On Oblique Random Forests

Bjoern H. Menze[1,2], B. Michael Kelm[1], Daniel N. Splitthoff[1], Ullrich Koethe[1], and Fred A. Hamprecht[1]

[1]Interdisciplinary Center for Scientific Computing, University of Heidelberg, Germany
[2]Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, USA

Abstract. In his original paper on random forests, Breiman proposed two different decision tree ensembles: one generated from "orthogonal" trees with thresholds on individual features in every split, and one from "oblique" trees separating the feature space by randomly oriented hyperplanes. In spite of a rising interest in the random forest framework, however, ensembles built from orthogonal trees (RF) have gained most, if not all, attention so far.
In the present work we propose to employ "oblique" random forests (oRF) built from multivariate trees which explicitly learn optimal split directions at internal nodes using linear discriminative models, rather than using random coefficients as the original oRF. This oRF outperforms RF, as well as other classifiers, on nearly all data sets but those with discrete factorial features. Learned node models perform distinctively better than random splits. An oRF feature importance score shows to be preferable over standard RF feature importance scores such as Gini or permutation importance. The topology of the oRF decision space appears to be smoother and better adapted to the data, resulting in improved generalization performance. Overall, the oRF propose here may be preferred over standard RF on most learning tasks involving numerical and spectral data.

1 Introduction

Random forests have gained popularity in high-dimensional and ill-posed classification and regression tasks, for example on micro-arrays [19], time series [37], or spectral data [39, 29], but also for inference in application such as image segmentation or object recognition in computer vision [9, 45]. Random forests are comparable in performance to many other non-linear learning algorithms. They often do well with little parameter tuning [16], and are able to identify relevant feature subsets even in the presence of a large amount of irrelevant predictors [6, 21, 25, 2]. More recently, additional properties of the random forest have gained interest, for example in feature selection [25, 34, 23, 44] or the explorative analysis of sample proximities [38].

The main idea of the random forest framework, proposed by Breiman in [5], is to learn many variable but unbiased base learners, and to reduce variance by pooling over a whole committee of predictors. This concept is familiar from bagging [3], when a large number of decision trees is learned from random subsets

of the training samples and their decisions are averaged in prediction. In random forests the correlation between individual base learners is further reduced by seeking at every node for the best prediction in a random subspace of the training data, similar to ideas from "random subspaces" [17] and "random splits" [10].

A random forest can be generated from two different kinds of trees. *Univariate decision trees* – such as CART or C4.5 – serve as base learners of the most popular random forest implementations. They separate feature space by hyperplanes that are *orthogonal* to single feature axes, resulting in the typical stair- or box-like decision surfaces of these classifiers. While this characteristic shape of the decision boundary might be advantageous for some data, one may argue that it is suboptimal for other – potentially leading to a substantial bias of the base learner. Collinear data with correlated features [23], for example, arising from spectra, time series, but also micro-arrays or image patches, may reside in subspaces that lie between the coordinate axes. In that case, class distributions may appear inseparable when marginal distributions are evaluated in the search for the best univariate split, and separating classes may require complex and deeply nested trees (Fig. 1). *Multivariate decision trees* – trees using decision surfaces at arbitrary, *oblique* orientations to the axes – may be better adapted to decisions in such subspaces [27], leading to decision boundaries that are less biased by geometrical constraints of the base learner. In his original paper [5], Breiman proposed the use of decision trees with oblique splits at random orientations, observing that this approach yielded "results never reached so far" on the data sets tested. Unlike their counterparts with univariate node models, however, these random forests have not met a wider interest, yet.

In the present work we propose to use random forests with regularized *oblique* model trees as base learners. Recent results on ensemble pruning showed an advantage of choosing base learners which are optimally adapted to the data [22]. So, rather than choosing *random* recursive splits, as Breiman suggests in [5], we focus on trees with *task optimal* recursive partitioning. In this we follow the idea also used, for example, in probabilistic boosting trees [43], which use "strong" discriminative model at each node to obtain overall better decision rules. Our approach is also related to "rotation forests" [33] where oblique split directions are sought from the principal components of feature subsets of the training data, reportedly improving results significantly in selected classification tasks. We differ from this approach by using supervised approaches to define optimal split directions. While we follow some of the ideas of [40, 41], we refrain from using global optimization techniques to train the oblique tree. Furthermore, we also perform experiments to understand the benefit of learned oblique node models. For splitting feature space at a node, a wide range of linear discriminative node models can be used, all employing somewhat different optimization objectives. We emphasize on regularized node models which may complement the good properties of the original random forest by Breiman [5] for classifying high-dimensional data with few samples – a domain where the random forest with orthogonal trees performs exceptionally well. Finally, we propose feature impor-

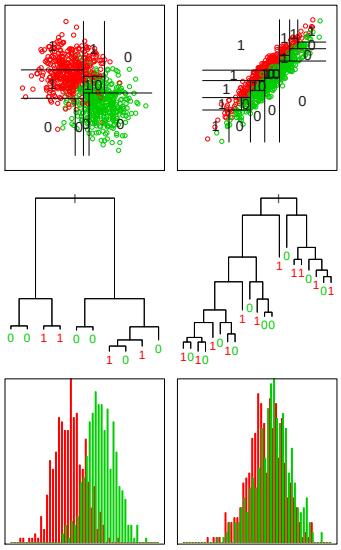

Fig. 1. Separation of correlated data using univariate decision trees. The left column shows a synthetic binary data set, the right column shows the same data after the samples have been artificially correlated. The top row shows the bivariate samples, and a segmentation using an orthogonal decision tree (black lines, and segments indicated by 0 and 1). The classification tree is visualized in the middle row. Marginal distributions of the horizontal axis (top row) are shown as histograms in the bottom row – these are the distributions evaluated at every node in a univariate decision tree. While the initial segmentation problem (left column) is simple, the strong overlap in the marginal distributions (right column) leads to highly nested decision rules and complex decision boundaries if features are correlated.

tance and visualization measures for the oblique random forest, similar to the ones proposed [5].

In the following we will shortly outline model-based *oblique random forests* (oRF) (Section 2). In subsequent experiments we will compare the oRF quantitatively against standard random forests with univariate trees (RF) and other related non-linear classifiers (Section 3), and try to understand properties of oblique random forests with learned node model (Section 4). We will finally propose feature importance and sample proximity measures derived from the oblique random forest (Section 5).

2 Oblique random forests

The oRF shares the ensemble generation processes with the "standard" RF [5]: For each tree a new set of samples is drawn randomly from the training data with replacement. At every recursive binary split, a new set of m_{try} features is sampled without replacement, and the optimal split is sought in the subspace spanned by these features. Differences to the standard procedure apply in the way optimal splits direction are sought at each node.

Oblique model trees For oRF, we rely on multivariate models for binary splits in each node. For a sample $\mathbf{x} = [x_1, \ldots, x_{m_{try}}]^T$ in a m_{try}-dimensional space, the decision f at the node m can be formulated as:

$$f_m(\mathbf{x}) : \beta_m^T \mathbf{x} > c_m \tag{1}$$

with coefficients β_m defining the projection for the split and threshold c_m. Inferring the optimal β_m is more difficult than the identification of a single optimal feature and an appropriate threshold for a split in the univariate case. Different criteria can be used to find the linear subspace (Fig. 2). Projections for

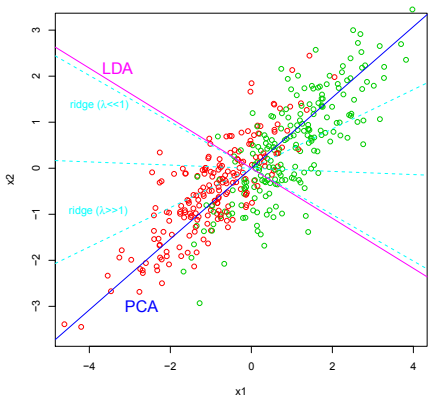

Fig. 2. Effect of regularization on split directions of the oblique node models. Lines represent normals of the discriminating hyperplanes. PCA seeks for oblique splits with maximal data support (first component shown here) and LDA for correlation with the class label. By using regularized regression a multitude of alternative subspaces with intermediate properties can be defined and evaluated. Regularization biases the optimal LDA-like projection towards one which has higher data-support, more directing towards the PCA-like solution.

linear splits may consider class label information only (as in logistic regression or linear discriminant analysis), they may align with the data variation (as with principal component analysis), or they may seek for an optimum in between, trading class label correlation and data support (as in ridge or partial least squares regression, or linear support vector machines). Constrained regression methods also enjoying popularity in the classification of high dimensional data. They perform well in tasks with less observations than predictors, and they may help to find splits when less than m_{try} samples reside in a node which often occurs in deep split nodes far from the root. We therefore choose ridge regression for our split model:

$$\beta_{ridge}(\lambda) \sim \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{2} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{P} |\beta_j|^2. \tag{2}$$

using regularization parameter λ. With this choice the node model is optimizing for [11]

$$\beta_{ridge}(\lambda') \sim \underset{||\beta=1||}{\operatorname{argmax}} corr^2(\beta X, Y) * \frac{var(\beta X)}{var(\beta X) + \lambda'}. \tag{3}$$

The regularization parameter λ allows the classifier to adapt to an optimal split direction β_{ridge} in between two extremes (Fig. 2): With $\lambda = 0$, β_{ridge} may point towards maximal correlation with class labels, similar to linear discriminant analysis (LDA). With $\lambda \gg 1$, it may point towards highest data variation and data support, similar to principal component analysis (PCA).

At each node m, the model parameter λ can be adapted to the out-of-bag samples available at that node. Then all samples X_m are projected into β_m and the optimal split-point c_m on the scores $s_m = X_m\beta_m$ is identified using the Gini impurity $I(s_m) = \frac{1}{2} * (1 - \sum_{k=0,1} P_k^2(s_m))$. The threshold c_m maximizing the decrease in Gini impurity (i.e., the minimum in $I(s_m < c_m) + I(s_m > c_m)$) is chosen and samples are separated accordingly. For both subsets the process of finding the best split is recursively iterated until both classes are separated.

Implementation of the oblique random forest In the random forest framework a large number of trees are combined. Two hyper-parameters control the generation of the decision trees in this ensemble: subspace dimensionality m_{try} and ensemble size n_{tree}. Parameter m_{try} determines the number of features sampled randomly at each individual node and the degree of randomness in the model generation. Parameter m_{try} has to be chosen to obtain a "sufficient" variability of the trees in the forest, ensuring that correlation between individual trees is as low as possible. In prediction new samples are pushed down each of the n_{tree} trees and are assigned the label in the terminal note and decisions can be pooled according to different schemes [32, 30]. Here we use the normalized number of votes, or probability $p \in [0, 1]$, which is relatively robust against over-fitting [5, 35, 32].

We implement two versions of the oRF in our experiment: a) *oRF-ridge* optimizing regularization parameter λ at every split, b) *oRF-lda* performing an unregularized LDA-like split at every node, and c) *oRF-rnd* with random oblique splits as proposed in [5]. Trees are generated as follows[1]: 1) For each tree a new set of samples is drawn randomly from the training data with replacement. 2) Random subspaces of dimension m_{try} are sampled without replacement for every node. 3) At every split, we scale variables to zero mean and unit variance to enhance the stability of the linear model. 4a) For *oRF-ridge*, ridge regression is tested using $\lambda = 10^i$ with $i = \{-5, -4, \ldots, 5\}$, and λ is optimized using the out-of-bag samples residing at the same node. 4b) For *oRF-lda* we set $\lambda = 0$ and use the resulting node model. 4c) For *oRF-rnd* we draw random values from a normal distribution (with zero mean and standard deviation of one) to obtain coefficients β of the node model beta under a similar prior distribution as in a L_2 constrained ridge regression. For all three oRF, samples are projected into subspace determined by the node model. 5) Optimal thresholds for splitting the data are sought using the Gini impurity on the fitted scores of the training data, and the samples are split accordingly. For each of the n_{tree} trees, steps 2)-5) are repeated until all classes are separated (oRF-lda, oRF-rnd), or no out-of-bag test samples are available for adapting λ any more (oRF-ridge).

3 Comparison of classification performances

In a first experiment we compare the performance of oRF, RF and other learning methods on different types of data to identify and compare specific properties of these classifiers.

Experimental data A number of binary benchmark problems are used for the evaluation: binary classification problems of the UCI data repository (data sets 1-15, Table 1), synthetic data sets (16-20) [4], binary groupings of handwritten digits (21-23) from the MNIST data base, binary data from detection problems in archaeological remote sensing (24-25) [26], from the analysis of magnetic resonance spectra for the detection of brain tumors or yeast infections (26-28, 32-36)

[1] Classifier publically available at `cran.r-project.org` in package *obliqueRF*.

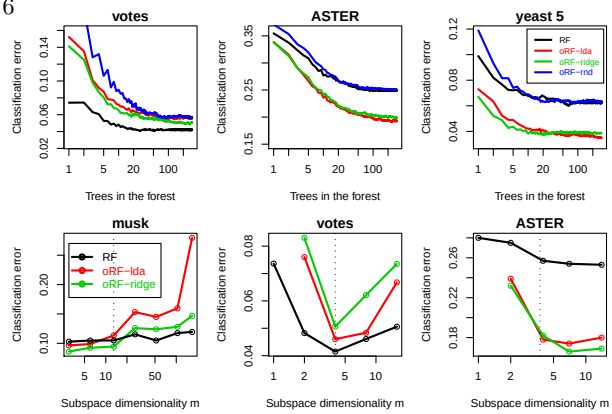

Fig. 3. Parametrization of the random forest. On most data sets default ensemble parameters for ensemble size n_{tree} (top) and subspace dimensionality m_{try} (indicated by the dotted vertical line, bottom) perform reasonably well.

[24, 23], from the analysis of infrared spectra for BSE detection from blood serum and food analysis (29-31, 37-40) [25, 23].

Alternative classifiers We choose seven alternative nonlinear classifiers for our comparison – most of them with some conceptual relation to oblique random forests. We test two other decision tree ensembles (Table 1): *adaboost* [12], the standard random forest (*RF*) [20] and a version of the extremely random forests [13] with $m_{try} = 1$ (*RF-rnd*), all relying on univariate decision trees. We also test the performance of single pruned classification trees (*CART*), support vector machines with radial basis function (*svm*) and a k-nearest-neighbors classifier (*knn*). For RF, the optimal number of trees is found in an internal cross-validation of the classification error testing $10 * 2^i$ trees with $i = 1 \dots 6$. Trees are grown to full depth. For CART, model complexity is optimized in a ten-fold cross-validation. For the support vector machine (*svm*) the kernel width is obtained in a three-fold cross-validation testing the quartiles of $|x - x'|^2$ [7] and regularization ("cost") is found in another 3-fold cross-validation evaluating the range of $\lambda = 10^{-5 \dots 5}$. A binomial model is modeled on top of the binary decisions [31]. Boosting is "discrete" adaboost, with exponential loss, and in conjunction with a bagging of the samples. We test the performance for $10 * 1 \dots 5$ iterations in a 3-fold internal cross-validation. The knn classifier is tested in a threefold cross-validation testing 2^i neighbors with $i = 0 \dots log_2(P)$.

Processing time Processing time is critical for the generation of large ensembles of classifiers. Using an implementation in native GNU R it took longer to train an oRF than training an RF (using a Fortran implementation), or the SVM (C); it was still faster than adaboost (C). Growing 100 oblique trees took about 10-100s for most of the learning tasks (on a standard personal computer) with *lda* at the lower end of this time range and *ridge* – requiring parameter tuning – at the upper. Training the node models on a subsampled set of observations, as done in computer vision applications [9, 45], may reduce the overall computation time significantly for large data sets.

Choice of oRF default parameters Random forests are relatively insensitive to the choice of model parameters, and we want to apply all oRF with the same default parameters. We test the effect of forest size, n_{tree}, on the overall classification performance of the classifiers (as determined in a ten-fold cross-validation;

Fig. 3, left). Overtraining by increasing n_{tree} could hardly be provoked for any random forest on any data set. Little improvement in classification accuracy is observed for $n_{tree} > 100$. We chose to set $n_{tree} = 300$ for all further experiments. The second tuning parameter is m_{try} – the number of features which are randomly sampled at each node. A popular default choice for RF is $m_{try}^{def} = \sqrt{P}$, with P being the number of features [20]. We find this default to perform reasonably well (Fig. 3, right), although larger differences can be observed for spectral data sets. In all further experiments we use $m_{try} = \sqrt{P}$.

Evaluation We apply all classifiers to the data set in ten repeats of a ten-fold cross-validation (Table 1) and evaluate both the mean accuracy and the receiver-operator-characteristics with its area-under-the-curve (ROC AUC). ROCs are calculated on the complete test data, and individually on each of the ten resampled data sets, to have a distribution of ten different ROCs which is used to measure significance of differences. To this end we use nonparametric statistical tests. We first identify the best method for a particular data set (defined by the highest mean classification accuracy, underlined in Table 1), and then test whether other classification results differed significantly from this one "best" result. We use paired Cox-Wilcoxon tests at 5% level (on the 100 test sets for classification accuracy and 10 repeats for AUC ROC) to compare the performance between the "best" classifier, and any other classifier applied to the same data set [18]. Classification results which do *not* differ significantly from the approach with the highest accuracy, are indicated by bold font in Table 1.

We analyze the general behavior of the algorithms over data which share certain properties. Trees with oblique and orthogonal perform differently on data with factorial and numerical features [8]. Random forest perform well in applications with many irrelevant predictors and few observations [23]. So, we group the data into three classes (Table 1): data with factorial or discrete features ("factorial data", data sets 1-10 in Table 1), data with continuous numerical features ("numerical data",11-20), and data with many correlated numerical features and few samples ("spectral data", 21-40).

Results are shown in Table 1, the frequency how often a particular method performed best or comparable to the best within the three data classes is reported in Table 2. The optimal choice for factorial data are methods evaluating univariate splits – adaboost, RF and in some classification tasks even CART performed best. They rank – in terms of classification performance – in front of all other methods, with adaboost being first. The advantage on factorial features may be expected as univariate trees consider the relative ordering of observations only. They are insensitive to the arbitrary assignment of numerical values to different factors. On numerical data oblique random forests perform slightly better than alternative classifiers (e.g., ada, SVM). Those oRF with regularized node model perform slightly better than the other two in terms of class separation (i.e, ROC). On spectral data, oRF is a clear first, both in terms of accuracy and ROC. SVM, RF or adaboost perform well on few data sets (*MRS tumor*, *IR yeast 2*) and here most oRF perform equally well. An advantage of oRF-ridge

		svm	knn	CART	adaboost	RF	RF-rnd	oRF-rnd	oRF-lda	oRF-ridge	svm	knn	CART	adaboost	RF	RF-rnd	oRF-rnd	oRF-lda	oRF-ridge
1	chess	2.8	5.4	2.4	**0.6**	1.4	52.2	23.5	4.1	2.7	0.3	1.2	0.9	**0**	0.1	33.3	0.9	0.4	0.3
2	credit	15.3	32	14.8	**12.9**	**12.7**	17.9	15.4	**13**	44.5	10.3	27.3	15.4	**6.7**	**6.8**	7.5	8.3	**7**	53.3
3	heart	23	35.5	21.3	**18.3**	**17.8**	40.6	20	**18**	**17.9**	15.7	30.9	20.5	11.4	10.4	13.5	12.7	**9.9**	**9.9**
4	hepatitis	20.9	21.6	**20.3**	**24.4**	**17.8**	**18.7**	**19.1**	**19.3**	20.6	47	42.2	28.7	18.9	**14.6**	19.5	16.2	15.7	50
5	monks3	3.6	**1.6**	**1.1**	**1.2**	2.4	43.3	5.7	3.5	**1.3**	1.4	**0.9**	1.7	**0.8**	1.1	5.8	**0.8**	1.1	1
6	promotergene	17.1	20.3	30.4	**9.6**	**11**	50.1	20.8	16.9	18.5	8.8	12	31.3	**3.9**	**4**	32	12.8	7.3	7.4
7	tic-tac-toe	12	16.1	26.5	**0**	4.7	65.3	12.7	15.2	14.8	4	32	31.1	**0**	0.3	73.3	4.9	3.5	2.8
8	titanic	**21.6**	23.3	23	**21.3**	22.7	30.2	26.8	22	24.7	20.1	20.1	25.5	**16.1**	16.8	22.1	20.3	17.6	18.6
9	votes	9.3	12.5	**4.4**	**4.5**	**4.2**	7.7	5.7	5.6	5.1	4.2	8.7	6.1	1.4	**0.9**	1.8	1.4	1.4	1.3
10	cancer	4.3	**3.4**	6.2	3.7	**3.3**	**3**	**2.8**	**3**	**3**	2.3	1.3	4.8	1	0.9	0.9	**0.7**	0.9	0.9
11	ionosphere	**5.1**	14	10.5	**6.4**	**6.5**	7.5	**5.6**	**5.4**	**5.6**	2	16.9	12.3	3.3	2.1	2.1	**1.7**	**1.7**	**1.6**
12	sonar	17.5	18.2	27.2	**13.1**	**16.1**	16.3	**14.4**	18	18.2	7.6	17.3	25.7	**6.1**	6.8	**6.4**	**5.8**	6.5	**6.2**
13	musk	10.4	14.6	19.6	**8.8**	10.6	17.9	10.2	10.6	10.4	3.8	12.9	15.9	**2.9**	4.4	5.3	3.7	5	4.5
14	liver	30.9	32	34.1	**26.8**	**26.8**	**27.8**	27.8	**25.8**	**26.2**	26.6	31.3	38.2	22.3	23.3	23.5	23.4	**21.4**	**21.3**
15	diabetes	30.7	31.7	34.2	**26.6**	**26.7**	**27**	**27.7**	**26.1**	**26.2**	26.8	30	37.7	22.3	23.3	23.3	23.1	**21.2**	**21.5**
16	ringnorm	**12.4**	17	19.8	13.2	17.5	17.9	18	18	18	**2.1**	25.5	40.9	7.2	24.7	14.2	17.9	36.3	35.4
17	spirals	**5.2**	**5.1**	10.3	6.1	6	6.1	**5.5**	**5.3**	**5.2**	**1.1**	1.5	6.9	2.2	2.1	2.1	1.3	1.4	1.3
18	threenorm	**13.3**	14.9	33.8	15.3	16.4	15.1	**14**	14.6	14.7	**6**	6.6	30.6	7.7	7.8	7.4	6.7	7.2	7
19	twonorm	**2.2**	**1.9**	21.6	3.4	3.5	2.8	**2.2**	**1.8**	**1.7**	0.2	0.2	18.5	0.4	0.5	0.3	0.2	**0.2**	**0.2**
20	circles	**2.1**	2.5	5.1	**2.4**	2.8	3.3	**1.7**	**1.9**	**1.8**	0.1	1.8	3.1	0.3	0.4	0.4	**0**	**0**	**0**
21	digits 2-4	1.3	**0.4**	5	2.4	1.7	50	1.5	2.9	2.7	**0.1**	0.3	3.4	0.3	0.1	5.9	0.1	0.5	0.6
22	digits 3-8	**2.3**	3.6	6	**3.1**	**3.2**	50	4.3	4.9	5.2	**0.3**	2.3	4.4	0.6	0.5	30.7	0.8	1.2	1.3
23	digits even-odd	**7.4**	**7.2**	15.8	9.1	**7.8**	50	9.8	16.2	16.6	**2.4**	6.2	14.6	3.2	**2.3**	47.2	3.3	9	9.4
24	RS SRTM	2.5	3.2	3	**0.8**	**0.7**	1.7	1.7	**0.7**	**1**	0.3	2.7	2.7	0.1	**0**	0.1	0.1	**0**	**0**
25	RS ASTER	25.5	29.8	32.7	22.9	24.9	27.5	25.1	**19.3**	**19.9**	18.7	25.4	30.1	17.4	19.6	20.8	18.6	**13.2**	14
26	MRS quality	**6.2**	6.7	18.1	6.7	7.7	9.3	8.4	**6.2**	**6.2**	2.1	2.5	15.5	2.4	2.1	2.7	2.4	1.9	**1.9**
27	MRS tumor 1	**11.2**	12	17.5	**10.6**	**10.4**	12.2	**10.7**	**10.6**	**11**	4.9	8.4	16.8	6.2	**4.6**	5.9	4.9	**4.6**	**4.7**
28	MRS tumor 1	**19.1**	**19.7**	22.3	**18.9**	**19**	**19.1**	**19.1**	**19.7**	**20.1**	28	29.5	35.4	**26.8**	**24.7**	**26**	25.9	26.9	**25.5**
29	IR BSE 1	22.5	23.1	25.2	22.1	20.5	23	23	**13.1**	**13.4**	22	33.6	39.3	27.4	21.3	26.8	24.5	**9.2**	9.9
30	IR BSE 2	27.9	42.9	24.1	25.9	25.7	29.6	34.5	**15**	**15.5**	20.5	41.5	30	20.7	18.5	25.1	27.3	**6.9**	**7.1**
31	IR BSE 3	27.1	40.5	25.2	33.5	24.6	30.8	35.4	**14.9**	**12.6**	20.3	39	31.5	18.2	18.5	26.1	29	**5**	**5.2**
32	MRS yeast 1	**4.3**	9	14.5	7.8	7.3	8.9	7.9	**4.1**	**4.3**	**1.2**	6.4	15.5	3.4	2.9	3.6	2.9	**1.2**	**1.2**
33	MRS yeast 2	**2.4**	**3.2**	9	8.3	3.9	4.4	**3.5**	**3**	**2.8**	**2.2**	4.3	11.1	4.5	3.6	2.8	2.9	2.7	2.8
34	MRS yeast 3	**3**	4.9	8.7	7.2	5	5.2	4.4	**3.2**	**3.2**	3.9	4.8	13.6	9.5	4.8	5	4.8	**3.2**	**3.2**
35	MRS yeast 4	9.7	11	15.7	15.8	12	14.1	13.3	**6.5**	**5.9**	**4.2**	14.2	26.9	6.1	8.6	8	7.1	4.5	**4.2**
36	MRS yeast 5	5	7.1	8	7.4	6.4	6.4	6.4	**3.5**	**3.9**	**3.2**	8.6	16.2	9.9	4.3	5.6	5.5	**3.4**	**3.1**
37	IR wine origin 1	**27.2**	40.7	**26**	**22.1**	**21.7**	**26.4**	29.6	**21.4**	**21.6**	23.5	40.6	28	16.9	**14.5**	19.6	23.8	**13.6**	**13.2**
38	IR wine origin 2	**25.5**	40.6	30.3	**21.8**	**21.1**	**25.4**	32.1	**25.5**	**22.6**	23.2	41.7	31.9	27.2	**15**	19.2	27.5	15.6	**13.9**
39	IR wine grape 1	17.1	40.3	14.7	18	**11.1**	21.9	25.1	**8.4**	**4.6**	6	38.4	18.8	7.6	3	10.2	16.7	**0**	**0**
40	IR wine grape 2	18	38.2	**15.3**	**12.5**	**10.3**	22.1	29.5	**11.6**	**11.1**	6.2	35.2	18.3	5.5	2.7	10.1	21.2	**0**	**0**

Table 1. Classification accuracy (left block) and 1−ROC AUC (right block) of the classifiers on the 40 data sets. Data sets are grouped in factorial, nominal and spectral data. Underlined is the best results, bold results do not differ significantly from the best.

over oRF-lda becomes apparent when comparing class separation. The somewhat weaker general performance of the unregularized oRF suggest that the overall performance may benefit to a large extend from the optimal choice of λ and a sufficiently strong regularization.

Overall, we find that random forests with orthogonal trees (RF) perform well on factorial data, but they are outperformed by oblique random forests (oRF) on numerical and spectral data. Oblique random forests with regularized node models (oRF-ridge) rank overall first, followed by oRF with unregularized model (oRF-lda), both outperforming Breiman's oRF with random split model (oRF-rnd). So, in any learning task which do not comprise discrete factorial features,

Ranking		1		2		3		4		5	
Factorial Data	AUC ROC	adaboost	(6)	RF	(4)	oRF-lda	(2)	oRF-rnd	(2)	knn	(1)
	Accuracy	adaboost	(9)	RF	(6)	oRF-lda	(4)	oRF-ridge	(3)	CART	(3)
Nominal Data	AUC ROC	oRF-ridge	(6)	oRF-lda	(5)	svm	(3)	oRF-rnd	(3)	adaboost	(2)
	Accuracy	oRF-rnd	(7)	oRF-ridge	(6)	oRF-lda	(6)	adaboost	(6)	svm	(6)
Spectral Data	AUC ROC	oRF-ridge	(14)	oRF-lda	(12)	svm	(7)	RF	(6)	adaboost	(1)
	Accuracy	oRF-ridge	(17)	oRF-lda	(17)	svm	(10)	RF	(9)	adaboost	(7)

Table 2. Ranking of the classifiers summarizing performance for the three different data classes. The figure shows how often a classifier performed best or similar to the best in Table 1. With the exception of of factorial data oblique random forests outperform the alternative classifiers. Overall oRF-ridge performs best.

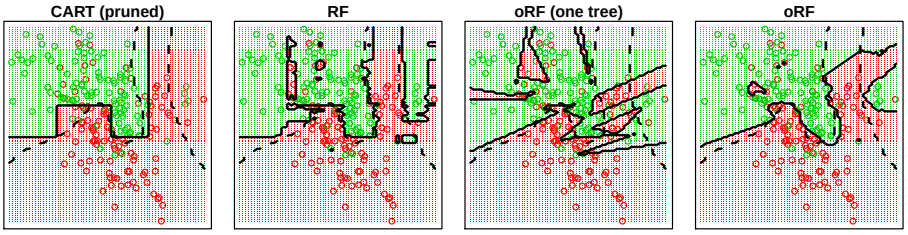

Fig. 4. Visualization of decision boundaries using the mixtures of Gaussians example from [16]. Colors indicate class assignments of the bivariate grid points when training the classifier using the indicated samples (circles). The solid black line represents the decision boundary learned. The dashed black line is Bayes optimal border between the distributions generating the training samples. Shown are single univariate classification tree (CART) and RF, and a single oblique tree and oRF. Differences can be observed in the topology of the decision boundary, and the way decision boundaries are extrapolated in regions with few observations. Note that CART has been optimized using cross-validation, while the others are applied out of the box.

the oRF may be advantageous over the regular RF and a better choice when an "out-of-the-box" learning algorithm is required.

4 Advantages of oblique model trees

In another set of experiments we want to shed light on specific advantages of the oRF classifier and properties of those types of data it performs well with.

Topology of the decision boundary To understand why RF and oRF behave differently, we visualize the actual class separation for the "mixture" data set [16] (Fig. 4). When testing a single pruned orthogonal tree (CART) we obtain a clear, binary separation of the feature space. However, even when pooling many trees the boundary imposed by the RF keep their blocked, or 'stair-like' topology. They extrapolate the decision boundary in space with few samples through axis-parallel, orthogonal splits. In addition to the blocked decision boundary, this somewhat arbitrary extrapolation may not be a natural choice in high dimensional tasks with few samples and correlated feature, e.g., on spectral data. The oRF adapts here closely to the training data (dots) and to the true boundary of the underlying distribution (dashed lines). In this the separation of the feature space imposed by the oRF is more similar to the SVM (not shown here) than to the RF, both favoring smooth decision boundaries. While for the RBF-kernel SVM the smoothness of this boundary is a parameter to be optimized during training, the random forest framework does not require to adjust any such parameter.

Advantage over univariate and random multivariate splits One may argue that RF suffers from significant bias imposed through the topological constraints of the decision boundaries of its base learners. We calculate bias and variance for results from Table 1 using ensemble votes as for regression tasks [16]. Fig. 5 (left) reports differences in bias and variance for RF and oRF, pooled over

the different data types. The advantage of RF on factorial data is due to a lower bias. The higher accuracy of oRF on spectral and numerical data correlates with a lower bias, too. Variance is slightly lower for oRF. A similar analysis can be done to compare the oRF with learned node model – proposed in this study – with the original oRF with random split (Fig. 5, right). Results show that the latter (oRF-rnd) suffers from high bias when compared to the first (oRF-ridge). This suggest that the higher variability of the oRF-rnd, and the resulting lower variability of the ensemble, does not trade off the disadvantage (or bias) of using random split directions.

Advantage on spectral data Oblique random forests perform well on spectral data which shows a strong correlation among the features (Fig. 1). To study this in a controlled fashion, we systematically increase correlation between features for selected data sets (Fig. 6) by adding a random scalar to the features of every single observation, drawn from a normal distribution. We then increase the size of this random offset stepwise by a multiplicative factor ($10^{\{0,1,\ldots,15\}}$). As visualized in Fig. 6, this stretches the samples successively along the intersecting line of the feature space. Both oRF and RF are tested on any of the 3*16 resulting data sets. Similar to our general evaluation, we calculate the AUC ROC.

We observe that the performance of the orthogonal random forest decrease at a certain point (Fig. 6, right), finally resulting in complete misclassification. On the *threenorm* data (Figure 6, left), the RF drops from more than 85% classification accuracy rapidly to a close to random prediction. At the same time the performance of the oRF remains practically unchanged. These results suggest, somewhat similar to our reasoning at the beginning, that in classification tasks with collinear samples the marginal distributions of the input variables x_i – i.e., the projections of the data into the subspaces evaluated by the orthogonal random forest in search for an optimal split – may lose their power to separate classes (compare the two marginal distributions in Fig. 1, for example).

5 Feature importance and sample proximity

The random forest framework provides additional tools which help to illustrate decision processes within the random forest ensemble, but which also have a value for exploratory data analysis on their own. This are importance scores

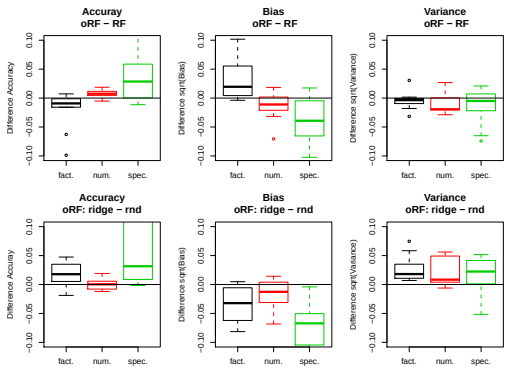

Fig. 5. Comparison of bias and variance between RF and oRF-ridge (left) and oRF-rnd and oRF-ridge (right) pooled over the data types. Boxplots indicate quartiles and outliers. Advantages in classification accuracy are dominated by bias. This benefits oRF-ridge for numerical and spectral data. The bias may arise from the topology of the base learner.

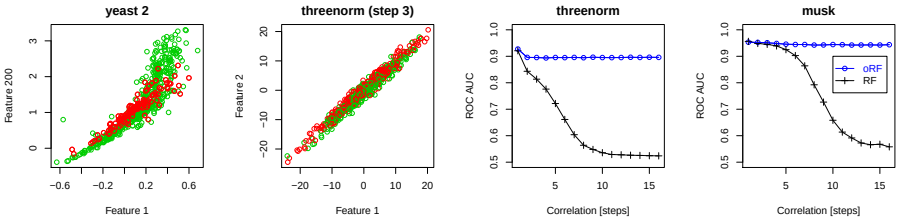

Fig. 6. Correlation between features (left) and performance on artificially correlated data (right). Shown is the 'natural' correlation between features in the *yeast 2* data set and samples of the *threenorm* data set with artificially induced correlation (left, corresponding to 'step 3' in the right figures). While the performance of the oRF remains nearly unchanged even on highly correlated data (right), RF fails beyond a certain point.

reporting the relevance of individual features [44] and sample proximity measures for visualizing similarities between individual samples and for mapping natural groupings within the data [38]. We propose similar tools for the oRF.

Feature importance Different feature importance measures exist for the random forest framework. One score is the "Gini importance" [5]. This measure is obtained during training by recording the decrease in Gini impurity for every variable, whenever a variable is selected in a split. Averaging this quantity, individually for each variable, and over all splits in a tree and all trees in the ensemble leads to the Gini importance score. An alternative measure is the "permutation importance". It is obtained by comparing, for each variable, the regular predictions of the classifier with those predictions obtained after randomly permuting the observations for this specific variable. Empirical studies suggest that both the feature rankings – the algorithmically motivated Gini importance, and the statistically defined permutation importance – correlate well in most classification tasks [1]. It has been observed that correlation between features may affect both Gini [23] and permutation importance [28]. For the oRF we obtain a feature relevance measure similar to the Gini importance by calculating analysis of variance (ANOVA) tables at every split in the oblique model tree, and by recording those variables which contribute significantly to the split (as expressed by a sufficiently small p-value, here $p \leq .01$). For every individual variable we record how often it was deemed significant in a split. This frequency, calculated from all splits over a sufficiently large number of trees, is the statistic we propose to use for visualizing feature importance in oblique random forests (Figure 7). We refer to it as "oRF importance" in the following.

We can compare "oRF importance" and "Gini importance". Feature selection is highly relevant in chemometric calibration and classification tasks. So, we show an example for the *BSE 1* data set in Figure 7, a data set where large differences in the performance of RF and oRF can be observed (Table 1). The Gini feature importance is obtained from an ensemble of 1500 trees, and the oRF feature importance using an ensemble with 9000 trees. (For computational convenience we use an oRF with logistic node model.) We chose such high numbers of trees to guarantee a sufficient sampling of all features with a larger ensemble for the

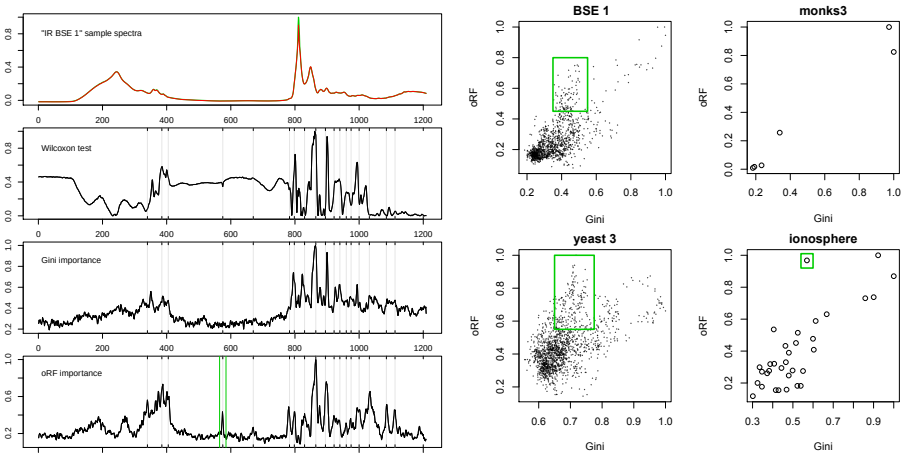

Fig. 7. Random forest feature importance – exemplified for a spectral data sets (BSE 1). The left plot shows different feature important measures, along with a representative spectrum of the data set. For all three tests high values indicate important features. The right plot compares RF and oRF feature importance scores for different data sets. The oRF also assigned importance to a number of additional features which are virtually ignored by the orthogonal RF (green boxes, for BSE 1 data also indicated by green lines).

oRF, as oblique model trees need fewer splits than orthogonal decision trees. We also compare both random forest scores with a *t*-test measuring class separation individually at every feature – similar to, for example, fMRI experiments, microarray, or tests on spectra [15]. We observe that both random forest measures show additional peaks when compared with the *t*-test that may be attributed to the presence of patterns which are of multivariate importance [23]. While both Gini and oRF importance show a broad correspondence for most spectral regions, a number of features appear to be relevant to the oRF only (Fig. 7, green boxes and spectra regions). We find such additional "peaks" for most spectral data sets, and some of the numerical data sets (*ionosphere*). It suggests that oblique splits allow the classifier to consider additional spectral regions during classification which are "invisible" to the traditional RF and, consequently, not used during classification.

Sample proximity The random forest framework allows one to measure a distance or "proximity" between observations, or samples, in a classification task [5]. During training of the classifier the out-of-bag samples are pushed down the tree. The frequency of how often each pair of samples ends up in the same terminal node – i.e., in the same partition of the feature space – is recorded. After a large number of trees has been learned, and every pair has been tested sufficiently often, these co-occurrence frequencies can be normalized and an affinity matrix is obtained [5, 38]. Similar to [5] the affinity matrix can be transformed to a distance matrix counting how often a pair of sample has *not* been in the same node, and the scores from projecting samples to the principal components of this distance matrix can be visualized. It may help to uncover natural groupings

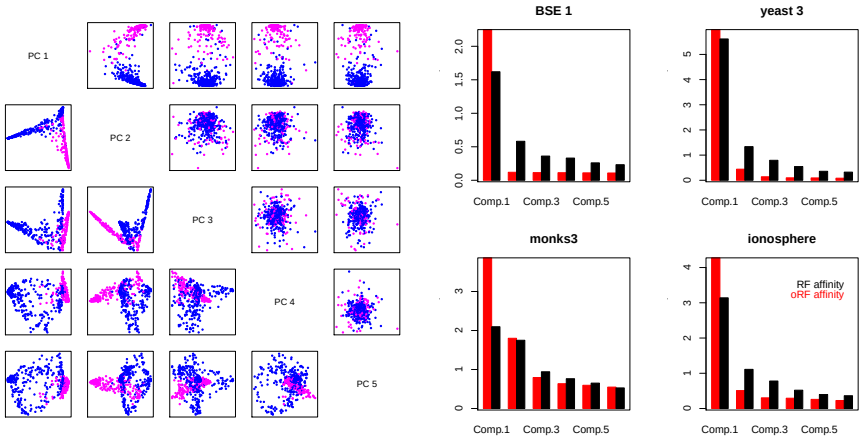

Fig. 8. Visualization of the random forest sample proximity. Shown are sample projected to the principal eigen-spaces of the proximity matrix of RF (left, lower triangle) and oRF (left, upper triangle) and the variation explained by RF (right, black) and oRF (right, red) for further data sets. The oRF is void of structures resulting from the topology of the base learner (*yeast 1*). For oRF, inter-sample differences of the dominating first eigen-direction correlate well with learned inter-class difference. Higher components show random noise and their contribution to inter-sample differences can be neglected here. The RF affinity matrix reveals complex structures even for higher components suggesting that rules by the oRF are much simpler than those induced by the RF. The oRF proximity may be highly preferable for visualizing natural groupings.

in the data and to identify consistently misclassified sample in a very intuitive fashion [38].

Figure 8 shows a projection of the samples to the principal eigen-spaces of their distance matrix for both a RF (lower triangle of the pairs plot matrix) and an oRF (upper triangle). Stark differences are visible: Inter-sample differences in the oRF proximity correlate well with the first eigen-direction of the distance matrix, representing the learned inter-class difference (Fig. 8, right). Higher components show random noise only and no further grouping can be observed. This is very different from the RF proximity, which reveals complex structures even for higher components. This may indicate that orthogonal trees separate feature space by splits which are not required by the learning task, but are imposed by the topology of the base learner.

The observation from the *yeast 2* data set – that oRF show complex inter-sample differences unrelated to the imposed inter-class separation task – holds true, to a lesser extent, for most data sets in our study. It is even visible from the variance explained along the eigen-directions of the affinity matrices. We find for all data sets the variance explained by the first eigen-space of the oRF to be much larger than the variance explained by the first eigen-space of the RF. For higher components the opposite is true indicating that some of the inter-sample differences in the RF are not due to plain inter-class differences imposed by the learning task, but due to a consistently over-complex segmentation of the features space by the splits of the orthogonal trees. Again, this implies that rules

by the oRF are much simpler than those induced by the RF and, supposedly, void of complex structures imposed superficially by the topology of the base learner.

6 Conclusion

Random forest with oblique node models have been somewhat neglected since proposed by Breiman in 2001. When replacing the random node model with learned model splits we obtain a classifier that can be implemented straightforwardly, and that performs well on a large range of data sets even with default parameterizations.

- We find random forests with orthogonal splits (RF) to perform well on factorial data. On all other data random forests with oblique node model (oRF) perform better. Here, a learned node model performs better than a random split, and a learned node model with adaptive regularization better than one without. On numerical and spectral data, oRFs outperform a range of alternative classifiers even with default parametrization. The oRF does exceptionally well in separating high dimensional distributions even when large correlations between features are present.
- The oRF can be used to define feature importance and sample proximity measures. They may be preferable in the analysis of learning tasks where oRF performs exceptionally well and shows less topological bias than RF. These are learning tasks with few samples, many irrelevant features and correlated predictors arising, for example, in biomedical diagnostics [19, 39, 29, 14, 25, 23]
- The inspection of the feature importance measure suggests that the oRF is able to consider information from variables which are "invisible" to univariate split models (Fig. 1) and void of structures reflecting constraints resulting from the geometry of the base learner. The ability to use these additional variables and considering their contribution to the classification problem illustrates why the oRF perform better than the "traditional" RF on numerical and spectral data.

These results may lead to further work. Oblique decision trees show similarities with deep learning architectures [36], and "reusing" scores from earlier splits for decisions in later nodes would even enhance this similarity. Reusing classification results and probabilities from earlier trees would be very similar to the "auto-context" idea from [42]. One may assume that mixing orthogonal and oblique splits within a tree would help to combine desired properties of both approaches.

References

1. Archer, K.J., Kimes, R.V.: Empirical characterization of random forest variable importance measures. Comput Stat Data Anal 52, 2249–2260 (2008)
2. Biau, G., Devroye, L., Lugosi, G.: Consistency of random forests and other averaging classifiers. J Mach Learn Res pp. 2015–2033 (2008)

3. Breiman, L.: Bagging predictors. Mach Learn 24, 123–140 (1996)
4. Breiman, L.: Arcing classifiers. Technical Report, UC Berkeley (1998)
5. Breiman, L.: Random forests. Mach Learn J 45, 5–32 (2001)
6. Breiman, L.: Consistency for a simple model of random forests. Tech. Rep. 670, UC Berkeley (2004)
7. Caputo, B., Sim, K., Furesjo, F., Smola, A.: Appearance-based object recognition using SVMs: which kernel should I use? In: Proc NIPS WS (2002)
8. Chan, K.Y., Loh, W.Y.: LOTUS: An algorithm for building accurate and comprehensible logistic regression trees. J Comp Graph Stat 13, 826–852 (2004)
9. Criminisi, A., Shotton, J., Bucciarelli, S.: Decision forests with long-range spatial context for organ localization in ct volumes. In: Proc MICCAI-PMMIA (2009)
10. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Mach Learn 40, 139–157 (2000)
11. Frank, I.E., Friedman, J.H.: A statistical view of some chemometrics regression tools. Technometrics 35, 109–135 (1993)
12. Freund, Y., Shapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Proc ECCL, LNCS 904 (1995)
13. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach Learn 63, 3–42 (2006)
14. Geurts, P., Fillet, M., de Seny, D., Meuwis, M.A., Malaise, M., Merville, M.P., Wehenkel., L.: Proteomic mass spectra classification using decision tree based ensemble methods. Bioinformatics 21, 313–845 (2005)
15. Hastie, T., Tibshirani, R., Eisen, M., Alizadeh, A., Levy, R., Staudt, L., Chan, W., Botstein, D., Brown, P.: Gene shaving as a method for identifying distinct sets of genes with similar expression patterns. Genome Biol 1, 1–8 (2000)
16. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, 2nd edn. (2009)
17. Ho, T.K.: The random subspace method for constructing decision forests. IEEE-T Patt Anal Mach Intell 20, 832–844 (1998)
18. Hothorn, T., Leisch, F., Zeileis, A., Hornik, K.: The design and analysis of benchmark experiments. Tech. rep., TU Vienna (2003)
19. Jiang, H., Deng, Y., Chen, H.S., Tao, L., Sha, Q., Chen, J., Tsai, C.J., Zhang, S.: Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. BMC Bioinformatics 5(81) (2004)
20. Liaw, A., Wiener, M.: Classification and regression by randomForest. R News 2, 18–22 (2002)
21. Lin, Y., Jeon, Y.: Random forests and adaptive nearest neighbors. J Am Stat Assoc 101, 578–590 (2006)
22. Martinez-Munoz, G., Hernandez-Lobato, D., Suarez, A.: An analysis of ensemble pruning techniques based on ordered aggregation. IEEE-T Pattern Anal Mach Intell 31, 245–259 (2009)
23. Menze, B.H., Kelm, B.M., Masuch, R., Himmelreich, U., Petrich, W., Hamprecht, F.A.: A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. BMC Bioinformatics 10, 213 (2009)
24. Menze, B.H., Lichy, M.P., Bachert, P., Kelm, B.M., Schlemmer, H.P., Hamprecht, F.A.: Optimal classification of long echo time in vivo magnetic resonance spectra in the detection of recurrent brain tumors. NMR Biomed 19, 599–610 (2006)

25. Menze, B.H., Petrich, W., Hamprecht, F.A.: Multivariate feature selection and hierarchical classification for infrared spectroscopy: serum-based detection of bovine spongiform encephalopathy. Anal Bioanal Chem 387, 801–1807 (2007)
26. Menze, B.H., Ur, J.A., Sherratt, A.G.: Detection of ancient settlement mounds – Archaeological survey based on the SRTM terrain model. Photogramm Engin Rem Sens 72, 321–327 (2006)
27. Murthy, S.K., Kasif, S., Salzberg, S.: A system for induction of oblique decision trees. J Artif Intell Res 2, 1–32 (1994)
28. Nicodemus, K., Malley, J., Strobl, C., Ziegler, A.: The behaviour of random forest permutation-based variable importance measures under predictor correlation. BMC Bioinformatics 11, 110 (2010)
29. Pal, M.: Random forest classifier for remote sensing classification. Intern J Remote Sensing 1, 217–222 (2005)
30. Pisetta, V., Jouve, P., Zighed, D.: Learning with ensembles of randomized trees : New insights. In: Proc ECML (2010)
31. Platt, J.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Smola, A., Bartlett, P., Schoelkopf, B., Schuurmans, D. (eds.) Advances in Large Margin Classifiers. MIT Press, Cambridge, MA (2000)
32. Robnik-Sikonja, M.: Improving random forests. In: Proc ECML (2004)
33. Rodriguez, J., Kuncheva, L., Alonso, C.: Rotation forest: A new classifier ensemble method. IEEE T Patt Anal Mach Intell 28, 1619–1630 (2006)
34. Saeys, Y., Abeel, T., Van de Peer, Y.: Robust feature selection using ensemble feature selection techniques. In: Proc ECML (2008)
35. Segal, M.R.: Machine learning benchmarks and random forest regression. Tech. rep., UC San Francisco (2004)
36. Sethi, I.K.: Entropy nets: from decision trees to neural networks. Proc IEEE 78, 1605–1613 (1990)
37. Shen, K.Q., Ong, C.J., Li, X.P., Zheng, H., Wilder-Smith, E.P.V.: A feature selection method for multi-level mental fatigue EEG classification. IEEE-T Biomed Engin 54, 1231–7 (2007), in press, epub ahead.
38. Su, X., Tsai, C.L., Wang, H., Nickerson, D.M., Li, B.: Subgroup analysis via recursive partitioning. J Mach Learn Res 10, 141–158 (2009)
39. Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P., Feuston, B.P.: Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. J Chem Inf Model 43, 1947–58 (2003)
40. Tan, P.J., Dowe, D.L., Webb, G.I., Y., X.: MML inference of oblique decision trees. In: Proc AJCAI. pp. 1082–1088 (2004)
41. Tan, P.J., Dowe, D.L.: Decision forests with oblique decision trees. In: Proc MICAI. pp. 593–603 (2006)
42. Tu, Z., Bai, X.: Auto-context and its application to high-level vision tasks and 3d brain image segmentation. IEEE-T Patt Anal Mach Intell 99(preprint) (2009)
43. Tu, Z.: Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In: Proc ICCV. pp. 1589–1596 (2005)
44. Tuv, E., Borisov, A., Runger, G., Torkkola, K.: Feature selection with ensembles, artificial variables, and redundancy elimination. J Mach Learn Res 10, 1341–1366 (2009)
45. Yao, B., Khosla, A., Fei-Fei, L.: Combining randomization and discrimination for fine-grained image categorization. In: Proc CVPR (2011)