An Ensemble of Optimal Trees for Class Membership Probability Estimation

Zardad Khan[1], Asma Gul[1], Osama Mahmoud[1], Miftahuddin Miftahuddin[1], Aris Perperoglou[1], Werner Adler[2], and Berthold Lausen[1]

[1] Department of Mathematical Sciences, University of Essex, Colchester, UK.
zkhan@essex.ac.uk

[2] Department of Biometry and Epidemiology, University of Erlangen-Nuremberg, Germany

Abstract. Machine learning methods can be used for estimating the class membership probability of an observation. We propose an ensemble of optimal trees in terms of their predictive performance. This ensemble is formed by selecting the best trees from a large initial set of trees grown by random forest. A proportion of trees is selected on the basis of their individual predictive performance on out-of-bag observations. The selected trees are further assessed for their collective performance on an independent training data set. This is done by adding the trees one by one starting from the highest predictive tree. A tree is selected for the final ensemble if it increases the predictive performance of the previously combined trees. The proposed method is compared with probability estimation tree, random forest and node harvest on a number of bench mark problems using Brier score as a performance measure. In addition to reducing the number of trees in the ensemble, our method gives better results in most of the cases. The results are supported by a simulation study.

Keywords

TREE SELECTION, ENSEMBLE METHODS, PROBABILITY ESTIMATION TREES

1 Introduction

The usual task of pattern recognition or discrimination is to make a simple statement about the group membership of an individual. For example, this simple statement about a tumour patient could be that he/she is having a malignant or a benign tumour. This might also be of interest to know the class membership probability of the individual which is an important biomedical application. It is usually required by surgeons, oncologists, pathologists, professionals involved in internal medicine and human genetics and pediatricians (Malley et al. (2012)). For instance, carrier probabilities are calculated in genetic counseling and treatment response probability is estimated in personalized medicine of every patient (Kruppa et al. (2012), Kruppa et al. (2014b)).

The logistic regression model is the standard and classical approach for estimating individual probabilities (Kruppa et al. (2012), Kruppa et al. (2014a)). A major problem with the logistic regression is the requirement of correct and full specification of the model. Misspecified model will give biased and inconsistent results.

Machine learning methods on the other hand can be used as non-parametric alternatives to the classical logistic regression models to avoid

the assumptions involved and to overcome the problem of misspecification. These methods have been utilized in various biomedical applications (Kruppa et al. (2012), Malley et al. (2012), Kruppa et al. (2014a)). Most of these methods are based on the idea of combining multiple models to build a strong model (Ali and Pazzani (1996), Hothorn and Lausen (2003)). Studies have shown that the generalization error can be reduced by combining the outputs of multiple models (Maclin and Opitz (2011)). In this paper, the possibility of creating an ensemble of optimal trees for class membership probability estimation is considered that is motivated by Brieman's (2001) upper bound for the overall prediction error of a random forest ensemble which is given by

$$PE^* \leq \bar{\rho}\, PE_j, \tag{1}$$

where $j = 1, 2, 3, ..., T$. T is the total number of trees in the forest, PE^* is the overall prediction error of a random forest, $\bar{\rho}$ is the the weighted correlation between residuals from two independent trees and PE_j is the prediction error of tree j in the forest. This relation indicates that individually accurate and diverse trees could make an efficient forest. Based on this intuition, trees are selected from a total of T trees grown on bootstrap samples drawn form a given learning data set. A similar approach is proposed in Gul et al. (2015) where the idea of random feature set selection and bagging is used with k-nearest neighbours classifiers for the issue of non-informative features in the data. We compare the method with k-nearest neighbours, tree, random forest (RF), node harvest (NH) (Meinshausen (2010)), and support vector machines for probability estimation. The rest of the paper is arranged as follows: Section 2 discusses the methods mentioned before; Section 3 describes the Brier score; Section 4 introduces our method; Section 5 gives experiments and results and conclusion is given in Section 6.

2 Probability Machines

Machine learning techniques that are used to give estimates of probability for the group membership in binary class problems are named probability machines by Malley et al. (2012). Here we briefly explain how kNN, tree, RF, NH and SVM could be used for estimating class membership probabilities before introducing our method, the Optimal Trees Ensemble (*OTE*).

2.1 Probability Estimation Trees (PETs)

To find the conditional probability, $P(\boldsymbol{Y}|\boldsymbol{X})$, of an individual belonging to a particular class, the steps are

1. On a bootstrap sample from the training data $\mathcal{L} = (\mathbf{X}, \mathbf{Y})$, grow a classification or regression tree.
2. Filter a test observation through the tree until it reaches to a leaf node Q'.
3. The proportion $p_i, i = 1, 2$ of an observations of a particular class in Q' is determined which is the required probability, where

$$p_i = \frac{\text{\# of } i\text{th class observations in } Q'}{\text{\# of observations in } Q'}.$$

2.2 Random Forest as Probability Machine

The Breiman (2001) random forest can effectively be used for estimating the conditional probability function $P(\mathbf{Y}|\mathbf{X})$ (Liaw and Wiener (2002)). To find the group membership probability $P(\boldsymbol{Y}|\mathbf{X})$, take the following steps.

1. Draw T bootstrap samples from the given training data $\mathcal{L} = (\mathbf{X}, \mathbf{Y})$ and grow T probability estimation trees.
2. A test observation is filtered through each tree until it reaches a leaf node.
3. The estimate of class probability is the average proportion of a class observations in the leaf nodes of all the trees where the test observation resides.

2.3 Node Harvest as Probability Machine

Node harvest, proposed by Meinshausen (2010), is a tree based algorithm that takes a large set of nodes as an initial ensemble and selects the most useful nodes for the final decision. Class membership probability of an observation is estimated as follows

1. Take a sufficiently large number of nodes from an initial tree ensemble.
2. Allow non-negative weights that take on values in the continuous interval [0,1] and select those nodes that are assigned the highest weights.
3. Remove nodes that are identical.
4. The estimate of class probability is the average proportion of a class observations in the selected nodes where the test observation resides.

2.4 *k*-Nearest Neighbours as Probability Machine

To estimate class membership probability of a test observation via kNN, the steps are

1. Compute the distance of a test observation from all the training instances.
2. Find k nearest instances to the test point according to the distance.
3. The estimate of the probability is the proportion of instances of a class in the k nearest neighbours.

2.5 Support Vector Machines (SVMs) for Probability Estimation

Given a training data set $\mathcal{L} = (\mathbf{X}, \mathbf{Y})$, support vector machines can be used to produce estimates of class membership probability instead of class labels. This is done by the implementation of Platt's posteriori probabilities (Platt (2000)) in several R packages, where the following sigmoid function is used.

$$p(y|\mathbf{X}) = \frac{1}{1 + exp(Af(\mathbf{X}) + B)}, \text{ where } f(\mathbf{X}) \text{ is a decision fuction.} \qquad (2)$$

A and B are the parameters to be estimated. For further information on this, see Platt (2000).

Before introducing the proposed ensemble, we explain the performance measure used in the algorithm and its comparison to other methods.

3 Assessment of the Probability Machines

We use the Brier score as performance measure which is generally used when the true probabilities are not available (Malley et al. (2012)). Gneiting and Raftery (2007) argued that the Brier score is a proper score and its minimum value can only be obtained if the estimated probabilities are taken exactly equal to the true unknown probabilities. It means that any probability machine having the smallest Brier score is estimating class probabilities in the best possible way. The Brier score is represented by the following equation.

$$BS = \mathbb{E}\left(\boldsymbol{Y} - P(\boldsymbol{Y}|\mathbf{X})\right)^2. \tag{3}$$

where $\boldsymbol{Y}$ is the state of the response variable in the 0,1 form for the two classes and $P(\boldsymbol{Y}|\mathbf{X})$ is the true unknown probability for the binary response given the features. An estimator for the above score is

$$\hat{BS} = \frac{\sum_{i=1}^{\#\text{ of test cases}} \left(y_i - \hat{P}(y_i|\mathbf{X})\right)^2}{\text{total \# of test cases}}. \tag{4}$$

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

4 The Ensemble of Optimal Trees, *OTE*

For obtaining the ensemble of best (accurate and diverse) trees, divide the given training data $\mathcal{L} = (\mathbf{X}, \mathbf{Y})$ randomly into two non overlapping parts, $\mathcal{L}_{\mathcal{B}} = (\mathbf{X_B}, \mathbf{Y_B})$ and $\mathcal{L}_{\mathcal{V}} = (\mathbf{X_V}, \mathbf{Y_V})$. Grow T trees on T bootstrap samples from $\mathcal{L}_{\mathcal{B}} = (\mathbf{X_B}, \mathbf{Y_B})$. Accurate and diverse trees are selected as follows

1. Estimate the error of each tree (growing by random forest without pruning) by using the out-of-bag (OOB) observations (observations left out from a bootstrap sample) as the validation data.
2. Arrange the trees in ascending order with respect to the prediction errors and take the first M trees.
3. To find diverse trees, the second best tree out of the M trees is combined with the best tree to get an ensemble of size two and see how they perform on $\mathcal{L}_{\mathcal{V}} = (\mathbf{X_V}, \mathbf{Y_V})$. Then the third best tree is added and the performance is measured and so on until the final Mth tree is added.
4. Tree $\hat{L}_k, k = 1, 2, 3, ..., M$ is selected if its addition to the ensemble without the kth tree fulfils the following criterion.
   - Let $BS^{\langle k-1\rangle}$ be the Brier score of the ensemble without the kth tree and $BS^{\langle k\rangle}$ be the Brier score of the ensemble including the kth tree, then tree $\hat{L}_k$ is selected if
   $$BS^{\langle k\rangle} < BS^{\langle k-1\rangle}. \tag{5}$$

To estimate class probability of an observation, apply steps 2 and 3 of random forest on the M selected trees. A simple illustrative flow chart of the steps is given in Figure 1.

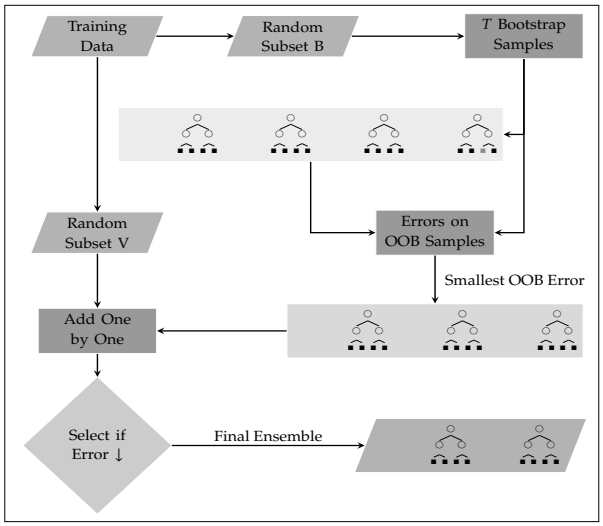


Fig. 1. A flow chart of the steps of *OTE* for probability estimation.

5 Experiments and Results

5.1 Simulation

We simulate data consisting of various structures to make the recognition problem slightly difficult for simple classifiers, kNN and PET for example. We aimed our method to perform better than the simple classifier and compete with the complex and powerful classifiers, SVM, random forest and node harvest in our study, in finding the structures. To this end we generate four models with a different number of tree components where all the components are partitioning the data set on a subset of the feature space. For each model we consider four different cases/complexity levels by altering the weights η_{ijk} of the tree nodes to move from highly non-uniform distributions (low entropy) to distributions with high entropy. Thus we get four different values of the Bayes error where the lowest Bayes error means a data set with meaningful patterns and the highest Bayes error indicates a data set with no patterns. Table 1 lists the various values of η_{ijk} used in model 1, 2, 3, and 4. Node weights for getting the four complexity levels are given in four columns of the table for $k = 1, 2, 3, 4$, for each model. All the four models are derived from the following 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$(0, 1)$, T being the total number of trees.

$$p(y|\mathbf{X}) = \frac{exp\left(b \times \left(\frac{\zeta_m}{T} - a\right)\right)}{1 + exp\left(b \times \left(\frac{\zeta_m}{T} - a\right)\right)}, \text{ where } \zeta_m = \sum_{t=1}^{T} \Upsilon_t. \tag{6}$$

$a, b \in \mathtt{R}$ are any arbitrary constants, $m = 1, 2, 3, 4$ is the model number and ζ_m's are $n \times 1$ vector of probabilities. T is the total number of trees in a model and Υ_t's are probabilities for a particular class in the response $\mathbf{Y}$ generated by different tree structures as follows

$$\begin{aligned}
\Upsilon_1 &= \eta_{11k} \times \mathbb{1}(x_1 \leq 0.5 \& x_3 \leq 0.5) + \eta_{12k} \times \mathbb{1}(x_1 \leq 0.5 \& x_3 > 0.5) \\
&\quad + \eta_{13k} \times \mathbb{1}(x_1 > 0.5 \& x_2 \leq 0.5) + \eta_{14k} \times \mathbb{1}(x_1 > 0.5 \& x_2 > 0.5), \\
\Upsilon_2 &= \eta_{21k} \times \mathbb{1}(x_4 \leq 0.5 \& x_6 \leq 0.5) + \eta_{22k} \times \mathbb{1}(x_4 \leq 0.5 \& x_6 > 0.5) \\
&\quad + \eta_{23k} \times \mathbb{1}(x_4 > 0.5 \& x_5 \leq 0.5) + \eta_{24k} \times \mathbb{1}(x_4 > 0.5 \& x_5 > 0.5), \\
\Upsilon_3 &= \eta_{31k} \times \mathbb{1}(x_7 \leq 0.5 \& x_8 \leq 0.5) + \eta_{32k} \times \mathbb{1}(x_7 \leq 0.5 \& x_8 > 0.5) \\
&\quad + \eta_{33k} \times \mathbb{1}(x_7 > 0.5 \& x_9 \leq 0.5) + \eta_{34k} \times \mathbb{1}(x_7 > 0.5 \& x_9 > 0.5), \\
\Upsilon_4 &= \eta_{41k} \times \mathbb{1}(x_{10} \leq 0.5 \& x_{11} \leq 0.5) + \eta_{42k} \times \mathbb{1}(x_{10} \leq 0.5 \& x_{11} > 0.5) \\
&\quad + \eta_{43k} \times \mathbb{1}(x_{10} > 0.5 \& x_{12} \leq 0.5) + \eta_{44k} \times \mathbb{1}(x_{10} > 0.5 \& x_{12} > 0.5), \\
\Upsilon_5 &= \eta_{51k} \times \mathbb{1}(x_{13} \leq 0.5 \& x_{14} \leq 0.5) + \eta_{52k} \times \mathbb{1}(x_{13} \leq 0.5 \& x_{14} > 0.5) \\
&\quad + \eta_{53k} \times \mathbb{1}(x_{13} > 0.5 \& x_{15} \leq 0.5) + \eta_{54k} \times \mathbb{1}(x13 > 0.5 \& x_{15} > 0.5), \\
\Upsilon_6 &= \eta_{61k} \times \mathbb{1}(x_{16} \leq 0.5 \& x_{17} \leq 0.5) + \eta_{62k} \times \mathbb{1}(x_{16} \leq 0.5 \& x_{17} > 0.5) \\
&\quad + \eta_{63k} \times \mathbb{1}(x_{16} > 0.5 \& x_{18} \leq 0.5) + \eta_{64k} \times \mathbb{1}(x_{16} > 0.5 \& x_{18} > 0.5),
\end{aligned}$$

where $0 < \eta_{ijk} < 1$ are weights given to the nodes of the trees, $k = 1, 2, 3, 4$. The four models use the following specifications for using (6)

Model 1

This model consists of 3 tree components each based on 3 variables. Therefore, $T = 3$, $\zeta_1 = \sum_{t=1}^{3} \Upsilon_t$ and $\mathbf{X}$ becomes a $n \times 9$ dimensional vector. A tree used in this model is shown in Figure 2.

Model 2

For this model we take $T = 4$ trees where $\zeta_2 = \sum_{t=1}^{4} \Upsilon_t$ and $\mathbf{X}$ becomes a $n \times 12$ dimensional vector.

Model 3

This model is based on $T = 5$ trees such that $\zeta_3 = \sum_{t=1}^{5} \Upsilon_t$ and $\mathbf{X}$ becomes a $n \times 15$ dimensional vector.

Model 4

This model consist of 6 tree components with $T = 6$, $\zeta_4 = \sum_{t=1}^{6} \Upsilon_t$ and $\mathbf{X}$ becomes a $n \times 18$ dimensional vector.

We see in Table 2 that tree, kNN, NH and SVM gave consistently poor performance as compared to RF and *OTE*. *OTE* gave comparable results with RF in most of the cases. Comparable/better results can be seen in the first of the four cases of all the remaining models. From these results, it follows that the proposed method can produce comparable results to random forest with a significant reduction in the ensemble size (given in the last column of Table 2) if there are some meaningful patterns in the data.

Table 1. Node weights, η_{ijk}, used in simulation models where i is tree number, j is node number in each tree and k is denoting a variant of the weights.

Model 1						Model 2						Model 3						Model 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

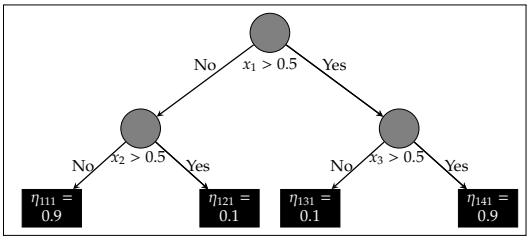


Fig. 2. A tree used in simulation model 1.

5.2 Bench Mark Problems

We considered 20 bench mark problems taken from various open sources. Dystrophy and Glaucoma data sets are taken from "ipred" R-package, Musk from "kernlab" R-package and Body data set is from "gclus" R-package. Appendicitis and SAHeart are from `http://sci2s.ugr.es/keel/dataset.php?cod=183`. Oil-Spill data is from `http://openml.org/d?from=180`. All the rest of the data sets are from UCI machine learning repository `http://archive.ics.uci.edu/ml/`. A brief description of these data is given in the first four columns of Table 3 where n is sample size and d is the number of features.

5.3 Experimental Setup and Results for Bench Mark Problems

The data sets are divided into two parts. The training part consisted of 90% of observations (of which 90% is used for bootstrapping and 10% for diversity check) and the remaining part is taken as the testing part. A total of 1000 runs are performed to calculate the average Brier score on all the data sets. The results are given in Table 3 where the average Brier scores of kNN, tree,

Table 2. Brier scores of *k*NN, tree, RF, NH, SVM and *OTE* on simulated data. The last column is the percentage reduction in ensemble size of *OTE* compared to RF.

Model	d	n	Bayes Error	kNN	Tree	RF	NH	SVM (Radial)	SVM (Linear)	SVM (Bessel)	SVM (Laplacian)	OptTreesEns	Reduction in Ensemble Size(%)
			0.09	0.16	0.09	0.10	0.12	0.13	0.13	0.14	0.13	0.08	90.7
			0.14	0.18	0.12	0.12	0.14	0.16	0.16	0.16	0.16	0.13	89.5
Model 1	9	1000	0.17	0.22	0.13	0.12	0.14	0.19	0.18	0.19	0.18	0.12	89.5
			0.33	0.27	0.23	0.22	0.22	0.23	0.23	0.23	0.22	0.23	90.8
			0.21	0.19	0.16	0.13	0.16	0.16	0.16	0.19	0.16	0.13	89.9
			0.24	0.21	0.18	0.15	0.17	0.17	0.17	0.20	0.17	0.15	89.7
Model 2	12	1000	0.28	0.24	0.21	0.18	0.2	0.20	0.20	0.22	0.20	0.19	89.7
			0.3	0.25	0.22	0.21	0.21	0.21	0.21	0.23	0.21	0.21	89.2
			0.15	0.21	0.17	0.14	0.18	0.16	0.16	0.25	0.16	0.14	90.7
			0.18	0.21	0.18	0.15	0.18	0.17	0.17	0.25	0.17	0.16	89.1
Model 3	15	1000	0.21	0.22	0.18	0.16	0.18	0.18	0.18	0.25	0.18	0.16	91.1
			0.24	0.24	0.2	0.19	0.2	0.19	0.19	0.25	0.19	0.18	89.9
			0.21	0.22	0.2	0.16	0.19	0.17	0.17	0.19	0.18	0.16	89.8
			0.22	0.23	0.2	0.16	0.19	0.18	0.18	0.20	0.19	0.17	89.3
Model 4	18	1000	0.25	0.25	0.22	0.18	0.2	0.20	0.20	0.21	0.22	0.18	90.5
			0.26	0.26	0.22	0.19	0.2	0.21	0.21	0.22	0.24	0.19	90.2

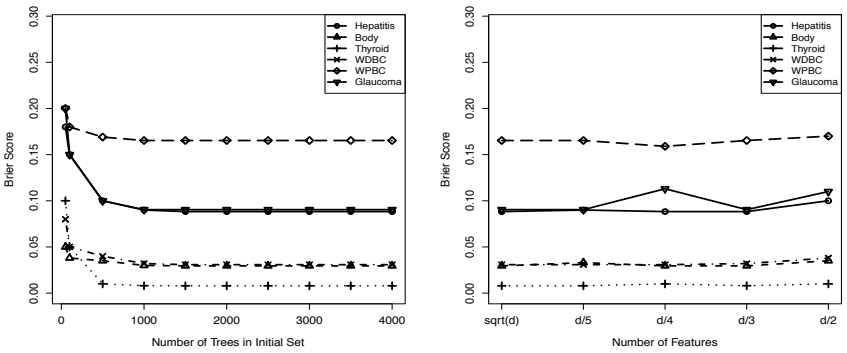
random forest, node harvest, SVM and *OTE* are given against each data set. Four kernels; Radial, Linear, Bessel and Laplacian, are considered for SVM with the rest of parameters on their default values in the "kernlab" R package. 10-fold cross validation is used for tuning the parameters of *k*NN, tree and RF. *k*NN is tuned for $k = 1, \ldots, 10$. For finding the optimal number of splits and the minimal optimal depth of the trees, values $(5, 10, 15, 20, 25, 30)$ are tried. For tuning the node size of RF, we tried values $(1, 5, 10, 15, 20, 25, 30)$, for *ntree*, $(500, 1000, 1500, 2000)$ and for tuning *mtry*, we tried $(\text{sqrt}(d), d/5, d/4, d/3, d/2)$ where d is the total number of features. Number of nodes in the initial set for NH is fixed at 1500. The result of the best performing method is given in bold. R package, version 3.1.0 is used in all the experiments. It is clear from Table 3 that *OTE* outperforms all the other methods on most of the data sets. The new method is giving the smallest Brier scores on 10 out of 20 data sets. On 4 data sets random forest gave the smallest Brier scores. On 1 data set, node harvest gave the best result while SVM gave the best performance on 5 data sets. A large number of trees in the initial set can be recommended under the available computational resources. For $T > 1000$ the results of the proposed method are invariant and the method converges afterwards for the data sets considered. This can be seen in Figure 3 (a). As shown in Figure 3 (b), class membership probability estimations by using *OTE* is unaffected by varying the number of features selected at random for splitting the nodes of the trees. This means that growing trees for the initial set through random forest or simple bootstrap technique might lead to very similar final ensembles.

6 Conclusion

We have proposed an ensemble of optimal trees, *OTE*, as a non-parametric method for estimating class membership probabilities in binary class problems. We compared probability estimation trees, random forest, node harvest

Table 3. Data sets summary (FT means feature type with R: real, I: integer and N: nominal number of features) and Brier scores of *k*NN, tree, random forest, node harvest, SVM and *OTE*. The best result is shown in bold.

Data Set	*n*	*d*	FT (R/I/N)	kNN	Tree	RF	NH	SVM (Radial)	SVM (Linear)	SVM (Bessel)	SVM (Laplacian)	OptTreesEns
Mammographic	830	5	(0/5/0)	0.1412	0.1229	0.1288	**0.1207**	0.1340	0.1252	0.1313	0.1354	0.1366
Dystrophy	209	5	(2/3/0)	0.1051	0.1344	0.0947	0.1161	0.0831	0.0872	0.0802	**0.0792**	0.0864
Monk3	122	6	(0/6/0)	0.0886	0.0687	0.0657	0.1817	0.0695	0.1570	0.0663	0.0938	**0.0610**
Appendicitis	106	7	(6/0/0)	0.1263	0.1354	**0.1199**	0.1165	0.1360	0.1257	0.1156	0.1178	0.1242
SAHeart	462	9	(5/3/1)	0.2092	0.2074	0.1895	0.1880	0.1850	**0.1794**	0.1966	0.1816	0.2006
tic-tac-toe	958	9	(0/0/9)	0.2279	0.1467	**0.0408**	0.1997	0.1483	0.2188	0.1200	0.1972	0.0437
Heart	303	13	(1/12/0)	0.2226	0.1683	**0.1231**	0.1441	0.1442	0.1278	0.1235	0.1247	0.1286
House vote	232	16	(0/0/16)	0.0655	0.0323	0.0293	0.0656	0.0299	0.0345	0.1580	0.0386	**0.0290**
Bands	365	19	(13/6/0)	0.2231	0.2549	0.1878	0.2240	0.1991	0.2028	0.2230	0.2107	**0.1814**
Hepatitis	80	20	(2/18/0)	0.3105	0.1378	0.0970	0.0950	0.0964	0.1042	0.1158	0.0894	**0.0883**
Parkinson	195	22	(22/0/0)	0.1151	0.1138	0.0676	0.0930	0.0763	0.1195	0.1544	0.0931	**0.0636**
Body	507	23	(22/1/0)	0.0190	0.0734	0.0311	0.0553	0.0124	**0.0120**	0.2377	0.0219	0.0295
Thyroid	9172	27	(3/2/22)	0.0305	0.0104	0.0084	0.0161	0.0388	0.0321	0.0572	0.0382	**0.0079**
WDBC	569	29	(29/0/0)	0.0541	0.0643	0.0311	0.0425	0.0266	**0.0212**	0.2034	0.0283	0.0308
WPBC	198	32	(30/2/0)	0.1825	0.2131	0.1679	0.1686	0.1603	**0.1542**	0.1806	0.1626	0.1653
Oil-Spill	937	49	(40/9/0)	0.0395	0.0334	0.0282	0.0293	0.0326	0.0373	0.0331	0.0364	**0.0274**
Spam base	4601	57	(55/2/0)	0.1744	0.0948	0.0383	0.0906	0.0730	0.0618	0.2407	0.0814	**0.0374**
Glaucoma	196	62	(62/0/0)	0.1365	0.1095	**0.0890**	0.0916	0.0941	0.1239	0.2193	0.1193	0.0904
Nki 70	144	76	(71/5/0)	0.1458	0.1410	0.1465	0.1473	0.1675	0.2024	0.2349	0.1832	**0.1329**
Musk	476	166	(0/166/0)	0.1420	0.1884	0.0963	0.1746	0.0956	0.1107	0.2470	0.1886	**0.0871**



(*a*) (*b*)

Fig. 3. (a):The effect of the number of trees in the initial set on *OTE*. (b): The effect of the number of features selected at random for splitting the nodes of the trees on *OTE*.

and the proposed *OTE* on a number of bench mark and simulated data sets. The proposed method outperformed *k*NN, tree, random forest, node harvest and SVM on most of the data sets. We also used tree style simulation models to generate data sets with several structures. The proposed method is observed to use fewer accurate and diverse trees and hence could be very helpful in reducing the number of trees in tree ensembles which might increase interpretability. The method is observed to be unaffected by varying the number of features selected at random for splitting the nodes of the trees and they could simply be grown using the simple bagging technique. The method is implemented in an R package ***OTE***. The proposed method could better be

used, in conjunction with some feature selection method, (Mahmoud et al. (2014a, 2014b), for example) in high dimensional settings.

References

ALI, K. M., and PAZZANI, M. J. (1996:) Error Reduction through Learning Multiple Descriptions. *Machine Learning, 24, 173–202*.

BREIMAN, L. (2001): Random Forests. *Machine Learning, 45, 5–32*.

GNEITING, T., and RAFTERY, A. E. (2007): Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association, 102, 359–378*.

GUL, A., KHAN, Z., MAHMOUD, O., PERPEROGLOU, A., MIFTAHUDDIN, M., ADLER, W. and LAUSEN, B. (2015): Ensemble of k-Nearest Neighbour Classifiers for Class Membership Probability Estimation. *In the Proceedings of European Conference on Data Analysis, 2014.*

HOTHORN, T. and LAUSEN, B. (2003): Double-Bagging: Combining Classifiers by Bootstrap Aggregation. *Pattern Recognition, 36, 1303–1309.*

KRUPPA, J., LIU, Y., BIAU, G., KOHLER, M., KONIG, I. R., MALLEY, J. d. and ZIEGLER, A. (2014a): Probability Estimation with Machine Learning Methods for Dichotomous and Multicategory Outcome: Theory. *Biometrical Journal, 56, 534–563.*.

KRUPPA, J., LIU, Y., DIENER, H. C., WEIMAR, C., KONIG, I. R. and ZIEGLER, A. (2014b): Probability Estimation with Machine Learning Methods for Dichotomous and Multicategory Outcome: Applications. *Biometrical Journal, 56, 564–583.*

KRUPPA, J., ZIEGLER, A. and KONIG, I. R. (2012): Risk Estimation and Risk Prediction Using Machine-Learning Methods. *Human Genetics, 131, 1639–1654*.

LIAW, A., and WIENER, M. (2002): Classification and Regression by Randomforest. *R News, 2, 18–22*.

MACLIN, R., and OPITZ, D. (2011): Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Research, 11, 169–189*.

MAHMOUD, O., HARRISON, A., PERPEROGLOU, A., GUL, A., KHAN, Z., METODIEV, M. V., and LAUSEN, B. (2014a): A Feature Selection Method for Classification within Functional Genomics Experiments based on the Proportional Overlapping Score. *BMC Bioinformatics, 15, 274.*

MAHMOUD, O., HARRISON, A., PERPEROGLOU, A., GUL, A., KHAN, Z., and LAUSEN, B. (2014b): propOverlap: Feature (Gene) Selection based on the Proportional Overlapping Scores. *R package version 1.0,* `http://CRAN.R-project.org/package=propOverlap`.

MALLEY, J., KRUPPA, J., DASGUPTA, A., MALLEY, K. and ZIEGLER, A. (2012): Probability Machines: Consistent Probability Estimation using Nonparametric Learning Machines. *Methods of Information in Medicine, 51, 74–81*.

MEINSHAUSEN, N., (2010): Node Harvest. *The Annals of Applied Statistics, 4, 2049–2072*.

PLATT, J. C. (2000): Probabilistic Outputs for Support Vector Machines and Comparison to Regularized Likelihood Methods. *Advances in Large Margin Classifiers, Eds. Cambridge MA: MIT Press, 61–74*.

R CORE TEAM (2014): R: A Language and Environment for Statistical Computing, `http://www.R-project.org/`.