A Novel Feature Subspace Selection Method in Random Forests for High Dimensional Data

Yisen Wang[*†‡], Shu-Tao Xia[*†]
[*]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[†]Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
[‡]Tsinghua National Laboratory for Information Science and Technology, Beijing, China
Email: wangys14@mails.tsinghua.edu.cn, xiast@sz.tsinghua.edu.cn

Abstract—**Random forests are a class of ensemble methods for classification and regression with randomizing mechanism in bagging instances and selecting feature subspace. For high dimensional data, the performance of random forests degenerates because of the random sampling feature subspace for each node in the construction of decision trees. To address the issue, in this paper, we propose a new Principal Component Analysis and Stratified Sampling based method, called PCA-SS, for feature subspace selection in random forests with high dimensional data. For each decision tree in the forests, we firstly create the training data by bagging instances and partition the feature set into several feature subsets. Principal Component Analysis (PCA) is applied on each feature subset to obtain transformed features. All the principal components are retained in order to preserve the variability information of the data. Secondly, depending on a certain principal components principle, the transformed features are partitioned into informative and less informative parts. When constructing each node of decision trees, a feature subspace is selected by stratified sampling method from the two parts. The PCA-SS based Random Forests algorithm, named PSRF, ensures enough informative features for each tree node, and it also increases the diversity between the trees to a certain extent. Experimental results demonstrate that the proposed PSRF significantly improves the performance of random forests when dealing with high dimensional data, compared with the state-of-the-art random forests algorithms.**

I. Introduction

Random forests are a type of ensemble methods for classification and regression that construct some identically randomized decision trees and make predictions by averaging the results from individual trees [1]. Random forests have developed an excellent reputation in the statistics and machine learning communities due to their high accuracy in various types of data [2]. However, high dimensional data have become more and more in the big data era as the ability to collect and store vast amounts of data becomes easier and increasingly common. In such situations, classical classification algorithms tend to become overwhelmed by the number of features and fail to get satisfactory results. Although random forests still can work, their performance cannot be comparable to theirs in moderate dimensional data.

It is well known that for high dimensional data only a small portion of features are truly informative [3]. However, random forests use a simple random sampling method to select feature subspace for each node when constructing each decision tree. Consequently, the selected feature subspace may contain few, if any, informative features and many less informative or non informative features. Thus, the decision tree inducted from such feature subspace will degrade its performance, and further influences the performance of random forests.

In this paper, to address the above issue, we propose a new feature subspace selection method in random forests for high dimensional data, which combines Principal Component Analysis technique (PCA) and Stratified Sampling method (SS) together, named by PCA-SS. Principal Component Analysis (PCA) is an unsupervised statistical technique that allows to extract axes of maximum variation (principal components) from the data [4]. Unlike the general usage of PCA, we reserve all the principal components except for the one whose magnitude is zero. That is to say, we keep all the information of the data without losing any potentially useful information. Stratified Sampling (SS) is a sampling method to introduce a stratification variable to divide the data into several subgroups and then randomly sample from each subgroup according to the ratio of target sample size to subgroup size, ensuring to obtain the representative sample of the data [5]. Moreover, in PCA-SS, it is exactly the principal component that serves as the stratification variable to divide the feature set. According to a specified principal components principle, the features transformed by PCA are divided into two parts, i.e. informative and less informative parts. Then, a feature subspace is selected through the stratified sampling method from these two parts for each node of the decision tree. In the following sections, we will describe the proposed PCA-SS for feature subspace selection in details.

In summary, the proposed PCA-SS not only reserves all the information from the original features, but also transforms the features to the directions of maximum variation of the data. Besides, PCA-SS selects the feature subspace in a stratified sampling manner which guarantees the enough informative features at any node of decision trees. Thus, the PCA-SS based random forests algorithm, named PSRF, can deal with the high dimensional data very well. Experimental results demonstrate that the proposed PSRF enhances the performance of random forests and outperforms other state-of-the-art random forests algorithms when handling high dimensional data.

The rest of the paper is organized as follows. Section II summarizes some related work. Section III reviews the original random forests algorithm. Section IV introduces the proposed

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

PCA-SS method and PSRF algorithm. Section V describes experimental setups and reports comparison results in details. Finally, Section VI concludes the paper.

II. Related work

To address the high dimensional data issue in random forests, numerous improved random forests algorithms have been proposed. Rotation Forests algorithm (RoF) introduces PCA technique into random forests in [6]. RoF randomly splits the features into several subsets and then PCA is applied to each subset which retains all the principal components. After that, the features are rotated accordingly to form the transformed features for the construction of decision trees in the forests. However, the feature subspace for each node is still selected through the random sampling method which yet suffers the problem of insufficient informative features for learning in the tree node. Besides, as a common feature extraction technique, there is another alternative usage of PCA shown in [7]. Their ensembles are built using the principal components calculated on the whole data. The first classifier uses the first S principal components. The second classifier uses the next S principal components and so on. Intuitively, the approach leads to that the latter decision trees are inducted from almost all less or non informative features which do not perform well and further hurt the ensemble performance. In addition, there is also some literature exposing PCA as an inadequate method for feature extraction and dimensionality reduction [8]. These kinds of algorithms only retain few large principal components of PCA. As only few principal components are retained, they all have the drawback that the most relevant discriminatory components corresponding to the small variance are possible to be discarded [9].

As a statistical sampling method, stratified sampling has been also adopted in a large number of literature about the random forests [10], [11], [12]. They mainly concentrate on how to partition the features into several parts properly based on different criteria. For example, Stratified Random Forests algorithm (SRF) is proposed in [11]. SRF uses the weight of Fisher Discriminant Projection [13] to divide the feature set into two subsets, i.e. strong informative and weak informative feature subsets. Similarly, a subspace selection Random Forests algorithm (ssRF) is presented in [12], which is based on a statistical criterion to partition the features into three parts. Firstly, ssRF applies p-value to assess the feature importance [14] on finding a cut-off between informative and less informative feature subsets. Secondly, the set of informative features is then further partitioned into two subsets, highly informative and informative feature subsets, using some statistical measures, e.g. Spearman rank test [15] for regression problem and χ^2 statistic [16] for classification problem. Moreover, there is another weighting features mechanism to replace stratified sampling method [17], which adds different weights to different features. The weights are calculated with respect to the correlation between the features and the class target. The resulting weights are treated as a probability by which a feature will be selected for inclusion in a feature subspace. However, the chances of introducing more correlated trees are also increased since the features with large weights are likely selected repeatedly.

III. Random forests

In this section, we briefly review the random forests framework. The more comprehensive review of the random forests methodology can refer [1] and [18].

Random forests were originally proposed by Breiman in 2001 [1], inspired by the previous work of the feature selection technique [19], random subspace method [20], and the approach of random split selection [21].

Generally, random forests are built by combining several decision trees, each of which is trained in isolation. To be specific, the random forests algorithm is usually described as follows: Given a data set $\mathcal{D} = \{(\boldsymbol{X_i}, Y_i), \boldsymbol{X_i} \in \mathbb{R}^D, Y_i \in \mathcal{Y}\}_{i=1}^N$, where $\boldsymbol{X_i}$ is one instance with D dimensional features and Y_i is the target, $\mathcal{Y} \in \{1, 2, \ldots, C\}$ with C being the number of classes. N is the number of training instances.

1) Use bagging [22] to draw M bootstrap data sets $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_M\}$ from the training data $\mathcal{D}$, where M is the number of trees in the forests.
2) For each data set $\mathcal{D}_m (m = 1, 2, \ldots, M)$, a decision tree is built using the Classification And Regression Tree algorithm (CART) [23]. At each node of the decision tree, randomly sample a subspace of p features $(p < D)$ from D features, and compute all possible split points based on the selected subspace of p features. The best split (e.g., the largest Gini impurity decrease) is utilized to partition the data and grow the tree. The procedure is applied recursively until the stopping criteria are satisfied, i.e., all data are pure with respect to the class; there are identical values for each feature; or the number of instances remaining in the node is less than a predefined threshold.
3) Combine the M decision trees into a random forests ensemble, and make the classification decision in a majority vote manner.

The performance of random forests is dependent on the performance of each decision tree and the diversity between the decision trees in the forests. Breiman formulates the generalization error (err) of random forests classifier [1], which is bounded by

$$err \leq \frac{\bar{\rho}(1 - s^2)}{s^2}, \tag{1}$$

where $\bar{\rho}$ is the average correlation between the decision trees and s is the average strength of the decision trees. Thus, the random forests algorithm with excellent generalization performance has two properties: high accuracy of each decision tree and low correlation (high diversity) between the decision trees. Specifically, Breiman employs two randomization procedures in random forests to achieve the diversity [1]. One is the training instances bagging for each individual tree, the other is the randomly selected subspace of features for splitting at each node within a tree.

IV. The proposed PCA-SS based random forests algorithm

In this section, we firstly analyze the drawback of random sampling method for feature subspace in original random forests. Secondly, we introduce the proposed PCA-SS method for feature subspace in random forests for high dimensional data. Finally, we describe the proposed PCA-SS based random forests algorithm (PSRF).

A. The Drawback of Random Sampling Method

As described above, when the number of possible features is huge and the percentage of truly informative features is small, one problem of the feature subspace selected by random sampling method arises.

Suppose that there are D features in the data and only H features are informative for the classification purpose. The remaining $D - H$ features are non informative. Then, at any node of the decision tree, p features are selected by resampling randomly and uniformly to form the feature subspace. The probability $\mathbb{P}$ of forming a subspace of p features $(p > 1)$ without any informative features is given by

$$\begin{aligned}\mathbb{P} = \frac{C^p_{D-H}}{C^p_D} &= \frac{(1-\frac{H}{D})\ldots(1-\frac{H}{D}-\frac{p}{D}-\frac{1}{D})}{(1-\frac{1}{D})\ldots(1-\frac{p}{D}-\frac{1}{D})} \\ &\approx \left(1-\frac{H}{D}\right)^p. \end{aligned} \tag{2}$$

In high dimensional data, we know that $D \gg H$, and therefore (2) is close to 1. That is to say, there is a high probability that a feature subspace created by the simple random sampling method will not contain any informative features. As a result, the performance of the base decision trees will degrade and further affect the average strength of decision trees. According to (1), it will reduce the performance of random forests.

In order to address the issue, we modify the selection of feature subspace from two aspects in this paper. One alteration is on feature transformation aspect, the other alteration is on stratified sampling aspect. Combing them together not only enhances the strength of the individual decision trees, but also increases the diversity between the decision trees.

B. Principal Components Analysis (PCA) Method

PCA is a common feature extraction technique which produces new extracted features from the original features [4]. Thus, it can be regarded as a procedure of the feature transformation. Given the training data $\mathcal{D} = \{(\boldsymbol{X_i}, Y_i), \boldsymbol{X_i} \in \mathbb{R}^D, Y_i \in \mathcal{Y}\}_{i=1}^N$ whose feature set is F, firstly, the bootstrap technique is carried out to draw a data set $\mathcal{D}_m (m = 1, 2, \ldots, M)$ with the same feature set F. Secondly, for each data set $\mathcal{D}_m$, F is randomly split into L subsets denoted by $\mathcal{D}_m^{(l)} (l = 1, 2, \ldots, L)$ with the feature subset $F^{(l)}$ (L is a parameter of the algorithm). The L subsets may be disjoint or intersecting. To maximize the chance for high diversity, we choose disjoint subsets. To be simplified, suppose that L is a factor of D so that each subset $\mathcal{D}_m^{(l)}$ contains $F^{(l)} = D/L$ features. Finally, the PCA is applied to each subset $\mathcal{D}_m^{(l)}$.

The reason behind the feature partition can be explained from two sides. One is to reduce the dimension of each data set $\mathcal{D}_m$ for PCA calculation, the other is to involve randomness into subset $\mathcal{D}_m^{(l)}$ to avoid similar results of PCA from the entire data set. The feature transformation procedure further increases the diversity of different trees beyond the diversity introduced by bootstrap technique.

In the following, we briefly introduce the procedure of PCA. Given the subset $\mathcal{D}_m^{(l)}$ with the dimension $N \times F^{(l)}$ whose rows represent instances and columns represent features, first of all, we calculate the mean of the subset by

$$\bar{\boldsymbol{x}} = \frac{1}{N} \sum_{\boldsymbol{x} \in \mathcal{D}_m^{(l)}} \boldsymbol{x}, \tag{3}$$

where $\boldsymbol{x}$ is an instance (row vector) of subset $\mathcal{D}_m^{(l)}$. Then, we compute the covariance matrix as follows:

$$\Sigma = \frac{1}{N-1} \sum_{\boldsymbol{x} \in \mathcal{D}_m^{(l)}} (\boldsymbol{x} - \bar{\boldsymbol{x}})^T (\boldsymbol{x} - \bar{\boldsymbol{x}}), \tag{4}$$

where Σ is exactly the covariance matrix of subset $\mathcal{D}_m^{(l)}$. After that, we can compute the eigenvectors $\{\boldsymbol{u}\}$ and eigenvalues $\{\lambda\}$ of the Σ. The eigenvectors are also called the principal components. There are many ready-made algorithms to compute the eigenvectors and eigenvalues of a matrix effectively and efficiently [24], [25], [26]. Through stacking the eigenvectors in columns in the decreasing order of the corresponding eigenvalues, we obtain the transformation matrix U. The transformed subset is

$$\mathcal{D}_m^{\prime(l)} = \mathcal{D}_m^{(l)} U. \tag{5}$$

Note that we reserve all the principal components due to the fact that some principal components may contain the discriminatory information for classification, though their eigenvalues may be small [9]. That is to say, unlike other usage of PCA, we do not lose any information of the data.

The above procedure is also applied to all the other subsets, finally, we get the transformed data set $\mathcal{D}'_m$ for each decision tree. Generally, decision trees are constructed by virtue of doing recursive splitting of data with splits based on a single feature. So by transforming the data to directions of maximum variation of the covariance matrix, it will be easier to make decision boundaries between the class distributions and further enhance the performance.

C. Stratified Sampling (SS) Method

As stated above, the key of the stratified sampling method is the criterion to partition the set into several subgroups. In this paper, we choose the principal components principle as the criterion. To be specific, it is the cumulative ratio of eigenvalues that divide the transformed feature set into informative and less informative parts. Note that principal components (eigenvectors) and eigenvalues have the correspondence of one to one. Generally, assuming $\lambda_1, \lambda_2, \ldots, \lambda_{F^{(l)}}$ are the eigenvalues of the covariance matrix Σ which have been

Algorithm 1 PSRF algorithm

```
1: Input: Training data $\mathcal{D}$, number of trees $M > 0$, number of feature subsets $L > 0$, the ratio $R$.
2: Output: The class label of the test instance $\boldsymbol{x}_t$.
3: for $m = 1, 2, \ldots, M$ do
4:     Use bagging to generate data set $\mathcal{D}_m$
5:     Split feature set $F$ of $\mathcal{D}_m$ into $L$ subsets.
6:     for each feature subset $F^{(l)}(l = 1, 2, \ldots, L)$ do
7:         Apply PCA to obtain transformed data using (3), (4) and (5).
8:         Divide the features into $A_1$ and $A_2$ parts according to the ratio $R$ using (6).
9:     end for
10:    Obtain the transformed data $\mathcal{D}'_m$ and two final feature sets $A_1$ and $A_2$
11:    while not satisfying the stop condition do
12:        Select feature subspace in a stratified sampling manner from $A_1$ and $A_2$ using (7) and (8).
13:        Compute all the possible splits based on the selected feature subspace
14:        Choose the split which achieves the largest Gini impurity decrease to split the data and grow the tree.
15:        Go to line 11 for recursively growing the tree
16:        // The stopping condition is the same as Breiman's random forests algorithm which is presented in Section III.
17:    end while
18:    //One of the decision trees in the forests is constructed completely.
19: end for
20: Transform the features of $\boldsymbol{x}_t$ and compute the estimate label through the majority vote from the trees in the random forests.
21: Return: The class label of $\boldsymbol{x}_t$
```

sorted in a decrease order, we define a cumulative ratio R of eigenvalues:

$$R = \frac{\sum_{j=1}^{r} \lambda_j}{\sum_{j=1}^{F^{(l)}} \lambda_j}. \tag{6}$$

The cumulative ratio R represents that we retain the $100R\%$ of the variance in the data if the former r eigenvalues are retained. R is also a parameter of the algorithm.

Therefore, according to the cumulative ratio R, for the transformed subset $\mathcal{D}_m^{'(l)}$, we assign the former r eigenvalues corresponding features to the informative part A_1 and the latter $F^{(l)} - r$ eigenvalues corresponding features to the less informative part A_2. The above procedure is also applied to other subset $\mathcal{D}_m^{'(l)}$ to get the final A_1 and A_2 parts.

The stratified sampling of a subspace with p features can now be accomplished by selecting individual features at random from the two parts. The features are selected in proportion to the relative sizes of the two parts. That is, we randomly selected

$$p_1 = p \times \frac{D_1}{D}, \tag{7}$$

$$p_2 = p - p_1, \tag{8}$$

where D_1 is the number of features in A_1, p_1 and p_2 are the number of samples from the two parts A_1 and A_2 respectively. These samples are then merged to form a feature subspace for tree construction. We specify p must contain at least one from each part. In this way, we can guarantee that the subspace at any node contains both informative and less informative features, therefore assuring a qualified tree.

D. PCA-SS Based Random Forests Algorithm (PSRF)

In this part, we summarize the above PCA-SS based random forests algorithm (PSRF) in a pseudo-code format in Algorithm 1.

The proposed PSRF algorithm combines PCA and stratified sampling method together. It improves the performance of random forests from two aspects. Firstly, PCA transforms the original features to a new feature space, which makes the classification easier. Secondly, stratified sampling ensures the feature subspace for each node contains enough informative features to learn, which accounts for the promotion in the performance especially for high dimensional data. Moreover, from a perspective of the generalization error bound in (1), PCA and stratified sampling methods not only increase the strength of base decision trees but also increase the diversity between the decision trees. Thus, the proposed PSRF can provide more accurate results than conventional and other variants of random forests algorithms, which will be demonstrated in the following section.

V. Experiments

We implement the proposed PSRF algorithm in Python referring to the scikit-learn package [27]. As for the comparison random forests algorithms, besides the original random forests, we also compare the proposed algorithm with other improved random forests algorithms for high dimensional data. The comparison algorithms are listed as follows:

TABLE I
Detailed information of the benchmark data sets

Dataset	Instances	Features	Classes
Glass [1]	214	10	7
Vehicle [1]	946	18	4
Image [1]	2310	19	2
Madelon [1]	2600	500	2
ColonTumor [2]	62	2000	2
CentralNervousSystem [2]	60	7129	2
Arcene [1]	200	10000	2
Amazon [1]	1500	10000	50
OvarianCancer [2]	253	15154	2
BreastCancer [2]	97	24481	2

[1] UCI data sets.
[2] Biomedical data sets.

TABLE II

CLASSIFICATION ACCURACY ($ACC\%$) OF DIFFERENT RANDOM FORESTS ALGORITHMS ON DIFFERENT DATA SETS

Dataset	RF (%)	RoF (%)	SRF (%)	ssRF (%)	PSRF (%)
Glass [1]	76.73	77.12	77.18	77.51	**79.02**
Vehicle [1]	74.95	76.25	75.04	75.53	**77.50**
Image-segmentation [1]	97.79	98.09	97.96	98.14	**98.26**
Madelon [2]	71.38	76.15	74.11	80.46	**88.35**
ColonTumor [2]	78.81	80.95	83.57	82.38	**85.48**
CentralNervousSystem [2]	61.67	63.33	66.67	68.33	**71.67**
Arcene [3]	79.50	81.09	82.40	81.52	**85.55**
Amazon [3]	63.67	68.40	66.07	67.67	**70.13**
OvatianCancer [3]	98.82	99.00	99.40	99.52	**99.93**
BreastCancer [3]	60.67	63.00	66.33	67.56	**71.89**
Mean	76.40	78.33	78.87	79.86	**83.07**

[1] Low dimensional data
[2] Moderate dimensional data
[3] High dimensional data

1) Rotation forests algorithm (RoF) [6]: PCA with random sampling feature subspace.
2) Stratified random forests algorithm (SRF) [11]: stratified sampling feature subspace without PCA.
3) Subspace selection random forests algorithm (ssRF) [12]: feature subspace sampled from subgroups divided by statistical measures in a stratified manner.
4) Random forests algorithm (RF) [1]: the original random forests.

Besides, in order to quantitatively evaluate the performance, we use the metric of classification accuracy (ACC).

A. Data Sets

A series of experiments are conducted on various data sets in Table I, including 6 UCI data sets [28] and 4 biomedical data sets from I^2R repository [29], remarked in superscript 1 and 2 respectively. Particularly, the biomedical data sets usually have high dimensional features and only a small number of instances. These data sets are ranked by the number of features from small to large. The table can be divided into three parts. The upper part includes three low dimensional data sets with binary and multi-class classification. The middle part includes three moderate dimensional data sets with binary classification. The bottom part includes four high dimensional data sets with binary and multi-class classification. In summary, these data sets are representative to test the performance of random forests from various aspects.

B. Experimental Setup

As PCA is defined for numeric features, discrete features should be converted to numeric ones for RoF and PSRF algorithms. The one-of-K or one-hot encoding is a common implementation in machine learning algorithms which replaces each categorical feature by K binary features encoded numerically as 0 and 1, where K is the number of possible categories of the feature. This encoding unifies all the inputs so that PCA can be carried out on any subsets of features.

As for the number of the decision trees M in the forests, we all set $M = 100$ for the above 5 algorithms. Besides, the size of a feature subspace for each node is set to $p = \sqrt{D}$ for all the algorithms. With respect to the number of feature subsets L in RoF and PSRF, we fix the number of features in each feature subset $F^{(l)} = 50$ rather than fix L directly. The ratio R in PSRF is set to $R = 0.8$. With respect to SRF, we employ the average weights of Fisher discriminant projection to be the stratification threshold as in [11]. With regard to ssRF, we keep the same parameters in [12], for example, the feature importance is computed over 30 times of random forests and a significance level of 0.05 is set to be the threshold for p-values.

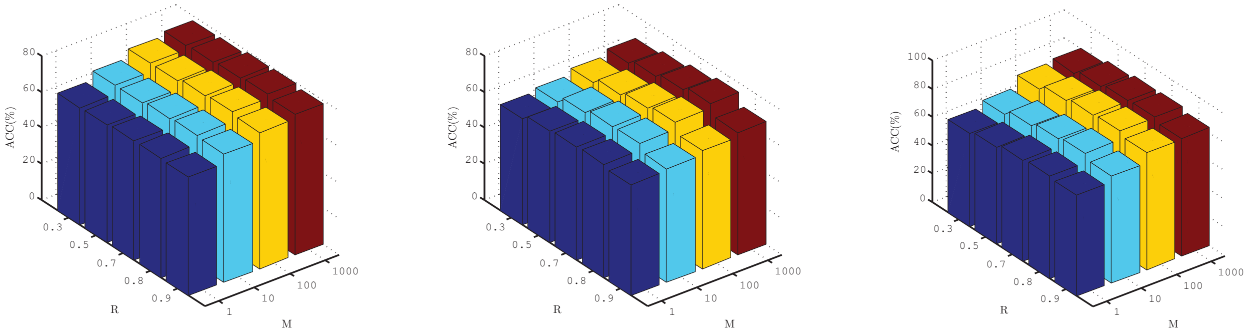
Finally, for each data set, a 10 times 10-fold cross validation is carried out to reduce the influence of randomness.

C. Comparisons of Classification Accuracy

Table II reports the detailed results of the proposed PSRF and the other random forests algorithms. The highest accuracy on each data set is in boldface. Besides, the mean accuracy on all data sets is also summarized at the bottom. It can be seen that PSRF achieves higher classification accuracy on all the data sets compared with RF, RoF, SRF and ssRF. The mean accuracy of PSRF (83.07%) is also higher than RF (76.40%), RoF (78.33%), SRF (78.87%) and ssRF (79.86%). Four Wilcoxon signed ranked tests [30] on accuracy, i.e. PSRF vs RF, RoF, SRF and ssRF respectively, all reject the null hypothesis of equal performance at a p-value less than 0.01. Thus, PSRF achieves significantly better performance than other random forests algorithms.

To be specific, on the top three low dimensional data sets, all the random forests algorithms obtain the comparable accuracy. The reason is that almost all the features in the low dimensional data set are informative, thus, the random sampling feature subspace method of the original random forests can handle these kinds of data set well. Therefore, the technique of PCA and stratified sampling will not provide significant promotion in the accuracy.

In the following seven moderate and high dimensional data sets, we can note that the performance of the original RF

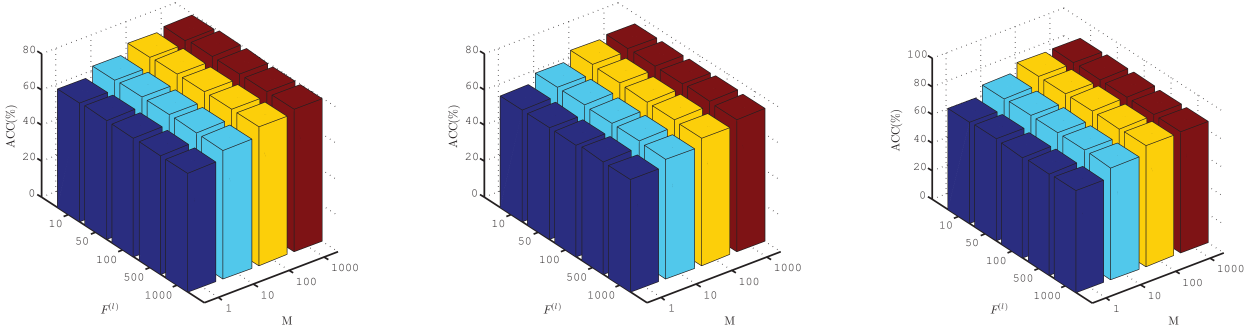


(a) Vehicle data set

(b) CentralNervousSystem data set

(c) Arcene data set

Fig. 1. Classification accuracy $(ACC\%)$ of PSRF with different R and M for various data sets $(F^{(l)} = 50)$



(a) Vehicle data set

(b) CentralNervousSystem data set

(c) Arcene data set

Fig. 2. Classification accuracy $(ACC\%)$ of PSRF with different $F^{(l)}$ and M for various data sets $(R = 0.8)$

becomes worse, and the advantages of the PCA and stratified sampling technique are shown. RoF, SRF, ssRF and PSRF all obtain higher classification accuracy compared with RF, and among them PSRF achieves the highest accuracy. Specifically, the remarkable promotions are indicated on the CentralNervousSystem and BreastCancer data sets, whose accuracies are improved up to 10% by PSRF comparing to RF. Moreover, comparing SRF with ssRF which are two kinds of stratified sampling algorithms using different subgroup partition criteria, we find that they obtain the similar results. The phenomenon indicates that different criteria to partition the feature into several subsets cannot supply more improvements beyond what the stratified sampling method can provide. That is to say, it is necessary to transform the original features to a new feature space for a further improvement in random forests. The performance of the algorithm RoF and PSRF have proven the above point about the usefulness of the feature transformation.

In summary, as shown in Table II, for high dimensional data sets, RoF with a single PCA technique improves the performance, and SRF as well as ssRF with a single stratified sampling technique also improve the performance. Therefore, it is natural that PSRF improves the performance one step further due to the fact that PSRF combines the benefits of PCA to transform features and stratified sampling to obtain representative feature subspace.

D. Cross-test for Parameter Settings

In addition, there are several parameters in PSRF, such as the number of the decision trees M, the number of feature subsets L, the size of the feature subspace p and the ratio R. Generally, the size of the feature subspace p has been analyzed a lot in the literature, thus we choose the default value $p = \sqrt{D}$ in [1]. As for the remaining three parameters, we conduct a series of experiments to evaluate the influence of parameters on the PSRF algorithm. As illustrated in subsection V-B, we use the parameter $F^{(l)}$ in place of L. Thus, two pairs of parameters are cross-tested. The first pair is the cumulative ratio R and the number of trees M, and the second pair is the number of features in each subset $F^{(l)}$ and the number of trees M.

We select three representative data sets which are composed of low, moderate and high dimensional data sets, i.e. Vehicle, CentralNervousSystem and Arcene data sets. We test R among

$\{0.3, 0.5, 0.7, 0.8, 0.9\}$, $F^{(l)}$ among $\{10, 50, 100, 500, 1000\}$ and M among $\{10^0, 10^1, 10^2, 10^3\}$.

The results are shown in Fig. 1 and 2. We find that the accuracy increases gradually tending to a stable value as the number of trees M increases in Fig. 1-2. As for the ratio R which is the criterion to partition the feature set into informative and less informative parts, we find that the accuracy first goes up and then goes down as the ratio R increases. It can be easily explained through recalling the procedure of stratified sampling in PSRF. The ratio R will affect the number of features in A_1 as well as A_2, and further influence the following stratified sampling procedure. If R is small, the informative subset is small which leads to a feature subspace with insufficient informative features, so the accuracy is low. If R is large, the feature subspace may ignore the discriminatory information contained in the less informative subset. Therefore, the ratio R plays a trade-off between the informative and less informative parts. Fig. 1 indicates that the best ratio R is 0.8, which is the same as our parameter settings in the subsection V-B. Moreover, Fig. 2 indicates that the accuracy almost keeps the same as $F^{(l)}$ changes. That is to say, PSRF is not sensitive to the parameter $F^{(l)}$.

VI. Conclusions

In this paper, we propose a novel feature subspace selection method in random forests for high dimensional data, namely Principal Components Analysis and Stratified Sampling method (PCA-SS), and a corresponding PCA-SS based random forests algorithm called PSRF. PSRF combines the advantages of PCA and stratified sampling method. PCA transforms the original features to a new feature space and stratified sampling ensures the feature subspace for each node contains enough informative features. In summary, PSRF not only increases the strength of base decision trees but also increases the diversity between the decision trees. Empirically, the performance of PSRF on several benchmark data sets demonstrates that PSRF outperforms other state-of-the-art random forests algorithms for high dimensional data.

Acknowledgments

This research is supported in part by the Major State Basic Research Development Program of China (973 Program, 2012CB315803), the National Natural Science Foundation of China (61371078), and the Research Fund for the Doctoral Program of Higher Education of China (20130002110051).

References

[1] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[2] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, "A comparison of decision tree ensemble creation techniques," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 1, pp. 173–180, 2007.

[3] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[4] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[5] C.-E. Särndal, B. Swensson, and J. Wretman, "Model assisted survey sampling (springer series in statistics)," 2003.

[6] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 10, pp. 1619–1630, 2006.

[7] M. Skurichina and R. P. Duin, "Combining feature subsets in feature selection," in *Multiple classifier systems*. Springer, 2005, pp. 165–175.

[8] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.

[9] A. M. Martínez and A. C. Kak, "Pca versus lda," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 2, pp. 228–233, 2001.

[10] Q. Wu, Y. Ye, Y. Liu, and M. K. Ng, "Snp selection and classification of genome-wide snp data using stratified sampling random forests," *NanoBioscience, IEEE Transactions on*, vol. 11, no. 3, pp. 216–227, 2012.

[11] Y. Ye, Q. Wu, J. Z. Huang, M. K. Ng, and X. Li, "Stratified sampling for feature subspace selection in random forests for high dimensional data," *Pattern Recognition*, vol. 46, no. 3, pp. 769–787, 2013.

[12] T.-T. Nguyen, H. Zhao, J. Z. Huang, T. T. Nguyen, and M. J. Li, "A new feature sampling method in random forests for predicting high-dimensional data," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2015, pp. 459–470.

[13] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero norm with linear models and kernel methods," *The Journal of Machine Learning Research*, vol. 3, pp. 1439–1461, 2003.

[14] T.-T. Nguyen, J. Z. Huang, and T. T. Nguyen, "Two-level quantile regression forests for bias correction in range prediction," *Machine Learning*, pp. 1–19, 2014.

[15] J. L. Myers, A. Well, and R. F. Lorch, *Research design and statistical analysis*. Routledge, 2010.

[16] P. E. Greenwood and M. S. Nikulin, *A guide to chi-squared testing*. John Wiley & Sons, 1996, vol. 280.

[17] D. Amaratunga, J. Cabrera, and Y.-S. Lee, "Enriched random forests," *Bioinformatics*, vol. 24, no. 18, pp. 2010–2014, 2008.

[18] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends® in Computer Graphics and Vision*, vol. 7, no. 2–3, pp. 81–227, 2012.

[19] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural computation*, vol. 9, no. 7, pp. 1545–1588, 1997.

[20] T. K. Ho, "The random subspace method for constructing decision forests," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 8, pp. 832–844, 1998.

[21] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning*, vol. 40, no. 2, pp. 139–157, 2000.

[22] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[23] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[24] D. M. Witten, R. Tibshirani, and T. Hastie, "A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis," *Biostatistics*, p. kxp008, 2009.

[25] P.-G. Martinsson, V. Rokhlin, and M. Tygert, "A randomized algorithm for the decomposition of matrices," *Applied and Computational Harmonic Analysis*, vol. 30, no. 1, pp. 47–68, 2011.

[26] F. P. Anaraki and S. Hughes, "Memory and computation efficient pca via very sparse random projections," in *Proceedings of The 31st International Conference on Machine Learning*, 2014, pp. 1341–1349.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[28] M. Lichman, "UCI machine learning repository," http://archive.ics.uci.edu/ml, 2013.

[29] J. Li, H. Liu, and L. Wong, "Mean-entropy discretized features are effective for classifying high-dimensional biomedical data," in *3rd ACM SIGKDD Workshop on Data Mining*. Citeseer, 2003.

[30] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.