Adv Data Anal Classif (2018) 12:953–972
https://doi.org/10.1007/s11634-018-0318-1

REGULAR ARTICLE

An efficient random forests algorithm for high dimensional data classification

Qiang Wang[1] · **Thanh-Tung Nguyen**[2,4] · **Joshua Z. Huang**[1] · **Thuy Thi Nguyen**[3]

Received: 15 December 2014 / Revised: 15 June 2017 / Accepted: 13 March 2018 /
Published online: 21 March 2018

Abstract In this paper, we propose a new random forest (RF) algorithm to deal with high dimensional data for classification using subspace feature sampling method and feature value searching. The new subspace sampling method maintains the diversity and randomness of the forest and enables one to generate trees with a lower prediction error. A greedy technique is used to handle cardinal categorical features for efficient node splitting when building decision trees in the forest. This allows trees to handle very high cardinality meanwhile reducing computational time in building the RF model. Extensive experiments on high dimensional real data sets including standard machine learning data sets and image data sets have been conducted. The results demonstrated that the proposed approach for learning RFs significantly reduced prediction errors and outperformed most existing RFs when dealing with high-dimensional data.

✉ Thanh-Tung Nguyen
tungnt@tlu.edu.vn

Qiang Wang
wangqiang@szu.edu.cn

Joshua Z. Huang
zx.huang@szu.edu.cn

Thuy Thi Nguyen
ntthuy@vnua.edu.vn

1 College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

2 Faculty of Computer Science and Engineering, Thuyloi University, 175 Tay Son, Dong Da, Hanoi, Vietnam

3 Faculty of Information Technology, Vietnam National University of Agriculture, Hanoi, Vietnam

4 Sorbonne Université, IRD, JEAI WARM, Unité de Modélisation Mathématiques et Informatique des Systèmes Complexes, UMMISCO, 93143 Bondy, France

Keywords Classification · Image classification · High dimensional data · Random forests · Data mining

Mathematics Subject Classification 68T01

1 Introduction

Recently, high dimensional data becomes very common in data mining and knowledge discovery applications, i.e., data with thousands to millions of both samples and features. Besides the huge number of samples that are collected, the high dimensional nature of data in many applications is a statistically challenge when the number of features is typically thousands of times larger than the number of samples, for examples in Genome-wide association data, DNA microarrays and digital images. Furthermore, the input data contains multiple data types, missing values, non-linear interactions among features and the distribution is not Gaussian. This phenomenon is known as the problem of *curse of dimensionality* (Donoho et al. 2000) because the large number of features can increase the noise of the data, many types of analysis and classification problems become significantly harder. State-of-the-art machine learning classification methods can work well for data sets of moderate feature size but they suffer when scaling for high dimensional data.

Random forests (RF) model (Ho 1998; Dietterich 2000; Breiman 2001) is an ensemble machine learning method composed by un-pruned decision trees. RF is widely used in data mining domain and achieved a good performance when dealing with both regression and classification problems (Breiman 2001; Dietterich 2000; Banfield et al. 2007). However, the performance of random forests suffers when applied to high dimensional data (Xu et al. 2012; Ye et al. 2013; Nguyen et al. 2015; Deng and Runger 2013) because the simple feature sampling method is used to build the model.

Given a training data set $\mathcal{L} = \{(X_i, Y_i), X \in \mathbb{R}^M, Y \in \mathcal{Y}\}_{i=1}^N$, where X_i are features (also called predictor variables) and Y is the target (also called response feature), $\mathcal{Y} \in \{1, 2, \ldots, C\}$ for a classification problem ($C \geq 2$), N and M are the number of training samples and features, respectively. A standard version of RF independently and uniformly resamples observations from the training data $\mathcal{L}$ to draw a bootstrap data set $\mathcal{L}^*$ from which a decision tree T^* is grown. Repeating this process K times produces a series of bootstrap data sets $\mathcal{L}_k^*$ and corresponding decision trees T_k^* ($k = 1, 2, \ldots, K$), afterwards, we aggregate all K trees to form a RF model.

Given an input $X = x$, the predicted value by the whole RF is obtained by aggregating the results given by individual trees. Let $\hat{f}_k(x)$ denote the prediction of unknown value y of input $x \in \mathbb{R}^M$ by the kth tree, we have

$$\hat{f}(x) = \arg\max_{y \in \mathcal{Y}} \left\{ \sum_{k=1}^{K} \mathcal{I}\left[\hat{f}_k(x) = y \right] \right\}, \tag{1}$$

where $\mathcal{I}(\cdot)$ denotes the indicator function and $\hat{f}(x)$ denotes the RF prediction.

Recently, Xu et al. (2012) and Ye et al. (2013) proposed novel random forests by weighting the input features and then selecting features to ensure that each subspace

always contains informative features. Their efficient RF algorithms can be used to classify multi-class data. Amaratunga et al. (2008) presented a feature weighted method for subspace sampling to deal with two-class problems, in which the t-test of variance analysis is used to compute weights for the features. However in these sampling methods, they used a simple linear function to calculate the correlation measures between the predictor features and the response feature. This measure was treated as a weight for each predictor feature. Feature ranking was based on these simple weights and feature interactions were neglected.

Genuer et al. (2010) proposed a new RF involving a ranking of explanatory features using the RF score of importance and a stepwise ascending feature introduction strategy. Deng and Runger (2013) proposed a guided regularized RF (GRRF) and the weights are calculated using RF to produce importance scores from the out-of-bag data, in which these weights are used to guide the feature selection process. They found that the least regularized subset selected by GRRF with minimal regularization ensures better accuracy than the complete feature set. In these approaches, a regular RF was used as a classifier due to the fact that their RF models may have higher variance than RF because the trees are correlated.

Tuv et al. (2009) presented an ensemble of decision trees for both classification and regression. The features are partitioned into important features and irrelevant ones using a cut-off point, which comes straight from the classical hypothesis test with null distribution obtained by random permutations. The complex feature interactions were provided in the context of multiple hypothesis test for high-dimensional problems. However, the information gain at higher levels of a tree in the forest is weighted differently than the information gain at lower levels of the tree. In fact, at lower levels of a tree, the information gain is reduced because of the effect of splits on different features at higher levels of the tree.

Our new approach addresses the above mentioned problems by employing different techniques for feature weighting subspace selection. Given a training data set $\mathcal{L}$, we first use a feature permutation technique to measure the importance of features and produce raw feature importance scores. Then we apply p-value assessment to find the cut-off between informative and uninformative features. The χ^2 statistic is used for the classification problem to find the subset of highly informative features. When sampling the feature subspace, features from the group of highly informative features are taken into account for splitting the data at a node. Since the subspace always contains highly informative features, it can guarantee a better split at the node, therefore assuring a qualified tree. This sampling method always provides enough highly informative features for the subspace features at any levels of the decision tree. By using just a new feature sampling method, the diversity and random of the forests in the Breiman's framework (Breiman 2001) are maintained.

For efficient node splitting when building decision trees in the forest for dealing with categorical features, an unique categorical value in the current candidate feature is chosen by the probability distribution. This is a greedy technique for searching the cut-point s^* to handle cardinal categorical features for efficient node splitting. It allows to reduce the computational time and trees can handle very high cardinality.

The above feature subspace selection and greedy searching schemes are used for building trees in our new learning random forests algorithm, called ssRF, for solv-

ing classification problems. The proposed approach achieves a better computational efficiency than that achieved by existing RF. Our experimental results have shown that with the proposed feature sampling method and new greedy search scheme, our random forests outperformed existing random forests in reduction of prediction errors when applied to high dimensional data, even though a small feature subspace size of $\lfloor \log_2(M) + 1 \rfloor$ is used.

2 Feature partition and subspace selection

2.1 Importance measure of features from a tree and random forest

Classification and Regression Tree (CART) method involves repeated binary splits of subsets of samples, namely nodes or leaves, into two child nodes. For each split, one predictor feature and a corresponding cut-point (chosen according to the node impurity criterion) are used to split a parent node into two child nodes (Breiman et al. 1984; Louppe et al. 2013).

Denote the parent node to be split as t in a decision tree, a split on feature X_j is determined by the decrease in node impurity $\Delta R(X_j, t)$. Consider splitting at a node t for a classification tree, when the response feature $Y \in \mathcal{L}$ contains C classes, $Y \in \{1, 2, \ldots, C\}$, the Gini index is used to reflect the node impurity $R(t)$. Let $\Phi_c(t)$ be the class frequency for class $c \in C$ in a node t. Let s denote a proposed split for a feature X_j that splits t into left and right children nodes t_L and t_R depending on whether cases $X_j \leq s$ or $X_j > s$; i.e., $t_L = \{X_{ij} \in t, X_{ij} \leq s\}$ and $t_R = \{X_{ij} \in t, X_{ij} > s\}$. The Gini node impurity for t is defined as

$$Gini(t) = \sum_{c=1}^{C} \Phi_c(t)[1 - \Phi_c(t)]. \tag{2}$$

The Gini index is given by

$$\Delta R(s, t) = Gini(t_L)p(t_L) + Gini(t_R)p(t_R). \tag{3}$$

where $Gini(t_L)$ and $Gini(t_R)$ are the Gini node impurity for t_L and t_R; $p(t_L) = N_L(t)/N(t)$ and $p(t_R) = N_R(t)/N(t)$ are the proportions of observations that go left and right; $N(t)$, $N_L(t)$, $N_R(t)$ are the total number of samples of t, t_L, t_R, respectively. To achieve a good split, we seek the cut-point maximizing the decrease in node impurity and equivalently minimizing $\Delta R(s, t)$ with respect to s (Breiman et al. 1984).

Let $IS_k(X_j)$ denote the importance score of feature X_j in a single decision tree T_k. We have

$$IS_k(X_j) = \sum_{t \in T_k} \Delta R(s, t).$$

Let IS_j be an importance score of feature X_j and IS_j is computed over all K trees in a random forest (Breiman 2001), defined as

$$IS_j = \sum_{k=1}^{K} IS_k(X_j)/K.$$

We can normalize IS_j to values in [0, 1] using the min-max normalization as follows:

$$VI_j = \frac{IS_j - min(IS_j)}{max(IS_j) - min(IS_j)}. \quad (4)$$

Having the raw importance scores VI_j determined by Eq. (4) we can evaluate the contributions of the features regarding predicting the response feature.

2.2 A new feature sampling method for subspace selection

In our approach we need to rank input features. We first compute importance scores for all features according to Eq. (4). Denote the feature set as $\mathcal{L}_X = \{X_j, j = 1, 2, \ldots, M\}$. We randomly permute all values in each feature to get a corresponding shadow feature set, denoted as $\mathcal{L}_A = \{A_j\}_1^M$. The shadow features do not have prediction power to the response feature. Following the feature permutation procedure recently presented in Nguyen et al. (2015), we ran RF R times from the extended data set $\{\mathcal{L}_X \cup \mathcal{L}_A, Y\}$ to get importance scores $VI^r_{X_j}$ and $VI^r_{A_j}$, and the comparison sample denoted as $V^* = max\{A_{rj},\ r = 1, \ldots, R\}$.

The unequal variance Welch's two-sample t-test is then used to compare the importance score of a feature with the maximum importance scores of generated shadows. The non-parametric statistical test is required because the importance scores across the replicates are not normally distributed. Having computed the t statistic, we compute the p-value for the feature and perform hypothesis test on $\overline{VI}_{X_j} > \overline{V}^*$. This test confirms that if a feature is important, it consistently scores higher than the shadow over multiple permutations. Therefore, any feature whose importance score is smaller than the maximum importance score of noisy features, is considered less important. Otherwise, it is considered important.

The p-value of a feature indicates the importance of the feature in the prediction. The lower the p-value of a feature is correlated the feature is to the response feature, and the more powerful the feature is in the prediction. The use of a p-value requires a feature to consistently score higher than the shadow features over multiple replicates. Given a statistical level, we can identify informative features from low-informative ones.

Given all p values of features, we set threshold λ as a turning parameter, for instance $\lambda = 0.05$. Any feature whose p-value is greater than λ is added to the low-informative feature subset denoted as X_l, the direct relationship with the Y values is assessed otherwise. We now consider the set of features $\widetilde{X} = \{\mathcal{L}_X \setminus X_l\}$ obtained from the input features after separating all low-informative features.

$\chi^2(X_j, Y)$ is used to test the association between the class label and each feature $X_j \in \widetilde{X}$. For the test of independence, a chi-squared probability of less than or equal to 0.05 is commonly interpreted for rejecting the hypothesis that the feature is independent of the response feature.

Let X_h denote a subset of highly informative features. All features X_j are added to X_h whose p-value from the results of χ^2-test is smaller than 0.05 and the remaining features are added into a group of mid-informative features denoted as X_m.

The three subsets of features X_h, X_m and X_l are obtained using Algorithm 1. At each node, we randomly select $mtry$ ($mtry > 1$) features from three separated groups. For a given subspace size, we choose proportions between highly informative, mid-informative and low-informative features that depends on the size of the three groups. $Mtry_{high} = \lceil mtry \times (M_{high}/M) \rceil$, $mtry_{mid} = \lceil mtry \times (M_{mid}/M) \rceil$ and $mtry_{low} = mtry - mtry_{high} - mtry_{mid}$, where M_{high} and M_{mid} are the number of features in X_h and X_m, respectively. These are merged to form the feature subspace for splitting the node. The Random Forests diversity is maintained by using this randomly selected subspaces.

Algorithm 1: Feature Partition

input : The training data set $\mathcal{L}$ and a random forest RF.
R, θ: The number of replicates and the threshold.
output: X_h, X_m and X_l.

Let $\mathbb{S}_X = \{\mathcal{L} \backslash Y\}$, $M = \|\mathbb{S}_X\|$.
for $r \leftarrow 1$ **to** R **do**
  $\mathbb{S}_A \leftarrow permute(\mathbb{S}_X)$.
  $\mathbb{S}_{X,A} = \mathbb{S}_X \cup \mathbb{S}_A$.
  Build RF model from $\mathbb{S}_{X,A}$ to produce $\{IS^r_{X_j}\}$,
  $\{IS^r_{A_j}\}$ and IS^{max}_A, $(j = 1, \ldots, M)$.
Set $\widetilde{X} = \emptyset$.
for $j \leftarrow 1$ **to** M **do**
  Compute Wilcoxon rank-sum test with IS_{X_j} and IS^{max}_A .
  Compute p_j values for each feature X_j.
  if $p_j \leq \theta$ **then**
    $\widetilde{X} = \widetilde{X} \cup X_j \; (X_j \in \mathbb{S}_X)$
Set $X_h = \emptyset$, $X_m = \emptyset$, $X_l = \emptyset$.
$X_l = \mathbb{S}_X \setminus \widetilde{X}$.
for $j \leftarrow 1$ **to** $\|\widetilde{X}\|$ **do**
  Compute χ^2-test with $\widetilde{X}$ and Y to get p_j value.
  if $(p_j < 0.05)$ **then**
    $X_h = X_h \cup X_j \; (X_j \in \widetilde{X})$
$X_m = \{\widetilde{X} \backslash X_h\}$.
return X_h, X_m, X_l

3 The proposed algorithm

3.1 The greedy search technique for node split

Consider the splitting of a node t into t_L, t_R based on the jth feature of X which is assumed here to be a categorical feature whose possible values $s \in S$, and a finite set $S \in \mathcal{L}_{X_j}$. The decrease in node impurity $\Delta R(s, t)$ at the categorical value s in t is cal-

culated by Eq. (3). The cut-point s^* is a split such that $\Delta R(s^*, t) = \max_{s \in S} \Delta R(s, t)$ (Breiman et al. 1984). In Breiman (2001) there are $2^{S-1} - 1$ possible splits of the

Algorithm 2: Random Greedy Approach

```
input : A categorical feature X_j
output: the cut-point s*
Set S = unique(L_{X_j})
S = randomize(S) /* randomize the order of S */
Initialize L^s = ∅ that sends data to the left branch.
for i ← 1 to ||S|| do
    s ← S_i
    L^s ← s
    /* Scan all categorical values s at node t in the current feature
    X_j */
    for m ← 1 to N(t) do
        if s == X_mj then
            Compute the decrease in impurity ΔR(s, t) according to Eq. (3), then calculate how
            good the split is.
            if the split is improved then
                L^s ← s /* keep s in L^s */
            else
                L^s = L^s \ s /* remove s from L^s */
s* ← L^s_1 /* The first element of L^s */
```

cardinal S categorical values. The computational complexity when obtaining splits can be reduced, $S - 1$ splits, if the values are sorted such that $\overline{Y}_1 \leq \overline{Y}_2 \leq \cdots \overline{Y}_S$ (see Breiman et al. 1984, Theorem 4.5, Section 9.4). However, using this approach, there is an exception when sorting values according to the average of the response values. In our approach, we take full advantage of Breiman's method for categorical data but do not use the sorted Y values. We use the method introduced by Viswanathan et al. (2011) to search the cut-point s^*. In each step, we choose the next unique categorical value in the current candidate feature according to the probability distribution. In this case, a categorical value s is tested to know how good the split is. We keep s if its split is improved, a random categorical value is chosen otherwise. This step will stop after S unique categorical values are tested. The random greedy approach is summarized in Algorithm 2. The motivation behind this new approach of searching the cut-point s^* is to achieve better computational efficiency than that achieved by existing methods. This approach reduces computational complexity and can handle very high cardinality.

3.2 The ssRF algorithm

The new feature subspace sampling method is now used to generate splits at the nodes of decision trees for building RF. The greedy random search is used to split a categorical feature at any level of trees.

Table 1 Description of the classification data sets sorted by the number of features

Data set	#Training	#Testing	#Features	#Classes
Fbis	1711	752	2000	17
Brain-tumor1	72	18	5920	5
Prostate-tumor	82	20	10,509	2
La2s	1855	845	12,432	5
Lung-cancer	162	41	12,600	5
11-tumors	139	35	12,533	11
La1s	1963	887	13,195	5
GCM	152	38	16,063	14

The new random forests algorithm ssRF consists of following steps:

1. Given $\mathcal{L}$, arrange highly related features and the weak important features from the less important ones in three feature subsets X_h, X_m and X_l obtained by Algorithm 1.
2. Sample the training set $\mathcal{L}$ with replacement to generate bagged samples $\mathcal{L}_k, k = 1, 2, \ldots, K$.
3. For each $\mathcal{L}_k$, build a classification tree T_k as follows:
   (a) At each node, select a subspace of $mtry$ ($mtry > 1$) features randomly and separately from X_l, X_m and X_h and use the subspace features as candidates for splitting the node.
   (b) Each tree is grown nondeterministically, without pruning until the minimum node size n_{min} is reached. For a categorical feature in a node of each tree, the greedy search in Algorithm 2 is used to search for the best split.
6. Given an input $X = x$, use Eq. (1) to predict the new sample for the classification problem.

4 Experiments and evaluation

4.1 Data sets

Table 1 lists the real data sets used to evaluate the performance of random forests models. The *Fbis* data set was compiled from the archive of the Foreign Broadcast Information Service and the *La1s*, *La2s* data sets were taken from the archive of the Los Angeles Times for TREC-5.[1]

Five gene data sets *Brain-tumor1, Prostate-tumor, Lung-cancer, 11-tumors* and *GCM* are taken from http://www.gems-system.org, http://www.upo.es/eps/bigs/datasets.html. Gene classification is an important problem in Genome-wide association data analysis. The data sets are sorted by the dimensionality.

[1] http://www.trec.nist.gov.

Image classification and object recognition are important problems in computer vision. Image data sets are well-known to be big and the feature space is usually high dimensional. This challenges machine learning methods. In the experiments we tested our system on four benchmark data sets, including: The *Caltech* categories data set,[2] the *Horse* data set,[3] the extended *YaleB* database (Georghiades et al. 2001) and the *AT&T ORL* data set (Samaria and Harter 1994). In the following we briefly summarize the characters of the image data sets and the extracted features for each of them that will be used in our experiments.

For the *Caltech* data set, we use a subset of 100 images from the *Caltech* face dataset and 100 images from the *Caltech* background data set following the setting in ICCV[4] and Ye et al. (2013). The extended *YaleB* database consists of 2414 face images of 38 individuals captured under various lighting conditions. Each image has been cropped to a size of 192×168 pixels and normalized. The *Horse* data consists of 170 images containing horses for the positive class and 170 images of the background for the negative class. The *AT&T ORL* data set includes of 400 face images of 40 persons.

Extraction of features for image data representation is a complex process. Traditionally, feature extraction is used in conjunction with a classifier and the quality of the extracted features strongly influences the performance of the classifier. One of the most well-know methods for image feature extraction is the bag-of-words (Zhang et al. 2007; Lepetit and Fua 2006). In our experiments, we use this type of features for the *Caltech* and the *Horse* data sets. Derived data sets with codebook sizes of 300, 500, 1000, 3000, 5000, 7000, 10,000, 12,000 and 15,000 were used. For the face data sets, we use two type of features: Eigenface (Turk and Pentland 1991) and the random features (randomly sample pixels from the images). These features used in our experiments are with dimensions of 30, 56, 120, and 504.

4.2 Evaluation measures and experimental settings

The performance of a random forests model was evaluated on a test data set with two measures that are the *Area under the curve* (AUC) and the *test accuracy*, defined as:

$$\text{Test Accuracy} = \frac{1}{N}\sum_{i=1}^{N} I\left(Q(d_i, y_i) - max_{j \neq y_i} Q(d_i, j) > 0\right), \tag{5}$$

where $\hat{f}_{\mathbb{D}_t}(x_i)$ is the prediction given $X = x$, $I(\cdot)$ is the indicator function; N is the number of samples in test data $\mathbb{D}_t$ and y_i indicates the true value of d_i; $Q(d_i, j) = \sum_{k=1}^{K} I(h_k(d_i) = j)$ is the number of votes for $d_i \in \mathbb{D}_t$ on class j, h_k is the kth tree classifier.

[2] http://www.vision.caltech.edu/html-files/archive.html.

[3] http://pascal.inrialpes.fr/data/horses/.

[4] http://people.csail.mit.edu/torralba/shortCourseRLOC/.

Table 2 Comparison of prediction performance (test error) of the random forests models

Data set	RF	GRRF	wsRF	ssRF
Fbis	.236	.239	.159	**.153**
Brain-tumor1	.155	.143	.143	**.110**
Prostate-tumor	.079	**.068**	.088	.069
La2s	.210	.174	.130	**.110**
Lung-cancer	.104	.079	.074	**.059**
11-tumors	.125	.121	.097	**.092**
La1s	.223	.196	.137	**.128**
GCM	.347	.331	.305	**.268**

Numbers in bold are the best results

The latest RF (Liaw and Wiener 2002), QRF (Meinshausen 2012) and GRRF (Deng 2013) R-packages were used in R environment to conduct these experiments. For the GRRF model, we used a value of 0.1 for the coefficient γ because GRRF(0.1) has shown competitive prediction performance in Deng and Runger (2013). The novel wsRF model (Xu et al. 2012) using the weighted sampling method was intended to solve the classification problem. The ssRF model with the new subspace sampling method and the greedy random approach is a new implementation. In that implementation, we called the corresponding R/C++ functions in R environment. The parameters R, $mtry$ and λ for pre-computation of feature partition in Algorithm 1 were 30, $\sqrt{M}$ and 0.05, respectively.

For the three data sets *Fbis*, *La2s*, *La1s*, the number of samples is fixed in the training and testing data, as shown in Table 1. From each training data set we built 10 random forest models; each of the classification model had 500 trees. The number of the minimum node size n_{min} was 1. The number of features-candidates was set with the default setting to $mtry = \lfloor \log_2(M) + 1 \rfloor$. For the gene data sets *Brain-tumor1*, *Prostate-tumor*, *Lung-cancer*, *11-tumors* and *GCM*. The 5-fold cross-validation was used to evaluate the prediction performance of the models on gene data sets. For the visual data sets, 10-fold cross-validation was used to evaluate the prediction performance of the models. From each fold, we built the models with 500 trees, the subspace size is fixed $mtry = \sqrt{M}$ and the gene partition was re-calculated on each training fold data set.

The average of test error of the models were computed according to Eq. (5), where $Test\ Error = 1 - Test\ Accuracy$. All experiments were conducted on the six 64-bit Linux machines, each one equipped with Intel$R$ Xeon$R$ CPU E5620 2.40 GHz, 16 cores, 4 MB cache, and 32 GB main memory. The ssRF and wsRF models were implemented as multi-thread processes, while other models were run as single-thread processes.

4.3 Experimental results on real data sets

Table 2 presents the average test error results of random forest models with 10 repetitions on the data sets. Table 3 shows the prediction test errors on the three data sets *Fbis*, *La2s*, *La1s* against the number of trees and features. Table 4 shows the prediction

Table 3 The prediction test error of the models against the number of trees K and features $mtry$ on the three data sets *Fbis*, *La2s*, *La1s*

Data set	Model	The number of trees					The number of features				
		K = 50	K = 100	K = 150	K = 200	K = 300	mtry = 10	mtry = 20	mtry = 30	mtry = 40	mtry = 50
Fbis	RF	.2307	.2241	.2254	.2261	.2279	.2434	.2351	.2156	.2303	.2187
	GRRF	.2394	.2407	.2287	.2314	.2340	.2527	.2101	.1955	.1862	.1981
	wsRF	.1689	**.1649**	.1622	**.1569**	.1618	**.1569**	.1702	.1636	.1715	.1715
	ssRF	**.1676**	.1676	**.1543**	.1689	**.1569**	.1822	**.1556**	**.1503**	**.1503**	**.1522**
La2s	RF	.2303	.2363	.2256	.2315	.2280	.2536	.1611	.1586	.1432	.1402
	GRRF	.2476	.2121	.2180	.2156	.2192	.2820	.1860	.1540	.1505	.1386
	wsRF	.1327	.1517	.1493	.1445	.1410	.1244	.1315	.1374	.1386	.1434
	ssRF	**.1078**	**.1066**	**.1102**	**.1185**	**.1090**	**.1149**	**.1102**	**.0995**	**.1002**	**.1014**
La1s	RF	.6708	.6697	.6731	.6742	.6488	.6776	.6032	.4543	.3337	.2052
	GRRF	.1928	.1759	.2063	.1849	.1966	.1905	.1691	.1612	.1577	.1409
	wsRF	**.1308**	.1353	.1330	.1353	.1488	**.1330**	**.1375**	**.1387**	.1364	.1398
	ssRF	.1354	**.1321**	**.1322**	**.1321**	**.1264**	.1477	.1432	.1443	**.1319**	**.1387**

Numbers in bold are the best results

Table 4 The prediction test error of the RF models against the number of trees K on the gene data sets

Gene data	Method	K = 100	K = 200	K = 300	K = 400	K = 500	K = 600	K = 700	K = 800	K = 900	K = 1000
Brain-tumor1	RF	.187	.145	.167	.179	.166	.177	.166	.166	.131	.155
	GRRF	.153	.156	.133	.145	.132	.155	.143	.133	.142	.133
	wsRF	.165	.145	.144	.167	.132	.121	.133	.122	.121	.122
	ssRF	**.132**	**.132**	**.122**	**.122**	**.121**	**.111**	**.121**	**.111**	**.109**	**.122**
Prostate-tumor	RF	.098	.097	.079	.108	.088	.107	.107	.098	.089	.089
	GRRF	**.078**	.088	**.069**	.081	.079	.078	.079	.079	.089	.089
	wsRF	.108	.068	.079	.089	.088	.078	.087	.079	.079	.079
	ssRF	.088	**.078**	.069	**.079**	**.059**	**.069**	**.078**	**.089**	**.069**	**.079**
Lung-cancer	RF	.093	.088	.079	.084	.074	.088	.083	.094	.088	.079
	GRRF	.078	.064	.069	.074	.069	.079	.068	.084	.083	.069
	wsRF	.063	.073	.064	.074	.059	.079	.068	.084	.073	.059
	ssRF	**.059**	**.064**	**.049**	**.064**	**.054**	**.069**	**.058**	**.074**	**.059**	**.054**
11-tumors	RF	.121	.121	.103	.132	.121	.114	.103	.127	.129	.116
	GRRF	.121	.107	.12	.121	.109	.109	.109	.116	.099	.11
	wsRF	.11	**.075**	.097	.081	.105	.102	.092	.092	.087	.092
	ssRF	**.076**	.075	**.086**	**.075**	**.092**	**.081**	**.075**	**.081**	**.071**	**.077**
GCM	RF	.358	.316	.334	.359	.333	.348	.326	.356	.327	.337
	GRRF	.327	.278	.298	.321	.306	.311	.299	.314	.316	.311
	wsRF	.337	.3	.319	.311	.285	.311	.305	.314	.3	.316
	ssRF	**.306**	**.253**	**.266**	**.289**	**.259**	**.264**	**.258**	**.261**	**.275**	**.279**

Numbers in bold are the best results

performance on the gene data sets. We can see that the ssRF model always provided good results and achieved lower prediction error when varying K and $mtry$ on both kind of data sets. In some cases where the ssRF model did not obtain the best results compared with the wsRF, GRRF models on *Fbis*, *La1s*, *Prostate-tumor* and *11-tumors* the data sets, the differences from the best results were minor.

The RF model require larger number of features to achieve the lower prediction error, as shown in the right part of Table 3. This means the RF model could achieve better prediction performance only if they are provided with a much larger feature subspace. For solving the classification problem, the size of the subspace in the default settings of RF (Liaw and Wiener 2002) was set to $mtry = \lfloor\sqrt{M}\rfloor$. With this size, the computational time for building a RF is still too high, especially for large high dimensional data. These empirical results indicated that, the ssRF model does not need many features in the subspace to achieve good prediction performance. For application on high dimensional data, when the ssRF model uses a subspace of features of size $mtry = \lfloor\log_2(M) + 1\rfloor$ features, the achieved results can be satisfactory. In general, when the feature subspace of the same size as the one suggested by Breiman is used, the ssRF model gives lower prediction error with a less computational time than those reported by Breiman. This achievement is considered to be one of the contributions in this work.

We also test the effect of the depth of the tree on the three data sets, the results have shown in Table 5, the minimum number of samples per leaf n_{min} was increased stepwise from 3 to 15 while holding other parameters ($K = 200$ and $mtry = \lfloor\log_2(M) + 1\rfloor$) fixed. It can be seen that the ssRF model outperformed other random forests models on all cases. The prediction test error of the ssRF model always produced the best results even though $n_{min} = 15$. These results demonstrated that, at lower levels of the tree, the gain is reduced because of the effect of splits on different features at higher levels of the tree. The other random forests models reduce the prediction accuracy dramatically while the ssRF model always is stable and produces the best results. This was because the feature weighting subspace sampling method was used in generating trees in the ssRF model. The selected subspace of features contains enough highly informative features at any levels of the decision tree. The effect of the new sampling method is clearly demonstrated in this result.

Figure 1 is provided to illustrate the speed of splitting of a categorical feature using the greedy random approach described in Algorithm 2. The feature subspace for splitting a node was $mtry = 1,\ 2$ and 3. In order to train the models, a categorical subset was selected from the *Fbis* data set and we increased the number of objects in this data set with 20 times. The RF model requires high number of iterations when obtaining categorical splits, which makes it computational time high. One worth noting remark is that the ssRF model uses the random greedy method to search the cut-point has almost linear relationship between the computational time and the number of samples. In addition, the fact that the *Fbis* subset has maximum 28 levels of categorical features. This means that the cost of evaluating all possible categorical splits is not too high when compared to our random greedy method. In domains containing categorical features with a large set of values, the advantage of our random greedy method would be even more evident.

Table 5 The prediction test errors of the classification random forests models against the number of samples per leaf n_{min}

Data set	Model	$n_{min} = 3$	$n_{min} = 5$	$n_{min} = 8$	$n_{min} = 10$	$n_{min} = 15$
Fbis	RF	.2358	.2513	.2959	.3231	.3711
	GRRF	.2388	.2509	.2972	.3197	.3656
	wsRF	.1885	.1997	.2005	.2289	.2317
	ssRF	**.1661**	**.1674**	**.1738**	**.1838**	**.2008**
La2s	RF	.2096	.2071	.2731	.3323	.4094
	GRRF	.1735	.1717	.2754	.3201	.3627
	wsRF	.1294	.1283	.1618	.1737	.1910
	ssRF	**.1109**	**.1101**	**.1107**	**.1139**	**.1316**
La1s	RF	.2230	.2237	.2635	.3526	.3968
	GRRF	.1950	.1921	.2956	.3577	.3968
	wsRF	.1464	.1566	.1664	.1839	.2027
	ssRF	**.1397**	**.1402**	**.1417**	**.1454**	**.1578**

Numbers in bold are the best results

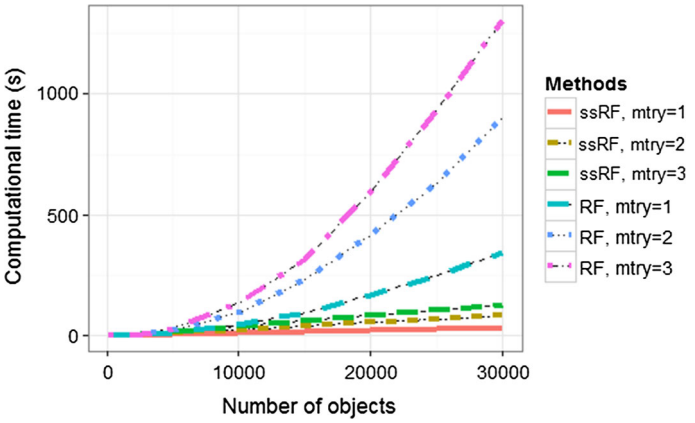

Fig. 1 Comparison of the computational time on splitting categorical data

Figures 2 and 3 show the box plots of the recognition rate (%) of the models on the *YaleB* and *ORL* image data sets using eigenface and random features, respectively. From these figures, we can observe that the ssRF and GRRF models always produced good results, the GRRF model demonstrated better results on some cases, for examples, *YaleB* + *Eigenface* 120 features, *YaleB* + *Eigenface* 504 features and *ORL* + *randomface* 504 features. The reason could be that truly informative features in this kind of data sets were many. Therefore, when the informative feature set was large, so the chance of selecting informative features in the subspace increased, which in turn increased the average recognition rates of the GRRF model. However, the ssRF model produced the best results in the remaining cases. The effect of the new approach for feature subspace selection is clearly demonstrated in these results, although these data sets are not very high dimensional.

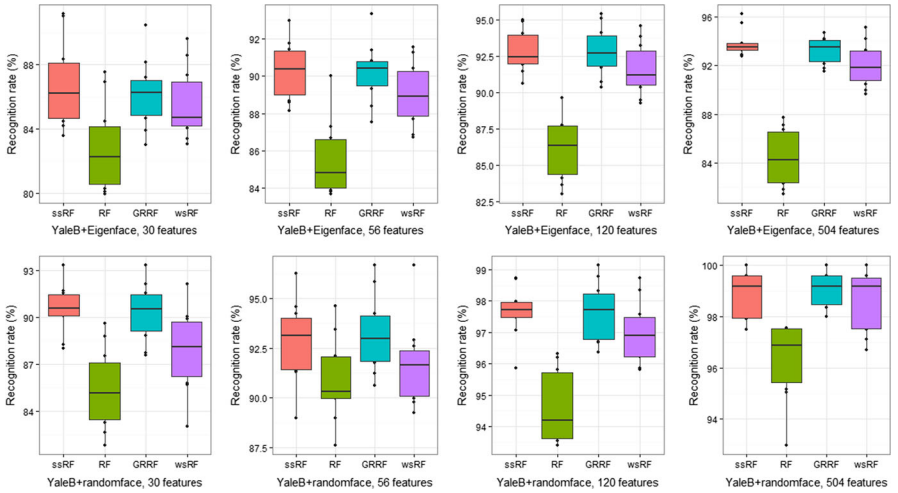

Fig. 2 Box plots of recognition rate (%) using Eigenface and random features on the YaleB data set

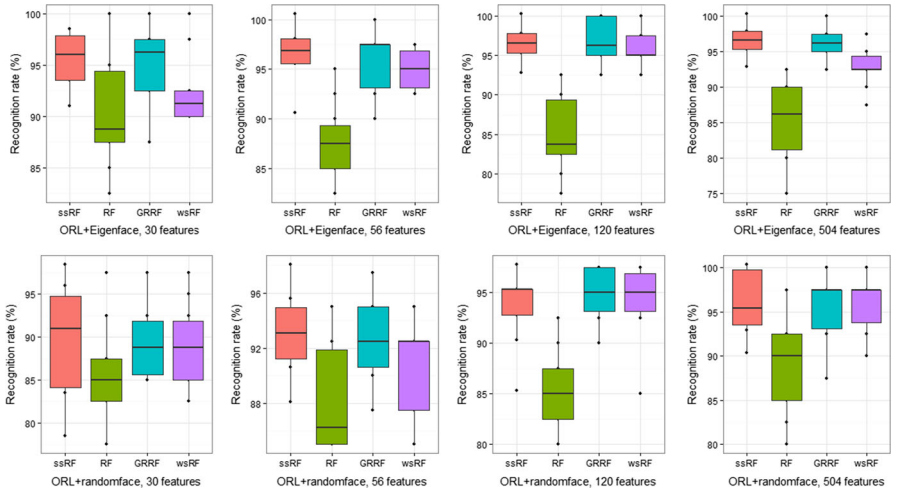

Fig. 3 Box plots of recognition rate (%) using Eigenface and random features on the ORL image data set

Table 6 shows the classification results in terms of accuracy (mean $\pm$ std-dev%) on the two image data sets, *Caltech* and *Horse*. Our proposed ssRF model with the new feature selection method is tested with different codebook sizes, from 300 to 15,000. We can see that the ssRF model consistently obtained highest results when varying the number of the used codebook sizes. Table 7 shows the results of our ssRF model in terms of AUC measure (mean $\pm$ std-dev%) on the two image data sets, *Caltech* and *Horse*. Again, the new proposed ssRF model is tested with different codebook sizes, from 300 to 15,000. It can be seen clearly that the ssRF model obtained highest results and outperformed other existing RF models, including traditional RF, recently proposed wsRF and GRRF. There is only one case when the GRRF model is

Table 6 Test accuracy results (mean ± std-dev%) of random forests models against the number of codebook size on the visual data sets

Data set	Model	300	500	1000	3000	5000	7000	10,000	12,000	15,000
Caltech	ssRF	**.93 ± .2**	**.95 ± .2**	.98 ± **.1**	**.98 ± .1**	**.96 ± .2**	.92 ± .2	**.87 ± .5**	**.86 ± .6**	**.82 ± .2**
	RF	.84 ± .4	.85 ± .7	.84 ± .6	.81 ± 1.3	.90 ± .5	.84 ± .7	.73 ± .6	.70 ± 1.4	.55 ± .2
	wsRF	.89 ± .3	.91 ± .2	.94 ± .2	.92 ± .7	.92 ± .7	.88 ± .3	.82 ± .6	.83 ± .8	.70 ± .5
	GRRF	.92 ± .1	.94 ± .2	.95 ± .2	.94 ± .2	.94 ± .2	**.93 ± .2**	.86 ± .4	.82 ± .6	.74 ± .2
Horse	ssRF	**.80 ± .3**	**.81 ± .2**	**.84 ± .4**	**.77 ± .7**	**.81 ± .5**	**.79 ± .6**	**.76 ± .3**	**.79 ± .3**	**.76 ± .2**
	RF	.70 ± .4	.69 ± .4	.75 ± .5	.66 ± .7	.69 ± .6	.68 ± .4	.7 ± .3	.66 ± .1	.61 ± .2
	wsRF	.68 ± .3	.66 ± .5	.72 ± .3	.69 ± .7	.66 ± .4	.67 ± .8	.72 ± .2	.66 ± .5	.66 ± .4
	GRRF	.74 ± .4	.72 ± .2	.76 ± .3	.72 ± .2	.74 ± .5	.72 ± .4	.73 ± .4	.71 ± .2	.71 ± .2

Numbers in bold are the best results

Table 7 AUC results (mean ± std-dev%) of random forests models against the number of codebook size on the visual data sets

Data set	Model	300	500	1000	3000	5000	7000	10,000	12,000	15,000
Caltech	ssRF	**1.0 ± .0**	**1.0 ± .0**	**1.0 ± .0**	**1.0 ± .0**	**1.0 ± .0**	.98 ± .1	**.98 ± .1**	**.99 ± .1**	**.93 ± .6**
	RF	.91 ± .3	.93 ± .6	.90 ± .3	.89 ± .9	.98 ± .0	.98 ± .1	.97 ± .1	.94 ± .1	.87 ± 1.3
	wsRF	.96 ± .2	.97 ± .1	.98 ± .0	.98 ± .0	.98 ± .0	.97 ± .1	.97 ± .1	.96 ± .2	.88 ± .5
	GRRF	.97 ± .1	.98 ± .0	.99 ± .0	.98 ± .0	.99 ± .0	**.98 ± .0**	.97 ± .0	.94 ± .1	.89 ± .3
Horse	ssRF	**.91 ± .0**	**.89 ± .0**	**.87 ± .0**	**.86 ± .0**	**.87 ± .0**	**.86 ± .0**	**.92 ± .0**	**.87 ± .0**	**.86 ± .0**
	RF	.76 ± .8	.78 ± .4	.81 ± .6	.75 ± .9	.76 ± .7	.76 ± .7	.79 ± .4	.75 ± .6	.73 ± .5
	wsRF	.74 ± .6	.74 ± .5	.80 ± .5	.74 ± .6	.75 ± .6	.73 ± .9	.79 ± .2	.71 ± 1.1	.72 ± .3
	GRRF	.85 ± .2	.84 ± .2	.83 ± .2	.81 ± .1	.79 ± .2	.80 ± .2	.84 ± .1	.76 ± .2	.77 ± .3

The value of bold indicates best result from the models

Table 8 The prediction test accuracy (mean ± std-dev%) of the models on the visual data sets against the number of trees K

Data set	Model	K = 20	K = 50	K = 80	K = 100	K = 200
YaleB ± eigenface	ssRF	.757 ± .1	**.857 ± .1**	**.881 ± .1**	.889 ± .0	.912 ± .0
(M = 504)	RF	.719 ± .1	.795 ± .1	.807 ± .1	.817 ± .1	.829 ± .1
	wsRF	**.776 ± .1**	.856 ± .0	.881 ± .0	.893 ± .0	.907 ± .0
	GRRF	.747 ± .0	.847 ± .1	.872 ± .0	**.896 ± .0**	**.919 ± .0**
YaleB ± randomface	ssRF	.947 ± .0	.976 ± .0	.980 ± .0	.982 ± .0	.986 ± .0
(M = 504)	RF	.880 ± .0	.926 ± .0	.941 ± .0	.949 ± .0	.961 ± .0
	wsRF	.954 ± .0	.979 ± .0	.982 ± .0	.981 ± .0	.984 ± .0
	GRRF	**.957 ± .0**	**.981 ± .0**	**.984 ± .0**	**.989 ± .0**	**.988 ± .0**
ORL ± eigenface	ssRF	.763 ± .6	.873 ± .3	**.918 ± .2**	**.933 ± .2**	**.948 ± .2**
(M = 504)	RF	.718 ± .2	.788 ± .4	.820 ± .3	.828 ± .3	.855 ± .5
	wsRF	**.783 ± .4**	**.888 ± .3**	.900 ± .1	.913 ± .2	.925 ± .2
	GRRF	.735 ± .6	.850 ± .2	.900 ± .1	.908 ± .3	.948 ± .1
ORL ± randomface	ssRF	**.878 ± .3**	.925 ± .2	**.955 ± .1**	.943 ± .1	**.960 ± .1**
(M = 504)	RF	.775 ± .3	.820 ± .7	.845 ± .2	.875 ± .2	.860 ± .2
	wsRF	.870 ± .5	**.938 ± .2**	.938 ± .0	**.950 ± .1**	.955 ± .1
	GRRF	.873 ± .1	.933 ± .1	.945 ± .1	.943 ± .1	.955 ± .1
Caltech	ssRF	**.950 ± .3**	**.975 ± .1**	**.965 ± .1**	**1.000 ± .0**	**.985 ± .1**
(M = 15,000)	RF	.565 ± .6	.575 ± .2	.545 ± .1	.555 ± .2	.535 ± .1
	wsRF	.685 ± 1.2	.665 ± .4	.705 ± 1.0	.675 ± .8	.695 ± 1.4
	GRRF	.790 ± .5	.816 ± .1	7.700 ± 5.1	.786 ± .4	.858 ± .3
Horse	ssRF	**.695 ± .4**	**.729 ± .2**	**.735 ± .6**	**.721 ± .5**	**.772 ± .1**
(M = 15,000)	RF	.556 ± .5	.588 ± .3	.600 ± .4	.606 ± .3	.626 ± .3
	wsRF	.574 ± .2	.647 ± .7	.629 ± .3	.621 ± .5	.641 ± .5
	GRRF	.666 ± .5	.660 ± .2	.669 ± .4	.617 ± .1	.695 ± .3

The number of feature dimensions in each data set is fixed. Numbers in bold are the best results

Table 9 AUC results (mean ± std-dev%) of random forest models against the number of trees K

Data set	Model	K = 20	K = 50	K = 80	K = 100	K = 200
Caltech	ssRF	**.995 ± .3**	**.999 ± .1**	**1.00 ± .1**	**1.00 ± .0**	**1.00 ± .1**
	RF	.851 ± .6	.817 ± .2	.826 ± .1	.865 ± .2	.864 ± .1
	wsRF	.841 ± 1.2	.845 ± .4	.834 ± 1.0	.850 ± .8	.870 ± 1.4
	GRRF	.846 ± .5	.860 ± .1	.862 ± 5.1	.908 ± .4	.923 ± .3
Horse	ssRF	**.849 ± .4**	**.887 ± .2**	**.895 ± .6**	**.898 ± .5**	**.897 ± .1**
	RF	.637 ± .5	.664 ± .3	.692 ± .4	.696 ± .3	.733 ± .3
	wsRF	.635 ± .2	.687 ± .7	.679 ± .3	.671 ± .5	.718 ± .5
	GRRF	.786 ± .5	.778 ± .2	.785 ± .4	.699 ± .1	.806 ± .3

The number of codebook size was 15,000. Numbers in bold are the best results

slightly better in both tables, the *Caltech* data set with codebook size of 700, but this is neglectable.

Table 8 reports the test results of the models when varying the number of trees K, the number of the used features M is fixed for each data set. One can see that for all of the tested random forests models, when the number of trees K in the forests is increased, the accuracy is improved. We are interested in the performance of the models when the number of the used features (or codebook sizes) varied. When the number of the used features was moderate, for example $M = 504$ in the face data sets, the ssRF model was comparable to state-of-the-art wsRF and GRRF models. However, when the codebook size was large, i.e. high dimensional data ($M = 15{,}000$ in our experiments), our ssRF model significantly surpassed those of stat-of-the-art models. More than that, the truly informative features sets detected and selected by our new feature subspace selection method are small. Using only a small subset of features significantly reduces computational time in building a random forests model.

The advance of the ssRF model is further confirmed by the AUC results shown in Table 9. One can see that with the high dimensional data (codebook size $M = 15{,}000$ in this experiment), when the number of trees K in the forests is varied, our ssRF model outperformed and was significantly better than all the above mentioned stat-of-the-art random forests models.

5 Conclusions

We have presented a new random forest algorithm based on the state-of-the-art RF for high dimensional data. In this algorithm, we propose a new approach for feature subspace selection and feature value searching in random forests to deal with high dimension of feature space for classification. Our first contribution is a new feature weighting subspace sampling method in RF. Our second contribution is a greedy technique to handle cardinal categorical features for efficient node splitting when building decision trees in the forest. This enables the tree to handle very high cardinality, to deal with missing values meanwhile reducing computational time. The small subspace size $mtry = \lfloor \log_2(M) + 1 \rfloor$ reported by Breiman can be used in our algorithm to get lower prediction error. With ssRF, the feature space is reduced and the performance for classification is preserved and improved. Experimental results have demonstrated the improvement of our ssRF in reduction of prediction errors in comparison with existing recent proposed random forests, and especially it performed well on high dimensional data.

Acknowledgements Part of this work was done while the author Thanh-Tung Nguyen was visiting the Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), Shenzhen 518055, and the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China.

References

Amaratunga D, Cabrera J, Lee YS (2008) Enriched random forests. Bioinformatics 24(18):2010–2014

Banfield RE, Hall LO, Bowyer KW, Kegelmeyer WP (2007) A comparison of decision tree ensemble creation techniques. IEEE Trans Pattern Anal Mach Intell 29:173–180

Breiman L (2001) Random forests. Mach Learn 45(1):5–32
Breiman L, Friedman J, Stone CJ, Olshen RA (1984) Classification and regression trees. CRC Press, Boca Raton
Deng H (2013) Guided random forest in the rrf package. arXiv preprint arXiv:1306.0237
Deng H, Runger G (2013) Gene selection with guided regularized random forest. Pattern Recognit 46(12):3483–3489
Dietterich TG (2000) An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Mach Learn 40(2):139–157
Donoho DL et al (2000) High-dimensional data analysis: the curses and blessings of dimensionality. AMS Math Challenges Lecture, pp 1–32
Genuer R, Poggi JM, Tuleau-Malot C (2010) Variable selection using random forests. Pattern Recognit Lett 31(14):2225–2236
Georghiades AS, Belhumeur PN, Kriegman D (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans Pattern Anal Mach Intell 23(6):643–660
Ho TK (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844
Lepetit V, Fua P (2006) Keypoint recognition using randomized trees. IEEE Trans Pattern Anal Mach Intell 28(9):1465–1479
Liaw A, Wiener M (2002) Classification and regression by random forest. R News 2(3):18–22
Louppe G, Wehenkel L, Sutera A, Geurts P (2013) Understanding variable importances in forests of randomized trees. In: Advances in neural information processing systems, pp 431–439
Meinshausen N (2012) quantregforest: quantile regression forests. R package version 02-3
Nguyen TT, Huang J, Nguyen T (2015) Two-level quantile regression forests for bias correction in range prediction. Mach Learn 101(1–3):325–343
Samaria FS, Harter AC (1994) Parameterisation of a stochastic model for human face identification. In: Proceedings of the second IEEE workshop on applications of computer vision. IEEE, pp 138–142
Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3(1):71–86
Tuv E, Borisov A, Runger G, Torkkola K (2009) Feature selection with ensembles, artificial variables, and redundancy elimination. J Mach Learn Res 10:1341–1366
Viswanathan V, Sen A, Chakraborty S (2011) Stochastic greedy algorithms: a leaning based approach to combinatorial optimization. Int J Adv Softw 4(1 and 2):1–11
Xu B, Huang JZ, Williams G, Wang Q, Ye Y (2012) Classifying very high-dimensional data with random forests built from small subspaces. Int J Data Warehous Min 8(2):44–63
Ye Y, Wu Q, Zhexue Huang J, Ng MK, Li X (2013) Stratified sampling for feature subspace selection in random forests for high dimensional data. Pattern Recognit 46(3):769–787
Zhang J, Marszałek M, Lazebnik S, Schmid C (2007) Local features and kernels for classification of texture and object categories: a comprehensive study. Int J Comput Vis 73(2):213–238