Rule Extraction from Random Forest: the RF+HC Methods

Conference Paper · June 2015

DOI: 10.1007/978-3-319-18356-5_20

CITATIONS
9

READS
2,579

2 authors:

Morteza Mashayekhi
University of Windsor
16 PUBLICATIONS **56** CITATIONS

SEE PROFILE

Robin Gras
University of Windsor
129 PUBLICATIONS **936** CITATIONS

SEE PROFILE

Rule Extraction from Random Forest: The RF+HC Methods

Morteza Mashayekhi(✉) and Robin Gras

School of Computer Science, University of Windsor, Windsor, ON, Canada
{mashaye,rgras}@uwindsor.ca
http://www.uwindsor.ca

Abstract. Random forest (RF) is a tree-based learning method, which exhibits a high ability to generalize on real data sets. Nevertheless, a possible limitation of RF is that it generates a forest consisting of many trees and rules, thus it is viewed as a black box model. In this paper, the RF+HC methods for rule extraction from RF are proposed. Once the RF is built, a hill climbing algorithm is used to search for a rule set such that it reduces the number of rules dramatically, which significantly improves comprehensibility of the underlying model built by RF. The proposed methods are evaluated on eighteen UCI and four microarray data sets. Our experimental results show that the proposed methods outperform one of the state-of-the-art methods in terms of scalability and comprehensibility while preserving the same level of accuracy.

Keywords: Rule extraction · Random forest · Hill climbing

1 Introduction

Random forest (RF) is an ensemble learning method for both classification and regression that constructs and integrates multiple decision trees at training step using bootstrapping. Additionally, it aggregates the outputs of all trees via plurality voting in order to classify a new input. It has few parameters to tune and it is robust against overfitting. It runs efficiently on large data sets and can handle thousands of input variables. Moreover, RF has an effective method for estimating missing data, and has some mechanisms to deal with unbalanced data sets [4]. In some applications, RF outperforms well-known classifiers such as support vector machines (SVMs) and neural networks (NNs) [5,18]. Despite good performance of RF in different domains, its major drawback is that, it generates a 'black box'model in the sense that it does not have the ability to explain and interpret the model in an understandable form [23,27] given that it generates a vast number of propositional if-then rules. As a result, ensemble predictors such as RF are very rarely used in domains where making transparent models is mandatory, such as predicting clinical outcomes [23]. In order to bear this limitation, the hypothesis generated by RF should be transformed into a more comprehensible representation.

D. Barbosa and E. Milios (Eds.): Canadian AI 2015, LNAI 9091, pp. 223–237, 2015.
DOI: 10.1007/978-3-319-18356-5_20

To obtain a comprehensible model which is simpler to interpret, accuracy is often sacrificed. This fact is normally referred to as the 'accuracy vs. comprehensibility tradeoff'. The importance of accuracy or comprehensibility is completely related to the application. One way to obtain a transparent model is to induce rules directly from the training set or to build a decision tree. However, another option is to take advantage of the good performance of the existing opaque models such as SVMs, RF, or NNs and generate rules based on them. This process is called rule extraction (RE), which is aimed at providing explanations for the predictive models' outputs. There are two different rule extraction methods based on an opaque model: decompositional and pedagogical [11]. Decompositional methods extract rules at the level of individual units of the prediction model such as neurons in neural networks, and therefore rely on the model's architecture. In contrast, in pedagogical approaches, the predictive model is only used to produce predictions.

In previous years, a high number of rule extraction methods using trained NNs and SVMs have been published (see [11] for a good survey). Nevertheless, in the case of the RF model, few research projects have been conducted. In this paper, the RF+HC methods for the interpretation of the RF model are proposed. The proposed methods can be treated as a decompositional rule extraction approach given that we employed all the generated rules by RF, which are dependent on the number of trees and also the tree structures in the RF.

This paper is organized as follows: the background description including foundation of the RF followed by a discussion of related research projects are explained in section 2. In section 3, the RF+HC methods are introduced. Experimental results of our methods applied to several data sets and comparisons are described in section 4. Finally, we present our conclusions along with possible future directions for our work.

2 Background

2.1 Random Forest

The RF is an ensemble learning method such that successive trees do not depend on the previous ones and each tree is constructed independently using a bootstrap sample of the data set. At the end, a majority voting procedure is used for making predictions. In addition, each node is split using the best feature among a subset of features (m) randomly chosen at that node. Parameter m is usually equal to $0.5 \times sqrt(n)$, $sqrt(n)$, or $2 \times sqrt(n)$ where n is the number of features. Error estimations are performed on a subset of data which are not included in the bootstrap sample at each bootstrap iteration (This subset is called out-of bag or OOB). RF can also estimate the importance of a feature by permutation of the values associated with a feature and comparing of the average OOB error before and after the permutation over all trees. However, it does not consider the dependency between features.

RF deserves to be considered as one of the important prediction methods because it demonstrates a high prediction accuracy and it can be used for clustering and feature selection applications as well [4,7,22]. Moreover, estimating

the out-of-bag error often eliminates the need for cross-validation. More importantly, as it generates a multitude of propositional if-then rules, which is the most widespread rule type in RE domain, it has a very high potential to provide clear explanations and interpretations of its underlying model.

2.2 Related Work

One of the projects focusing on this topic was conducted by Zhang et al. [27] to search for the smallest RF. Although their method is not a rule extraction strategy, it seeks out a sub-forest that can achieve the accuracy of a large RF. They used three different measures in order to determine the importance of trees in terms of their predictive power. The experimental results demonstrate that such a sub-forest with performance as good as a large forest exists. Latinne et al. [13] attempted to reduce the number of trees in RF using the McNemar test of significance on the prediction outputs of the trees. Similarly, others tried to select an optimal sub-set of decision trees in RF [2,16,26]. These methods are not really rule extraction methods and mostly concentrate on reducing the number of decision trees in the RF or in a similar ensemble method such as bagging.

There are also some other methods to increase comprehensibility of an ensemble or RF by compacting them into one decision tree. For example, a single decision tree was used to approximate an ensemble of decision trees [24]. In this method, class distributions were estimated from the ensemble in order to determine the tests to be used in the new tree. A similar method was employed to approximate the RF with just one decision tree [12]. The aim was to generate a weaker but transparent model using combinations of regular training data and test data initially labeled by the RF.

Other methods with different approaches were proposed to select an optimal set of rules generated by RF [9,17]. More specifically, Liu et al. [14,15] used RF as an ensemble of rules and proposed a joint rule extraction and feature selection method (CRF) based on 1-norm regularized RF, using sparse encoding and solving an optimization problem applying linear programming method.

3 RF + HC Methods

RE can be expressed as an optimization problem [8] and one solution of this problem is to apply heuristic search methods. These methods overcome the complexity of finding the best rule set, which is an NP-hard problem.

In this section, we present our algorithm (Algorithm 1) to extract comprehensible rules from RF as follows. The algorithm consists of four parts: In the first part, RF is constructed and all the rules in the forest are extracted into the *Rs* set. The second part of the algorithm computes the score of all rules based on the *RsCoverage*, a sparse matrix that shows which rules cover each sample and its corresponding class label. Afterwards, the scores are assigned to the rules in order to control the rule selection process, which can be based on different factors such as accuracy and rule coverage. We used equation (1) that has been shown to be a promising fitness function [20]:

Algorithm 1. RF+HC

```
Input: trainSet, testSet,iniRuleNo, treeNo
Step 1: // Construct Random Forest
   RF = trainRF( trainSet, treeNo );
   Rs = getAllTerminalNodes (RF);
Step 2: // Compute rules coverage
   m = size( trainSet );
   n = size(Rs);
   RsCoverage=zeros( m, n);
   foreach sample in trainSet {
       foreach rule in Rs
           if match(rule, sample)
               RsCoverage(sample, rule) = class;
   }
   RScore = ruleScore (RsCoverage);
Step 3: // Repeat the HC method to obtain best rules
   iniRs = getRuleSet(RScore, n, iniRuleNo);
   impRs = iniRs; bestRs=iniRs;
   for i=1 to MaxIteration {
       impRs = HeuristicSearch (impRs, RScore);
       if Acc_impRs >Acc_bestRs
           bestRs = impRs;
       impRs = getRuleSet( RScore, n, iniRuleNo);
   }
Step 4: // Calculate the accuracy on test set
   calcPerformance(testSet,bestRs);
```

$$ruleScore_1 = \frac{cc - ic}{cc + ic} + \frac{cc}{ic + k} \tag{1}$$

In this formula, cc (correct classification) is the number of training samples that are covered and correctly classified by the rule. Variable ic (incorrect classification) is the number of incorrectly classified training samples that are covered by the rule. Finally, k is a predefined positive constant value. In our case k=4, though other values can be used as it is mostly to avoid the denominator becomes zero and there is no significant change in the results by modifying k). This scoring function ensures the retention of the rules with higher classification accuracy and higher coverage and to remove the noisy rules. Obviously, other fitness measures can be used instead. One possibility would be to employ the rule score based on metrics such as number of features in the extracted rule set and number of antecedents to increase the quality of rules in terms of comprehensibility. In the third step of the algorithm, a fitness proportionate selection method is used *iniRuleNo* times to generate an initial rule set (*iniRs*) with a probability to select a rule proportional to its score. In order to search the RF rules space,

we used the random-restart stochastic hill climbing method, which gives a local optimum point of the search space based on the random start locations.

Any other search methods such as simulated annealing or genetic algorithm can be applied instead of HeuristicSearch function in the algorithm. We repeated the search with a predefined maximum number of iterations (*MaxIteration*), each time with a new initial rule set. This can compensate some of the deficiencies in hill climbing due to the randomized and incomplete search strategy [21]. The hill climbing algorithm, searches for the best neighbor, the one with the highest score, of the current location based on equation (1) in the search space and by changing (adding/removing) one rule to the current rule set. For adding/removing a rule, we used the same fitness proportionate selection procedure that was employed for producing the *iniRs*. The hill climbing score function was defined based only on the overall accuracy because the scoring schema of the second step already took into account both rule coverage and rule accuracy. If the new movement in the rule set space improves the score value, that change is retained. Otherwise it is discarded and then another neighbor in the rule space is sought. We repeat this step for a pre-defined maximum number of iterations (*MaxIteration*). Finally, in the fourth step, we apply the best extracted rule set on the test set to evaluate the generalization ability of the extracted rules.

One of the RF characteristics is that there is no pruning while it is constructed. Therefore, we expect to have long rules (with a large number of antecedents) in the rule set as well as in the extracted rule set using the proposed algorithm. Having long rules damages the interpretability of the model and thus it should be considered in the applications for which the interpretation of the rules is important. Therefore, we proposed the second algorithm, which is basically similar to Algorithm 1 except that a modified version of the rule score function (i.e. equation (2)) was used, where *rl* shows rule length or number of antecedents. We called the new method RF+HC_CMPR. In the RF+HC_CMPR method more generalized rules (shorter length rules with higher accuracy) have higher priority than the more specialized rules (the longer rules with lower accuracy) based on the following equation:

$$ruleScore_2 = ruleScore_1 + \frac{cc}{rl} \tag{2}$$

The inputs of the proposed methods are the training/test sets, initial number of rules (*iniRuleNo*) and the number of trees in the RF (see Algorithm 1). Variable *iniRuleNo* adjusts the tradeoff between accuracy and comprehensibility. In cases where prediction ability is important, higher values are used and in cases where the interpretation of the underlying model is important lower values should be used. For the implementation, we used Matlab as the same as the source code available for the CRF method.

4 Experiments and Discussion

To compare our proposed methods with other methods, we also applied CRF [14,15] and RF on 22 different data sets. Different criteria have been proposed to

evaluate a RE algorithm [11]. For instance, accuracy is defined as the ability of extracted rules to predict unseen test sets. Another major factor is comprehensibility, which is not easy to measure due to the subjective nature of this concept. There are different factors that are used to determine comprehensibility such as, the number of rules and the average number of antecedents. Another desirable characteristic of a RE method is its potential to be applicable to a wide range of applications. If a RE algorithm is applicable to data sets with a large number of samples, features, or classes then it is said to be scalable. This scalability notion includes time and algorithm complexity.

In our work, we measured the average accuracy of 10 times 3-fold cross-validation (by randomizing the data set for every repetition) for evaluating accuracy, as it gives more accurate results in compare to one time k-fold cross-validation. This measure demonstrates the prediction and generalization ability of the extracted rules. Majority voting is used to classify a sample when more than one rule covers a sample. We assumed a default rule such that the samples not covered by any of the extracted rules are simply assigned to the high frequency class in the dataset. In the RF+HC methods, due to their stochastic nature, we repeated the whole procedure 10 times and computed the average results along with their standard deviations. For the CRF method, 10 different values for the lambda parameter (which indicates the tradeoff between the number of rules and accuracy) were used. To determine these values, we did a few pilot runs with each data set separately. To determine the best lambda, a cross-validation step is incorporated in the CRF method such that it selects the lambda value, which gives the minimum error for cross-validation. In order to show the comprehensibility of the methods, we considered the number of rules, maximum rule length, and total number of antecedents in the extracted rule set. For the CRF method, these values are related to the lambda parameter value, which gives the lowest cross-validation error. On the other hand, those of RF+HC methods are related to 10 repetitions of the process. All the values are rounded to the closest integer value.

Scalability is one of the most important evaluation metrics often overlooked in most of the RE methods such as the CRF method. We measured the computational time as a metric to evaluate the scalability. To have a fair comparison, we used 10 different lambdas in the CRF method and we divided the required time to find the best lambda by 10. This means that we only considered the time for cross-validation using the best lambda plus the time for training and test steps. We considered the cross-validation time because it is an important part of the CRF method , which finds the best lambda in each iteration of the algorithm. At each iteration, that includes cross-validation, optimization, and feature selection, the features in the extracted rules are kept and the rest are removed. In the case of RF+HC, we repeated each experiment 10 times and then divided the overall time by 10. We also considered the hill climbing repetition time (*MaxIteration*) in order to calculate the computation time.

As the input of the proposed algorithms, we should specify the initial number of rules (*iniRuleNo*) for each data set. We used 500 decision trees to build RF

with m=$sqrt(n)$, which is a default value mostly used in the literature, where (m) is the number of features randomly chosen at that node for splitting. In the random-restart hill climbing, we repeated hill climbing from 10 initial rule sets. We took *MaxIteration = 500* in all of our experiments. Higher values provide hill climbing with more opportunities for likely improving the rule set score, although it did not happen in our case. For comparing the proposed methods with CRF in terms of performance, comprehensibility, and computation time complexity, we used Wilcoxon and Friedman tests as suggested in [6].

4.1 Data Sets

We used 22 data sets with various characteristics in terms of the number of features, the number of samples, and the number of classes to observe how the performance of the proposed methods varies depending on the data set type. Eighteen data sets were taken from UCI machine learning repository [3] and another four data sets (Golub [10], Colon [1], Nutt [19], Veer [25]), which are gene expression microarray data sets. The extreme cases are Veer with 24188 features, Magic with 19020 samples, and Yeast and Cardio with 10 classes (see Table 1).

Table 1. Data sets along with their characteristics

Data set	Feature#	Class#	Sample#
Breast Cancer	9	2	699
Magic	10	2	19020
Musk Clean1	166	2	476
Wine	13	2	178
Wine Quality	11	6	1599
Iris	4	3	150
Yeast	8	10	1485
Cardiography	20	10	1726
Balance Scale	4	3	625
Cmc	9	3	1473
Glass	9	6	214
Haberman	3	2	306
Iono	34	2	351
Segmentation	19	7	210
Tae	5	3	151
Zoo	16	7	101
Ecoli	7	8	336
Spam	57	2	4601
Golub	5147	2	72
Colon	2000	2	62
Glimo Nutt	12625	2	50
Veer	24188	2	77

4.2 Accuracy and Generalization Ability

On average, both the RF+HC and RF+HC_CMPR methods gave almost the same level of accuracy as the CRF method with marginal differences (Table 2). Moreover, all three methods obtained 96% of the RF accuracy for the whole data sets on average. For some datasets, they demonstrated higher accuracy than RF such as for Tae, Cmc, and Golub with RF+HC or Tae and Clean with CRF method. A similar result was observed in [28] when the authors used a NN ensemble to extract the rules, observing higher accuracy for extracted rules than for the underlying model.

The generalization ability of RF+HC is due to the selection of the high score rules in RF and it is also due to some level of stochasticity, which results in assigning odds to the rules with low scores in the training set, but they may be important for unseen data. Comparing the accuracy of CRF method with the proposed methods revealed that the null hypothesis with $\alpha = 0.05$ cannot be rejected with $z = 0.41$ (CRF vs. FR+HC) and $z = 0.42$ (CRF vs. RF+HC_CMPR), while the critical z value is -1.96 in Wilcoxon test. Therefore, the difference is not significant, which proves that two methods are equivalent in terms of accuracy.

Table 2. Percentage accuracy of the RF+HC, RF+HC_CMPR, CRF, and RF methods on the selected data sets along with the standard deviations in parenthesis

Data set	RF+HC	RF+HC_CMPR	CRF	RF
Cancer	96.18 (0.32)	96.23 (1.56)	95.71 (1.01)	96.65 (1.75)
Magic	85.37 (0.46)	85.6 (0.28)	83.65 (1.3)	88.12 (0.3)
Clean	81.34 (3.25)	83.17 (4.3)	88.45 (1.55)	88.68 (2.18)
Wine	92.07 (3.29)	95.93 (1.8)	91.93 (5.91)	98.99 (0.9)
Wineqlty	65.13 (1.93)	62 (1.8)	62.79 (0.57)	68.59 (3.47)
Iris	93.36 (2.4)	94.12 (3.25)	94.4 (2.61)	96.40 (1.67)
Yeast	59.98 (1.5)	61.3 (0.7)	55.02 (2.75)	62.02 (1)
Cardio	81.74 (0.82)	82 (0.6)	84.01 (0.84)	85.67 (2.19)
BalancS	84.48 (0.52)	83.75 (2.36)	82.87 (2.86)	87.24 (1.6)
Cmc	52.87 (0.99)	52.6 (2.5)	49.42 (3.65)	52.46 (2.57)
Glass	74.33 (2.7)	73.75 (7.3)	72.77 (2.15)	78.02 (7.51)
Haber	67.69 (2.1)	69.14 (1.7)	70.2 (4.42)	73.92 (4.2)
Iono	90.14 (3.53)	91.9 (3.3)	91.45 (1.6)	93.16 (1.9)
Segment	87.54 (1.86)	89.97 (2.4)	88.86 (3.7)	93.14 (2.1)
Tae	57.60 (3.46)	53.45 (4.1)	62.29 (4.8)	55.60 (1)
Zoo	91.33 (9.6)	92.96 (5.87)	93.94 (8.2)	97.02 (2)
Ecoli	84.2 (3.11)	79.9 (4)	86.67 (11.54)	86.96 (1.74)
Spam	94.04 (0.71)	94.33 (0.5)	94.2 (1.05)	95.24 (0.3)
Golub	93.00 (6.7)	87.25 (7.6)	86.11 (9.62)	92.5 (4.5)
Colon	74.76 (5.26)	76.1 (3.9)	82.46 (17.94)	75.00 (11.85)
Glimo	64.11 (4.26)	66.3 (7.36)	54.9 (8.99)	71.69 (14.47)
Veer	58.27 (7.88)	63.11 (8)	60.97 (8.99)	66.43 (13.76)

4.3 Comprehensibility

Although a feature selection phase was incorporated in the CRF method, our methods were superior in the number of extracted rules in all the data sets except the Golub data set (Table 3). The number of rules extracted by RF+HC or RF+HC_CMPR in average are 0.6% of the total number of rules in RF while that of CRF is 11.66%, which demonstrates very good improvement compared to RF and CRF . The proposed methods significantly reduced the number of rules in comparison to CRF ($z = -4.06$) and as a result improved the comprehensibility. However, the difference in terms of rule numbers for the two proposed methods was not significant ($z = -1.89$). There is one dataset, i.e. Golub, for which CRF extracted only one rule. In such cases, the extracted rule is related to one class and it can only explain that class. However, there is no information and interpretation regarding the other class(es). Therefore, we believe that this type of rule set is not fully comprehensible as it cannot describe the underlying model completely. We found an issue in the implementation of the CRF method, which will affect the results. When the number of rules is reported, only the rules with the weights greater than a threshold (in this case 10e-6) are considered. However, all the extracted rules are used to do prediction for the test set which is not correct. The CRF results in Table 3 corresponds to the correct number of rules.

We used the modified version of the rule score function (i.e. equation (2)) in order to give higher priority to the more generalized rules. Table 3 shows the comparison between the original algorithm and RF+HC_CMPR. The results showed that RF+HC_CMPR have a stronger impact on the maximum rule length and also on the total number of antecedents (42% and 18% decrease respectively) in the rule set in comparison with RF+HC. In addition, we observed no significant change in the accuracy. These results indicate that RF+HC_CMPR improves the comprehensibility significantly ($z = -4.16$).

Comparing the CRF method with the two proposed methods using Wilcoxon test (critical z=-1.96) indicates that RF+HC had a significant lower maximum rule length ($z = -3.13$) and also number of antecedents ($z = -4.07$) in compare to CRF. RF+HC_CMPR was superior in all data sets in terms of maximum rule length ($z = -4.09$) and number of antecedents ($z = -4.07$) except for the maximum rule length for Golub.

One important aspect of comprehensibility is the number of rules extracted from an underlying model. However, we have to consider the importance of the tradeoff between accuracy and comprehensibility. The extracted rules should not only be concise but also have good performance on unseen samples. This is, in fact, the main objective of rule extraction. Therefore, a good rule extraction method should consider two facts simultaneously: comprehensibility and generalization ability, although it should be adjustable based on the application. For example, for the Magic dataset, RF generates 608155 rules with approximately 88% accuracy. This number of rules shows the complexity of the model for this dataset. RF+HC methods extract only about 0.4% of the RF rules and give about 85% accuracy for this data set. We still can generate fewer rules by decreasing *iniRuleNo*, although it will reduce accuracy. Therefore, what needs to

be considered in order to have a fair judgment is the combination of the number of rules and accuracy. The results we have presented in this paper correspond to the smallest number of rules in order to achieve a level of accuracy as close as possible to the level of accuracy for RF. We provided the samples of extracted rules for two data sets in the Table 4.

Table 3. Each cell shows 'Number of extracted rules / Maximum length of rule / Total number of antecedents'in each method. The values in bold show the best results.

Data set	RF+HC	RF+HC_CMPR	CRF	RF
Cancer	36/8/159	**33/6/129**	463/9/1940	12075/13/65869
Magic	2604/8/8186	**2597/3/5697**	3182/37/50668	608155/58/8514170
Clean	83/15/586	**78/10/473**	104/18/947	18392/20/150309
Wine	16/8/64	**14/5/55**	176/7/619	7590/10/26784
Wineqlty	**1258**/21/12301	1259/**12/10526**	2282/24/22256	138889/30/1757860
Iris	13/6/39	**11/5/28**	43/5/145	4202/9/13222
Yeast	**1037**/25/13460	1303/**13/11621**	1836/27/18430	126936/32/1469328
Cardio	1609/20/15720	**1606/11/12951**	2121/20/19003	126412/22/1150839
BalancS	88/9/471	**83/5/339**	360/11/1768	19764/13/124447
Cmc	332/16/2390	**322/10/1818**	2025/19/14695	74257/22/754197
Glass	88/13/398	**59/8/335**	10050/12/30662	16530/16/115932
Haber	28/13/165	**25/8/140**	410/16/2417	19697/18/142512
Iono	41/11/193	**36/7/145**	155/12/784	10641/14/57312
Segment	**42**/10/267	54/**6/175**	11134/13/24065	9905/12/59837
Tae	91/13/495	**76/8/359**	177/13/997	14437/16/93530
Zoo	16/6/66	**15/4/51**	185/7/608	4954/9/17615
Ecoli	**138**/11/762	141/**7/649**	8900/14/29421	16761/16/105260
Spam	476/34/5076	**473/21/4228**	1154/41/14852	118878/44/1455859
Golub	9/3/18	6/**2**/10	**1**/3/**3**	2322/4/4939
Colon	**17**/4/46	19/**3/39**	27/5/85	2620/6/8154
Glimo	**9**/3/23	12/**2/20**	17/4/47	1953/4/4716
Veer	18/4/45	**17/3/33**	39/5/128	3254/6/8513

4.4 Complexity and Scalability

We found a significant difference in terms of computational time between our methods and CRF ($z = -4.07$). For all data sets, the RF+HC methods were superior to CRF with the exception of the Iris data set, which had only a one-second difference (Table 5). More specifically, in some cases with large numbers of classes such as Yeast, Glass, Ecoli, and Segment, our methods were 136, 310, 518, and 842 times faster than CRF respectively. We observed the same circumstance for data sets with a large number of samples such as Magic, Spam, and CMC such that RF+HC and RF+HC_CMPR were 13, 18, and 130 times faster than CRF. On average, the overhead time for the proposed methods and CRF method was 1.12, and 11.8 times respectively relative to RF time.

Moreover, we observed more computational time for CRF when there was a larger number of classes (Table 5) because the CRF method considers c classifiers

Table 4. Sample of rules extracted by the RF+HC_CMPR method from Iris and Golub data sets (The features in the data sets are shown by "V" and a subscripted number. The consequence of each rule is specified by a class label, for example, "Class 1". The value in the parenthesis is the score of the rule based on equation 2. Acc. is test set accuracy).

Iris (Acc. 98%)
$V_4 \leq 0.80$: Class 1 (38.50)
$V_3 \leq 2.70$: Class 1 (38.50)
$V_3 \leq 2.60$: Class 1 (38.50)
$V_3 \leq 4.85$ & $V_3 > 2.70$: Class 2 (20.88)
$V_2 \leq 3.05$ & $V_3 \leq 4.75$ & $V_3 > 2.45$ & $V_2 > 2.55$: Class 2 (10.50)
$V_4 > 0.80$ & $V_2 > 2.95$ & $V_4 \leq 1.70$ & $V_3 \leq 5.15$: Class 2 (5.50)
$V_4 > 1.60$: Class 3 (43.50)
Golub (Acc. 100%)
$V_{1727} > 1570.00$: Class 1 (35.73)
$V_{4572} > 1116.50$ & $V_{737} \leq 526.50$: Class 1 (18.25)
$V_{3607} \leq 13177.00$ & $V_{4005} > 44.50$: Class 1 (18.81)
$V_{4969} \leq 540.50$: Class 2 (21.00)
$V_{4648} > 489.00$: Class 2 (16.46)
$V_{4929} > 5863.00$: Class 2 (12.25)
$V_{1556} > 2699.00$ & $V_{3595} \leq 2939.50$: Class 2 (10.67)
$V_{1394} \leq 63.50$ & $V_{3776} \leq 211.00$: Class 2 (9.31)
$V_{4594} \leq 530.50$: Class 2 (6.67)

(c is number of classes) and finds a weight vector for each class. When there are a relatively large number of samples and a large number of classes simultaneously, the CRF method has an even worse performance. In addition, a large number of features can increase the computational time as CRF has a repeating feature selection step. However, in RF+HC methods, the overhead time on top of RF in RF+HC method has a strong linear correlation with the number of samples in the data sets ($R^2 = 0.994$).

4.5 Overall Comparison and Major Contributions

The major contributions of the proposed methods in comparison to RF are that they refine RF in selecting the most valuable rules, which leads to a huge decrement in the number of rules i.e. 0.6% of the random forest rules, while at the same time attaining 96% of the RF accuracy with a reasonable overhead time on top of RF time. In addition, both methods improved the comprehensibility in comparison with CRF while retaining the same accuracy. RF+HC decreased the number of rules, the maximum rule length, and the total number of antecedents by 27%, 16%, and 49% respectively in average. RF+HC_CMPR also reduced them by 25%, 50%, and 59%. The RF+HC methods decreased the computational time in 21 of the 22 data sets. Moreover, for the data sets with a large

Table 5. Computational time for RF+HC, RF+HC_CMPR, CRF, and RF in second

Dataset	RF+HC	RF+HC_ CMPR	CRF	RF
Cancer	16	16	36	5
Magic	1409	1425	19338	1050
Clean	34	34	118	26
Wine	4	5	13	1
Wineqlty	52	56	5317	17
Iris	4	9	3	1
Yeast	46	49	6276	15
Cardio	80	83	6410	31
BalancS	17	17	233	4
Cmc	36	36	4696	14
Glass	5	5	1551	1
Haber	10	10	15	2
Iono	9	9	24	3
Segment	7	7	5900	2
Tae	5	5	14	1
Zoo	3	3	14	1
Ecoli	9	9	4669	2
Spam	236	239	4479	166
Golub	230	230	253	228
Colon	56	56	62	54
Glimo	633	633	720	631
Veer	3165	3165	3558	3162

number of samples and/or a large number of classes, they were much faster (up to about 800 times) in terms of the computational time. Table 6 summarizes the overall comparisons of RF+HC and RF+HC_CMPR with the CRF method. The numbers in the table specify the average rank of each method for Friedman test computed for the mentioned criteria in the table, where lower value demonstrates the better method. The Friedman test showed significant difference between the average ranks and the mean rank for each criterion. However, the difference was marginal for the accuracy as it was also confirmed by the Wilcoxon test. These results show that our proposed methods are better than the CRF in terms of

Table 6. Comparison summary for different methods. The values are the average rank with the standard deviation in the parenthesis.

	RF+HC	RF+HC_CMPR	CRF
Accuracy	2.23 (0.81)	1.73 (0.7)	2.05 (0.9)
Rule#	1.77 (0.53)	1.32 (0.48)	2.91 (0.43)
Time	1.34 (0.24)	1.7 (0.37)	2.95 (0.21)
MaxCond	2.11 (0.26)	1.02 (0.11)	2.86 (0.35)
Cond#	2 (0)	1 (0)	3 (0)

number of rules, computational time, maximum rule length, and also number of antecedents while they keep level of accuracy as the same as CRF method.

5 Conclusions and Future Works

In this paper, we introduced new rule extraction methods from RF. Experimental results showed that these methods are superior to the CRF method in terms of comprehensibility while keeping the same level of accuracy. In addition, our methods are much more scalable than the state-of-the-art method, CRF and they can be applied more generally and on data sets with various characteristics.

This work can be extended in several different directions in future research. We plan to compare the proposed methods with other related methods, especially the ones described in [2,17,26]. Another possible direction would be improving the rule score and fitness function based on other metrics such as number of features in the extracted rule set and number of antecedents to increase the quality of rules in terms of comprehensibility. Yet another direction is to examine other heuristic search methods such as simulated annealing, tabu search, and genetic algorithms in order to find better sets of rules than those obtained with hill climbing.

Acknowledgments. This research was supported by the CRC grant 950-2-3617 and NSERC grant ORGPIN 341854. We greatly appreciate Brian MacPherson for his comments on this paper.

References

1. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences **96**(12), 6745–6750 (1999)
2. Bernard, S., Heutte, L., Adam, S.: On the selection of decision trees in random forests. In: International Joint Conference on Neural Networks, IJCNN 2009, pp. 302–307. IEEE (2009)
3. Blake, C., Keogh, E., Merz, C.J.: Uci repository of machine learning data bases MLRepository. html (1998). www.ics.uci.edu/mlearn
4. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)
5. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd International Conference on Machine Learning, ICML 2006, pp. 161–168. ACM (2006)
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)

7. Díaz-Uriarte, R., Andres, S.A.D.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics **7**(1), 3 (2006)
8. Friedman, J.H., Fisher, N.I.: Bump hunting in high-dimensional data. Statistics and Computing **9**(2), 123–143 (1999)
9. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. The Annals of Applied Statistics, 916–954 (2008)
10. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science **286**(5439), 531–537 (1999)
11. Huysmans, J., Baesens, B., Vanthienen, J.: Using rule extraction to improve the comprehensibility of predictive models. DTEW-KBI_0612, 1–55 (2006)
12. Johansson, U., Sonstrod, C., Lofstrom, T.: One tree to explain them all. In: 2011 IEEE Congress on Evolutionary Computation (CEC), pp. 1444–1451. IEEE (2011)
13. Latinne, P., Debeir, O., Decaestecker, C.: Limiting the number of trees in random forests. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 178–187. Springer, Heidelberg (2001)
14. Liu, S., Patel, R.Y., Daga, P.R., Liu, H., Fu, G., Doerksen, R., Chen, Y., Wilkins, D.: Multi-class joint rule extraction and feature selection for biological data. In: 2011 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 476–481. IEEE (2011)
15. Liu, S., Patel, R.Y., Daga, P.R., Liu, H., Fu, G., Doerksen, R.J., Chen, Y., Wilkins, D.E.: Combined rule extraction and feature elimination in supervised classification. IEEE Transactions on NanoBioscience **11**(3), 228–236 (2012)
16. Martinez-Muoz, G., Hernández-Lobato, D., Suárez, A.: An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(2), 245–259 (2009)
17. Meinshausen, N.: Node harvest. The Annals of Applied Statistics, 2049–2072 (2010)
18. Näppi, J.J., Regge, D., Yoshida, H.: Comparative performance of random forest and support vector machine classifiers for detection of colorectal lesions in ct colonography. In: Yoshida, H., Sakas, G., Linguraru, M.G. (eds.) Abdominal Imaging. LNCS, vol. 7029, pp. 27–34. Springer, Heidelberg (2012)
19. Nutt, C.L., Mani, D.R., Betensky, R.A., Pablo Tamayo, J., Cairncross, G., Ladd, C., Pohl, U., Hartmann, C., McLaughlin, M.E., Batchelor, T.T., et al.: Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. Cancer Research **63**(7), 1602–1607 (2003)
20. Sarkar, B.K., Sana, S.S., Chaudhuri, K.: A genetic algorithm-based rule extraction system. Applied Soft Computing **12**(1), 238–254 (2012)
21. Selman, B., Gomes, C.P.: Hill-climbing search. Encyclopedia of Cognitive Science (2006)
22. Shi, T., Horvath, S.: Unsupervised learning with random forest predictors. Journal of Computational and Graphical Statistics **15**(1) (2006)
23. Song, L., Langfelder, P., Horvath, S.: Random generalized linear model: a highly accurate and interpretable ensemble predictor. BMC Bioinformatics **14**(1), 5 (2013)
24. Van Assche, A., Blockeel, H.: Seeing the forest through the trees: learning a comprehensible model from an ensemble. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 418–429. Springer, Heidelberg (2007)

25. Veer, L.J., Dai, H., Vijver, J.V.D., He, Y.D., Hart, A.A.M., Mao, M., Peterse, H.L., Kooy, K., Marton, M.J., Witteveen, A.T., et al.: Gene expression profiling predicts clinical outcome of breast cancer. Nature **415**(6871), 530–536 (2002)
26. Yang, F., Wei-hang, L., Luo, L., Li, T.: Margin optimization based pruning for random forest. Neurocomputing **94**, 54–63 (2012)
27. Zhang, H., Wang, M.: Search for the smallest random forest. Statistics and its Interface **2**(3), 381 (2009)
28. Zhou, Z.-H., Jiang, Y., Chen, S.-F.: Extracting symbolic rules from trained neural network ensembles. Ai Communications **16**(1), 3–15 (2003)