An Ensemble of Optimal Trees for Classification and Regression (OTE)

Zardad Khan[a,b,*], Asma Gul[b,c], Aris Perperoglou[b], Miftahuddin Miftahuddin[b,f], Osama Mahmoud[b,d], Werner Adler[e], Berthold Lausen[b,**],

[a] *Department of Statistics, Abdul Wali Khan University, Mardan, Pakistan*
[b] *Department of Mathematical Sciences, University of Essex, Colchester CO4 3SQ, UK*
[c] *Department of Statististics, Shaheed Benazir Bhutto Women University Peshawar, Pakistan*
[d] *School of Oral & Dental Sciences, University of Bristol, UK*
[e] *Department of Biometry and Epidemiology, University of Erlangen-Nuremberg, Germany*
[f] *College of Science, Syiah Kuala University - Banda Aceh, Indonesia*

Abstract

Predictive performance of a random forest ensemble is highly associated with the strength of individual trees and their diversity. Ensemble of a small number of accurate and diverse trees, if prediction accuracy is not compromised, will also reduce computational burden. We investigate the idea of integrating trees that are accurate and diverse. For this purpose, we utilize out-of-bag observation as validation sample from the training bootstrap samples to choose the best trees based on their individual performance and then assess these trees for diversity using Brier score. Starting from the first best tree, a tree is selected for the final ensemble if its addition to the forest reduces error of the trees that have already been added. A total of 35 bench mark problems on classification and regression are used to assess the performance of the proposed method and compare it with kNN, tree, random forest, node harvest and support vector machine. We compute unexplained variances and classification error rates for all the methods on the corresponding data sets. Our experiments reveal that the size of the ensemble is reduced significantly and better results are obtained in most of the cases. For further verification, a simulation study is also given where four tree

*zardadkhan@awkum.edu.pk
**blausen@essex.ac.uk

style scenarios are considered to generate data sets with several structures.

Keywords: classification and regression trees, random forest, ensemble methods, accuracy and diversity.

1. Introduction

Many studies have suggested that combining weak models leads to efficient ensembles [1, 2, 3, 4, 5, 6, 7] that are used frequently in many real world problems[8, 9, 10, 11]. Combining the outputs of multiple classifiers also reduces generalization error [2, 3, 12, 4]. Ensemble methods are effective in that different types of models have different inductive biases where such diversity reduces variance-error while not increasing the bias error [13, 14, 15].

Extending this notion, Breiman [16] suggested growing a large number, T for instance, of classification and regression trees. Trees are grown on bootstrap samples form a given training data $\mathcal{L} = (\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)\}$. The $\mathbf{x_i}$ are observations on d features and y values are from real line and a set of known classes $(1, 2, 3, ..., K)$ in cases of regression and classification, respectively. Breiman called this method as random forest.

As the number of trees in random forest is often very large, there has been a significant work done on the problem of minimizing this number to reduce computational cost without decreasing prediction accuracy[17, 18, 19, 20].

Overall prediction error of a random forest is highly associated with the strength of individual trees and their diversity in the forest. This idea is backed by Breiman's[16] upper bound for the overall prediction error of random forest given by

$$\widehat{Err} \leq \bar{\rho}\,\widehat{err}_j, \tag{1}$$

where $j = 1, 2, 3, ..., T$, T denotes the number of all trees, $\widehat{Err}$ is the overall prediction error of the forest, $\bar{\rho}$ represents weighted correlation between residuals from two independent trees and $\widehat{err}_j$ is the prediction error of the jth tree in the forest.

Based on the above discussion, this article proposes to select the best trees,in terms individual accuracy and diversity, from a large ensemble grown by random forest. Using 35 benchmark data sets, the results from the new method are compared with those of kNN, tree classifier, random forest, node harvest and support vector machine. For further verification, a simulation study is also given where data sets with many tree structures are generated. The rest of the paper is organized as follows. The proposed method and the underlying algorithm are given in section 2, experiments and results based on benchmark and simulated data sets are given in section 3. Finally, section 4 gives the conclusion of the paper.

2. OTE: Optimal Trees Ensemble

Random forest refines bagging by introducing additional randomness in the base models, trees, by drawing subsets of the predictor set for partitioning the nodes of a tree[4] . This article investigates the possibility of further refinement by proposing the method of trees selection on the basis of their individual accuracy and diversity using unexplained variance and Brier score [21] in cases of regression and classification respectively. To this end, we partition the given training data $\mathcal{L} = (\mathbf{X}, \mathbf{Y})$ randomly into two non overlapping portions, $\mathcal{L}_{\mathcal{B}} = (\mathbf{X_B}, \mathbf{Y_B})$ and $\mathcal{L}_{\mathcal{V}} = (\mathbf{X_V}, \mathbf{Y_V})$. Grow T classification or regression trees on T bootstrap samples from the first portion $\mathcal{L}_{\mathcal{B}} = (\mathbf{X_B}, \mathbf{Y_B})$. While doing so, select a random sample of $p < d$ features from the entire set of d predictors. This inculcates additional randomness in the trees. Due to bootstraping, there will be some observations left out of the samples which are called out-of-bag (OOB) observations. These observations take no part in the training of tree. These observatons can be utilized in two ways:

1. In case of regression, out-of-bag observations are used to estimate unexplained variances of each tree grown on a bootstrap sample. Trees are then ranked in ascending order whith respect to their unexplained variances and the top ranked M trees are chosen.

2. In case of classification, out-of-bag observations are used to estimate error rates of the trees. Trees are then ranked in ascending order whith respect to their error rates and the top ranked M trees are chosen.

A diversity check is carried out as follows

1. Starting from the two top ranked trees, successive ranked trees are added one by one to see how they perform on the independent validation data, $\mathcal{L}_{\mathcal{V}} = (\mathbf{X}_{\mathbf{V}}, \mathbf{Y}_{\mathbf{V}})$. This is done until the last Mth tree is added.
2. Select tree $\hat{L}_k, k = 1, 2, 3, ..., M$ if its inclusion to the ensemble without the kth tree satisfys the following two criteria given for regression and classification respectively.
    (a) In regression case, let $\mathcal{U}.\mathcal{EXP}^{\langle -k \rangle}$ be the unexplained variance of the ensemble not having the kth tree and $\mathcal{U}.\mathcal{EXP}^{\langle +k \rangle}$ be the unexplained variance of the ensemble with kth tree included, then tree $\hat{L}_k$ is chosen if

    $$\mathcal{U}.\mathcal{EXP}^{\langle +k \rangle} < \mathcal{U}.\mathcal{EXP}^{\langle -k \rangle}.$$

    (b) In classification case, let $\hat{\mathcal{BS}}^{\langle -k \rangle}$ be the Brier score of the ensemble not having the kth tree and $\hat{\mathcal{BS}}^{\langle +k \rangle}$ be the Brier score of the ensemble with kth tree included, then tree $\hat{L}_k$ is chosen if

    $$\hat{\mathcal{BS}}^{\langle +k \rangle} < \hat{\mathcal{BS}}^{\langle -k \rangle},$$

    where

    $$\hat{\mathcal{BS}} = \frac{\sum_{i=1}^{\text{\# of test cases}} \left(y_i - \hat{P}(y_i|\mathbf{X})\right)^2}{\text{total \# of test instances}},$$

    y_i is the state of y_i for observation i in the $(0, 1)$ form and $\hat{P}(y|\mathbf{X})$ is the binary response probability estimate given the features.

These trees, named as optimal trees, are then combined and are allowed to vote, in case of classification, or average, in case of regression, for new/test data. The resultant ensemble is named as optimal trees ensemble, *OTE*.

2.1. The Algorithm

Steps of the proposed algorithm both for regression and classification are

1. Take T bootstrap samples from the given portion of the training data $\mathcal{L}_{\mathcal{B}} = (\mathbf{X_B}, \mathbf{Y_B})$.
2. Grow regression/classification trees on all the bootstrap samples using random forest technique.
3. Choose M trees with the smallest individual prediction error on the training data.
4. Add the M selected trees one by one and select a tree if it improves performance on validation data, $\mathcal{L}_{\mathcal{V}} = (\mathbf{X_V}, \mathbf{Y_V})$, using unexplained variance and Brier score in cases of regression and classification as the respective performance measures.
5. Combine and allow the trees to vote, in case of classification, or average, in case of regression, for new/test data.

An illustrative flow chart of the proposed algorithm can be seen in Figure 1.

An algorithm based on a similar idea has previously been proposed where instead of classification and regression trees, probability estimation trees are used [22]. The ensemble of probability estimation trees is used for estimating class membership probabilities in binary class problems. Ensembles selection for kNN classifiers have also been proposed recently where in addition to individual accuracy, the kNN models are grown on random subsets of the feature set instead of considering the entire space [23, 24].

3. Experiments and Results

3.1. Simulation

This section presents four simulation scenarios each consisting of various tree structures. The aim is to make the recognition problem slightly difficult for classifiers like kNN and CART, and to provide a challenging task for the most complex method like SVMs and random forest. In each of the scenarios, four

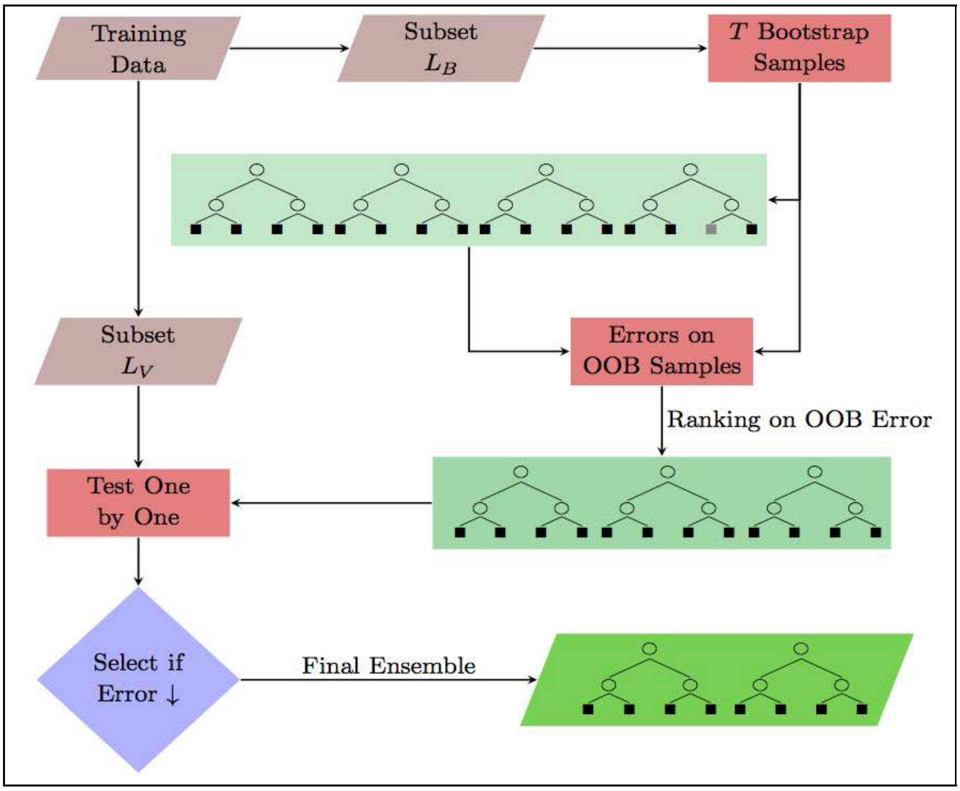

Figure 1: Flow chart of OTE for regression and classification

different complexity levels are considered by changing the weights η_{ijk} of the tree nodes. Consequently, four different values of the Bayes error are obtained where the lowest Bayes error indicates a data set with meaningful patterns and the highest Bayes error means a data set with no patterns. Table 1 gives various values of η_{ijk} used in Scenarios 1, 2, 3, and 4. Node weights for obtaining the complexity levels are listed in four columns of the table for $k = 1, 2, 3, 4$, for each model. A generic equation for producing class probabilities of the bernoulli response $\mathbf{Y} = \text{Bernoulli}(p)$ given the $n \times 3T$ dimensional vector $\mathbf{X}$ of n iid observations from Uniform$(0, 1)$ is.

$$p(y|\mathbf{X}) = \frac{exp\left(c_2 \times \left(\frac{\mathcal{Z}_m}{T} - c_1\right)\right)}{1 + exp\left(c_2 \times \left(\frac{\mathcal{Z}_m}{T} - c_1\right)\right)}, \text{ where } \mathcal{Z}_m = \sum_{t=1}^{T} \hat{p}_t. \qquad (2)$$

c_1 and c_2 are some arbitrary constants, $m = 1, 2, 3, 4$ is scenario number and $\mathcal{Z}_m$'s are $n \times 1$ probability vectors. T is the total number of trees used in a scenario and $\hat{p}_t$'s are class probabilities for a particular response in $\mathbf{Y}$. These probabilities are generated by the following tree structures

$$\begin{aligned}
\hat{p}_1 &= \eta_{11k} \times \mathbf{1}_{(\mathbf{x_1 \leq 0.5 \& x_3 \leq 0.5})} + \eta_{\mathbf{12k}} \times \mathbf{1}_{(\mathbf{x_1 \leq 0.5 \& x_3 > 0.5})} + \eta_{\mathbf{13k}} \times \mathbf{1}_{(\mathbf{x_1 > 0.5 \& x_2 \leq 0.5})} \\
&\quad + \eta_{14k} \times \mathbf{1}_{(x_1 > 0.5 \& x_2 > 0.5)}, \\
\hat{p}_2 &= \eta_{21k} \times \mathbf{1}_{(x_4 \leq 0.5 \& x_6 \leq 0.5)} + \eta_{22k} \times \mathbf{1}_{(x_4 \leq 0.5 \& x_6 > 0.5)} + \eta_{23k} \times \mathbf{1}_{(x_4 > 0.5 \& x_5 \leq 0.5)} \\
&\quad + \eta_{24k} \times \mathbf{1}_{(x_4 > 0.5 \& x_5 > 0.5)}, \\
\hat{p}_3 &= \eta_{31k} \times \mathbf{1}_{(x_7 \leq 0.5 \& x_8 \leq 0.5)} + \eta_{32k} \times \mathbf{1}_{(x_7 \leq 0.5 \& x_8 > 0.5)} + \eta_{33k} \times \mathbf{1}_{(x_7 > 0.5 \& x_9 \leq 0.5)} \\
&\quad + \eta_{34k} \times \mathbf{1}_{(x_7 > 0.5 \& x_9 > 0.5)}, \\
\hat{p}_4 &= \eta_{41k} \times \mathbf{1}_{(x_{10} \leq 0.5 \& x_{11} \leq 0.5)} + \eta_{42k} \times \mathbf{1}_{(x_{10} \leq 0.5 \& x_{11} > 0.5)} + \eta_{43k} \times \mathbf{1}_{(x_{10} > 0.5 \& x_{12} \leq 0.5)} \\
&\quad + \eta_{44k} \times \mathbf{1}_{(x_{10} > 0.5 \& x_{12} > 0.5)}, \\
\hat{p}_5 &= \eta_{51k} \times \mathbf{1}_{(x_{13} \leq 0.5 \& x_{14} \leq 0.5)} + \eta_{52k} \times \mathbf{1}_{(x_{13} \leq 0.5 \& x_{14} > 0.5)} + \eta_{53k} \times \mathbf{1}_{(x_{13} > 0.5 \& x_{15} \leq 0.5)} \\
&\quad + \eta_{54k} \times \mathbf{1}_{(x13 > 0.5 \& x_{15} > 0.5)}, \\
\hat{p}_6 &= \eta_{61k} \times \mathbf{1}_{(x_{16} \leq 0.5 \& x_{17} \leq 0.5)} + \eta_{62k} \times \mathbf{1}_{(x_{16} \leq 0.5 \& x_{17} > 0.5)} + \eta_{63k} \times \mathbf{1}_{(x_{16} > 0.5 \& x_{18} \leq 0.5)} \\
&\quad + \eta_{64k} \times \mathbf{1}_{(x16 > 0.5 \& x_{18} > 0.5)},
\end{aligned}$$

where $0 < \eta_{ijk} < 1$ are weights given to to the nodes of the trees, $k = 1, 2, 3, 4$. The four scenarios use the following specifications for using (2)

3.1.1. Scenario 1

This scenario consists of 3 tree components each grown on 3 variables which follows that, $T = 3$, $\mathcal{Z}_1 = \sum_{t=1}^{3} \hat{p}_t$ and $\mathbf{X}$ becomes a $n \times 9$ dimensional vector.

3.1.2. Scenario 2

In this scenario we take a total of $T = 4$ trees where $\mathcal{Z}_2 = \sum_{t=1}^{4} \hat{p}_t$ such that $\mathbf{X}$ becomes a $n \times 12$ dimensional vector.

3.1.3. Scenario 3

This scenario is based on $T = 5$ trees such that $\mathcal{Z}_3 = \sum_{t=1}^{5} \hat{p}_t$ and $\mathbf{X}$ becomes a $n \times 15$ dimensional vector.

3.1.4. Scenario 4

This scenario consists of 6 tree components which follows that, $T = 6$, $\mathcal{Z}_4 = \sum_{t=1}^{6} \hat{p}_t$ and $\mathbf{X}$ becomes a $n \times 18$ dimensional vector.

To understand how the trees are grown in the above simulation scenarios, a tree used in simulation Scenario 1.1 is given in Figure 2.

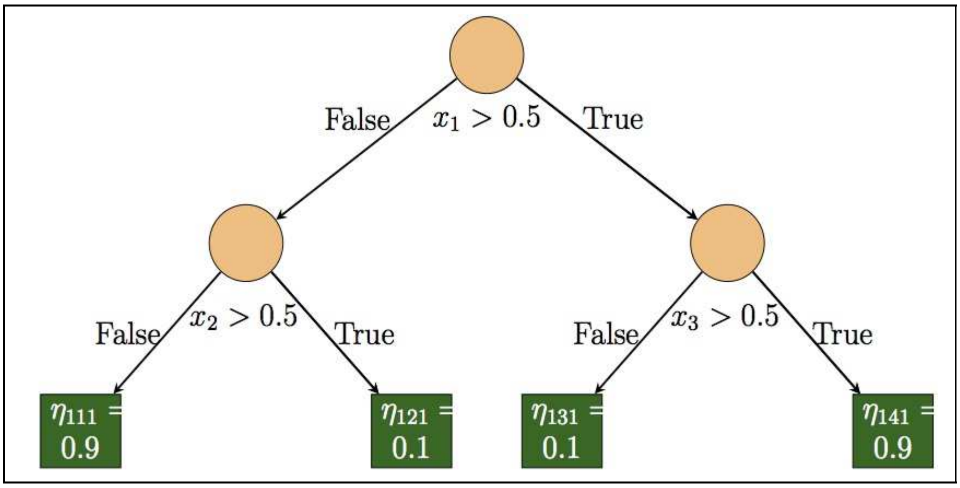

Figure 2: One of the trees used in simulation Scenario 1.1

Table 1: Node weights, η_{ijk}, used in simulation scenarios where i is tree number, j is node number in each tree and k is denoting a variant of the weights for the four complexity levels for all the scenarios.

Scenario 1						Scenario 2						Scenario 3						Scenario 4					
		k						k						k						k			
i	j	1	2	3	4	i	j	1	2	3	4	i	j	1	2	3	4	i	j	1	2	3	4
1	1	0.9	0.8	0.7	0.6	1	1	0.9	0.8	0.7	0.6	1	1	0.9	0.9	0.9	0.8	1	1	0.9	0.9	0.9	0.8
	2	0.1	0.2	0.3	0.4		2	0.1	0.2	0.3	0.4		2	0.1	0.1	0.1	0.2		2	0.1	0.1	0.1	0.2
	3	0.1	0.2	0.3	0.4		3	0.1	0.2	0.3	0.4		3	0.1	0.1	0.1	0.2		3	0.1	0.1	0.1	0.2
	4	0.9	0.8	0.7	0.6		4	0.9	0.8	0.7	0.6		4	0.9	0.9	0.9	0.8		4	0.9	0.9	0.9	0.8
2	1	0.9	0.8	0.7	0.6	2	1	0.9	0.8	0.7	0.6	2	1	0.9	0.9	0.9	0.8	2	1	0.9	0.9	0.9	0.8
	2	0.1	0.2	0.3	0.4		2	0.1	0.2	0.3	0.4		2	0.1	0.1	0.1	0.2		2	0.1	0.1	0.1	0.2
	3	0.1	0.2	0.3	0.4		3	0.1	0.2	0.3	0.4		3	0.1	0.1	0.1	0.2		3	0.1	0.1	0.1	0.2
	4	0.9	0.8	0.7	0.6		4	0.9	0.8	0.7	0.6		4	0.9	0.9	0.9	0.8		4	0.9	0.9	0.9	0.8
3	1	0.9	0.8	0.7	0.6	3	1	0.9	0.8	0.7	0.6	3	1	0.9	0.8	0.7	0.7	3	1	0.9	0.9	0.9	0.8
	2	0.1	0.2	0.3	0.4		2	0.1	0.2	0.3	0.4		2	0.1	0.2	0.3	0.3		2	0.1	0.1	0.1	0.2
	3	0.1	0.2	0.3	0.4		3	0.1	0.2	0.3	0.4		3	0.1	0.2	0.3	0.3		3	0.1	0.1	0.1	0.2
	4	0.9	0.8	0.7	0.6		4	0.9	0.8	0.7	0.6		4	0.9	0.8	0.7	0.7		4	0.9	0.9	0.9	0.8
						4	1	0.9	0.8	0.7	0.6	4	1	0.9	0.8	0.7	0.7	4	1	0.9	0.8	0.7	0.7
							2	0.1	0.2	0.3	0.4		2	0.1	0.2	0.3	0.3		2	0.1	0.2	0.3	0.3
							3	0.1	0.2	0.3	0.4		3	0.1	0.2	0.3	0.3		3	0.1	0.2	0.3	0.3
							4	0.9	0.8	0.7	0.6		4	0.9	0.8	0.7	0.7		4	0.9	0.8	0.7	0.7
												5	1	0.9	0.8	0.7	0.7	5	1	0.9	0.8	0.7	0.6
													2	0.1	0.2	0.3	0.3		2	0.1	0.2	0.3	0.4
													3	0.1	0.2	0.3	0.3		3	0.1	0.2	0.3	0.4
													4	0.9	0.8	0.7	0.7		4	0.9	0.8	0.7	0.6
																		6	1	0.9	0.8	0.7	0.6
																			2	0.1	0.2	0.3	0.4
																			3	0.1	0.2	0.3	0.4
																			4	0.9	0.8	0.7	0.6

Table 2: Classification error (in %age) of kNN, tree, random forest, node harvest, SVM and OTE. The forth column of the table shows Bayes error for each model. The last column is the percentage reduction in the size of OTE compared to random forest

Model	d	n	Bayes	kNN	Tree	RF	NH	SVM	SVM	SVM	SVM	OTE	Reduction in
			Error					(Radial)	(Linear)	(Bessel)	(Laplacian)		Ensemble Size (%)
Scenario 1	9	1000	9.0	22	9.9	9.6	9.8	19	19	19	19	9.5	91
			14	26	15	15	15	22	22	23	22	15	90
			17	32	18	18	21	28	28	28	28	18	90
			33	42	36	35	36	37	37	38	37	37	91
Scenario 2	12	1000	21	29	22	21	21	24	23	30	24	21	90
			24	31	25	24	24	26	26	32	26	23	90
			28	36	30	28	29	31	30	36	31	29	90
			30	39	32	32	32	33	33	38	33	32	89
Scenario 3	15	1000	15	31	22	18	22	24	24	55	24	18	91
			18	32	24	21	24	26	25	55	26	22	89
			21	34	25	23	27	27	27	55	27	24	91
			24	36	29	28	29	29	29	54	30	28	90
Scenario 4	18	1000	21	34	28	23	25	25	25	72	27	22	90
			22	35	27	23	26	27	27	71	28	24	89
			25	39	31	26	29	31	31	67	35	27	90
			26	40	31	28	30	32	32	68	36	29	90

The values of c_1 and c_2 are fixed at 0.5 and 15, respectively, in all the scenarios for all variants. A total of $n = 1000$ observation are generated using the above setup. kNN, CART, random forest, node harvest, SVM and *OTE* are trained by using 90% of the data as training data (of which 90% is for bootstring and 10% for diversity check, in the case of *OTE*) and then applying the remaining 10% data as test data for testing purpose. A total of 1000 realizations are made under each scenario. The results obtained in all the scenarios are given in Table 2. Node weights are changed in a manner that could make the patterns in the data less meaningful and thus getting a higher Bayes error. This can be observed in the fourth column of Table 2, where each scenario has four different values of the Bayes error. As anticipated, kNN and tree classifiers have the highest percentage errors in all the four scenarios. Random forest and *OTE* performed quite similarly with slight variations in few cases. In cases where the models have the highest Bayes error, the results of random forest are better or comparable with those of *OTE*. In all the remaing cases where the Bayes error is the smallest, *OTE* is better or comparable with random forest. SVM performed very similarly to kNN and tree. Percentage reduction in ensemble size of *OTE* is also shown in the last column of the table. This follows that *OTE* could be very helpful in decreasing the size of the ensemble thus reducing storage costs.

The box plots given in Figure 3 reveal that the best results of *OTE* can be observed in Figure (a) where a data set with meaningful tree structures is generated. Figure (d) is the worst example of *OTE* where the Bayes error is the highest (i.e. 33%), and where the data have no meaningful tree structures.

3.2. Benchmark Problems

For assessing the performance of *OTE* on benchmark problems, we have considered 35 data sets out of which 14 are regression and 21 classification problems. A brief summary of the data sets is given in Table 3. The upper portion of table 3 is a summary of regression problems whereas the lower portion is a summary of classification problems.

Table 3: Data sets for classification and regression with total number of observations n, number of features d and feature type; F: real, I: integer and N: nominal features in a data set. Sources are also given.

Data Set	n	d	**Feature type (R/I/N)**	**Source**
			Regression	
Bone	485	3	(1/1/1)	[25, 26]
Galaxy	323	4	(4/0/0)	[25, 27]
Friedman	1200	5	(5/0/0)	[28]
CPU	209	7	(7/0/0)	[29]
Concrete	103	7	(7/0/0)	[29]
Abalone	4177	8	(7/0/1)	[29]
MPG	398	8	(2/2/4)	[29]
Stock	950	9	(9/0/0)	http://funapp.cs.bilkent.edu.tr/DataSets/
Wine	1599	11	(11/0/0)	[29]
Ozone	203	12	(9/0/3)	[30]
Housing	506	13	(12/0/1)	[31]
Pollution	60	15	(7/8/0)	http://openml.org/
Treasury	1049	15	(15/0/0)	http://sci2s.ugr.es/keel/dataset.php?cod=42
Baseball	337	16	(2/14/0)	http://sci2s.ugr.es/keel/dataset.php?cod=76#sub2
			Classification	
Mammographic	830	5	(0/5/0)	http://sci2s.ugr.es/keel/category.php?cat=clas
Dystrophy	209	5	(2/3/0)	[32]
Monk3	122	6	(0/6/0)	[29]
Appendicitis	106	7	(7/0/0)	http://sci2s.ugr.es/keel/dataset.php?cod=183
SAHeart	462	9	(5/3/1)	http://sci2s.ugr.es/keel/dataset.php?cod=184#sub1
Tic-Tac-Toe	958	9	(0/0/9)	[29]
Heart	303	13	(1/12/0)	[29]
House vote	232	16	(0/0/16)	[29]
Bands	365	19	(13/6/0)	http://sci2s.ugr.es/keel/dataset.php?cod=184#sub1
Hepatitis	80	20	(2/18/0)	[29]
Parkinson	195	22	(22/0/0)	[29]
Body	507	23	(22/1/0)	[33]
Thyroid	9172	27	(3/2/22)	[29]
WDBC	569	29	(29/0/0)	[29]
WPBC	198	32	(30/2/0)	[29]
Oil-Spill	937	49	(40/9/0)	http://openml.org/
Spam base	4601	57	(55/2/0)	[29]
Glaucoma	196	62	(62/0/0)	[32]
Nki 70	144	76	(71/5/0)	[34]
Musk	476	166	(0/166/0)	[35]

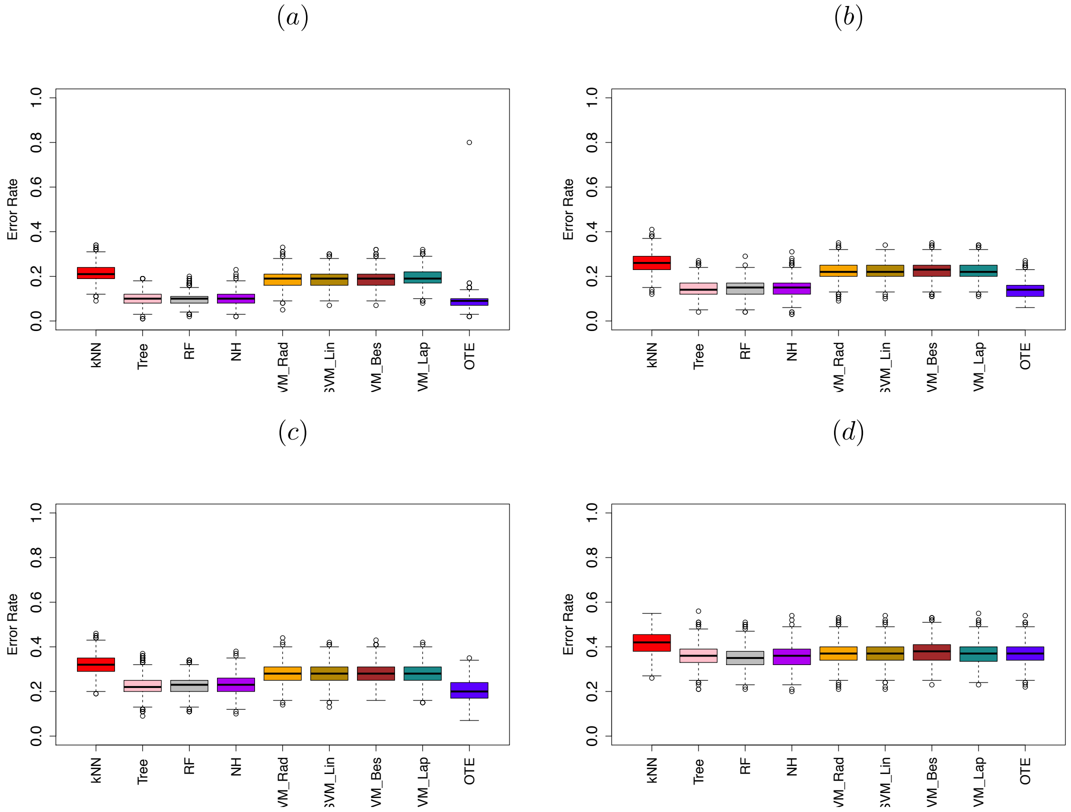

Figure 3: Box plots for kNN, tree, random forest (RF), node harvest (NH), SVM and (OTE) on the data simulated in Scenario 1. (a): simulation with Bayes error 9%, (b): simulation with Bayes error 14%, (c): simulation with Bayes error 17% and (d): simulation with Bayes error 33%. The best results of OTE can be seen in fugure (a) where the model produces a data with almost perfect tree structures. Figure (d) is the worst example of OTE

3.3. Experimental Setup for Benchmark Data Sets

Experiments carried out on the 35 data set are designed as follows. Each data set is divided into two parts, a training part and testing part. The training part consists of 90% of the total data while the testing part consists of the remaining 10% of the data. A total of $T = 1500$ independent classification and regression trees are grown on bootstrap samples from the (90% of) training data along with randomly selecting p features for splitting the nodes of the trees. The remaining 10% of training data is used for diversity check. In the cases of both regression and classification, the number p of features is kept constant at $p = \sqrt(d)$ for all data sets. The best of the total T trees are selected by using the method given in Section 2 and are used as the final ensemble (M is taken as 20% of T). Testing part of the data is applied on the final ensemble and a total of 1000 runs are carried out for each data set. Final result is the average of all these 1000 runs.

For tuning various parameters of CART, we used the R-Function "tune.rpart" available within the R-Package "e1071". We tried various values, (5,10,15,20,25,30) for finding the optimal number of splits and the minimal optimal depth of the trees.

For tuning the hyper parameters, *nodesize*, *ntree* and *mtry* of random forest, we used the function "tune.randomForest" available with in the R-Package "e1071" as used by [36]. For tuning the node size we tried values (1,5,10,15,20,25,30), for tuning ntree we tried values (500,1000,1500,2000) and for tuning mtry, we tried (sqrt(d), d/5, d/4, d/3, d/2). We tried all the possible values of mrty where $d < 12$.

The only parameter in the node harvest estimator is the number of nodes in the initial ensemble and for its large values the results are insensitive [18]. Meinshausen [18] showed for various data sets that initial ensemble size greater than 1000 yields almost the same results. In our experiments we kept this value fixed at 1500. In case of SVM, automatic estimation of sigma was used available with in the R package "kernlab". The rest of the parameters are kept at default values.

The same set of training and test data is used for tree, random forest, node harvest, SVM and our proposed method. Average unexplained variances and classification errors, for regression and classification respectively, are noted down for all the four methods on the data sets. All the experiments are done using R-Program version 3.0.2 [37]. The results are given in tables 4 and 5 for regression and classification respectively.

Table 4: Unexplained variances for regression data sets from kNN, tree, random forest, node harvest, SVM and *OTE*. The unexplained variance of the best performing method for the corresponding data set is shown in bold.

Data Set	n	d	kNN	Tree	RF	NH	SVM	SVM	SVM	SVM	OTE
							(Radial)	(Linear)	(Bessel)	(Laplacian)	
Bone	485	3	0.8932	0.7058	0.6601	0.6632	**0.6292**	0.7908	0.7369	0.6329	0.6454
Galaxy	323	4	0.0285	0.0952	0.0275	0.0686	**0.0253**	0.1153	0.0356	0.0262	0.0261
Friedman	1200	5	0.1373	0.3871	0.1212	0.4452	**0.0559**	0.2828	0.0849	0.0657	0.1364
CPU	209	7	0.1058	0.2838	0.0646	0.2659	0.3898	0.0916	0.2861	0.3143	**0.0600**
Concrete	103	7	0.3720	0.4989	0.2174	0.4307	0.0700	0.1743	**0.0623**	0.1806	0.2342
Abalone	4177	8	0.5347	0.5673	**0.4386**	0.6083	0.4410	0.4904	0.4433	0.4418	0.4473
MPG	398	8	0.3230	0.2301	0.1259	0.1990	0.1358	0.2066	0.1435	0.1359	**0.1203**
Stock	950	9	**0.0102**	0.0942	0.0121	0.1192	0.0153	0.1373	0.0274	0.0142	0.0110
Wine	1599	11	0.8975	0.7140	**0.4933**	0.7044	0.5980	0.6653	0.8991	0.5859	0.5072
Ozone	203	12	0.6430	0.4366	0.3061	0.3642	**0.2488**	0.3528	0.7967	0.2750	0.3016
Housing	506	13	0.4696	0.2821	0.1190	0.2477	0.1756	0.3055	0.8824	0.1853	**0.1160**
Pollution	60	15	0.9500	0.9500	0.6779	0.7728	0.6942	0.8144	0.9500	0.7326	**0.6653**
Treasury	1049	15	0.0075	0.0405	0.0040	0.0574	0.0062	0.0060	0.0077	0.0070	**0.0039**
Baseball	337	16	0.6931	0.3513	0.3434	0.3908	0.3641	0.3818	0.8765	0.3641	**0.3329**

Table 5: Classification error rates of kNN, tree, random forest, node harvest, SVM and *OTE*. The result of the best performing method for the corresponding data set is shown in bold.

Data Set	n	d	kNN	Tree	RF	NH	SVM	SVM	SVM	SVM	OTE
							(Radial)	(Linear)	(Bessel)	(Laplacian)	
Mammographic	830	5	0.1901	0.1631	0.1670	**0.1579**	0.1910	0.1750	0.1875	0.1863	0.1711
Dystrophy	209	5	0.1172	0.1482	0.1154	0.1470	0.0999	0.1122	0.1070	**0.0997**	0.1182
Monk3	122	6	0.1226	0.0773	**0.0728**	0.2699	0.0953	0.2254	0.0928	0.0938	0.0731
Appendicitis	106	7	0.1423	0.1640	0.1455	**0.1380**	0.2245	0.1726	0.1905	0.1650	0.1500
SAHeart	462	9	0.3363	0.2911	0.2897	**0.2762**	0.3075	0.3080	0.3332	0.3139	0.3178
Tic-Tac-Toe	958	9	0.3617	0.1082	**0.0317**	0.2861	0.2078	0.3948	0.1725	0.1972	0.0353
Heart	303	13	0.3500	0.2108	0.1629	0.1892	0.2342	0.1745	**0.1612**	0.1719	0.1743
House Vote	232	16	0.0825	0.0345	**0.0322**	0.1020	0.0330	0.0470	0.2211	0.0529	0.0340
Bands	365	19	0.3196	0.3683	0.2683	0.3647	0.3669	0.3202	0.4724	0.5573	**0.2601**
Hepatitis	80	20	0.3831	0.1868	0.1385	0.1296	0.1406	0.1568	0.5629	0.1490	**0.1229**
Parkinson	195	22	0.1620	0.1456	0.0894	0.1235	0.1385	0.1941	0.2838	0.1928	**0.0859**
Body	507	23	0.0226	0.0788	0.0395	0.0744	0.0156	**0.0136**	0.5505	0.0219	0.0380
Thyroid	9172	27	0.0388	0.0126	**0.0100**	0.0203	0.1113	0.0310	0.2936	0.0834	**0.0100**
WDBC	569	29	0.0671	0.0686	0.0388	0.0525	0.0415	**0.0264**	0.6297	0.0403	0.0375
WPBC	198	32	0.2413	0.2815	0.1958	0.2282	0.2848	0.2881	0.5684	0.3084	**0.1921**
Oil-Spill	937	49	0.0435	0.0366	0.0330	0.0360	0.0756	0.1400	0.0387	0.1467	**0.0321**
Spam base	4601	58	0.1747	0.1083	0.0469	0.0944	0.0941	0.0725	0.4820	0.1020	**0.0460**
Sonar	208	60	0.1790	0.2879	0.1615	0.2390	0.1710	0.2505	0.5300	0.2698	**0.1600**
Glaucoma	196	62	0.1934	0.1237	0.1052	0.1154	0.1108	0.1565	0.6397	0.1664	**0.1051**
Nki 70	144	76	0.1827	0.1683	0.1466	0.1448	0.2664	0.3381	0.4260	0.4089	**0.1399**
Musk	476	166	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

3.4. Discussion

The results given in tables 4 and 5 show that the proposed method is performing better than the other methods on many of the data sets. In the case of regression problems, our method is giving better results than the other methods considered on 7 data sets out of a total of 14 data sets, whereas on 2 data sets, Wine and Abalone, random forest gives the best performance. On 5 of the data sets, Bone, Galaxy, Freidman, and Ozone, SVM with radial kernel and Concrete with Bessel kernel gave the best results. Tree and kNN are unsurprisingly the worst performers in all the methods with the exception of Stock data set where kNN is the best.

In the case of classification problems, the new method is giving better results than the other methods considered on 10 data sets out of a total of 21 data sets and comparable to random forest on 1 data set. On 3 data sets, random forest gives the best performance. On three of the data sets, Mammographic, Appendicitis and SAHeart, node harvest classifier gives the best result among all other methods. SVM is better than the others on 4 data sets.

Overall, the proposed method gave better results on 15 data sets and comparable results on 2 data set.

We kept all our parameters in the ensemble fixed for the sake of simplicity. Searching for the optimal total number T of trees grown before the selection process, the percentage M of best trees selected at the first phase, node size and the number of features for splitting the nodes might further improve our results. Large values are recommended for the size of the initial set under the available computation resources and a value of $T \geq 1500$ is expected to work well in general. This can be seen in Figure 4 that show the effect of the number of trees in the initial set on (a): unexplained variance and (b): misclassification error for the data sets given using OTE.

One important parameter of the our method is the number M of best trees selected at the first phase for the final ensemble. Various values of M reveal different behaviour of the method. We considered the effect of $M = (1\%, 5\%, 10\%, 20\%, ..., 70\%)$ of the total T trees on the the method for both re-

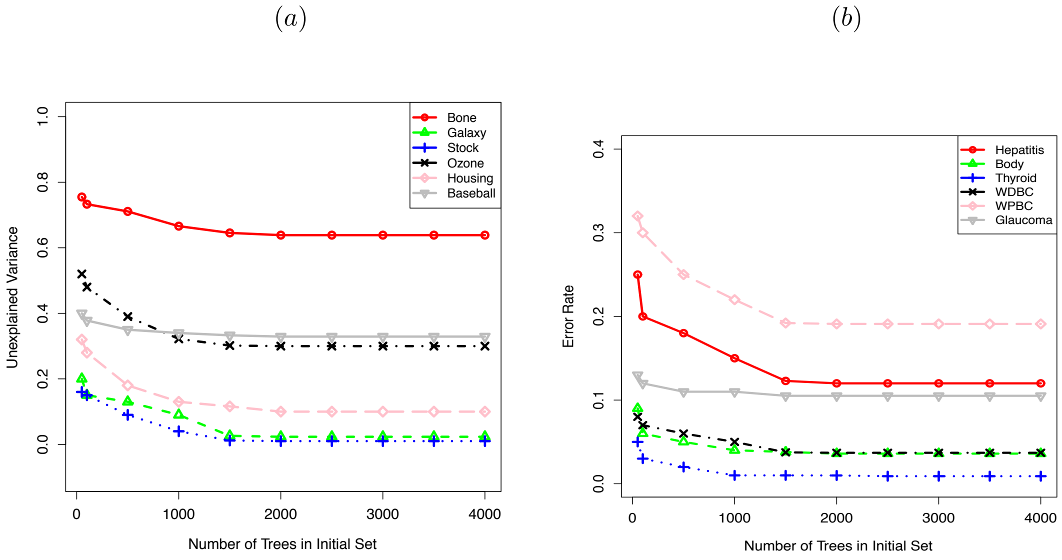

Figure 4: The effect of the number of trees in the initial set on (a): unexplained variance and (b): misclassification error for the data sets given using *OTE*. In both the cases, number of trees larger than 1500 can be recommended

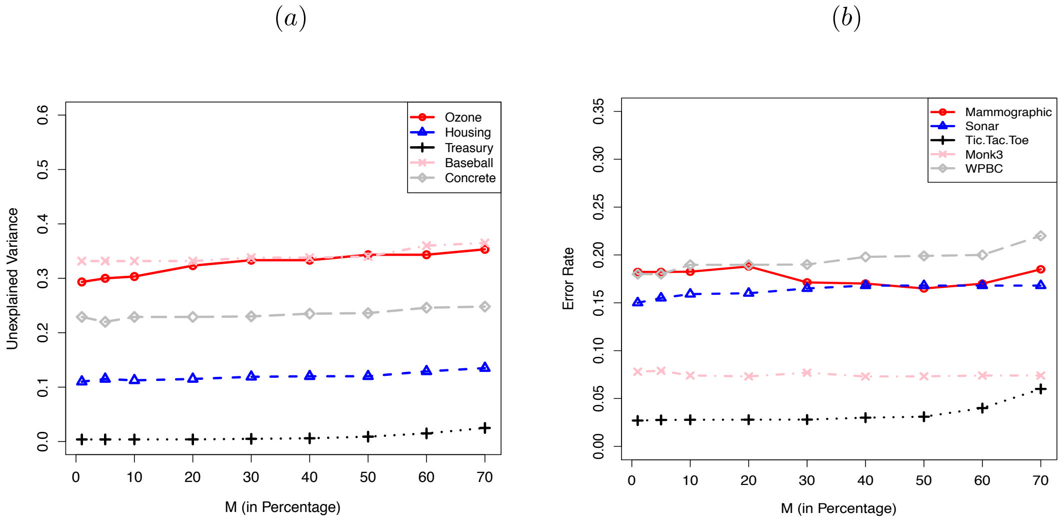

Figure 5: Effect of M on the unexplained variances, (Fig. (a)), and error rate (Fig. (b)), of the data sets shown using *OTE*. The value of M in percentage is on the x-axis and unexplained variance on the y-axis.

gression and classification as shown in Figure 5. It is clear from figure 5 that the highest accuracy is obtained by using only a small portion, $1\% - 10\%$, of the total trees that are individually strong which is further reduced in the second phase. This may significantly decrease the storage costs of the ensemble while increasing/without loosing accuracy. On the other hand, having a large number of trees may not only increase storage costs of the resulting ensemble but also decrease the overall prediction accuracy of the ensemble. This can be seen in Figure 5 in the cases of Concrete, WPBC and Ozone data sets where the best results are obtained at about less than 5% best trees of the total trees at the first phase. This might be due to the reason that in such cases the possibility of having poor trees is high if the size of ensemble is large and trees are simply grown with out considering their individual and collective behaviours.

We also looked at the effect of various numbers $p = \sqrt{d}, \frac{d}{5}, \frac{d}{4}, \frac{d}{3}, \frac{d}{2}$ of features selected at random for splitting the nodes of the trees on the unexplained variances and classification error in the cases of both regression and classification, respectively, for some data sets. The graph is shown in Figure 6. The only reason that random forest is considered as an improvement over bagging is the inclusion of additional randomness by randomly selecting a subset of features for splitting the nodes of the tree. The effect of this randomness can be seen in Figure 6 where different values of p results in different unexplained variances/classification errors for the data sets. For example in the case of Ozone data, selecting a higher value of p adversely affects the performance. For some data sets, WPBC for example, selecting large p results in better performance.

4. Conclusion

The possibility of selecting best trees from an original ensemble of a large number of trees, and combining them together to vote/average for the response is considered. The new method is applied on 35 data sets consisting of 14 regression problems and 21 classification problems. The ensemble performed better than kNN, tree, random forest, node harvest and SVM on many of the

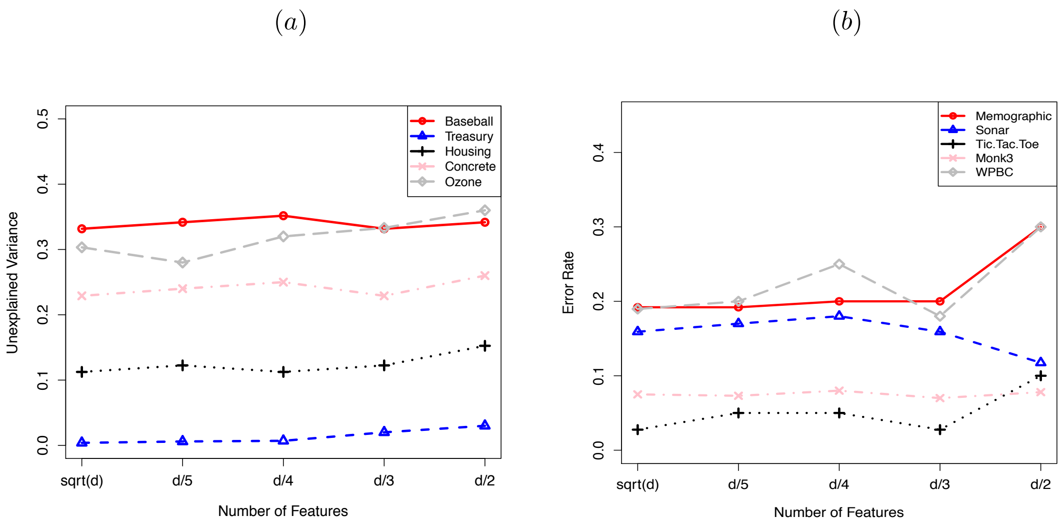

Figure 6: Effect of the number of features (on x-axis) selected at random for splitting the nodes of the trees on the unexplained variance (Fig. (a)), and error rate (Fig. (b)) for the data sets shown using *OTE*.

data sets. The intuition for the better performance of the new method is that if the base learners in the ensemble are individually accurate and diverse, then their ensemble must give better or at least comparable results as compared to the one consisting of weak learners. This might also be due to the reason that there could be various different meaningful structures present in the data that could not be captured by an ordinary algorithm. Our method tries to find these meaningful structures in the data and ignore those that only increase the error.

Our simulation reveals that the method can find meaningful patterns in the data as effectively as other complex methods might do.

Even if one could get comparable results by using a few strong and diverse base learners to those based upon thousands of weak base learners should be welcomed. This might be very helpful in in reducing the associated storage costs of tree forests with little or no loss of prediction accuracy.

The method is implemented in an R-Package called "*OTE*" [38].

The fact that we use the out-of-bag sample for choosing the best learners at the first place, there might be a chance of not properly assessing the individual learners and thus selecting weak learners for the final ensemble. One could investigate the possibility of choosing the individual learners by using some other criteria, cross validation for example. The use of some variable selection methods, [39, 40, 41, 42, 43], might, in conjunction with our method, lead to further improvements.

References

[1] R. Schapire, The strength of weak learnability, Machine learning 5 (2) (1990) 197–227.

[2] P. Domingos, Using partitioning to speed up specific-to-general rule induction, in: Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models, Citeseer, 1996, pp. 29–34.

[3] J. Quinlan, Bagging, boosting, and c4. 5, in: Proceedings of the National Conference on Artificial Intelligence, 1996, pp. 725–730.

[4] R. Maclin, D. Opitz, Popular ensemble methods: An empirical study, Journal of Artificial Research 11 (2011) 169–189.

[5] T. Hothorn, B. Lausen, Double-bagging: Combining classifiers by bootstrap aggregation, Pattern Recognition 36 (6) (2003) 1303–1309.

[6] T. T. Nguyen, T. T. T. Nguyen, X. C. Pham, A. W.-C. Liew, A novel combining classifier method based on variational inference, Pattern Recognition 49 (2016) 198–212.

[7] R. Younsi, A. Bagnall, Ensembles of random sphere cover classifiers, Pattern Recognition 49 (2016) 213–225.

[8] D. Ravì, M. Bober, G. Farinella, M. Guarnera, S. Battiato, Semantic segmentation of images exploiting dct based features and random forest, Pattern Recognition 52 (2016) 260–273.

[9] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, An ensemble of filters and classifiers for microarray data classification, Pattern Recognition 45 (1) (2012) 531–539.

[10] M. Bhardwaj, V. Bhatnagar, K. Sharma, Cost-effectiveness of classification ensembles, Pattern Recognition 57 (2016) 84–96.

[11] Y. Quan, Y. Xu, Y. Sun, Y. Huang, Supervised dictionary learning with multiple classifier integration, Pattern Recognition 55 (2016) 247–260.

[12] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: Bagging, boosting, and variants, Machine learning 36 (1) (1999) 105–139.

[13] T. Mitchell, Machine learning, Burr Ridge, IL: McGraw Hill.

[14] K. Tumer, J. Ghosh, Error correlation and error reduction in ensemble classifiers, Connection science 8 (3-4) (1996) 385–404.

[15] K. Ali, M. Pazzani, Error reduction through learning multiple descriptions, Machine Learning 24 (3) (1996) 173–202.

[16] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[17] S. Bernard, L. Heutte, S. Adam, On the selection of decision trees in random forests, in: International Joint Conference on Neural Networks, IEEE, 2009, pp. 302–307.

[18] N. Meinshausen, Node harvest, The Annals of Applied Statistics 4 (4) (2010) 2049–2072.

[19] T. Oshiro, P. Perez, J. Baranauskas, How many trees in a random forest?, Machine Learning and Data Mining in Pattern Recognition (2012) 154–168.

[20] P. Latinne, O. Debeir, C. Decaestecker, Limiting the number of trees in random forests, in: Multiple Classifier Systems: Second International Workshop, MCS 2001 Cambridge, UK, July 2-4, 2001 Proceedings, Vol. 2, Springer Science & Business Media, 2001, p. 178.

[21] G. W. Brier, Verification of forecasts expressed in terms of probability, Monthly weather review 78 (1) (1950) 1–3.

[22] Z. Khan, A. Gul, O. Mahmoud, M. Miftahuddin, A. Perperoglou, W. Adler, B. Lausen, An ensemble of optimal trees for class membership probability estimation, in: Analysis of Large and Complex Data, Springer, 2016, pp. 395–409.

[23] A. Gul, A. Perperoglou, Z. Khan, O. Mahmoud, M. Miftahuddin, W. Adler, B. Lausen, Ensemble of a subset of knn classifiers, Advances in Data Analysis and Classification (2016) 1–14.

[24] A. Gul, Z. Khan, A. Perperoglou, O. Mahmoud, M. Miftahuddin, W. Adler, B. Lausen, Ensemble of subset of k-nearest neighbours models for class membership probability estimation, in: Analysis of Large and Complex Data, Springer, 2016, pp. 411–421.

[25] K. Halvorsen, ElemStatLearn: Data sets, functions and examples, r package version 2012.04-0 (2012).
URL `http://CRAN.R-project.org/package=ElemStatLearn`

[26] L. K. Bachrach, T. Hastie, M.-C. Wang, B. Narasimhan, R. Marcus, Bone mineral acquisition in healthy asian, hispanic, black, and caucasian youth: a longitudinal study, Journal of Clinical Endocrinology & Metabolism 84 (12) (1999) 4702–4712.

[27] R. Buta, The structure and dynamics of ringed galaxies. iii-surface photometry and kinematics of the ringed nonbarred spiral ngc 7531, The Astrophysical Journal Supplement Series 64 (1987) 1–37.

[28] J. H. Friedman, Multivariate adaptive regression splines, The annals of statistics (1991) 1–67.

[29] K. Bache, M. Lichman, UCI machine learning repository (2013).
URL `http://archive.ics.uci.edu/ml`

[30] F. Leisch, E. Dimitriadou, mlbench: Machine Learning Benchmark Problems, r package version 2.1-1 (2010).

[31] N. Meinshausen, nodeHarvest: Node Harvest for regression and classification, r package version 0.6 (2013).
URL http://CRAN.R-project.org/package=nodeHarvest

[32] A. Peters, T. Hothorn, ipred: Improved Predictors, r package version 0.9-1 (2012).
URL http://CRAN.R-project.org/package=ipred

[33] C. Hurley, gclus: Clustering Graphics, r package version 1.3.1 (2012).
URL http://CRAN.R-project.org/package=gclus

[34] J. J. Goeman, penalized: Penalized generalized linear models., penalized R package, version 0.9-42 (2012).
URL http://CRAN.R-project.org/package=penalized

[35] A. Karatzoglou, A. Smola, K. Hornik, A. Zeileis, kernlab – an S4 package for kernel methods in R, Journal of Statistical Software 11 (9) (2004) 1–20.
URL http://www.jstatsoft.org/v11/i09/

[36] W. Adler, A. Peters, B. Lausen, et al., Comparison of classifiers applied to confocal scanning laser ophthalmoscopy data, Methods of information in medicine 47 (1) (2008) 38–46.

[37] R Core Team, R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria (2014).
URL http://www.R-project.org/

[38] A. P. O. M. W. A. M. Zardad Khan, Asma Gul, B. Lausen, OTE: Optimal Trees Ensembles, r package version 1.0 (2014).
URL https://cran.r-project.org/package=OTE

[39] A. Hapfelmeier, K. Ulm, A new variable selection approach using random forests, Computational Statistics & Data Analysis 60 (0) (2013) 50 – 69. doi:http://dx.doi.org/10.1016/j.csda.2012.09.020.

[40] O. Mahmoud, A. Harrison, A. Perperoglou, A. Gul, Z. Khan, M. V. Metodiev, B. Lausen, A feature selection method for classification within functional genomics experiments based on the proportional overlapping score, BMC Bioinformatics 15 (1) (2014) 274.

[41] O. Mahmoud, A. Harrison, A. Perperoglou, A. Gul, Z. Khan, B. Lausen, propOverlap: Feature (gene) selection based on the Proportional Overlapping Scores, r package version 1.0 (2014).
URL http://CRAN.R-project.org/package=propOverlap

[42] K. Kim, H. Lin, J. Y. Choi, K. Choi, A design framework for hierarchical ensemble of multiple feature extractors and multiple classifiers, Pattern Recognition 52 (2016) 1–16.

[43] J. Calvo-Zaragoza, J. J. Valero-Mas, J. R. Rico-Juan, Improving knn multi-label classification in prototype selection scenarios using class proposals, Pattern Recognition 48 (5) (2015) 1608–1622.

Acknowlegments

We acknowledge support from grant number ES/L011859/1, from The Business and Local Government Data Research Centre, funded by the Economic and Social Research Council to provide researchers and analysts with secure data services.