive instances covered by R1
        n1: number of negative instances covered by R1
```

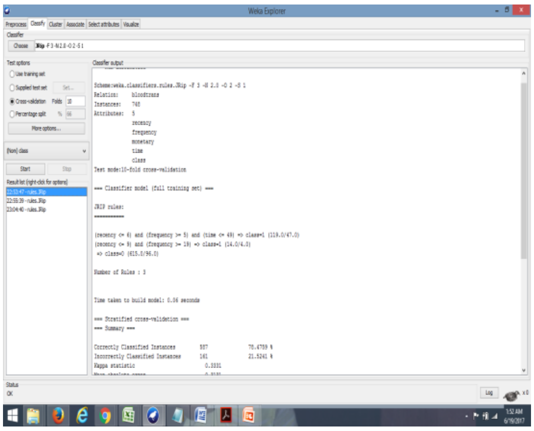
Figure 1: Ripper Algorithm

4. Experiments results with Datasets

4.1 Blood Transfusion Service Center Data Set (BISCD):

The data in the Dataset is taken from the Blood Transfusion Service Center in Taiwan. It is an example for classification Problem. The dataset consists of four attributes, namely Recency (months since Last Donation), Frequency (Total number of Donation), Monetary (Total Blood donated), and Time(Months since first Donation). The dataset consists of 748 instances with two classes with a binary variable value 1 stand for donating blood and 0 stands for not donating blood [6].

In the dataset with 748 instances, 178 instances for class having value 1 and 571 instances for class having value 0. In Figure 2, we could find 2 rules generated for positive class and keeping negative class as default.



Choose JRip -F 3 -N 2.0 -O 2 -S 1

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

22:53:47 - rules.JRip
22:55:39 - rules.JRip
23:04:40 - rules.JRip

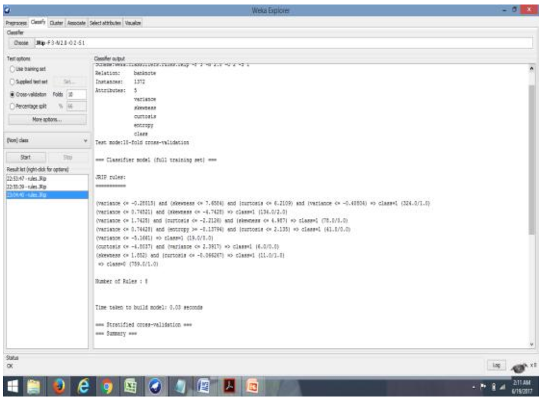
```
Scheme:weka.classifiers.rules.JRip -F 3 -N 2.0 -O 2 -S 1
Relation:     bloodtrans
Instances:    748
Attributes:   5
              recency
              frequency
              monetary
              time
              class
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

JRIP rules:
===========

(recency <= 6) and (frequency >= 5) and (time <= 49) => class=1 (119.0/47.0)
(recency <= 9) and (frequency >= 19) => class=1 (14.0/4.0)
 => class=0 (615.0/96.0)

Number of Rules : 3


Time taken to build model: 0.06 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         587              78.4759 %
Incorrectly Classified Instances       161              21.5241 %
Kappa statistic                          0.3331
```

Figure 2. Implementation of BTSC Dataset in WEKA 3.6 Tool.

4.2 Banknote Authentication Dataset (BAD):

The data in the dataset are extracted from images for authentication procedure for banking notes. Wavelet Transformation are used to extract features from the images. There are four attributes namely variance, skewness, curtosis, entropy. The dataset consists of 1372 instances with two classes having binary value of 1 and 0 [7].

In 1372 instances, 610 instances are for a class having value 1 and 762 instances all for class having value 0. In Figure 3, It is find that seven rules are generated for positive class and keeping negative class as default.



JRip -F 3 -N 2.0 -O 2 -S 1

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

```
Scheme:weka.classifiers.rules.JRip -F 3 -N 2.0 -O 2 -S 1
Relation:     banknote
Instances:    1372
Attributes:   5
              variance
              skewness
              curtosis
              entropy
              class
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

JRIP rules:
===========

(variance <= -0.28015) and (skewness <= 7.6584) and (curtosis <= 6.2109) and (variance <= -0.40804) => class=1 (324.0/1.0)
(variance <= 0.74521) and (skewness <= -4.7428) => class=1 (136.0/2.0)
(variance <= 1.7425) and (curtosis <= -2.2126) and (skewness <= 4.987) => class=1 (78.0/3.0)
(variance <= 0.74428) and (entropy >= -0.13794) and (curtosis <= 2.135) => class=1 (41.0/0.0)
(variance <= -5.1661) => class=1 (19.0/0.0)
(curtosis <= -4.8037) and (variance <= 2.3917) => class=1 (6.0/0.0)
(skewness <= 1.852) and (curtosis <= -0.066267) => class=1 (11.0/1.0)
 => class=0 (759.0/1.0)

Number of Rules : 8


Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===
```

Figure 3. Weka 3.6 Implementation of Banknote Authentication Dataset in WEKA 3.6

5. Discussion

Based on the experimental results, it is observed that the rules are generated for positive class (smaller Set) for any number of instances. The range of instances can be smaller dataset to larger dataset.

6. Conclusion

Ripper algorithm is easy to understand and classification rules can be easily generated only for positive class. The algorithm is unsuitable for generating rules for negative class at any situation.

References:

1. Sagar . S. Nikam , A Comparative study of classification Techniques in Data mining Algorithms, Oriental Journal of Computer Science and Technology, ISSN : 0974-6471 Online ISSN : 2320-8481
2. Svetlana S. Aksenova, Machine Learning with WEKA, http://csed.sggs.ac.in/csed/sites/default/files/WEKA%20Explorer%20Tutorial.pdf
3. http://www.gtbit.org/downloads/dwdmsem6/dwdmsem6lman.pdf
4. S.Vijayarani and M.Divya, "An efficient Algorithm for generating classification rules" , International Journal of Computer Science and Technology, ISSN:0976-8491(Online)|ISSN:2229-4333(Print), Vol 2, Issue, Oct-Dec.2011.
5. https://www.users.cs.umn.edu/~kumar/dmbook/dmslides/chap5_alternative_classification.pdf
6. https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center
7. https://archive.ics.uci.edu/ml/datasets/banknote+authentication