Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/

(RESEARCH ARTICLE)

Check for updates

Comparison estimating of classification error rate in decision tree: Data mining

Yousef M. T. El Gimati *

Statistics Department, Faculty of Science University of Benghazi, Libya.

Global Journal of Engineering and Technology Advances, 2021, 07(02), 067–082

Publication history: Received on 06 April 2021; revised on 09 May 2021; accepted on 12 May 2021

Article DOI: https://doi.org/10.30574/gjeta.2021.7.2.0068

Abstract

Decision Tree (DT) typically splitting criteria using one variable at a time. In this way, the final decision partition has boundaries that are parallel to axes. An observation is misclassified when it falls in a region which does not have the same class membership. Misclassification rate in classification tree is defined as the proportion of observations classified to the wrong class while in the regression tree is defined as a mean squared error. In this paper, we present two of the important methods for estimating the misclassification (error) rate in decision trees, as we know that all classification procedures, including decision trees, can produce errors.

Constructed DT model by using a training dataset and tested it based on an independent test dataset. There are several procedures for estimating the error rate of decision tree-structured classifiers, as K-fold cross-validation and bootstrap estimates. This comparison aimed to characterize the performance of the two methods in terms of test error rates based on real datasets. The results indicate that 10-fold cross-validation and bootstrap yield a tree fairly close to the best available measured by tree size.

Keywords: Cross-validation; Bootstrap; Misclassification; Training error; Test error; Tree size

1. Introduction

Decision Trees DTs are one of the data mining methods, widely studied and applied to data mining tasks. However, data mining contains many methods, and more popular methods of data mining are DTs, artificial neural networks nearest neighbor method, and genetic algorithms. In this paper, we consider the classification of learning data which the resulting classifier is DT. The main objective of this research is to compare cross-validation and bootstrap techniques as two important criteria of choice and assessment of statistical prediction rules and give an objective assessment of their strengths and weakness in real-life data, especially misclassification rate for DTs. This article concerns estimating the error rate of a DT that has been constructed from a training set of data. The training set consists of a distribution for a pair of random variables $\ell = \{(x_i, y_i), i = 1,2, \dots, n\}$, where x_i is a vector of predictors and is either class label or numerical response. For example, x_i might describe a medical patient's age, weight, sex, previous disease history, and so on, and y_i might indicate whether the *patient* survived a certain operation (Mclachlan, 1992) [1]. On the basis of the training set, a DT $c_l(x)$ is constructed. The intention is to use $c_l(x_0)$ to predict a future unobserved response y_0 on the basis of its predictor vector x_0, where x_0 is new observation (Eforn, 1979) [2] and (Hastie, et al, 2017) [3].

1.1. How Does Decision Tree Construction?

This section gives an overview of the procedure of DT construction. In brief, the construction of a DT, classification rule centers on the definition of four major elements. These are Splitting Rule, Assignment of each terminal node to a class, Choosing Right-Sized Tree, which nodes are terminal nodes (pruning rule), and specifying the criteria for predictive

* Corresponding author: Yousef M. T. El Gimati
Statistics Department, Faculty of Science University of Benghazi, Libya.

accuracy. The steps in the tree building process involve growing a large tree (a tree with a large number of nodes), combining some of the branches of this large tree to generate a series of sub-tree s of different sizes (varying numbers of nodes), and selecting an optimal tree via the application of "measures of accuracy of the tree".

1.2. Splitting Rule

The process starts with a training set consisting, which has a known class or label ("male" or "female," for example). The goal is to build a tree that distinguishes among the classes. For simplicity, assume that there are only two target classes and that each split is binary partitioning. To choose the best splitter at a node, the algorithm considers each independent variable in turn. In essence, each variable is sorted. Then, every possible split is tried and considered, the best split is the one that produces the largest decrease in diversity (minimum deviance) of the classification label within each partition (this is just another way of saying "the increase of homogeneity"). This is repeated for all independent variables, and the winner is chosen as the best splitter for that node. The process is continued at the next node and, in this manner, a full tree is generated. In general terms, the splits at each node will be found that generate the greatest improvement in predictive accuracy (minimal error rate).

1.3. Assignment Rule

Basically, the terminal nodes are assigned to the classes that have the highest probabilities. These probabilities are usually estimated via the respective sample relative frequency $\frac{n_j}{n}$, where n_j is the number of observations in class *j*. This is a simple majority membership, i.e. assign to the terminal node (leaf node) the label of the class that has most samples at that terminal node. Note that a leaf node is said to be pure if all the training observations are belonging to the same class (Breiman et al, 1984) [4].

1.4. Pruning Procedure

Pruning the method most widely used for obtaining right-sized trees, was proposed by (Breiman et al, 1984) [4]. They suggested the following procedure: build the complete tree (a tree in which splitting no leaf node further will improve the accuracy of the training data) and then remove sub-tree that is not contributing significantly towards generalization accuracy. Then a sequence of smaller trees can be created by pruning the initial large tree, wherein the pruning process, splits that were made are removed and a tree having a fewer number of nodes is produced. The accuracies of the members of this sequence of a sub-tree (really a finite sequence of the nested sub-tree) since the first tree produced by pruning, is a sub-tree of the original tree, and a second pruning step creates a sub-tree of the first sub-tree, so on, are then compared using good estimates of their misclassification rates (either based on cross-validation or obtained by bootstrap). A specific way to create a useful sequence of different sized trees is to use minimum cost-complexity pruning. In this process, a nested sequence of a sub-tree of the initial large tree is created by weakest-link cutting. With weakest-link cutting (pruning), all of the nodes that arise from a specific nonterminal node are pruned off (leaving that specific node as a terminal node). Now, letting be the re-substitution estimate of the misclassification rate of a tree, T, and $|T|$ be the number of terminal nodes of the tree, for each $\alpha \geq 0$ the *cost-complexity* measure, $R_\alpha(T)$, for a tree, T , is given by $R_\alpha(T) = R(T) + \alpha|T|$.

1.5. Heuristics of Bias-Variance in DTs

A decision tree imposes a partitioning of the attribute space that can be represented as a collection of regions. When the attribute space is split into a small number of partitions, as a result of a small number of terminal nodes, the fit is poor. We refer to this lack of fitting of the attribute space as bias. When the attribute space is split into many small partitions, as a result of a large number of terminal nodes, the bias is small. In other words, these small partitions are more likely to have a majority of the wrong class. This latter type of error is referred to as variance. Thus, the tradeoff between bias and variance is an important characteristic of decision trees.

The concept of the test error, including the training error, is illustrated in Figure 1 (Hast, et al, 2017) [3]. The graph of the training error tends to decrease whenever the size of the tree is increased, typically converging to zero. However, a model with zero training error has been over-fitted to the training data and in that case, the prediction model will have a large variance. Typically, the estimated test error starts high with a small number of terminal nodes, reaches a shallow minimum region, and eventually increases with the tree size. Note that the test error rate often follows a U- shaped pattern.

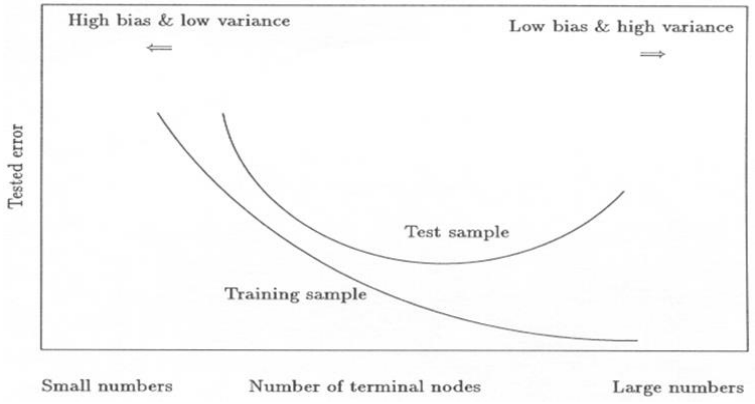

Figure 1 Typical plot of test and training error as a function of terminal nodes. Few numbers of terminal nodes produce high bias & low variance whereas large numbers of terminal nodes produce low bias & high variance.

1.6. Stable and Unstable algorithms

Breiman (1996b) [5] argues that many prediction algorithms are unstable, in that small changes in the training set may lead to large changes in the resulting partitions. In other words, there is variance due to the instability of the training sets themselves, leading to an increase in variance. In several cases, unstable algorithms such as DTs are characterized by high variance, while algorithms like linear discriminant analysis are characterized by low variance. As the training set changes, the algorithms can differ significantly from each other, but stable algorithms do not change much and will tend to be the same, and the variance will be small with possibly large bias. In general, the instability of an algorithm depends on many factors, for example, the distribution of data used to construct the classifier and the sensitivity of the classifier to the size and the composition of the training dataset.

2. Estimating the Misclassification (Error) Rate

The misclassification (error) rate of a decision tree is the probability of incorrectly classifying a randomly selected instance, for a randomly selected instance, where the probability distribution over the instance space is the same as the distribution that was used to select instances for the training set.

There are several methods available to estimated test error or (generalization error). The most common statistical methods are re-substitution, cross-validation, and bootstrap. A method that is more suitable for intermediate sample sizes is cross-validation. A closely related method, which is used for small sample sizes, is the bootstrap procedures, which had been considerable work in the literature on cross-validation and bootstrap for error rate estimation. See, for example, Stone (1974) [6] and Efron (1979) [2]. A good general discussion can be found in Efron (1983) [7], Efron & Tibshirani (1993) [8], and Hastie et al, (2017) [3].

It is important to introduce the loss function in this context for measuring errors between the predicted value c and the actual response y , which is denoted by $L[y, c]$. The choice of L plays an important role in defining bias, variance, and the prediction error for the model $c_l(x)$ that represents the predicted value at x with respect to all possible values in ℓ. Here, we are particularly interested in the categorical response, where both y and c are either 0 or 1, i.e. two classes. Typical choices are misclassification error as

$$L[y,c] = \begin{cases} 0 & if\ c = y \\ 1 & if\ c \neq y \end{cases}$$

Also, where both y and c are continuous *i.e.* y and $c \in \Re$. Typical choices are Mean Square Error is

$$L = [y,c] = E[y-c]^2, \quad \text{Here } L = [y,c] \in \Re$$

Assume that the observations $\ell_i = (x_i, y_i),\ i = 1,2,\dots,n$ in the training set are a random sample from some distribution F ,

$$\ell(\ell_1, \ell_2, \ldots, \ell_n) \overset{iid}{\sim} F \quad (1)$$

and that $\ell_0 = (x_{0,} y_0)$ is independent draw from F (called test sample).The test error rate (Err) of the decision tree is its probability of incorrectly classifying a randomly selected future case $\ell_0 = (x_{0,} y_0)$ Hence, we can be define the test error rate of the model c_l is

$$Err = Err\left(\ell, F\right) = E\ L[y_0, c_l(x_0)] \quad (2)$$

This is the expected prediction error over an independent test sample, also referred to as generalization error or test error. The notation E_{0F} indicates that only $\ell_0 = (x_{0,} y_0)$ is random in Equation (2). In this section, we discuss the issues of error rate estimation through three popular methods that are available for estimating errors of misclassification: re-substitution, cross-validation, and bootstrap methods. Among the three approaches, we have compared the latter two approaches on artificial examples and real datasets.

2.1. Re-substitution Method

The re-substitution error rate or training error (or apparent error) has been proposed and used in the past. This error is the average loss over the training sample. One approach is to use the entire training data to select our classifier and estimate the error rate, this naïve approach has two fundamental problems the final model will normally over fit the training data and the error rate estimate will be overly optimistic (lower than the true error rate), and it is defined as

$$\overline{err} = Err(\ell, \hat{F}) = \frac{1}{n}\sum_{i=1}^{n} L\left[y_i, c_i(x_i)\right]$$

The notation $\hat{F}$ indicates the empirical distribution that puts probability $\frac{1}{n}$ on each observation $\ell_i = \{(x_i, y_i)\}$, *i = 1, 2…n,* $\overline{err}$ tends to be biased downward as an estimate of Err , because the error is estimated using the same training sample that is used to construct the model. One way of avoiding over-fitting in the apparent error rate as a consequence of the classifier being tested on the same data from which it has been trained or constructed is to use a holdout method. The available information is divided into disjoint training and test subsets. The classifier is constructed from the training subset and then tested on the test subset. Obviously, this method requires large samples. However, techniques of estimation, such as cross-validation and bootstrap methods, that eliminate the need for a separate test set.

2.2. Cross-validation Method

Cross-Validation (CV) is the traditional method and most commonly used for estimating prediction error, provides a nearly unbiased estimating of the future error rate. However, the low bias of cross-validation is often paid for by high variability. Cross-validation (Stone, 1974) [6] avoids the over-fitting problem by removing the data point to be predicted from the training set.

The leave-one-out cross-validation estimate of Err is defined as:

$$\hat{Err}^{CV(1)} = \frac{1}{n}\sum_{i=1}^{n} L\left[y_i, c_{l_{(i)}}(x_i)\right]$$

Where $\ell_{(i)} = \{(x_1, y_1) \ldots (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}) \ldots (x_n, y_n)\}$ is the training set (except point i) used to build the model $c_{l_{(i)}}(x_i)$ and results are tested on the point i which was left out.

$\hat{Err}^{CV(1)}$ is leave-one-out cross-validation estimating the error rate of prediction rule (decision trees), that approximately unbiased for the true prediction error rate, but can have high variance? "Leave-one-out" is a more elaborate and expensive version of cross-validation that involves leaving out all possible subsets of n cases.

Also, leave-one-out cross-validation has high variance if the prediction rule is unstable: the reason is the leave-one-out training sets are too similar to the full dataset. In general, in *K*-fold cross-validation, you divide the data into *K* subsets of (approximately) equal size. You train the data *K* times, each time leaving out one of the subsets from training, but using only the omitted subset to compute whatever error criterion interests you. If *n* = *KH*, where *K* = total number of subsets and *H* = total number of observations per subset. This is called "leave-one-out" cross-validation version $\hat{Err}^{CV(k)}$ partitions the training set into *K* parts, hence (*K-1*) of these subsets are used to train the model and remains to test the model, *K*-fold version $\hat{Err}^{CV(k)}$ is defined as

$$\hat{Err}^{CV(k)} = \frac{1}{n}\sum_{k=1}^{K}\sum_{v=1}^{V} L[y_{(k-1)V+v}, c_{l_{(v)}}(x_{(k-1)V+v})]$$

Where $\ell_{(v)} = \{(x_1, y_1) \ldots (x_{(k-1)V}, y_{(k-1)V}), (x_{(k)V+1}, y_{(k)V+1}) \ldots (x_n, y_n)\}$ is the training set used to build the model $c_{l_{(v)}}(x_{(k-1)V+v})$ and *k* indexes the subset left out which used to testing the results.

2.3. Bootstrap Method

Bootstrap is a good tool for estimating the true prediction Err. The bootstrap procedure was introduced by Efron (1979) [2] and is fully described in Efron and Tibshirani (1993) [8]. The bootstrap, like cross-validation, is a resampling technique to create different training sets for each model by resampling the original training set. The bootstrap has other important advantages besides providing more accurate point estimates for prediction error. The bootstrap replications also provide a direct assessment of variability for estimated parameters in the prediction rule. For example, see Efron and Tibshirani (1993) [8].

One approach, uses the original training set $\ell_i = (x_i, y_i),\ i = 1,2, \ldots, n$ consisting of *n* observations $\ell^*_i = (x_i^*, y_i^*)$ as the training set. The basic idea is to draw datasets randomly with replacement from the training set, each sample has the same size as the original training set, where *m=n*. Let $\hat{F}$ be the empirical distribution putting mass $\frac{1}{n}$ on each observed case, such that

$$\ell^*(\ell^*_1, \ell^*_2, \ldots, \ell^*_m) \overset{iid}{\sim} \hat{F} \qquad (3)$$

Suppose ℓ^* be a random sample of size *m* taken *iid* with replacement from $\hat{F}$, where $\ell^*_i = (x_i^*, y_i^*)$ is a single random observation. This procedure fits the model in question on a set of bootstrap samples ℓ^*_i (*b* = 1,..., *B*). If $c_{l^{*b}}$ is the predicted value at x_i from the model fitted to the *b*th bootstrap sample, our estimate is defined as

$$\hat{Err}_1^{boot} = \frac{1}{B}\frac{1}{n}\sum_{b=1}^{B}\sum_{i=1}^{n} L[y_i, c_{l^{*b}}(x_i)]$$

However, this procedure does not provide a good estimate in general. The reason is that both the bootstrap samples and training set have observations in common. This overlap can make the prediction unrealistically good. Here, in this work, another approach is pursued by using an independent test set, $l_0 = \{(x_{0i}, y_{0i})$, *i =1, ...,* $n_0\}$ consists of n_0 observations $l_{0i} = (x_{i0}, y_{i0})$ from the training set, so that our estimate is defined as

$$\hat{Err}_2^{boot} = \frac{1}{B}\frac{1}{n_0}\sum_{b=1}^{B}\sum_{i=1}^{n} L[y_{0i}, c_{l^{*b}}(x_{0i})]$$

This can be done for the artificial example as the underlying concept is known. In real-world datasets (underlying concept is not known), we generated ℓ^* of size *m*, as in Equation (3), from the original training set ℓ of size n as in

Equation (1), where $m < n$. A tree is grown using l^{*b} , (b = 1,..., B) and tested $(l \setminus l^{*b})$={ (x_i', y_i') , i =1, ..., $n'(b)$ that an observation in l but not in l^{*b} , see for example (El Gimati, 2020) [9]., our estimate is

$$\hat{Err}_3^{boot} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{n'(b)}\sum_{i=1}^{n'(b)} L[y_i', c_{l^{*b}}(x_i')]$$

However, bootstrap samples are generated by resampling $\hat{F}$ with replacement, while cross-validation resamples $\hat{F}$ without replacement. Also, a bootstrap sample is created by a sampling of size m uniformly from the original training set of size n which gives $\binom{n}{m}$ possible samples, whereas the CV is generated by a random number that depends on the division into folds.

3. An artificial Example

In this section, both methods of error estimation outlined above are applied to an artificial dataset with random noise. The simulated data are derived from three-dimensional concentric spheres, three input features are defined by

$$x_1 = r\cos(\theta),\ x_2 = r\sin(\theta)\cos(\phi) \text{ and } x_3 = r\sin(\theta)\sin(\phi)$$

where $\theta \sim u(0,\pi), \phi \sim u(0,2\pi)$ and $r \sim u(0,4)$

In this case, there are four layers, each layer being defined as the subset of observations labeled by 0 or 1, as class 1 $r \in \{(0,1] \cup (2,3]\}$ and class 2 if, $r \in \{(1,2] \cup (3,4]\}$, respectively, as the response variables. So that there are approximately equal observations in each layer.

The dataset of size 600 observations is used with random noise, which can be obtained by exchanging one class label with another class label, with a probability of 0.08. An independent test set is drawn of 5000 observations by using the same procedure as the training set with the same amount of noise added.

To see how well the estimation methods perform, we sampled observations from the dataset (uniformly with and without replacement) and created a training set with desired fold for cross-validation and several bootstrap sample sizes. We then grew the tree using the R tree algorithm rpart. The tree classifier for CV is then trained on all subsets except the subset for the test, whereas the bootstrap uses an independent test set for estimating the test error rate. The bootstrap procedure uses 75 bootstrap samples of size m.

3.1. Results from Artificial Dataset

In our example, V-fold (V=3, 5, 7 and 10) CV is based on the assumption that the training set is randomly divided into V equal parts, of which (V-1) are used to grow the tree and one part is used to test the validity of the model. For example, a 10-fold cross-validation simple means that 10 trees have to be grown. On the other hand, bootstrap sample sizes (m=100, 200, 300, and 400) are used to grow the tree and test it on a test set; for each bootstrap sample size, B=75 trees have to be grown. Note that Bayes error rate for this example is 8%.

Figure (2) shows different folds cross-validation. The four plots (top) show the test error rate as a function of the cost-complexity criterion, while the four plots (bottom) are a function of the tree size which refers to the number of terminal nodes. This figure shows that the test error rate starts high for small cost-complexity or tree size, decreases as the cost-complexity or tree size increases, reaches a shallow minimum region, and then eventually increases slowly with the cost-complexity or the tree size. In both cases, the behavior of the error rate curves is approximately similar, which is consistent with Figure (1).

This scenario gives less accuracy for both V-fold=3 and 5 with low variability, whilst the performance of V -fold=7 and 10 is much more accurate than V -fold=3 and 5, but has high variability. These results indicate that V -fold=7 and 10 give approximately accurate estimates of the test error with acceptable high variability, which means that it is satisfactory for choosing the correct model between 4 and 7 terminal node trees, whereas cost-complexity is between about 0.04 and 0.06 for choosing the correct tree.

Figure (3) illustrates the results of the bootstrap method for different sample sizes. This figure shows the test error rate as a function of the cost-complexity of the four plots (top) and the tree size of the four plots (bottom). The behavior observed is similar to that under cross-validation. We observe low variability with reasonable bootstrap sample size, which seems to offer considerably improved estimation in sample sizes 300 and 400 observations. From the plots, it is acceptable for choosing the correct model between 4 and 6 terminal nodes, while cost-complexity is between about 0.02 and 0.04, which is satisfactory for choosing the correct tree. To focus on ideas, it will be useful to compare these two methods using different datasets.

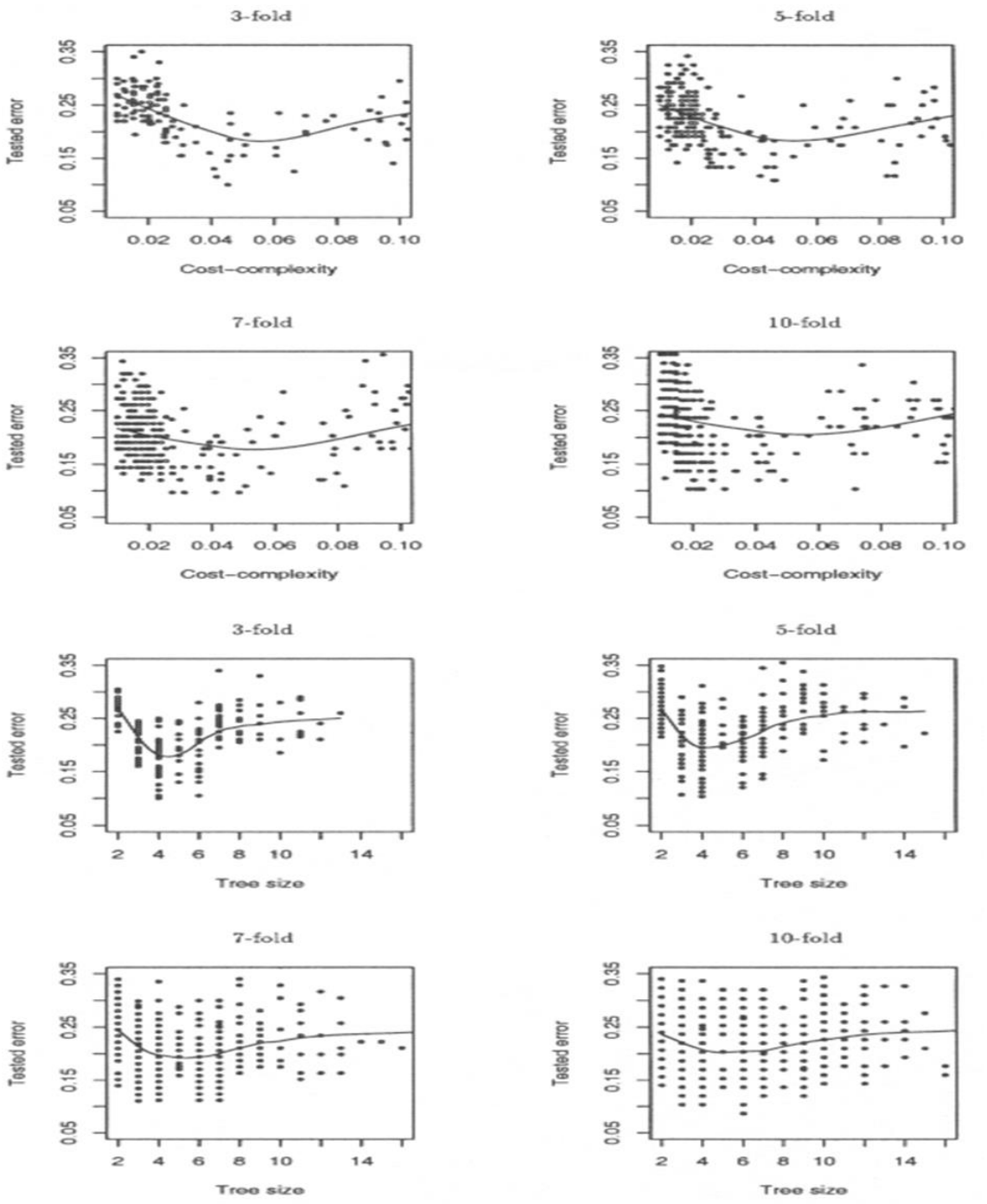

Figure 2 The top four plots present the test error rates as a function of cost-complexity, while the bottom four plots are a function of tree size, by using the CV method for simulated data (10 datasets). Each point corresponds to a V-fold CV. The line obtained from scattering smooth.

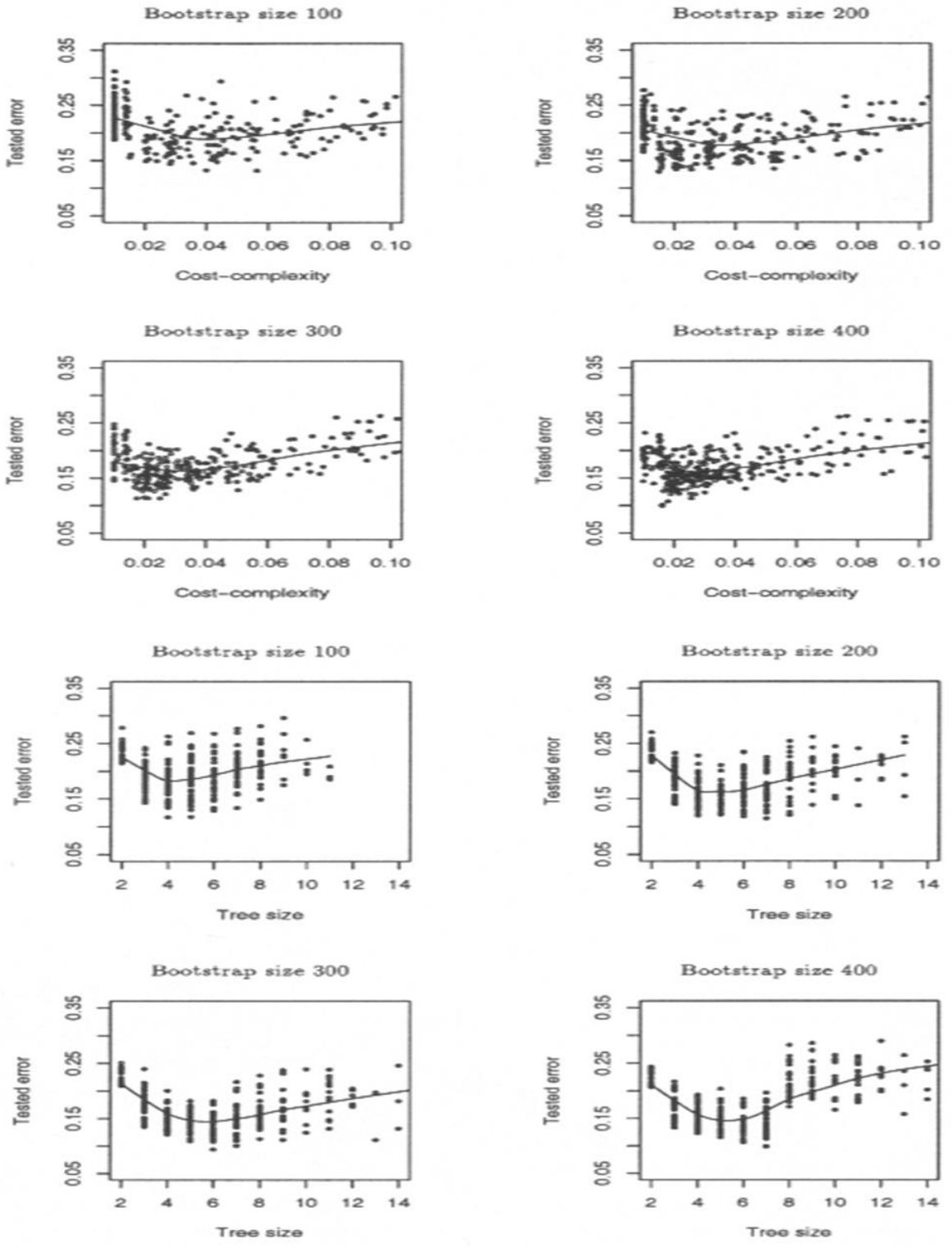

Figure 3 The top four plots present the test error rates as a function of cost-complexity, while the bottom four plots are a function of tree size, by using the bootstrap method for simulated data (10 datasets). Each point corresponds to the bootstrap sample. The line obtained from scattering smooth.

3.2. Comparing Cross-Validation and Bootstrap

Efron (1979) [2] shows an example of the bootstrap method outperforming the cross-validation method, but this is with simple linear discriminant analysis. Breiman (1984) [4] conducted experiments using cross-validation for decision tree

pruning. They chose 10-fold cross-validation for the CART program and claimed it was satisfactory for choosing the correct tree. In this context, the question naturally arises as to which of these two methods, cross-validation or bootstrap, is better in estimating the classification error in decision trees which leads to help us in choosing between the classifiers.

To illustrate this comparison, suppose we have ten datasets of size 1000 generated from the same model which is described in Section (2), with the same amount of noise for each dataset, but with different random seed to generate different datasets. An independent test set is drawn of 5000 observations by using the same procedure as for training set with the same amount of noise added. These experiments are performed based on bias versus the other methods to investigate which method is willing to tradeoff bias in order to reduce variance to use for a model selection.

In this example, we estimate the bias by using *V*-fold (*V*=3 and 10) CV and bootstrap methods. We calculate the true error rate from the test dataset of size 5000. The *V*-fold cross-validation estimate of which (*V-1*) is used to train the model, and results are tested on the subset which was left out of the training set.

For the bootstrap method, we generate *B*=50 different training sets of size *m* (=500). A model (classifier) is then learned on each training set as well as tested on the test set. We use the predictions made by these models to estimate the true error rates. Then we calculate bias, which is the difference between the estimated error rate and the true error rate. Figure (4) shows the bias as a function of the methods. From the boxplots of default pruned trees, it is clear that 3-fold has a large bias as well as high variability, whereas 10-fold has less bias than 3-fold, with reasonable variance. The bootstrap has nearly identical bias with 10-fold and acceptable variance. The tree stump boxplots show 10-fold and bootstrap are nearly identical for bias, but with a high variance of 10-fold, while 3-fold has larger bias than 10-fold and bootstrap, with acceptable variability. Overall, bias has a statistically significant decrease between 3-fold and bootstrap, giving a significant probability of 0.039 and 0.048 at the 5% significance level in both default pruned and tree stumps, respectively. We have concluded that the bootstrap method has significantly better results than those with cross-validation. However, 10-fold cross-validation is comparable to the bootstrap method, especially for default pruned trees. Note that significant test results are obtained by ANOVA.

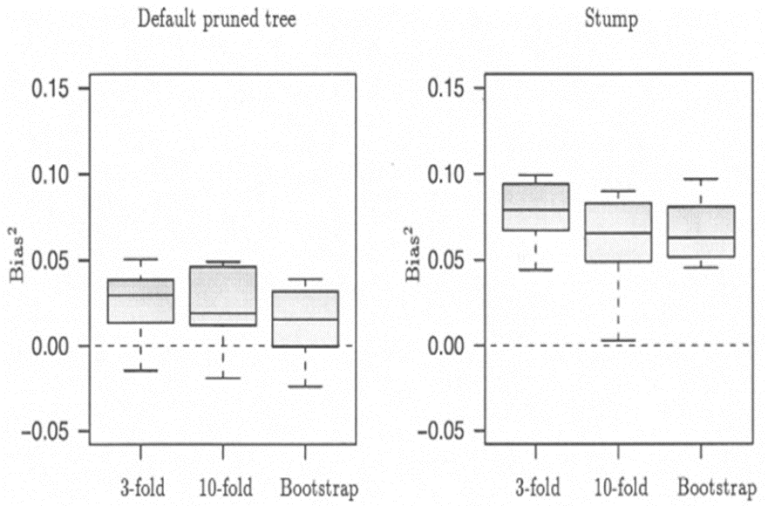

Figure 4 Boxplots of bias-squared over ten datasets with random noise on average for V-fold=3,10 and bootstrap sample of size 500 observations. The left panel presents default pruning, whereas the right panel presents two terminal nodes (stump).

4. Real data: Anæmic Libyan infants

According to population estimates, over one billion people in the world have anæmia. The term anæmia, is generally used in clinical medicine, refers to a below-normal concentration of hæmoglobin in the blood. Anæmia may be diagnosed with confidence when the hæmoglobin concentration is abnormally low in per unit volume of circulating blood as a result of a reduction of oxygen-carrying capacity of the blood. Hæmoglobin is a complex protein of iron-containing *hæme* and *globin* groups. Dynamic interaction between them gives hæmoglobin its unique properties in the reversible transfer of oxygen.

Hæmatologic examination of the blood is a routine procedure to determine the presence of anæmia in patients with major illnesses. The normal range at one year of age is between 11 and 12 g/l. An infant whose value is below 11 g/l can be considered to be suffering from anæmia. It is clear that anæmia is a deficiency of hæmoglobin concentration below normal.

Anæmia is still a good health indicator, especially in a developing country like Libya. In developed countries, there has been a sharp decline due to better living conditions and medical facilities, but the incidence is still high among neonatal infants in the northeastern part of Libya, as observed by El Ojali (1994) [10]. Thus, more research is needed about anæmia in infancy and the risk factors in order to lead to more effective control of the disease in the study area.

The main objective of this analysis is to discriminate between neonatal infants who acquire anæmia and those who do not, and hence to identify the factors associated with such discrimination by using decision trees. Here, we would also like to construct our model so that it gives a good bias-variance compromise with minimum test error which gives an optimal model complexity by using cross-validation and bootstrap techniques. We also compare the performance of the two methods for a comparison estimating of classification error rate in decision trees.

4.1. Source and Limitation of Dataset

This study is based on the data supplied by El Ojali (1994) [10] of the Department of Laboratory Medicine, University of Benghazi. The survey includes 510 anæmic Libyan infants in their first year of life and 135 normal healthy infants of the same age group as a control. An infant is considered anæmic when his or her hæmoglobin level is below 11 g/l. Out of 510 anæmic infants, 366 (72%) were from the children's hospital in Benghazi, 50 (9.8%) from Tobruk, 51 (10%) from Darana, and 43 (8.4%) from Beyda. The control group includes 135 healthy infants whose hæmoglobin level is within the normal range. These infants attend the Medical Child Health Care Centre for each city, and out of 135 healthy infants, 60 (44%) were from Benghazi, 25 (19%) from Tobruk, 30 (22%) from Darana, and 20 (15%) from Beyda. The supplied data include some information regarding biomedical tests of samples and symptoms of the disease, but we did not include this information in the analysis since the variables are considered irrelevant to the objective of our work. Thus the total number of the variables that are found appropriate to include in the analysis is 15. These variables, measured for each infant and their different categories are briefly discussed below.

4.1.1. Neonatal infants variable

The neonatal infants under study have been grouped into two classes: (1) Anæmic class (disease class) which consists of 510 anæmic infants, and (2) Healthy class (control class) which consists of 135 normal healthy infants.

4.1.2. Demographic variables

The five demographic variables consist of age (in months), sex, family size, birth weight, and present weight. The variable sex is categorical and has been included in the analysis by coding as Male and Female. The remaining variables are all quantitative and are included directly in the analysis.

4.1.3. Socio-economic variables

This group consists of four variables which describe the education of the mother, education of the father, place of residence, and family income. Each of the variables education of mother and education of father has been divided into five categories and assigned a code for each, such as illiterate (I), primary (P), middle (M), secondary (S), and university (U). These categories have been included in the analysis. The place of residence is a categorical variable and is included in the analysis as Benghazi (Ben), Tobruk (Tab), Darana (Dar), and Beyda (Bey). The variable income is quantitative and was included in the analysis directly.

4.1.4. Nutrition variables

This group consists of four variables: duration of breastfeeding, duration of bottle feeding, duration of mixed feeding (breast and bottle), and duration of solid feeding. All these variables are quantitative (in months) and are included directly in the analysis. Note that the sum of these variables is equal to the present age of the infant.

4.1.5. Antenatal variable

A single variable which indicates that the mother was anæmic during pregnancy and has been included in the analysis by coding Y for being anæmic and N otherwise. The objective of including this variable in the analysis is to find out whether anæmic mother during pregnancy has any effect on anæmia among neonatal infants.

4.2. Classification Tree for Anæmic Infants

In this section, we are interested in constructing decision trees automatically using the R tree algorithm rpart from the above data. The dataset consists of 510 anæmic infants (patient) and 135 normal healthy infants (control) with fourteen predictor variables.

A decision tree that could have been constructed is given in Figure(5), with a re-substitution error rate of about 10%. Each node of the decision tree consists of either a test that partitions the data or a decision. Once a tree is constructed from data, it can be used to classify observations of an unknown category (patient or control). In the figure, each decision node is labeled control (C) or patient (P), with the number of observations assigned to each class.

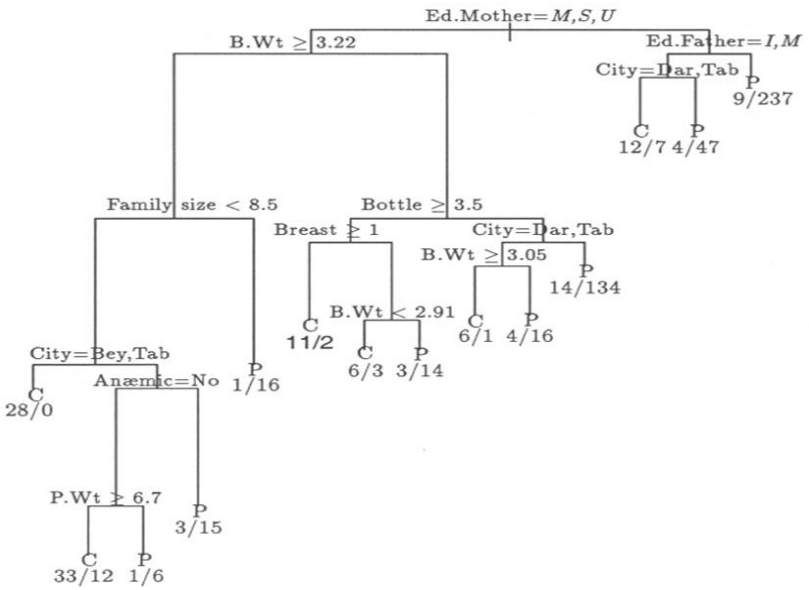

Figure 5 A decision tree using rpart tree with two classes' patient and control of Libyan infants in their first year of life. Nodes at the end of the tree (terminal node) are decision nodes and under each decision, the node gives the numbers of cases for control (C) and patient (P) respectively.

4.3. Pruned Tree for Anæmic Infants

As a rule, a pruned tree may produce a higher misclassification error on the training data than a large tree. A large tree may have a low error rate, but it could introduce *overfitting* to the data. Many researchers have found that judicious pruning results in both smaller and more accurate tree classifiers. Figure (6) (top) shows the same tree after pruning has taken place, with a re-substitution error rate of about 15.9%, whereas the default error rate is about 21%. Note that no further pruning for the sub-tree (birth weight) is possible, as beyond that level all observations go to one class which is not a legitimate tree.

The first split of the pruned tree is based on the mother's education. This variable is most important, in the sense of discriminating anæmic neonatal infants between two classes having disease and control. The two branches generated at the root relate to mother's education with levels of education, middle (M), secondary (S), and university (U) lead to birth weight, whereas education levels illiterate (I) and primary (P) leads to a node at the right, most of the observations are classified to class having disease, which means that mother's education levels, illiterate (I) and primary (P), are more likely to have anæmic infant.

The next split relates to birth weight -- infants with birth weight less than 3.22 kg are likely to be anæmic. Infants with a birth weight more than or equal to 3.22 kg are further split according to family size. A family with more than (8.5) 9 members is more likely to have the disease than a smaller family.

Figure (6) (bottom) shows a partition based on the sub-tree with root node given by node two of a large tree, i.e. children of mothers who have levels of education, middle (M), secondary (S), and university (U). The partition between birth weight and family size gives three partitions corresponding to three terminal nodes in the sub-tree, and the labels in

each partition refer to control (C) and patient (P). Note that displaying the partition is only possible for one or two continuous predictors on a 2D plot. If the tree contains one predictor, the predicted value of the first class is plotted against the predictor over its range in the dataset. When the tree contains two predictors a plot is made of the space covered by those two predictors and the partition made by the tree is superimposed.

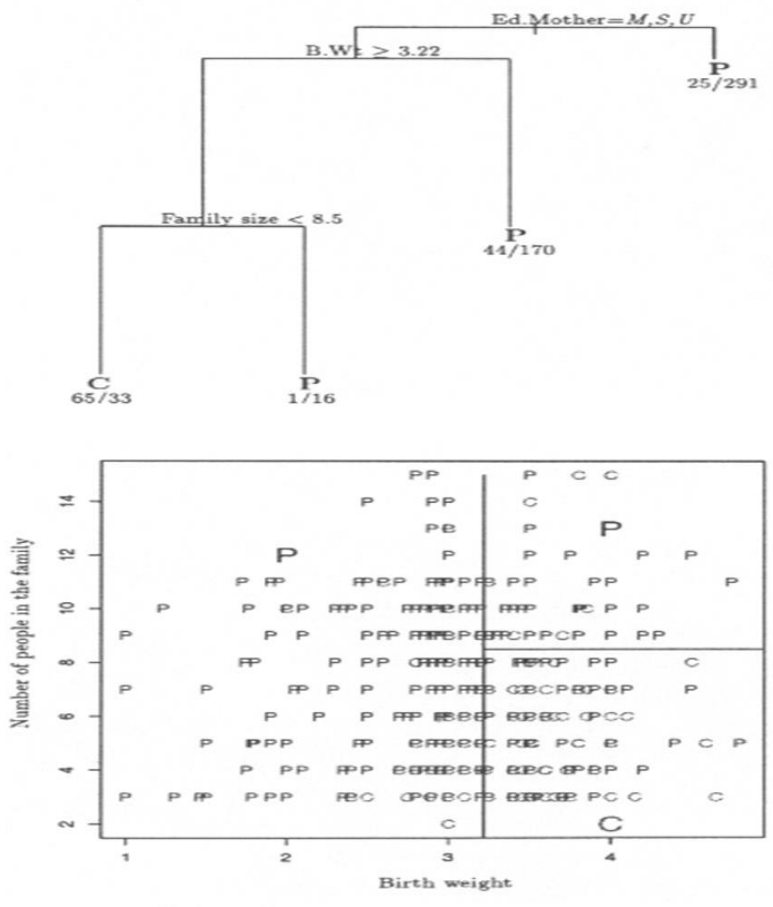

Figure 6 A decision tree (top) after pruning of Libyan infants. The double under each terminal node gives the number of cases of control (C) and patient (P). the partition (bottom) of birth weight versus family size gives three partitions corresponding to three-leaf nodes in the sub-tree and labels in each partition refer to control (C) and patient (P).

4.4. Results from Anæmic Infants for Comparison Estimating Error Rate

In this section, we show the results of running the two methods outlined in Section (2) to the anæmic Libyan infants' dataset. For analysis of data from patients and healthy, we sampled observations from the training set with and without replacement and created a training set of the desired sizes, as well as of the desired folds. We then grew the tree on part of the training set and tested on the remaining part of the training set for the bootstrap procedure, as explained in Section (2.3). This procedure was repeated for 75 bootstrap samples. Also, we applied the V-fold cross-validation technique on the basis of 3,5,7, and 10, as described in Section (2.2).

Figures (7) and (8) summarize the results of this dataset. The task is the same as before. From these figures, we can observe that the test error rates follow the same pattern as shown in Section (1). The result of interest here is that when the complexity of trees is measured by cost-complexity, the cross-validation has lower variability than when the complexity of trees is measured by tree size. These results indicate that the performance of V-fold=7 and 10 is better than V-fold=3 and 5. Here, it looks as if a tree can be selected between 5 and 8 terminal nodes, whereas cost-complexity is between about 0.02 and 0.04 for the tree selection.

As can be observed from Figure (8), the bootstrap method is a better scheme when compared to cross-validation. This comes as no surprise since we observed similar results in our experiment with random noisy data in Section (3). Moreover, from the same figure, we can note that the performance of bootstrap samples (300 and 400) is better than the other sizes (100 and 200). In this case, the tree can be selected between 4 and 6 terminal node trees, while cost-

complexity is between about 0.02 and 0.05 for the correct tree. Our results demonstrate that cross-validation and bootstrap yield a tree fairly close to the best available measured by cost-complexity, whereas the bootstrap method seems to be uniformly better than cross-validation measured by tree size. Note that all the plots for both methods are consistent with Figure (1) and follow a U-shaped pattern.

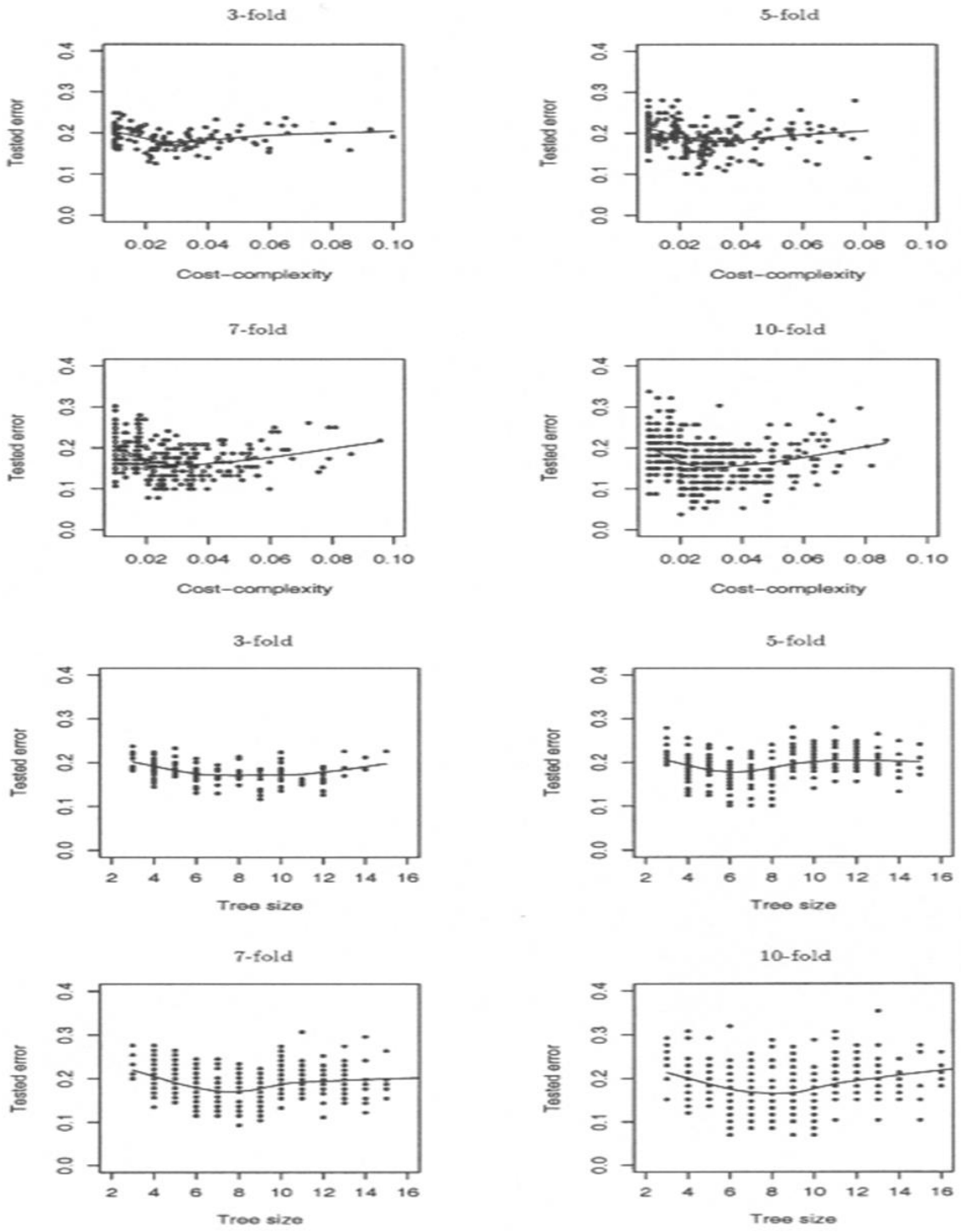

Figure 7 The top four plots present the test error rates as a function of cost-complexity, while the bottom four plots are a function of tree size, by using the cross-validation method of a real dataset. Each point corresponds to a V-fold CV. Line obtained from scattering smooth.

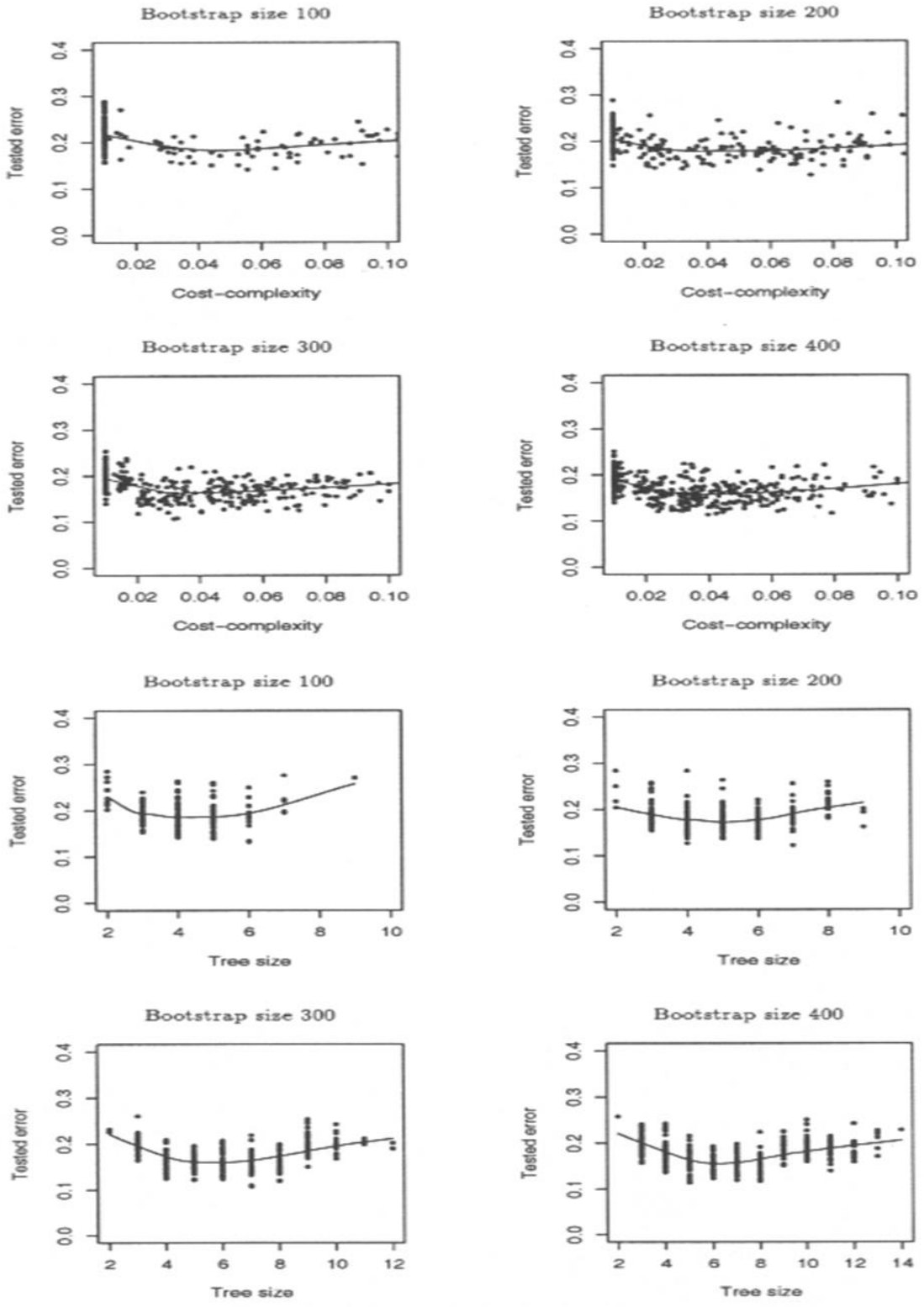

Figure 8 The top four plots present the test error rates as a function of cost-complexity, while the bottom four plots are a function of tree size, by using the bootstrap method of a real dataset. Each point corresponds to the bootstrap sample. The line obtained from scattering smooth.

5. Conclusion

It is always very educational and exciting to apply classification techniques like decision trees in new problems. This is because each significant real-world classification problem has its own properties, requirements, and challenges. A study of classification trees using resampling methods in real-world datasets may help to identify what kinds of structure information are most useful for specific problems. We reviewed resampling techniques for estimating the prediction error, including cross-validation and bootstrap, and showed examples, each one involving the complexity of trees when the test error rate is measured by cost-complexity and the number of terminal nodes. Breiman (1984) [4] note that, in their empirical trials with CART, the test error rates often follow the U-shaped pattern. This is confirmed by our results in both the artificial and anæmic infants datasets. Comparison is made between the two approaches for ten artificial datasets with random noise from the same model, but with different random seeds to generate different datasets (to ensure accurate estimates of error rate). Using the R tree algorithm rpart for large trees and stump (two terminal nodes) showed 3-fold cross-validation has larger bias than 10-fold cross-validation and the bootstrap, with a statistically significant increase in bias. Thus the bootstrap method seems to be uniformly better than cross-validation, for both bias and variance. By examining the two methods on the artificial example and anæmic infants dataset with different folds for cross-validation and sample sizes for bootstrap, our results indicate that the bootstrap method is a generally better approach in both simulated and anæmic infants datasets than those with cross-validation, especially for sample sizes (m=300 and 400). However, V-fold=10 and 7 of anæmic infants dataset are comparable to bootstrap when the estimation test error rate is measured by cost-complexity for the correct tree.

The natural question arises why the bootstrap method is a better approach (in decision trees). One possible explanation is because a small change in the training dataset can result in a very different series of splits. It is leading to an increase in the variance contribution to the test error rate. We would normally expect that the bootstrap method can substantially reduce the variance.

For analysis of anæmic infants dataset using decision trees as described in Section (5), the decision trees of large and pruned trees have re-substitution error rates of about 10% and 15.9%, respectively, whereas the default error rate is about 21%. The error rates decrease by about 52% and 33%, respectively. The decision tree shows that the education of the mother is the most important factor for discriminating between patients and normal healthy infants. It is followed by birth weight -- infants, and the family size, which are the most useful information of the anæmic infants' dataset.

Potential research directions

In this paper, the resampling tree-based methods represent a general approach to model assessment and selection. As is known in any real experiment, the data do not come without problems. We advocate the use of prediction-based resampling approaches in new problems, which is useful to uncover the structure of the data. We feel that continued application of existing techniques to new situations is a prerequisite for progress in data exploration technology.

Compliance with ethical standards

Acknowledgments

Firstly, I express my gratitude to Pro. Charles Taylor (Leeds University) for his scientific advice of this work, secondly, we would like to acknowledge the Research and Consulting Centre (RCC), University of Benghazi, Libya for supporting this work.

References

[1] Mclachlan G. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York. 1992.

[2] Efron B. Bootstrap method: another look a the jacknife. *Annlas of Statistics.* 1979; 7: 1-26.

[3] Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning - Data Mining*, Inference, and Prediction. Springer Series in Statistics. New York. 2017.

[4] Breiman L, Friedman JH, Olshen R, Stone CJ. *Classification and Regression Trees*. Belmont, California. 1984.

[5] Breiman L. Bias, variance and arcing classifier. Technical report, Statistics Department, University of California, Berkeley. 1996b.

[6] Stone MA. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society.* 1974*;* 36: 111-47.

[7] Efron B. Estimating the error rate of a prediction rule: improvement on cross-validation, *Journal of the American statistical Association 78, 316-331*. 1983.

[8] Efron B, Tibhirani R. *An Introduction to The Bootstrap*. Chapman & Hall, London. 1993.

[9] El Gimati Y. Weighted Bagging in Decision Trees: Data Mining. JINAV: *Journal of Information and Visualization.* 2020*;* 1(1): 1-14.

[10] El Ojali. Pattern of anæmic among Libyan infants of northeastern Libya. Master's dissertation, Medicine Department, Garyounis University. 1994.