

# Gene Selection With Guided Regularized Random Forest

Houtao Deng

*Intuit, Mountain View, CA, USA*

George Runger

*Arizona State University, Tempe, AZ, USA*

---

## Abstract

The regularized random forest (RRF) was recently proposed for feature selection by building only one ensemble. In RRF the features are evaluated on a part of the training data at each tree node. We derive an upper bound for the number of distinct Gini information gain values in a node, and show that many features can share the same information gain at a node with a small number of instances and a large number of features. Therefore, in a node with a small number of instances, RRF is likely to select a feature not strongly relevant.

Here an enhanced RRF, referred to as the guided RRF (GRRF), is proposed. In GRRF, the importance scores from an ordinary random forest (RF) are used to guide the feature selection process in RRF. Experiments on 10 gene data sets show that the accuracy performance of GRRF is, in general, more robust than RRF when their parameters change. GRRF is computationally efficient, can select compact feature subsets, and has competitive accuracy performance, compared to RRF, varSelRF and LASSO logistic regression (with evaluations from an RF classifier). Also, RF applied to the features selected by RRF with the minimal regularization outperforms RF applied to all the features for most of the data sets considered here. Therefore, if accuracy is considered more important than the size of the feature

---

\*This research was partially supported by ONR grant N00014-09-1-0656.

*Email addresses:* `hdeng3@asu.com` (Houtao Deng), `george.runger@asu.edu` (George Runger)

subset, RRF with the minimal regularization may be considered. We use the accuracy performance of RF, a strong classifier, to evaluate feature selection methods, and illustrate that weak classifiers are less capable of capturing the information contained in a feature subset. Both RRF and GRRF were implemented in the “RRF” R package available at CRAN, the official R package archive.

*Keywords:* classification; feature selection; random forest; variable selection.

---

## 1. Introduction

Given a training data set consisting of  $N$  instances,  $P$  predictor variables/features  $X_i (i = 1, \dots, P)$  and the class  $Y \in \{1, 2, \dots, C\}$ , the objective of feature selection is to select a compact variable/feature subset without loss of predictive information about  $Y$ . Note feature selection selects a subset of the original feature set, and, therefore, may be more interpretable than feature extraction (e.g., principal component analysis [15] and partial least squares regression [8]) which creates new features based on transformations or combinations of the original feature set [13]. Feature selection has been widely used in many applications such as gene selection [23, 30, 18] as it can moderate the curse of dimensionality, improve interpretability [10] and avoid the effort to analyze irrelevant or redundant features.

Information-theoretic measures such as symmetrical uncertainty and mutual information can measure the degree of association between a pair of variables and have been successfully used for feature selection, e.g., CFS (correlation-based feature selection) [12] and FCBF (fast correlation-based filter) [29]. However, these measures are limited to two variables and do not capture high-order interactions between variables well. For example, the measures can not capture the exclusive OR relationship  $Y = XOR(X_1, X_2)$ , in which neither  $X_1$  nor  $X_2$  is predictive individually, but  $X_1$  and  $X_2$  together can correctly determine  $Y$  [14].

LASSO logistic regression [25] and recursive feature elimination with a linear SVM (SVM-RFE) [11] are well-known feature selection methods based on classifiers. These methods assume a linear relationship between the log odds of the class and the predictor variables (LASSO logistic regression) or between the class and the predictor variables (linear SVM). Furthermore, before using these methods, one often needs to preprocess the data such as

transforming categorical variables to binary variables or normalizing variables of different scales.

A random forest (RF)[3] classifier has been commonly used for measuring feature importance [17]. An RF naturally handles numerical and categorical variables, different scales, interactions and nonlinearities, etc. Although the RF feature importance scores can be used to select  $K$  features with the highest importance scores *individually*, there could be redundancy among the  $K$  features. Consequently, the selected features subset can differ from the best *combination* of  $K$ -features, that is, the best feature subset. Similarly, Boruta [22], a method based on RF, aims to select a set of relevant features, which is different from the objective of a relevant and also non-redundant feature subset.

A feature selection method based on RF, varSelRF[27], has become popular. varSelRF consists of multiple iterations and eliminates the feature(s) with the least importance score(s) at each iteration. Since eliminating one feature at each iteration is computationally expensive, the authors considered eliminating a fraction, e.g.,  $1/5$ , of the features at each iteration. However, when there is a large number of features, many features are eliminated at each iteration, and, thus, useful features, but with small importance scores, can be eliminated.

The ACE algorithm [26] is another ensembles-based feature selection method. It was shown to be effective, but it is more computationally demanding than the simpler approaches considered here. It requires multiple forests to be constructed, along with multiple gradient boosted trees [6].

Recently, the regularized random forest (RRF) was proposed for feature selection with one ensemble [4], instead of multiple ensembles [27, 26]. However, in RRF the features are evaluated on a part of the training data at each tree node and the feature selection process may be greedy.

Here we analyze a feature evaluation issue that occurs in all the usual splitting algorithms at tree nodes with a small number of training instances. To solve this issue, we propose the guided RRF (GRRF) method, in which the importance scores from an ordinary RF are used to guide the feature selection process in RRF. Since the importance scores from an RF are aggregated from all the trees based on all the training data, GRRF is expected to perform better than RRF.

Section 2 presents previous work. Section 3 discusses the node sparsity issue when evaluating features at tree nodes with a small number of training instances. Section 4 describes the GRRF method. Section 5 presents and

discusses the experimental results. Section 6 concludes this work.

## 2. Background

### 2.1. Variable importance scores from Random Forest

A random forest (RF) [3] is a supervised learner that consists of multiple decision trees, each of which grown on a bootstrap sample from the original training data. The Gini index at node  $v$ ,  $Gini(v)$ , is defined as

$$Gini(v) = \sum_{c=1}^C \hat{p}_c^v (1 - \hat{p}_c^v)$$

where  $\hat{p}_c^v$  is the proportion of class- $c$  observations at node  $v$ . The Gini information gain of  $X_i$  for splitting node  $v$ ,  $Gain(X_i, v)$ , is the difference between the impurity at the node  $v$  and the weighted average of impurities at each child node of  $v$ . That is,

$$Gain(X_i, v) = Gini(X_i, v) - w_L Gini(X_i, v^L) - w_R Gini(X_i, v^R)$$

where  $v^L$  and  $v^R$  are the left and right child nodes of  $v$ , respectively, and  $w_L$  and  $w_R$  are the proportions of instances assigned to the left and right child nodes. At each node, a random set of  $mtry$  features out of  $P$  is evaluated, and the feature with the maximum  $Gain(X_i, v)$  is used for splitting the node  $v$ .

The importance score for variable  $X_i$  can be calculated as

$$Imp_i = \frac{1}{ntree} \sum_{v \in S_{X_i}} Gain(X_i, v) \quad (1)$$

where  $S_{X_i}$  is the set of nodes split by  $X_i$  in the RF with  $ntree$  trees. The RF importance scores are commonly used to evaluate the contributions of the features regarding predicting the classes.

### 2.2. Regularized Random Forest

The regularized random forest (RRF) [4] applies the tree regularization framework to RF and can select a compact feature subset. While RRF is

---

**Algorithm 1:** Feature selection at node  $v$ .

---

```

input :  $F$  and  $\lambda$ 
output:  $F$ 
1  $gain^* \leftarrow 0, count \leftarrow 0, f^* \leftarrow -1, f_1 \leftarrow \{1, 2, 3, \dots, P\}, f_2 \leftarrow \emptyset$ 
2 while  $f_1 \neq \emptyset$  do
3    $m \leftarrow select(f_1)$  //randomly select feature index  $m$  from  $f_1$ 
4    $f_2 \leftarrow \{f_2, m\}$  //add  $m$  to  $f_2$ 
5    $f_1 \leftarrow f_1 - m$  //remove  $m$  from  $f_1$ 
6    $gain_R(X_m, v) \leftarrow 0$ 
7   if  $X_m \in F$  then
8      $gain_R(X_m, v) \leftarrow gain(X_m, v)$  //calculate  $gain_R$  for all variables in  $F$ 
9   end
10  if  $X_m \notin F$  and  $count < \lceil \sqrt{P} \rceil$  then
11     $gain_R(X_m) \leftarrow \lambda \cdot gain(X_m)$  //regularize the gain if the variable is not in  $F$ 
12     $count \leftarrow count + 1$ 
13  end
14  if  $gain_R(X_m, v) > gain^*$  then
15     $gain^* \leftarrow gain_R(X_m, v), f^* \leftarrow m$ 
16  end
17 if  $f^* \neq -1$  and  $f^* \notin F$  then
18    $F \leftarrow \{F, f^*\}$ 
19 end
20 return  $F$ 

```

---

built in a way similar to RF, the main difference is that the regularized information gain,  $Gain_R(X_i, v)$ , is used in RRF

$$Gain_R(X_i, v) = \begin{cases} \lambda \cdot Gain(X_i, v) & i \notin F \\ Gain(X_i, v) & i \in F \end{cases} \quad (2)$$

where  $F$  is the set of indices of features used for splitting in previous nodes and is an empty set at the root node in the first tree. Here  $\lambda \in (0, 1]$  is called the penalty coefficient. When  $i \notin F$ , the coefficient penalizes the  $i^{th}$  feature for splitting node  $v$ . A smaller  $\lambda$  leads to a larger penalty. RRF uses  $Gain_R(X_i, v)$  at each node, and adds the index of a new feature to  $F$  if the feature adds enough predictive information to the selected features.

RRF with  $\lambda = 1$ , referred to as RRF(1), has the minimum regularization. Still, a new feature has to be more informative at a node than the features already selected to enter the feature subset. The feature subset selected by RRF(1) is called the *least regularized subset*, indicating the minimal regularization from RRF.

Figure 1 illustrates the feature selection process. The nodes in the forest are visited sequentially (from the left to the right and from the top to the

bottom). The indices of three distinct features used for splitting are added to  $F$  in Tree 1, and the index of  $X_5$  is added to  $F$  in Tree 2. Algorithm 1 shows the feature selection process at each node.

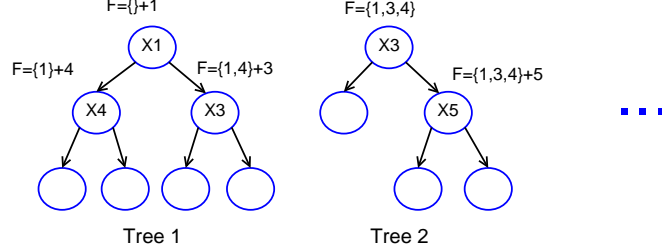


Figure 1: The feature selection procedure of RRF. The non-leaf nodes are marked with the splitting variables. Three distinct features used for splitting are added to  $F$  in Tree 1, and one feature  $X_5$  is added to  $F$  in Tree 2.

Similar to  $Gain_R(\cdot)$ , a penalized information gain was used to suppress spurious interaction effects in the rules extracted from tree models [7]. The objective of Friedman and Popescu [7] was different from the goal of a compact feature subset here. Also, the regularization used in Friedman and Popescu [7] only reduces the redundancy in each path from the root node to a leaf node, but the features extracted from such tree models can still be redundant.

### 3. The Node Sparsity Issue

This section discusses an issue of evaluating features in a tree node with a small number of instances, referred to as the *node sparsity issue* here. The number of instances decreases as the instances are split recursively in a tree, and, therefore, this issue can commonly occur in tree-based models.

In a tree node, the Gini information gain or other information-theoretic measures are commonly used to evaluate and compare different features. However, the measures calculated from a node with a small number of instances may not be able to distinguish the features with different predictive information. In the following we establish an upper bound for the number of distinct values of Gini information gain for binary classification.

Let  $D(f)$  denote the number of distinct values of a function  $f$  (defined over a specified range),  $N$  denote the number of instances at a non-leaf node  $v$  and  $N \geq 2$  (otherwise it cannot be split). For simplicity, in the following

we also assume  $N$  is even. The following procedure is similar when  $N$  is odd. Let  $N_1$  and  $N_2$  be the number of instances of class 1 and class 2, respectively, at the node. Let  $L$  denote the number of instances at the left child node, and let  $L_1$  and  $L_2$  denote the number of instances for class 1 and class 2 at the left child node, respectively, with similar notations  $R$ ,  $R_1$  and  $R_2$  for the right child node. Note  $L \geq 1$  and  $R \geq 1$ . The notation  $L_1L_2$  denotes the product function where  $L_1$  and  $L_2$  assume values in the feasible domain  $0 \leq L_1, L_2 \leq L$  and  $L_1 + L_2 = L$ . Then we have the following lemmas and theorem.

**Lemma 1.** *An upper bound of the number of distinct values of  $L_1L_2$  is  $\lceil (L+1)/2 \rceil$ . That is,  $D(L_1L_2) \leq \lceil (L+1)/2 \rceil$ , where  $\lceil \cdot \rceil$  denotes the ceiling.*

*Proof.*  $L_1$  has at most  $L+1$  values ( $L_1 \in \{0, 1, \dots, L\}$ ). Now let  $l_1l_2$  and  $m_1m_2$  be two realizations of  $L_1L_2$ , and let  $l_1l_2 = m_1m_2$ , we have

$$\begin{aligned} & (L - l_1)l_1 = (L - m_1)m_1 \\ \Leftrightarrow & Ll_1 - l_1^2 - Lm_1 + m_1^2 = 0 \\ \Leftrightarrow & L(l_1 - m_1) + (m_1 + l_1)(m_1 - l_1) = 0 \\ \Leftrightarrow & (l_1 - m_1)(L - l_1 - m_1) = 0 \\ \Leftrightarrow & l_1 = m_1 \text{ or } l_1 = L - m_1 \end{aligned}$$

Therefore, we obtain an upper bound for the number of distinct values for  $L_1L_2$  as  $D(L_1L_2) \leq \lceil (L+1)/2 \rceil$ .  $\square$

**Lemma 2.**  $D(L_1L_2/L) \leq N(N+2)/4 - 1$ .

*Proof.* For each  $L$ ,  $L_1L_2/L$  has at most  $D(L_1L_2)$  distinct values. Because  $1 \leq L \leq N-1$ , an upper bound for  $D(L_1L_2/L)$  is derived as follows

$$D(L_1L_2/L) \leq \lceil 2/2 \rceil + \lceil 3/2 \rceil + \dots + \lceil N/2 \rceil = N(N+2)/4 - 1$$

$\square$

**Lemma 3.**  $D(L_1L_2/L + R_1R_2/R) \leq N(N+2)/4 - 1$

*Proof.* Because  $L_1L_2/L$  and  $R_1R_2/R$  are symmetric, the upper bound is the same as for one term. That is,  $D(L_1L_2/L + R_1R_2/R) \leq N(N+2)/4 - 1$   $\square$

**Theorem 1.** *For binary classification, an upper bound of the number of distinct information gain values is  $N(N+2)/4 - 1$ .*

*Proof.* The information gain for splitting node  $v$  is

$$Gain(v) = Gini(v) - w_L Gini(v^L) - w_R Gini(v^R)$$

Because  $Gini(v)$  is a constant at node  $v$ , we only need to consider  $D(w_L Gini(v^L) + w_R Gini(v^R))$ . For two classes

$$\begin{aligned} & w_L Gini(v^L) \\ &= (L/N)(L_1/L \cdot L_2/L + L_2/L \cdot L_1/L) \\ &= 2(L/N)(L_1 L_2)/L^2 \end{aligned}$$

Then

$$\begin{aligned} & w_L Gini(v^L) + w_R Gini(v^R) \\ &= 2(L/N)(L_1 L_2)/L^2 + 2(R/N)(R_1 R_2)/R^2 \\ &= (2/N)(L_1 L_2/L) + (2/N)(R_1 R_2/R) \end{aligned}$$

Because  $N$  is a constant, we have  $D(w_L Gini(v^L) + w_R Gini(v^R)) = D(L_1 L_2/L + R_1 R_2/R)$ . According to Lemma 3,  $D(L_1 L_2/L + R_1 R_2/R) \leq N(N+2)/4 - 1$ .  $\square$

Note similar conclusions may be applied to other information-theoretic measures such as the regularized information gain in Equation 2.

Consequently, when  $N$  is small, the number of distinct Gini information gain values is small. For a large number of variables, many could have the same Gini information gain. For example, at a node with 10 instances there are at most 29 distinct Gini information gain values for binary classification problems. For 1000 genes with two classes, there are at least  $1000 - 29 = 971$  genes having the same information gain as other genes. The number of instances can be even smaller than 10 in a node of an RF or RRF as each tree is grown completely.

In RRF, a feature with the maximum regularized information gain is added at a node based on only the instances at that node. When multiple features have the maximum regularized information gain, one of these features is randomly selected. As discussed above, at a node with only a small number of instances less relevant, or redundant features may be selected. An additional metric is useful to help distinguish the features. In the following section, we introduce the guided regularized random forest that leverages the importance scores calculated from an RF based on all the training data.



#### 4. Guided Regularized Random Forest

The guided RRF (GRRF) uses the importance scores from a preliminary RF to guide the feature selection process of RRF. Because the importance scores from an RF are aggregated from all the trees of the RF based on all the training data, GRRF may be able to handle the node sparsity issue discussed in the previous section.

A normalized importance score is defined as

$$Imp'_i = \frac{Imp_i}{\max_{j=1}^P Imp_j} \quad (3)$$

where  $Imp_i$  is the importance score from an RF (defined in Equation 1). Here  $0 \leq Imp'_i \leq 1$ . Instead of assigning the same penalty coefficient to all features in RRF, GRRF assigns a penalty coefficient to each feature. That is,

$$Gain_R(X_i, v) = \begin{cases} \lambda_i Gain(X_i, v) & X_i \notin F \\ Gain(X_i, v) & X_i \in F \end{cases} \quad (4)$$

where  $\lambda_i \in (0, 1]$  is the coefficient for  $X_i$  ( $i \in \{1, \dots, P\}$ ) and is calculated based on the importance score of  $X_i$  from an ordinary RF. That is,

$$\lambda_i = (1 - \gamma)\lambda_0 + \gamma Imp'_i \quad (5)$$

where  $\lambda_0 \in (0, 1]$  controls the degree of regularization and is called the base coefficient, and  $\gamma \in [0, 1]$  controls the weight of the normalized importance score and is called the importance coefficient. Note that RRF is a special case of GRRF with  $\gamma = 0$ . Given  $\lambda_0$  and  $\gamma$ , a feature with a larger importance score has a larger  $\lambda_i$ , and, therefore, is penalized less.

In our experiments we found the feature subset size can be effectively controlled by changing either  $\lambda_0$  or  $\gamma$ , but changing the latter often leads to better performance in terms of classification accuracy. To reduce the number of parameters of GRRF, we fix  $\lambda_0$  to be 1 and consider  $\gamma$  as the only parameter for GRRF. With  $\lambda_0 = 1$ , we have

$$\lambda_i = (1 - \gamma) + \gamma Imp'_i = 1 - \gamma(1 - Imp'_i) \quad (6)$$

For a feature  $X_i$  that does not have the maximum importance score ( $Imp'_i \neq 1$ ), a larger  $\gamma$  leads to a smaller  $\lambda_i$  and, thus, a larger penalty on  $Gain(X_i, v)$  when  $X_i$  has not been used in the nodes prior to node  $v$ . Consequently,  $\gamma$  is essentially the degree of regularization. Furthermore, GRRF with  $\gamma = 0$  is equivalent to RRF(1), with the minimal regularization.

## 5. Experiments

We implemented the RRF and GRRF algorithms in the “RRF” R package based on the “randomForest” package [17]. The “RRF” package is available from CRAN (<http://cran.r-project.org/>), the official R package archive.

Work by [27] showed that the accuracy performance of RF is largely independent of the number of trees (between 1000 and 40000 trees), and  $mtry = \sqrt{P}$  is often a reasonable choice. Therefore, we used  $ntree = 1000$  and  $mtry = \sqrt{P}$  for the classifier RF and the feature selection methods RRF, GRRF and varSelRF. Guided by initial experiments, randomly sampling 63% of the data instances (the same as the default setting in the “randomForest” package) without replacement was used in RRF and GRRF. We also evaluated two well-known feature selection methods: varSelRF available in the “varSelRF” R package and LASSO logistic regression available in the “glmnet” R package. Unless otherwise specified, the default values in the packages were used. Also note  $\lambda$  is the parameter of RRF, and as discussed,  $\gamma$  is considered as the only parameter of GRRF here.

Table 1: The number of groups identified, and the number of irrelevant or redundant features selected for different algorithms.

Method	# Groups				# Irrelevant or Redundant Features			
	LASSO	varSelRF	RRF	GRRF	LASSO	varSelRF	RRF	GRRF
Average	4	4.95	5	4.95	4.3	4.95	4.95	0.75
Std. err	0.000	0.050	0.000	0.050	0.147	0.050	0.050	0.123

### 5.1. Simulated Data Sets

We start with a simulated data set generated by the following procedure. First generate 10 independent variables:  $X_1, \dots, X_{10}$ , each is uniformly distributed in the interval  $[0,1]$ . Variable  $Y$  is calculated by the formula

$$Y = 10 \sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + e$$

where  $e$  follows a standard normal distribution. Note the above data generation procedure was described in [5]. We simulated 1000 instances and then calculated the median of  $Y$  as  $\bar{y}$ . We then labeled class 2 to the instances with  $Y > \bar{y}$ , and class 1 otherwise, so that it becomes a classification problem. Furthermore, we added five other variables with  $X_{11}=X_1$ ,  $X_{12}=X_2$ ,  $X_{13}=X_3$ ,  $X_{14}=X_4$ ,  $X_{15}=X_5$ . Consequently, the feature selection solution is  $\{(X_1|X_{11}) \ \& \ (X_2|X_{12}) \ \& \ (X_3|X_{13}) \ \& \ (X_4|X_{14}) \ \& \ (X_5|X_{15})\}$ ,

Table 2: Summary of the data sets.

Data set	Reference	# Examples	# Features	# classes
adenocarcinoma	[20]	76	9868	2
brain	[19]	42	5597	5
breast.2.class	[28]	77	4869	2
breast.3.class	[28]	95	4869	3
colon	[2]	62	2000	2
leukemia	[9]	38	3051	2
lymphoma	[1]	62	4026	3
nci60	[21]	61	5244	8
prostate	[24]	102	6033	2
srbct	[16]	63	2308	4

where  $\&$  stands for “and” and  $(X_1|X_{11})$  indicates that one and only one from this group should be selected. For example, for the data set considered here, both  $(X_1, X_2, X_3, X_4, X_5)$  and  $(X_{11}, X_2, X_{13}, X_4, X_5)$  are correct solutions. However,  $(X_1, X_3, X_4, X_5)$  misses the group  $(X_2|X_{12})$ . Furthermore,  $(X_1, X_2, X_3, X_4, X_5, X_{11})$  has a redundant variable because  $X_{11}$  is redundant to  $X_1$ .

We simulated 20 replicates of the above data set, and then applied GRRF, RRF and two well-known methods varSelRF and LASSO logistic regression to the data sets. Here  $\gamma$  of GRRF was selected from  $\{0.4, 0.5, 0.6\}$ ,  $\lambda$  of RRF was selected from  $\{0.6, 0.7, 0.8\}$ , and the regularization parameter of LASSO logistic regression was selected from  $\{0.01, 0.02, 0.03, 0.05, 0.1, 0.2\}$ , all by 10-fold CV.

The results are shown in Table 1. LASSO logistic regression identifies the least number of groups on average (4). The methods varSelRF, RRF and GRRF identify almost all the groups, but varSelRF and RRF select 4.95, on average, irrelevant or redundant variables, and GRRF only selects 0.75 irrelevant or redundant variables. This experiment shows GRRF’s potential in selecting a relevant and non-redundant feature subset.

### 5.2. Gene Data Sets

The 10 gene expression data sets analyzed by Uriarte and de Andres [27] are considered in this section. The data sets are summarized in Table 2. For each data set, a feature selection algorithm was applied to 2/3 of the instances (selected randomly) to select a feature subset. Then a classifier was applied to the feature subset. The error rate is obtained by applying the classifier to the other 1/3 of the instances. This procedure was conducted 100 times with different random seeds, and the average size of feature subsets and the average error rate over 100 runs were calculated. In the experiments, we considered

Table 3: The number of features selected in the least regularized subset selected by RRF (i.e., RRF(1)), the total number of original features (“All”), and the error rates of RF applied to the least regularized subset and all the features. The win-lose-tie results of “All” compared to RRF(1) are shown. Here “o” or “•” represents a significant difference at the 0.05 level, according to the paired t-test. RRF(1) uses many fewer features than the original features, and wins on 7 data sets. There are significant differences for four data sets.

	Number of features		Average error rates		
	RRF(1)	All	RRF(1)-RF	All-RF	
adenocarcinoma	86	9868	0.158	0.159	
brain	97	5597	0.159	0.170	
breast.2.class	210	4869	0.352	0.371	o
breast.3.class	253	4869	0.397	0.415	o
colon	92	2000	0.162	0.158	
leukemia	24	3051	0.053	0.064	
lymphoma	31	4026	0.018	0.006	•
nci	197	5244	0.332	0.321	
prostate	88	6033	0.082	0.109	o
srbct	51	2308	0.027	0.032	
win-lose-tie	-	-	-	3-7-0	

Table 4: The total number of original features (“All”) and the average number of features (from 100 replicates for each data set) selected by different methods. All feature selection methods are able to select a small number of features. Here GRRF(0.1) and RRF(0.9) select a similar number of features, but it is shown later that GRRF(0.1) is more accurate than RRF(0.9) (with an RF classifier) for most data sets.

	All	GRRF(0.1)	GRRF(0.2)	RRF(0.9)	varSelRF	LASSO
adenocarcinoma	9868	20	15	23	4	2
brain	5597	22	12	27	28	24
breast.2.class	4869	59	25	60	7	7
breast.3.class	4869	77	31	78	12	18
colon	2000	29	13	27	4	7
leukemia	3051	6	4	6	2	8
lymphoma	4026	5	4	4	81	25
nci	5244	63	26	61	60	53
prostate	6033	18	13	19	6	12
srbct	2308	13	9	14	34	28

RRF, GRRF, varSelRF and LASSO logistic regression as the feature selection algorithms, and random forest (RF) and C4.5 as the classifiers.

### 5.2.1. Feature selection and classification

First compare the accuracy of RF applied to all the features, denoted as “All”, and the least regularized subset (i.e., features selected by RRF(1)). The number of features and the error rates are shown in Table 3. The win-lose-tie results of “All” compared to RRF(1) are also shown in the tables. To investigate the statistical significance of these results, we applied the paired

Table 5: The error rates of RF applied to the feature subsets selected by GRRF(0.1), GRRF(0.2), RRF(0.9), varSelRF and LASSO logistic regression, respectively (to three decimal places). The win-lose-tie results of each competitor compared to GRRF(0.1) with RF are shown. Here “o” or “•” represents a significant difference between a method and GRRF(0.1) with RF at the 0.05 level, according to the paired t-test. Here GRRF(0.1) leads to competitive accuracy performance, compared to GRRF(0.2), RRF(0.9), LASSO and varSelRF.

	GRRF(0.1) -RF	GRRF(0.2) -RF	RRF(0.9) -RF	varSelRF -RF	LASSO -RF
adenocarcinoma	0.169	0.168	0.160	0.212	0.189
brain	0.214	0.259	0.234	0.231	0.259
breast.2.class	0.345	0.359	0.367	0.386	0.366
breast.3.class	0.387	0.403	0.410	0.418	0.400
colon	0.175	0.186	0.190	0.232	0.180
leukemia	0.080	0.093	0.091	0.107	0.076
lymphoma	0.067	0.076	0.098	0.022	0.009
nci	0.389	0.452	0.405	0.418	0.396
prostate	0.085	0.085	0.101	0.085	0.088
srbc	0.064	0.072	0.074	0.035	0.007
win-lose-tie	-	1-9-0	1-9-0	3-7-0	3-7-0

Table 6: The error rates of C4.5 applied to the feature subsets selected by GRRF(0.1), GRRF(0.2), RRF(0.9), varSelRF and LASSO logistic regression, respectively. The win-lose-tie results of each competitor compared to GRRF(0.1) with C4.5 are calculated. Here “o” or “•” represents a significant difference between a method and GRRF(0.1) with C4.5 at the 0.05 level, according to the paired t-test. The GRRF methods and the other methods have perform similarly in terms of the accuracy with C4.5. As expected, C4.5 has noticeably higher error rates than RF, shown in Table 5.

	GRRF(0.1) -C4.5	GRRF(0.2) -C4.5	RRF(0.9) -C4.5	varSelRF -C4.5	LASSO -C4.5
adenocarcinoma	0.260	0.251	0.241	0.248	0.218
brain	0.438	0.428	0.418	0.394	0.476
breast.2.class	0.402	0.411	0.412	0.413	0.391
breast.3.class	0.497	0.494	0.491	0.481	0.482
colon	0.274	0.262	0.298	0.273	0.242
leukemia	0.133	0.157	0.138	0.146	0.166
lymphoma	0.126	0.125	0.126	0.152	0.138
nci	0.626	0.636	0.658	0.641	0.646
prostate	0.157	0.138	0.154	0.115	0.144
srbc	0.235	0.208	0.207	0.209	0.194
win-lose-tie	-	7-3-0	5-5-0	6-4-0	6-4-0

t-test to the error rates of the two methods from 100 replicates for each data set. The data sets with a significant difference at the 0.05 level are marked with “o” or “•” in the table.

The least regularized subset not only has many fewer features than “All”, but also leads to better accuracy performance on 7 data sets out of 10, and the error rates are significantly different on 4 data sets. Therefore, RRF(1) not only improves interpretability by reducing the number of features, but also can improve the accuracy performance of classification, even for RF, considered as a strong classifier capable of handling irrelevant and relevant variables [4]. It should also be noted that although the least regularized subset is much smaller than the original feature set, the size may still be considered large in some cases, e.g., more than 200 features for the breast.2.class data set. GRRF and RRF with larger regularization, investigated in the following experiments, are able to further reduce the number of features.

Next compare GRRF to RRF and two well-known methods: varSelRF and LASSO logistic regression. The regularization parameter of LASSO logistic regression was selected from  $\{0.01, 0.02, 0.03, 0.05, 0.1, 0.2\}$  by 10-fold CV. Here  $\gamma \in \{0.1, 0.2\}$  was used for GRRF (i.e., GRRF(0.1) or GRRF(0.2)), and  $\lambda = 0.9$  was used for RRF (i.e., RRF(0.9)). We used a fixed parameter setting for GRRF or RRF, as the parameter sensitivity analysis in the following section shows a consistent trend that GRRF or RRF tends to select more features and also tends to be more accurate, for a smaller  $\gamma$  or a larger  $\lambda$ . We chose these parameters so that a reasonably small number of features can be selected. One can also use cross-validation error to determine an appropriate parameter value customized for each data set and potentially improve these results.

The total number of original features and the average number of features selected by each feature selection method are shown in Table 4. All the feature selection methods are able to select a small number of features.

The average error rates of RF applied to all the features (“All”) and the subsets selected by different feature selection methods are shown in Table 5. GRRF(0.1) with RF outperforms GRRF(0.2) with RF on 9 data sets out of 10, 5 of which have significant differences at the 0.05 level. Even though GRRF(0.1) selects more features than GRRF(0.2), the sizes of the feature subsets are reasonably small (all less than 80 features).

GRRF(0.1) and RRF(0.9) select a similar number of features. However, GRRF(0.1) with RF outperforms RRF(0.9) with RF on 9 data sets, 5 of which have significant differences at the 0.05 level. Consequently, GRRF

selects stronger features than RRF, which is consistent with the simulated experiments. According to the discussion in Section 3, a feature with the maximum Gini information gain in a node with a small number instances may not be truly strong. Yet RRF adds this feature to the subset.

GRRF(0.1) with RF outperforms varSelRF with RF on 7 data sets, 6 of which have significant differences. Therefore, GRRF(0.1) may be more favorable than varSelRF for the data sets considered here. Also, as shown in the following section, GRRF has a clear advantage over varSelRF in terms of computational time. Furthermore, GRRF(0.1) with RF outperforms LASSO logistic regression with RF on 7 data sets, 3 of which have significant differences. The accuracy performance may be improved by applying a logistic regression model to the features selected by LASSO logistic regression. However, tree models like GRRF have a few desirable properties compared to LASSO logistic regression: they can naturally handle mixed categorical and numerical features, and multiple classes, etc.

The average error rates of C4.5 applied to all the features (“All”) and the subsets selected by different feature selection methods are shown in Table 6. It can be seen that the error rates of C4.5 are clearly higher than RF shown in Table 5. Indeed, RF has been considered as a stronger classifier than C4.5. Interestingly, the differences between the methods in terms of C4.5 are smaller than the RF results. As mentioned by [4], a relatively weaker classifier is less capable of capturing information from data than a stronger classifier. Consequently, a feature subset that includes strong features, but misses the features of small contributions may not affect the accuracy of C4.5 much, but can affect the accuracy of RF. A weak classifier should only be used for evaluating a feature subset if that classifier is actually used for classification after feature selection. However, if a strong classifier is used for classification after feature selection, or the objective is to evaluate the information contained in the feature subset, a strong classifier should be considered [4].

### 5.2.2. *Parameter Sensitivity and Computational Time*

We investigated the performance of RRF and GRRF with different parameter settings:  $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$  for RRF and  $\gamma \in \{0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0\}$  for GRRF. The parameter values are arranged by the degree of regularization in a decreasing order for both methods.

The sizes of the feature subsets, averaged over 100 replicates, for each

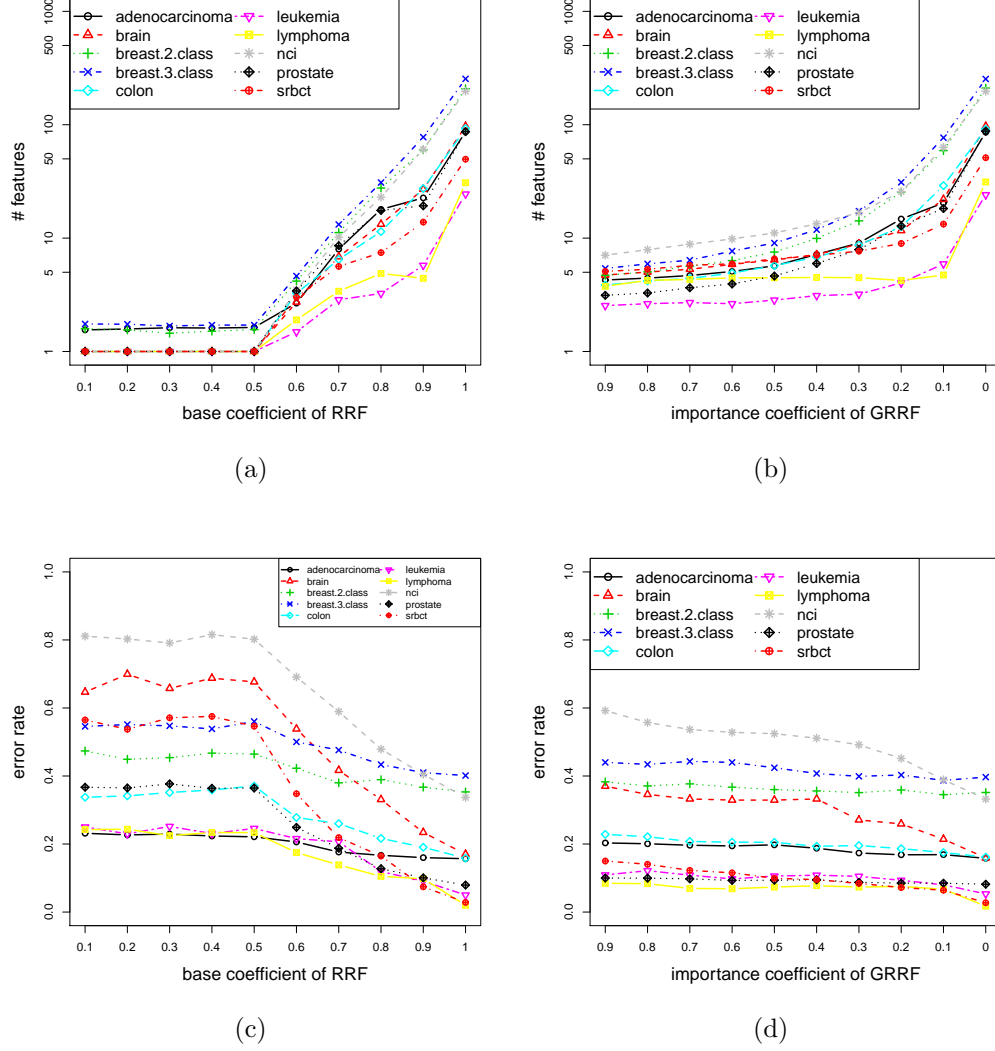


Figure 2: The performance of RRF for selected values of  $\lambda$  and GRRF for selected values of  $\gamma$ . Figure 2(a) shows the number of features selected by RRF. A smaller  $\lambda$  leads to fewer features. Figure 2(b) shows the number of features selected by GRRF. A larger  $\gamma$  leads to fewer features. Figure 2(c) shows the error rates of RF applied to the feature subsets selected by RRF for different  $\lambda$ . The error rates tend to decrease as  $\lambda$  increases. Figure 2(d) shows the error rates of RF applied to the feature subsets selected by GRRF for different  $\gamma$ . The error rates tend to decrease as  $\gamma$  decreases, but are reasonably robust.



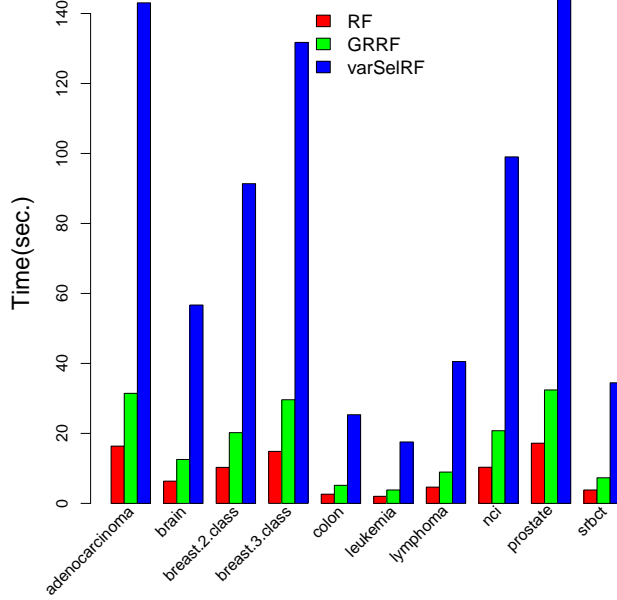


Figure 3: Computational time of RF, GRRF and varSelRF. GRRF is more computationally efficient than varSelRF. GRRF takes about twice as much as RF as it builds two ensembles.

parameter setting of RRF and GRRF, are shown in Figures 2(a) and 2(b), respectively. The number of features tends to increase as  $\lambda$  increases for RRF, or as  $\gamma$  decreases for GRRF. The consistent trend illustrates that one can control the size of the feature subset by adjusting the parameters.

The error rates, averaged over 100 replicates, for each parameter setting of RRF and GRRF are shown in Figures 2(c) and 2(d), respectively. In general, the error rates tend to decrease as  $\lambda$  increases for RRF, or as  $\gamma$  decreases for GRRF. However, for most data sets, the error rates of GRRF seem to be reasonably robust to the changes of  $\gamma$ . As mentioned, RRF(1) and GRRF(0) are equivalent, and, therefore, they have similar number of features and error rates for every data set (differing only by random selections).

The computational time of the RF-based methods: RF, GRRF and varSelRF is shown in Figure 3. As expected, RF was fast for training on these data sets. GRRF builds two ensemble models, and, thus, the computational time of GRRF is only about twice as much as RF. However, varSelRF needs to

build multiple ensemble models, and the computational advantage of GRRF over varSelRF is clear.

## 6. Conclusions

We derive an upper bound for the number of distinct Gini information gain values in a tree node for binary classification problems. The upper bound indicates that the Gini information gain may not be able to distinguish features at nodes with a small number of instances, which poses a challenge for RRF that selects a feature only using the instances at a node. Motivated by this node sparsity issue, we propose an enhanced method called the guided, regularized random forest (GRRF), in which a preliminary random forest is used to generate the initial variable importance scores to guide the regularized feature selection process of RRF. The importance scores from the preliminary RF help GRRF select better features when many features share the same maximal Gini information gain at a node. For the experiments here, GRRF is more favorable than RRF, computationally efficient, selects a small set of features, and has competitive accuracy performance.

Feature selection eliminates irrelevant or redundant features, but also may eliminate features of small importance. This may not affect the performance of a weak classifier which is less capable of capturing small information, but may affect the performance of a strong classifier such as RF [4]. Still, we found that the least regularized subset selected by RRF with the minimal regularization produces better accuracy performance than the complete feature set.

Finally we note that although RRF and GRRF can be used as classifiers, they are designed for feature selection. The trees in RRF and GRRF are not built independently as the features selected in previous trees have an impact on the trees built later. Therefore, as a classifier, RRF or GRRF may have a higher variance than RF because the trees are correlated. Consequently, in this work we applied RF on the feature subset selected by GRRF or RRF for classification.

## References

- [1] A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L. Yang, G.E.

- Marti, T. Moore, J. Hudson, L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Greiner, D.D. Weisenburger, J.O. Armitage, R. Warnke, R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botstein, P.O. Brown, L.M. Staudt, Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling., *Nature* 403 (2000) 503–511.
- [2] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine, 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 of the United States of America* 96 (1999) 6745–6750.
  - [3] L. Breiman, Random forests, *Machine Learning* 45 (2001) 5–32.
  - [4] H. Deng, G.C. Runger, Feature selection via regularized trees, in: *The 2012 International Joint Conference on Neural Networks (IJCNN)*, IEEE.
  - [5] J.H. Friedman, Multivariate adaptive regression splines, *The annals of statistics* (1991) 1–67.
  - [6] J.H. Friedman, Greedy function approximation: A gradient boosting machine, *Annals of Statistics* 29 (2001) 1189–1232.
  - [7] J.H. Friedman, B.E. Popescu, Predictive learning via rule ensembles, *Annals of Applied Statistics* 2 (2008) 916–954.
  - [8] P. Geladi, B. Kowalski, Partial least-squares regression: a tutorial, *Analytica chimica acta* 185 (1986) 1–17.
  - [9] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring., *Science* 286 (1999) 531–537.
  - [10] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, *Journal of Machine Learning Research* 3 (2003) 1157–1182.

- [11] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, *Machine Learning* 46 (2002) 389–422.
- [12] M.A. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: *Proceedings of the 17th International Conference on Machine Learning*, pp. 359–366.
- [13] A. Jain, D. Zongker, Feature selection: Evaluation, application, and small sample performance, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19 (1997) 153–158.
- [14] A. Jakulin, I. Bratko, Analyzing attribute dependencies, *Knowledge Discovery in Databases: PKDD 2003* (2003) 229–240.
- [15] I. Jolliffe, *MyiLibrary, Principal component analysis*, volume 2, Wiley Online Library, 2002.
- [16] J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, P.S. Meltzer, Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, *Nature Medicine* 7 (2001) 673–679.
- [17] A. Liaw, M. Wiener, Classification and regression by randomforest, *R News* 2 (2002) 18–22.
- [18] H. Liu, L. Liu, H. Zhang, Ensemble gene selection for cancer classification, *Pattern Recognition* 43 (2010) 2763–2772.
- [19] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J.Y.H. Kim, L.C. Goumnerova, P.M. Black, C. Lau, J.C. Allen, D. Zagzag, J.M. Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D.N. Louis, J.P. Mesirov, E.S. Lander, T.R. Golub, Prediction of central nervous system embryonal tumour outcome based on gene expression., *Nature* 415 (2002) 436–442.
- [20] S. Ramaswamy, K.N. Ross, E.S. Lander, T.R. Golub, A molecular signature of metastasis in primary solid tumors., *Nature genetics* 33 (2003) 49–54.

- [21] D.T. Ross, U. Scherf, M.B. Eisen, C.M. Perou, C. Rees, P. Spellman, V. Iyer, S.S. Jeffrey, M.V. de Rijn, M. Waltham, A. Pergamenschikov, J.C. Lee, D. Lashkari, D. Shalon, T.G. Myers, J.N. Weinstein, D. Botstein, P.O. Brown, Systematic variation in gene expression patterns in human cancer cell lines, *Nature Genetics* 24 (2000) 227–35.
- [22] W. Rudnicki, M. Kurska, Feature selection with the boruta package, *Journal of Statistical Software* 36 (2010).
- [23] R. Ruiz, J. Riquelme, J. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray data for cancer classification, *Pattern Recognition* 39 (2006) 2383–2392.
- [24] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, P. Tamayo, A.A. Renshaw, A.V. D’Amico, J.P. Richie, E.S. Lander, M. Loda, P.W. Kantoff, T.R. Golub, W.R. Sellers, Gene expression correlates of clinical prostate cancer behavior., *Cancer Cell* 1 (2002) 203–209.
- [25] R. Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society. Series B (Methodological)* (1996) 267–288.
- [26] E. Tuv, A. Borisov, G. Runger, K. Torkkola, Feature selection with ensembles, artificial variables, and redundancy elimination, *Journal of Machine Learning Research* 10 (2009) 1341–1366.
- [27] R.D. Uriarte, S.A. de Andres, Gene selection and classification of microarray data using random forest, *BMC Bioinformatics* 7 (2006) 3+.
- [28] L.J. van ’t Veer, H. Dai, M.J. van de Vijver, Y.D. He, A.A.M. Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, S.H. Friend, Gene expression profiling predicts clinical outcome of breast cancer, *Nature* 415 (2002) 530–536.
- [29] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, *Journal of Machine Learning Research* 5 (2004) 1205–1224.
- [30] Z. Zhu, Y. Ong, M. Dash, Markov blanket-embedded genetic algorithm for gene selection, *Pattern Recognition* 40 (2007) 3236–3248.