BCSIR

Available online at www.banglajol.info

Bangladesh J. Sci. Ind. Res. **44(2)**,171-180, 2009

BANGLADESH JOURNAL OF SCIENTIFIC AND INDUSTRIAL RESEARCH

E-mail: bjsir07@gmail.com

One-RM : An Improved One -Rule Classifier

Abdur Mahmood[a] and Wei Lei[b]

[a]University of Dhaka, Dhaka-1000, Bangladesh and [b]University of Melbourne, Australia

Abstract

One-R algorithm is a simple algorithm which exhibits quite good predictive accuracy for a large class of data. When compared to the more complex algorithms having better predictive accuracy, One-R provides the baseline accuracy for testing new machine learning algorithms. However, the simplicity of One-R means that it has there is a compromise between accuracy and complexity. Often, the accuracy of One-R can be further increased without making it significantly complex. The resulting algorithm as proposed in this paper, One-RM performs equal to One-R in most of the cases and sometimes outperforms One-R by significant margin. Theoretical analysis suggests that One-RM used in conjunction with One-R always performs either better or equal to One-R. Experimental analysis shows that One-RM is a viable alternative to One-R when used as a separate classification rule.

Key words: One-RM, One-R algorithm, Algorithm, Accuracy and Complexity.

1. Introduction

One of the main directions in machine learning research is the improvement of classification accuracy (Dietterich, 1997). To improve the classification accuracy many complex algorithms have been devised. But many of these algorithms do not offer as much accuracy as they are complex. Interestingly, in a seminal paper titled “Very Simple classification rules perform well on most commonly used datasets” (Holte, 1993). It was shown that in most datasets a very good and acceptable accuracy can be obtained by using very simple classification techniques. Such a technique known as One-R potentially increases. (Holmes and Nevill-Manning, 1995 and Nevill-Manning *et al*, 1995) the classification accuracy of an algorithm by performing feature subset selection on a dataset.

Although One-R is not the best classifier available, it is generally used, along with other methods such as, Fisher’s measure of “attribute dependence” (Fisher, 1987; Fisher and Schlimmer, 1988) as predictors of accuracy. So, One-R is widely used as a benchmark system for ranking or evaluating machine learning systems on popular datasets.

Researchers often favor One-R over simpler and naive benchmark systems such as “baseline accuracy”, which reflects the percentage of examples of the most frequent class of a dataset.

In this paper, we propose a new l-rule known as One-RM, which takes into account prediction class based average error attribute values to determine the best attribute for classification.

In Section 2, we discuss about the One-R algorithm, and its merits when compared to other more complex but accurate classifier rules. We also point to the inherent difficulties of One-R, its un-deterministic approach to attribute values having the same least average error and in case of attributes whose values have equal number of predicted class examples.

In Section 3, we present our algorithm One-RM, and discuss its complexity. We address the two problems highlighted in Section 2. One-RM chooses a different method to choose the best attribute and breaks the tie in favor of the attribute chosen by One-R. A variation of One-RM, which also decides on the predicted class of a given attribute under some circumstances based on the majority class, is suggested as an imporvement on One-R. Furthermore, we choose to use a different approach for missing values in the dataset.

In Section 4, we present the extensive experimentation done on the algorithms and 16 popular machine learning datasets. For each dataset we have used One-R, One-RM and One-RM* (explained later) to predict the class of examples with 100 fold cross validation for all except One-RM*. The

*Corresponding Author, Email:

results of the experiment are presented as tables.

In Section 5, an attempt is made to analyze some of the results obtained in Section 4. By using comparative analysis, we show that One-RM is an equally feasible alternative algorithm to One-R, and can be used when a slightly better accuracy is required at the cost of little computation.

In Section 6, we conclude with remarks to our algorithm and its shortcomings and point towards possible future work.

In the next section we briefly summarize the existing One-R algorithm with an illustration.

2. Background

2.1 Basic Structure of One-R Algorithm

One-R algorithm takes a set of instances as input, each with several attributes and a class. It infers a rule as the output from the given set of examples. This rule is based on a single attribute which is the most accurate in predicting the given class (Nevill-Manning, *et al,* 1995), in other words, it generats a one-level decision tree. Fig. 1 shows the One-R algorithm. The basic idea of the One-R algorithm is to choose an attribute with the lowest error rate. It first takes an attribute, and for each value of this attribute, assigns it to the most frequent class that this value yields. Then the number of errors of this attribute value corresponds to a different class. The sum of errors for all values of a particular attribute forms its total error. The attribute having the least of all total errors is chosen as the One-R rule.

To see how One-R algorithm generates

1. for each attribute
2. for each value of that attribute, make a rules as follows
3. count how often each class appears
4. find the most frequent class
5. make the rule assign that class to this attribute-value
6. calculate the accuracy of the rules
7. choose the rules with the highest accuracy (lowest error rate)

Fig. 1 Pseudocode of the One-R algorithm

Rule for each value of an attribute and a rule set for each attribute and how it chooses the final rule set, consider the following small illustrative dataset in Table I. The table shows the weather data in order to decide whether to play a certain game or not. The features or attributes for the predictive class variable play are outlook, temperature, humidity and windy.

To classify on the final column, play, One-R generates four sets of rules, one for each attribute, as shown in Table II. And the final output rule set is: outlook: sunny →no overcast →yes rainy →yes

2.2 Discretization scheme of numeric attributes

It is common for a machine learning algorithm to focus primarily on nominal value attributes (Michalski and Stepp,

1983). However, many real-life datasets exist which involve continuous value attribute. In order to apply machine learning algorithms to these datasets, the continuous features must first be discretized. One-R utilizes a simple supervised discretization method to convert attributes from numeric to nominal form.

Several alternatives exist in discretization methods. (Nevill, *et al*, 1995) suggest the use of iterative merging of partitions based on improving the accuracy of the splits. A broad study on discretization schemes is done in (Dougherty, *et al*, 1995). It concludes that the accuracy of a classifier, for example, Naive Bayes could be increased with the use of right kind of entropy-based discretization method prior to learning. The following example illustrates how One-R processes numeric values. The example represents values of the temperature attribute from the numeric version of the weather data. The process of discretization for a minimum bucket size of 3 is achieved in four stages. In the first stage we just assign the predictive class values for each instance.

In the second stage we create bins by placing breakpoints after every third or later value when we can find a majority in the class label. In this the first bin has a size of 4 and the class label is ‘yes’

In the third stage, we notice that the first value of the second bin is the same as the class label of the first bin. Therefore, the first bin is extended to include this instance, extending the partiton size from 4 to 5. Similarly, the second partition is generated. The majority class in this partition is also ‘yes’

In the final stage, it is noticed that the first two bins belong to the same class, therefore, they are merged to form one large bin with calss label ‘yes’ without loss of meaning. The last bin is arbitrarily chosen as ‘no’ since there are equal number of ‘yes’ and ‘no’ labeled instances

Therefore, the final rule set for this sequence of values is:

if temperature <=77.5 then class = yes

if temperature >77.5 then class = no

2.3 Discretization scheme of numeric attributes

One-R treats missing values as a new attribute value. This implies that all the missing values are useful for prediction.

In some cases, this assumption may be inappropriate. For example, if there are a large number of missing values One-

R would treat this attribute as having high predication accuracy, which is a mistake.

We have proposed a better method for handling missing values in the next section.

An example of how missing values are treated in One-R is given below.

As shown in the Table III. there are five instances whose values of attribute `windy' are missing. One-R treats missing value as a new attribute value of `windy'. Now the attribute values of `windy' are `True', `False' and `Missing'. Then the rule set for attribute `windy' is as follows.

Table IV shows the problems that may arise from having missing values represented as separate values. Note that in Table IV, there are 5 instances whose values are missing. Out of these there are 3 "no" class labels and 2 "yes" class labels. Based on this simple majority the class label for the missing values has been assinged "no". However, there may not be any practical relationship between missing value of an attribute and the class value of the instance. We try to resolve this problem using a different approach in the next section.

3. An Improved One rule algorithm - One-RM

Limitations of One-R Algorithm: Despite its popularity, the One-R algorithm also has some limitations. The two main problems of One-R are: the arbitrary choice made for attributes with similar average error values and a random choice for assigning attribute-class rules when the attributes have the same support values for all classes.

Arbitrary selection of rule

We see that in Table II. 'A' in the class column indicates that s random choice has been made between the final rule outlook and humidity, because both have equal total errors 4/14. 'B' indicates that an arbitrary choice has been made to decide the class an attribute value would fall into whose examples fall equally into the two classes. For example, the number of occurrences of class 'yes' and 'no' for 'hot' are equal. One-R randomly chooses class 'no'

Arbitrary selection of attribute-class:

One-R assigns an attribute value to a class based on the most frequent class. However, if there is more than one class with the same highest frequency, then it arbitrarily decides the class. Consider the weather data from Table I, for the attribute windy, there are three instances with 'false' value and three instances with 'true' value with class 'no'. Since, there is a tie; One-R chooses 'no' arbitrarily.

A similar problem may also arise in datasets having more than two class values. Consider a hypothetical example in Table V, where there are three classes 'A', 'B', and 'C', and an attribute X having 'True' as a value. A tie may also occur if there 3 instances supporting each of the three classes, or 3 in any two and 0 in the third class (Table VI)

Table III. Missing values of attribute windy

Windy	Play
False	no
True	no
False	yes
?	no
?	no
False	yes
False	yes
True	no
?	no
True	yes
False	no
False	yes
False	yes
True	yes
True	yes
?	yes
?	yes
False	yes
True	no

Table V. A three-way tie

Class	Instances	Tie
A	3	Among all three
B	3	attributes
C	3	

Table VI. A three-way tie

Class	Instances	Tie
A	3	
B	3	Between A and B
C	3	

Next, we address each of these key problems and present our improved algorithm One-RM.

3.1 Decision based rule selection

a. Class based average error

We propose a new method of choosing an attribute. The rule selection criterion is based on the average error of all attribute values that cover or correspond to a certain class. In other words, the One-RM rule is the attribute having the least average error based on the prediction class. In case of a tie, the total error is also taken into consideration similar to One-R. An illustrative example is given in Section 3.3.

b. Function mapping attribute values to class value

As a solution to the problem of mapping attribute values to prediction class, we propose that when there is a tie among attributes having high coverage for every class values, we break the tie by assigning the attribute of the majority class of the dataset. For example, the majority class is ‘yes’ as shown in Table VII.

Table VII. Coverage for classes in the weather data

Class	Coverage
yes	9
no	5

So, in the case of Table II, instead of ‘No’, we choose ‘Yes’ for attribute Windy, value ‘True’ and attribute ‘Outlook’ value ‘Hot’. Table VIII shows how it is done for the attribute Windy.

Table VIII. Choosing the majority class in case of a tie-breaker

Attr	Rules		Number of examples			error	total
	Value	Class	Corr	Err	Cov	rate	err
Windy	false	Yes	6	2	8	2/8	5/14
	true	Yes #	3	3	6	3/6	

3.2 Missing values

OneR’s approach to missing values is based on the assumption that all missing values are useful information for classification. We argue that a missing attribute value offers little additional information. Therefore, we do not take the missing values into account in the training process. However, during the classification process, if the attribute value is missing we classify these instances into the majority class, which is the class with the most occurrences. For example, in Table IV, when we find a missing value for windy, we classify it as the majority class ‘Yes’

3.3 One-RM: An illustration

Recall the weather dataset in Table I. The total coverage of the two class values in this example dataset according to Table VII are IX for 'Yes' and 5 for 'No'. So the majority class is 'Yes'. Table VIII shows how One-RM decides which attribute should be the best rule. For the attribute outlook, One-RM proceeds in the same way as One-R i.e., assign each of the value of the attribute outlook to a value of the predictive class, in this case either 'Yes' or 'No' based on the support of the attribute value. Similarly, the rest of the attributes are assigned a predictive class value. From Table II, Sunny is assigned to 'No', Overcast to 'Yes' and Rainy to 'Yes' . The error values of Overcast and Rainy are (0/4) and (2/5) respectively. Since, both Overcast and Rainy are assigned to the class 'Yes', we need to calculate their average error, i.e., (0/4+2/5)/2=0.20. Similarly, Sunny is the only attribute which is assigned to the class value 'No', its average error value is set to the original error value of 0.40. We can now combine the class based average errors for the attribute Outlook, i.e., 0.20 + 0.40 = 0.60. Similarly, the total average error for the attribute Temperature is 0.36, Humidity is 0.58, and Windy is 0.38. Clearly, the attribute with the least total average error is Humidity (0.58), therefore, Humidity is selected as the One-RM rule.

Table IX. Evaluating the attributes with class based error rate

Attributes	Class	Class based error	
		Average Error	Total avg error
Outlook	Yes	(0/4+2/5)/2=0.20)	(0.20+0.40)=0.60
	No	2/5 = 0.40	
Temperature	Yes	(2/4+2/6+1/4)/3= 0.36	(0.36+0)=0.36
	No	0	
Humidity	Yes	1/7 = 0.14	(0.14+0.43)=0.58
	No	3/7 = 0.43	
Windy	Yes	0	(0+0.38)=0.38
	No	(2/8+3/6)/2=0.38	

Algorithm: One-RM

Fig. 3 shows the algorithm for generating One-RM rule from a dataset. First, for each attribute and its different values we find out the most frequent class for those values. If there are two values with the equal frequency then the tie is broken by choosing the class which has the highest coverage in the dataset. The number of candidate rules at this stage is simply the number of different attribute values. Calculate the average class based error rate by taking the average of the errors induced by the attribute rules.

1. for each attribute
2 for each value of that attribute, make a rule as follows:
3. count how often each class appears
4. find the most frequent class
5. if there are more than one class of the most frequency
6. then choose the class with the most coverage of this training set
7. end if
8. make the rule assign that class to this attribute-value
9. calculate the error rate for this value
10. end for
11. calculate the class based error rate
12. end for
13. choose the rules with the lowest class based error

Fig. 3. Pseudocode for the One-RM algorithm

The algorithm, in Fig. 3, clearly favors those attributes having values more than there are classes. This idea about taking the average is a natural step of normalizing the effects of summing errors without consideration for the number of different values of the prediction attribute. Holte meticulously pointed out that a l-rule learner, such as OneR faces performs poorly when the dataset has few attributes having more values that there are classes. OneR does well in cases where continuous attributes are many.

4. Experimental evaluation

4.1 Description of datasets

To test the difference in classification performance between One-R and One-RM, thirteen datasets are selected from the

UCI repository of datasets (Blake and Merz, 1998). These datasets have been chosen as they are widely used in machine learning research, and also because they have been used in studies in the context of One-R algorithm (Holte, 1993). The context of the datasets and their corresponding filenames are shown in Table X.

Table XI describes the datasets in terms of the following characterstics.

Number of Class Values: Shows how many different values there in the prediction class attribute.

Baseline Accuracy: It is the ratio, in percentage, of the number of most frequently occurring class and total number of instances.

B.A = (Majority Class Instances/Total Instances)*100%

Missing Values: Shows if there are instances with attribute value (s) missing.

Number of distinct attribute values:

There are two types of attribute values-Continuous (numeric) and Discrete (nominal). The count in the continuous column shows how many attribute have numeric values. Rest of the columns indicates how many attributes there are with N values, where $2 \leq N \leq 6$ are the column headings.

We only consider those numeric attributes as continuous which have more than 6 distinct values.

Total: Total is the sum of all attributes. If there is a mismatch between the sum of the attribute and the total, because an attribute has not been reported, then it means that the attribute has the same value for all instances. For example, the MU dataset, the columns add up to 21. This means that there is one attribute which has a fixed value for all the examples.

4.2 Comparison of classification accuracy

Experiment 1: All algorithms (except One- RMW, as explained in the next section) were run with 100 fold cross-validation where the number of instances were greater than or equal to 100, 50 fold crossvalidation where instances were greater or equal to 50 and 10 fold cross-validation for datasets having 10 or fewer instances.

Table XII shows the percentage of prediction accuracy for One-R and One-RM on the thirteen datasets. The result is promising; One-RM performs as well as One-R in most of the cases (6 datasets), and performs better than One-R in 5 of the 7 remaining cases.

Experiment 2: Table XII shows the comparison among One-R, One-RM, One- RMW, and J48 algorithm. The J48 algorithm used is the one available with the WEKA package (Holmes, *et al*, 1994). All the parameters had their default

values. One-RMW is not a new algorithm, but One-RM applied to the same training set for determination of accuracy. Naturally, this gives the greatest accuracy possible for One-R under practical conditions. This is done to show how One-RM can be used, much like One-R (Holte, 1993) to predict the accuracy of a more complex algorithm. One- RMW

Table XII. Comparison between One-R and One RM classification accuracy

	Datasets						
	BC	HD	IR	CH	VO	LA	HO
One-R	69.2	73.5	93.3	65.9	95.6	6 8.4	81.5
One-RM	70.6	73.9	95.3	66.1	9 5.6	71.9	72.5
	HE	HY	LY	MU	GL	CG	
One-R	84.5	96.4	74.3	98.5	57.4	66.1	
One-RM	83.2	96.4	74.3	98.5	57.4	70.0	

Table XIII. Comparison among One-R, One-RM, One-RMW and J48 algorithms.

	Datasets						
	BC	HD	IR	CH	VO	LA	HO
One-R	69.2	73.5	93.3	65.9	95.6	68.4	81.5
One-RM	70.6	73.9	95.3	66.1	95.6	71.9	72.5
One-RMW	72.4	76.6	95.3	66.1	95.6	75.4	73.4
J48	75.9	75.9	95.3	99.5	96.8	77.2	85.3
	HE	HY	LY	MU	GL	CG	
One-R	84.5	96.4	74.3	98.5	57.4	66.1	
One-RM	83.2	96.4	74.3	98.5	57.4	70.0	
One-RMW	83.2	96.9	75.7	98.5	61.7	71.1	
J48	80.6	85.3	79.1	100	66.4	70.2	

outperforms One-R in every datasets except HO. One-RM outperforms J48 in HY, HE, CG, and HD. In CG and HD, it is the best classifier. Results for One-RMW show that a single value attribute can be potentially a good predictor of accuracy. One-RM performs as well as J48 in IR, and very nearly equal in CG, VO, and does better than J48 in HY, but in some cases it does worse than J48 as expected.

5. Analysis of Results

In Table XIV we present three parameters which help us to have a better understanding of the experimental results.

Missing Values: Observe that in order to compare performance of One-R and One-RM and to understand their difference more closely, we need to evaluate them on a dataset that has no missing values. This is important because, One-R and One-RM employ different techniques to handle missing values.

Selected Attribute: It is a fact that two ‘one rules’ which select the same attribute will perform equally well. This means that if One-R and One-RM choose the same attribute as their rules then their performance should be equal regardless of treatment of the missing values. Only when the two algorithms choose different attributes, should we be able to find any difference in performances. This should be true for all experiments done solely on the training set, without the cross-validation. The result is consistent with the hypothesis.

There is a risk of generalizing this rule to prove the inverse argument, i.e., if One-R and One-RM have the same performance, then they may not choose the same attribute. This is due to the difference in the rule selection algorithms of the two algorithms. In addition, different attributes may have the same prediction accuracy.

Performance: Prediction accuracy of the techniques is measured in numerical values with a precision of two places after the decimal. The performance measure does not take into account other factors, such as, confusion matrix, lift chart, ROC chart, F-measures, etc.

Analysis of the performances of algorithms on individdual datasets:

BC: One-RM’s approach to missing values takes effect

IR, LY, MU: One-R and One-RM use different criterion of rule selection but select the same attrivute (rule set).

CH, CG: One-RM selects a better attribute (rule set).

VO, HY : One-R and One-RM use different criterion of rule selection but select the same attribute (rule set), also they adopt different approaches to missing values but get the same results.

Table XIV. Analysis of performance based on attribute characteristics

	Dataset	Selected attribute		Performance
Name	missing	One-R	One-RM	
BC	yes	inv-nodes	inv-nodes	OneR< OneRm
HD	yes	Thal	number_of_vessels_colored	OneR<OneRm
IR	no	Petallength	Petallength	OneR = OneRm
CH	no	Bxqsq	Rimmx	OneR< OneRm
VO	yes	Physician-fee-freeze	physician-fee-freeze	OneR = OneRm
LA	yes	wage_increase first_year	wage_increase_second-year	OneR< OneRm
HO	yes	Abdomen	Surgery	OneR>OneRm
HE	yes	ASCITES	ALBUMIN	OneR>OneRm
HY	yes	TSH	TSH	OneR>OneRm
LY	no	Changes_in_node	changes_in_node	OneR = OneRm
MU	no	Odor	Odor	OneR = OneRm
GL	no	Al	Al	OneR = OneRm
CG	no	credit_amount	Duration	OneR < OneRm

LA, HD: In these two cases, One-RM's performance is better than One-R's, however; we cannot identify which approach of One-RM causes this increase in performance, necessitating further experiment. We modify One-RM's approach to missing values and use the original approach of One-R. We call this algorithm OneRm*. We use this algorithm to test LA. HD, which gives the results given in Table XV.

Table XV. Quantitative evaluation on LA and HD datasets

	OneR	OneRm*	OneRm
LA	68.4	71.9	71.9
HD	73.5	73.5	73.9

Table XV helps us conclude that for LA, One-RM's new selection scheme is responsible of the increase in accuracy. On the other hand, in HD, One-RM's approach to missing values takes effect.

Experimental tools and environment

Hardware	CPU: Pentium IV 2.4G RAM: 256M
Operating system supporting software	Microsoft windows 2000 SP3 WEKA 3.2.3
Test setings	Test methods: Cross-validation Cross-validation folds: 100 (if the number of instances>100) 50 (if the number of instances> 50) 10 (if the number of instances> 10)

Conclusion

One-R is a valuable tool which provides the baseline accuracy for tesing new machine learning algorithms. However, because of the simplicity of One-R it offers less accuracy in its prediction. Certain aspects of One-R are also ambiguous and implementations may differ on issues such as treatment of missing values and ties in error values of attributes. Our algorithm, One-RM, which is a derivative of One-R resolves all of these issues and performs much better than One-R in many cases. One-RM performs equal to One-R in some eases, however it out performs one-R by significant margin in the rest. Theoretical analysis suggests that One-RM used in conjunction with One-R always performs either better or equal to One-R. Experimental analysis shows that One-RM is a viable alternative to One-R when used as a separate classification rule.

Reference

Blake C., and Merz C. (1998) UCI Repostory of machine learning databases [ics. uci, edu/mlearn/ML repository. html] Iroine, CA: University of California Department of Information and Computer Scicnce, **85:** p. 90.

Dietterch, (1997) T Machine learning research: Four current directions. AI Magazine, **18** (4): 97-136.

Dougherty J., Kohavi R., and Sahamk M. (1995) Supervised and unsupervised discretization of continu-

ous features. in procedings of the Twelfth International Conference of Machine Learning: Morgan Kaufann, Publisher, San Farancisco, CA.

Fisher D. (1987) Knowledge acquisition via incremental conceptual clustering. Machine Learning. **2**(2): 139-172,

Fisher D., and Schlimmer J. (1988) Concept simplification and prediction accuracy in proceedings of the Fifth international conference on Machine Learning Morgan Kaufmann Pub.

Holmes G., Donkin A., and Witten I. (1994) WEAKA: a machine learning workbench. Intelligent Information Systems.Proceedings of the 1994 Second Australian and New Zealand Conference on 1994: p. 357-36l.

Holmes G.., and Nevill-Manning C. (1995) Feature selection via the discovery of simple classification rules in Proceedings of the symposium on Intelligent Data Analysis.

Holte R, (1993) Very Simple Classfication Rules Perform Well on Most Commonly Used Datasets. Machine Learning. **11** (1): 63-90.

Mickalski R., and Stepp R. (1983) Learning from observation: Conceptual clustering. Machine Learning; An Artificial Intelligence Approach.

Nevill-Manning C., Holmes G., and Witten I. (1995) The Development of Holte's IR Classifier. Proceedings of the 2nd New Zealand Two-Stream International Conference on Artificial Neural Networks and Expert Systems.

Received : August 04, 2008;
Accepted : September 14, 2008