Inf Technol Manag (2009) 10:223–233
DOI 10.1007/s10799-009-0052-7

# A finite-sample simulation study of cross validation in tree-based models

**Seoung Bum Kim · Xiaoming Huo · Kwok-Leung Tsui**

Published online: 1 July 2009


**Abstract** Cross validation (CV) has been widely used for choosing and evaluating statistical models. The main purpose of this study is to explore the behavior of CV in tree-based models. We achieve this goal by an experimental approach, which compares a cross-validated tree classifier with the Bayes classifier that is ideal for the underlying distribution. The main observation of this study is that the difference between the testing and training errors from a cross-validated tree classifier and the Bayes classifier empirically has a linear regression relation. The slope and the coefficient of determination of the regression model can serve as performance measure of a cross-validated tree classifier. Moreover, simulation reveals that the performance of a cross-validated tree classifier depends on the geometry, parameters of the underlying distributions, and sample sizes. Our study can explain, evaluate, and justify the use of CV in tree-based models when the sample size is relatively small.

**Keywords** Cross validation · Bayes classifier · Trees-based models

S. B. Kim (✉)
Department of Industrial Systems and Information Engineering, Korea University, Seoul, Korea
e-mail: seoungbum.kim@gmail.com; sbkim@uta.edu

X. Huo · K.-L. Tsui
Department of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, USA

X. Huo
e-mail: xiaoming@isye.gatech.edu

K.-L. Tsui
e-mail: ktsui@isye.gatech.edu

## 1 Introduction

Cross validation (CV) was described as early as in Stone (1974). It has been of tremendous interest to characterize why and how a CV method works. In statistics, most of the theoretical works on CV concentrate on *regression* rather than *classification*. Some well cited works include Efron (1983, 1986), Shao (1993, 1996, 1998), and Zhang (1992, 1993a, b). One result of particular interest is Zhang's distributional description of the CV error for linear regression models. For the problems of model selection and error prediction in linear models, certain forms of CV are shown to be equivalent to well known model selection criteria such as the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and the $C_p$ statistics. Based on this framework, good performance of CV and asymptotic convergence can be established.

In regression, the risk function is continuous. Hence, it is relatively easy to study the behavior of CV. However, in classification problems, discontinuity of categorical responses makes it hard to establish an equivalence between CV and some existing criteria. Despite this difficulty, a number of research have been done to explore the performance of CV (in particular, leave-one-out CV) in classification problems. Leave-one-out CV reserves one data point and utilizes the remaining $m - 1$ points to train a model, where $m$ is the number of data points. This process is repeated for all $m$ data points to obtain the estimate of true error. Most works provided bounds on the accuracy of the leave-one-out CV estimate. Rogers and Wagner (1978) and Devroye and Wagner (1979a, b) obtained exponential bounds of the leave-one-out CV estimate for a $k$-nearest neighbor algorithm within the *Probably Approximately Correct* framework. More precisely, they provided the bound of the probability that the leave-one-out CV

error departs from the estimate of true error. Kearns and Ron (1999) derived sanity-check bounds for the leave-one-out CV estimator, showing that the bounds from the leave-one-out CV estimate are not worse than that of the training error estimate.

Despite its popularity, the leave-one-out CV has some shortcomings. Most obvious disadvantage is the high computational cost, because the process requires training the model for every data point. Moreover, leave-one-out CV estimate yields high variance in spite of its low bias mainly due to the use of similar training set in each CV step. Therefore, the leave-one-out CV is not recommended when the learning algorithm is instable. Hastie et al. (2001) pointed out that the CV estimate in tree-based models can underestimate the true error, because the reserved testing set strongly affects determining of the optimal tree, and thus recommended five or ten-fold CV as a good compromise between variance and bias. Kohavi (1995), Zhang (1992), and Breiman and Spector (1992) also showed that 10-fold CV produces smaller variance than the leave-one-out CV. Thus, for instable algorithms (like tree-based), the 10-fold CV is more desirable than the leave-one-out CV to estimate the true error.

This study focuses on the behavior of 10-fold CV estimate in tree-based models. The main questions addressed in this paper are (1) how well does CV in tree-based models estimate test error? (2) How the CV performance varies with different situations? We answer those questions using an experimental approach. The following is a synopsis.

1. *Bayes classifier* Given the distribution of a point cloud, based on the likelihood ratio approach of Neyman–Pearson, an optimal classifier can be derived. Such a classifier is known as a *Bayes classifier*. Such a classifier is in theory the best possible classifier.
2. *Cross-validated classifier* Given a training set, a classifier can be trained by minimizing average error rate that is given in the form of CV. The description of the CV error rate is presented in Sect. 2. This classifier is called a *cross-validated classifier*.
3. *Training and testing errors* Both classifiers (Bayes and cross-validated) can be applied to the training and testing sets. A smaller error rate on the training set does not necessarily mean optimality, because it may be introduced by over-fitting. For a classifier, equality between training error and testing error may be desirable. Moreover, if the Bayes classifier is applied to both training and testing sets, the difference between the two error rates should be small since the difference is only affected by sampling error. On the other hand, if the testing-to-training error difference is large, the randomly sampled data does not reflect the underlying distribution which suggests that the classifier selected is inappropriate.
4. *Methodology evaluation* Based on the above, the following procedure is deployed to evaluate the performance of a cross-validated classifier. The difference between the training error and the testing error is calculated for the cross-validated classifier. Let $e_{1,A}$ and $e_{2,A}$ denote the *training* error of the Bayes and the cross-validated classifiers respectively, where
   - "1" stands for Bayes classifiers,
   - "2" stands for cross-validated classifiers, and
   - "$A$" stands for the training set. Let $e_{1,B}$ and $e_{2,B}$ denote the two corresponding *testing* error rates, where
   - "$B$" stands for the testing set. We consider the differences:

$$D_2 = e_{2,B} - e_{2,A} \quad \text{vs.} \quad D_1 = e_{1,B} - e_{1,A}.$$

5. *Main observation* The main observation is that the above two quantities have a roughly statistically linear relationship. This is more evident in Fig. 5. We have

$$D_1 = a + c \cdot D_2 + \varepsilon, \tag{1}$$

where the constant $c$, $|c| \leq 1$, depends on the underlying distribution and $a$ is the constant value. The random variable $\varepsilon$ has zero mean and seemingly normally distributed.

Note that the asymptotic behavior of $D_1$ and $D_2$ are somewhat known. The cross-validated classifier will converge to the Bayes classifier in most cases (Anthony and Holden 1998); see also Sect. 3. Hence $D_1$ and $D_2$ tend to be equal. This paper is to study their behavior in the finite-sample cases: when the sample size is not large. It is interesting to find that a simple linear regression model seems to characterize the relation between $D_1$ and $D_2$ nicely. The impact of the geometry of decision boundary, the parameter of the underlying distribution, and sample size are also examined in the simulations.

The rest of the paper is organized as follows. Section 2 describes the tree-based models and their relation with CV. Section 3 presents some theoretical analysis. Section 4 describes the simulation results. Section 5 ends the paper with concluding remarks and suggestions for future study.

## 2 Cross validation in tree-based models

### 2.1 The cross-validation principle

Suppose we have two disjoint sets: training set and testing set. The former set is used to learn the model and the latter to evaluate the performance of the trained model. Let $A$ denote the training set of size $N_A$ and $B$ the testing set of size $N_B$. Let us consider a $k$-fold CV and $\alpha$ is an algorithmic

parameter of a model. If we denote $e_\alpha^{(-i)}$ as the error rate when excluding the $i$th folder during CV, the cross-validating error at $\alpha$ is given as

$$CV(A;\alpha) = \frac{1}{k}\sum_{i=1}^{k} e_\alpha^{(-i)}.$$

The principle of CV is to choose an $\alpha$ such that $CV(A;\alpha)$ is minimized:

$$\alpha_0 = \text{argmin}_\alpha CV(A;\alpha).$$

Let $T_{\alpha_0}(A)$ denote the model that is built by using $\alpha = \alpha_0$ together with the training sample $A$. The training error based on CV can be expressed as $e_{CV}(T_{\alpha_0}(A), A)$ (identical with $e_{2,A}$), which denotes the error rate when the model $T_{\alpha_0}(A)$ is applied to the training data $A$. The testing error can be represented as $e_T(T_{\alpha_0}(A),B)$ (identical with $e_{2,B}$), which denotes the error rate when the model $T_{\alpha_0}(A)$ is applied to the testing data $B$.

## 2.2 Complexity-penalized loss function

Tree-based models are very popular in the data mining community because they provide interpretable rules and logic statements that enable more intelligent decision making. In general, tree modeling involves two major steps: tree growing and tree pruning. Tree growing searches over the whole dataset to find the splitting points that lead to the greatest improvements for a specified score function. After the tree reaches the full-grown stage when no further improvement is possible, one prunes back the tree to identify the right-sized tree that provides the minimum error when the tree is applied to the testing dataset (Hastie et al. 2001).

CV is adopted in the tree-pruning step. Among many tree-pruning algorithms, cost-complexity tree-pruning (CCP) (Breiman et al. 1984) and frontier-based tree-pruning (FBP) (Huo et al. 2006) algorithms utilize the principle of CV. The main idea of both CCP and FBP is to consider a complexity-penalized loss function (CPLF) and search the possible set of a penalizing parameter to find the optimal tree using CV. The CPLF has the form

$$L(T_b) + \alpha|T_b|, \tag{2}$$

where $L(T_b)$ is the loss function associated with the tree $T_b$, $|T_b|$ is the size of the tree, which is defined as the number of terminal nodes, and $\alpha$ is a parameter that controls the size of trees. We then solve the following:

$$T_{b_0}(\alpha) = \text{argmin}_{T_b} \quad L(T_b) + \alpha|T_b|.$$

## 2.3 Integration with cross validation

Despite the identical objective function, CCP and FBP use different algorithms to find an optimal tree. FBP is more advantageous to study the behavior of the CV because it can be utilized to implement the principle of cross validation more faithfully. We begin with a brief description on how the FBP algorithm can be used in implementing CV. Suppose the observations are

$$\{(x_1,y_1),(x_2,y_2),(x_3,y_3),\ldots,(x_N,y_N)\},$$

where $N$ is the number of observations, $x_i$'s are predictor variables, and $y_i$'s are responses. Suppose the above set is equally partitioned into $k$ subsets:

$$S_1 \cup S_2 \cup \cdots \cup S_k.$$

At each step, we reserve one subset (e.g., $S_i$) for testing and use the remaining subsets to prune the tree. The core idea of the FBP algorithm is illustrated in Fig. 1. For a given $\alpha$, when the target size of the tree is $m$, the minimum value of CPLF is $c_m + m\alpha$, where $c_m$ is a constant. The first step of FBP is to list CPLF in each node of the tree using a bottom-up tree-pruning algorithm. Then all the information is summarized at the root node as a list of CPLFs. Thus, the number of CPLFs at the root node should be equal to that of terminal nodes of the tree. The next step is to plot all the CPLF at the root node in a Cartesian plane (Fig. 1). The $x$-axis is the range of $\alpha$ and the $y$-axis is the value of the CPLFs. The lower bound of these CPLFs can be obtained as a form of a piecewise linear function and denoted as $f_{-i}(\alpha)$, where $f_{-i}(\alpha)$ is the minimum value of (2) without testing subset $S_i$.

For each value of the parameter $\alpha$, the optimal subtree can be obtained. Each model is then applied to the reserved testing set. The error rate in testing can be computed and is denoted by $e_{-i}(\alpha)$. Note that functions $f_{-i}(\alpha)$ and $e_{-i}(\alpha)$ are of the same variable. Because function $f_{-i}(\alpha)$ is a piecewise linear function, it is not hard to prove that function $e_{-i}(\alpha)$ is also a piecewise step function. The principle of CV is to find the $\alpha$ that minimizes the average of $e_{-i}$'s,

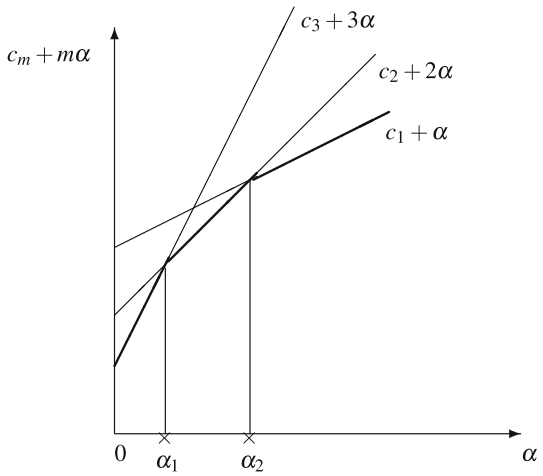


**Fig. 1** An illustration of the frontier-based tree-pruning algorithm

$$\frac{1}{k}\sum_{i=1}^{k} e_{-i}(\alpha).$$

The tree size corresponding to the optimal $\alpha$ is the final tree. Figure 2 shows how the error rates ($e_{CV}(A;\ \alpha)$) vary with $\alpha$. The lowest part of the step function indicates the optimal $\alpha$.

## 3 Analysis

### 3.1 Error difference for the Bayes classifier

Some distributional analysis is given here. The dataset is divided into a training set and a testing set. An oracle knows the underlying distribution and is able to derive a classifier that is statistically optimal—having the minimum expected prediction error rate overall, following the idea of Neymann–Pearson. Such a classifier is known as the Bayes classifier (BC). Note that this classifier does not depend on the sample data. Let $e_{1,*}$ denote the error rate by applying the BC to data *. We can obtain $e_{1,A}$ and $e_{1,B}$ using the following equations:

$$e_{1,A} = \frac{1}{N_A}\sum_{i\in A} I(\hat{Y}_{BC}(X_i), Y_i), \tag{3}$$

$$e_{1,B} = \frac{1}{N_B}\sum_{i\in B} I(\hat{Y}_{BC}(X_i), Y_i), \tag{4}$$

where $N_*$ is the size of dataset * and $I$ is a 0-1 loss function defined as follow:

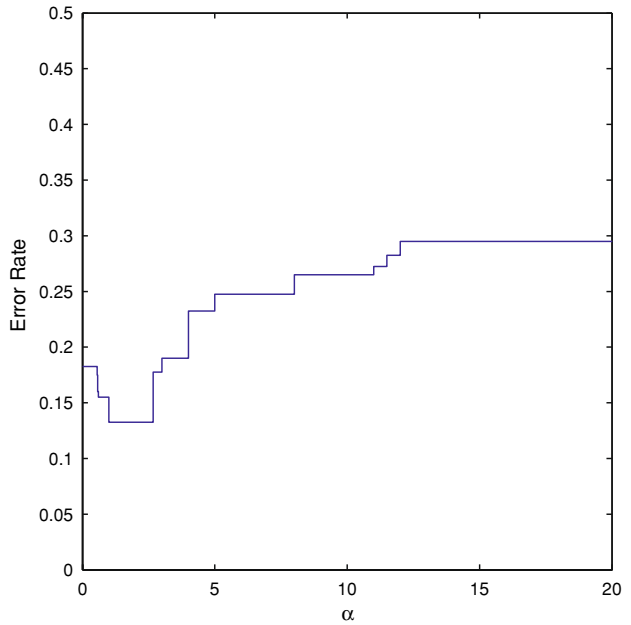


**Fig. 2** The range of the optimal $\alpha$ that produces the smallest error rate. Note that the minimum is not unique

$$I(\hat{Y}(X), Y) = \begin{cases} 0 & \text{if} \quad Y = \hat{Y}(X), \\ 1 & \text{if} \quad Y \neq \hat{Y}(X), \end{cases}$$

and $\hat{Y}_{BC}(\cdot)$ is the Bayes classifier. We can compute their difference:

$$D_1 = e_{1,B} - e_{1,A}. \tag{5}$$

Distribution of $D_1$ can be characterized by the following proposition, simply being derived from the central limit theorem.

**Proposition 1** *Suppose that the minimum misclassification error is a constant p within the state space. When errors $e_{1,A}$ and $e_{1,B}$ are defined as in (3) and (4), we have $e_{1,A} \sim \mathcal{N}(p, \sigma^2_{e_{1,A}})$ and $e_{1,B} \sim \mathcal{N}(p, \sigma^2_{e_{1,B}})$ where p is the true risk. Moreover, we have $D_1 = e_{1,B} - e_{1,A} \sim \mathcal{N}(0, \sigma^2_{D_1})$.*

*Proof* For $i \in A$, $I(\hat{Y}_{BC}(X_i), Y_i)$ can be considered as an independent and identically distributed Bernoulli random variable with success probability $p$, where $p$ also is the true risk; owing to a fixed decision boundary of the BC. Therefore $\sum_{i\in A} I(\hat{Y}_{BC}(X_i), Y_i)$ follows a binomial distribution with parameters $N_A$ and $p$. This can be approximated by a normal distribution, $\mathcal{N}(N_A \cdot p, N_A \cdot p(1-p))$. Thus $e_{1,A}$ can be described as $\mathcal{N}\left(p, \frac{p(1-p)}{N_A}\right)$. Similarly, we have $e_{1,B} \sim \mathcal{N}\left(p, \frac{p(1-p)}{N_B}\right)$. Furthermore, because the difference of two normal distributions is also a normal distribution, $D_1 = e_{1,B} - e_{1,A}$ will also approximately follow a normal distribution with mean 0 and variance $\frac{p(1-p)}{N_A} + \frac{p(1-p)}{N_B}$.

The assumption of a constant $p$ is stringent, however it is consistent with the settings in the simulations. The same results holds in more general cases. For illustration purpose, we choose not to pursue more general results. Note in the above setting, $D_1$ provides the minimum asymptotic variance among all unbiased estimators. We choose not to articulate it either; such a result is known in mathematical statistics.

### 3.2 Error difference for the cross-validated classifier

Incorporating CV in tree-based models yields a classifier, called a cross-validated tree (CVT) classifier. In the present study, we constructed CVTs based on the FBP algorithm, which is described in Sects. 2 and 3.

Let $e_{2,A}$ denote the training error based on CV, which is the error rate by applying CVT to the training data:

$$e_{2,A} = \frac{1}{N_A}\sum_{i\in A} I(\hat{Y}_{CVT}(X_i), Y_i). \tag{6}$$

Let $e_{2,B}$ denote the testing error when the CVT is applied to the testing data:

$$e_{2,B} = \frac{1}{N_B} \sum_{i \in B} I(\hat{Y}_{CVT}(X_i), Y_i). \quad (7)$$

A quantity similar to the one in (5) is

$$D_2 = e_{2,B} - e_{2,A}. \quad (8)$$

Similar to the aforementioned proposition, one can see that $e_{2,B}$ is approximately normally distributed because $e_{2,B}$ can be viewed as a sum of i.i.d. Bernoulli random variables. It is less evident that $e_{2,A}$ satisfies an approximate normal distribution as well. Note that the CVT depends on the training data. Our simulations show that such a dependence is ignorable and $e_{2,A}$ can be considered approximately normally distributed as well. Figure 3 displays the normal probability plots of $e_{2,A}$, $e_{2,B}$, and $D_2$ generated by a relatively small number of sample points (i.e., 200). All three plots do not depart substantially from linearity, suggesting that the error distributions are normal.

### 3.3 Convergence of CVT

We will argue that under amenable situations, the CVT should converge to the optimal Bayes classifier. Under this convergence, similar to the argument in Sect. 1, one can show that $D_2$ asymptotically satisfies a normal distribution with zero mean. In fact, *asymptotically*, $D_1$ and $D_2$ should be equal!

We now present our convergence analysis. Since we consider a tree-based binary classification model, the statistical formulation can be summarized as follows: Consider a univariate response random variable $Y = 0$ or $Y = 1$ and a random vector $X$ (which is called predictor). Recall $(X_i, Y_i)$ denotes the $i$th observed pair of the predictor and the response. Suppose there is a geometrically manageable (which will be specified later) subset $\Omega$. If $X \in \Omega$, we have $Pr(Y(X) = 1) = p = 1 - Pr(Y(X) = 0)$, where probability $p > 1/2$; if $X \notin \Omega$, then we have $Pr(Y(X) = 0) = p = 1 - Pr(Y(X) = 1)$. Under this model, apparently the Bayes classifier is

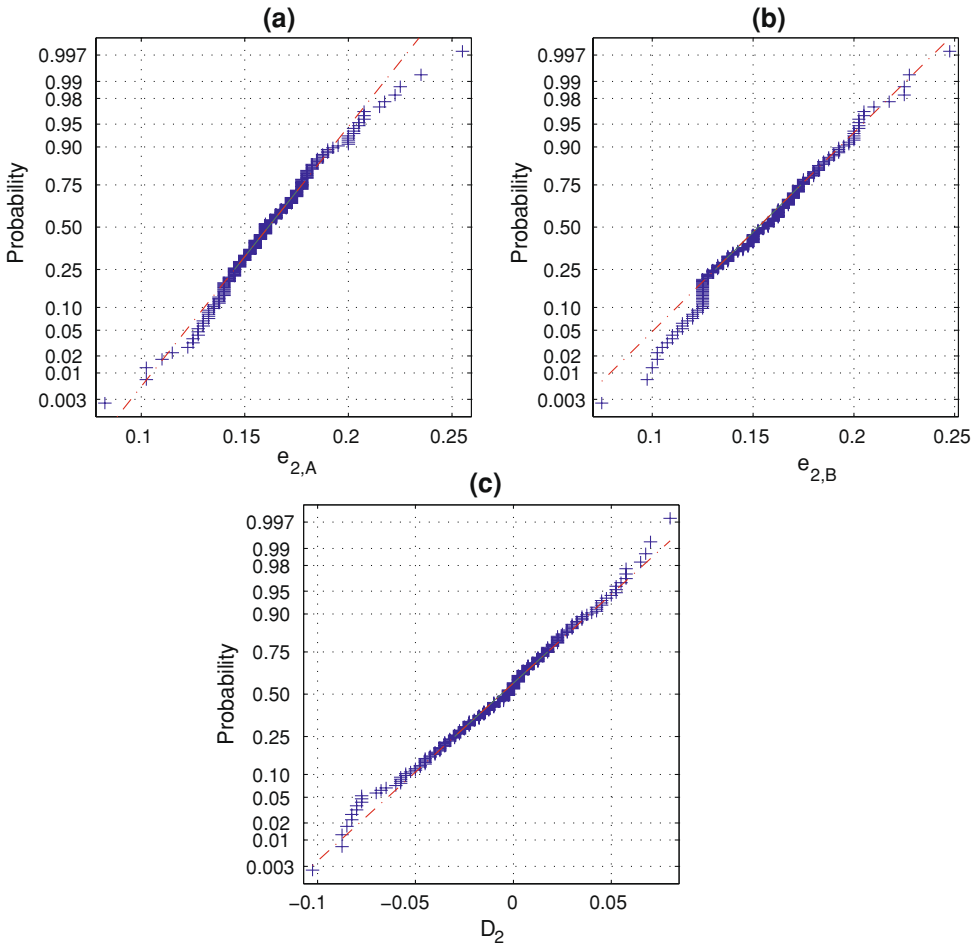


**Fig. 3** Normal probability plots for the errors from CVT based on 200 sample points: **a** training errors from CVT ($e_{2,A}$), **b** testing errors from CVT ($e_{2,B}$), and **c** differences ($D_2 = e_{2,B} - e_{2,A}$)

$$T_{BC}(X) = \begin{cases} 1, & x \in \Omega, \\ 0, & x \notin \Omega. \end{cases}$$

Moreover, by definition, we have that $T_{BC}$ minimizes the following

$$EI[T(X) \neq Y] = Pr[T(X) \neq Y],$$

where $E$ stands for expectation, $I$[*] is an indicator function of event *, and $Pr$[*] stands for the probability of event *.

Denote a partition of the training set $X_A$ with 10 folders as $X_A = X_{(1)} \cup \cdots \cup X_{(10)}$, where $X_{(k)}$, $k = 1, 2, \ldots, 10$ are 10 mutually exclusive and roughly equally sized subsets of $X_A$. Denote $X_{-k} = X_A \setminus X_{(k)}, 1 \leq k \leq 10$, to be the subset of $X_A$ after eliminating $X_{(k)}$. Let $T(X_{-k}; \alpha)$ be the tree-based binary classifier that is obtained by applying the algorithm described in Sect. 2 to subset $X_{(k)}$ with parameter $\alpha$. The cross-validation criterion picks the $\alpha$ that minimizes the following:

$$\sum_{k=1}^{10} \sum_{i \in X_{(k)}} I[T(X_{-k}; \alpha)(X_i) \neq Y_i]/|X_A|, \tag{9}$$

where $T(X_{-k}; \alpha)(X_i)$ is the predicted value of the classifier $T(X_{-k}; \alpha)$ at $X_i$, and $|X_A|$ denote the cardinality of set $X_A$.

We consider two properties, under which we argue that the CVT classifier converges to the Bayes classifier. First, recall that in the algorithm in Sect. 2, we first build a big tree, then adopts a fast algorithm to prune the large tree. The underlying geometric region $\Omega$ should be representable by a subtree model; i.e., a classifier according to an admissible subtree of the initial large tree is close to the classifier that is defined on $\Omega$, e.g., $T_{BC}$. Otherwise, the tree-based approach has no hope to converge to the optimal classifier.

**Property 1** *When the sample size is large enough, an admissible subtree of the largest possible tree (which is built according to the entire sample) leads to a classifier which has statistically similar performance of a classifier that is based merely on whether or not the predictor X is inside Ω, e.g., the Bayes classifier $T_{BC}$.*

We also must assume that distribution of the predictor $X$ is controllable such that when the sample size is large, the optimal classifier is achieved for similar values of $\alpha$.

**Property 2** *The distribution of the predictor X is "reasonable" such that when the total sample size for training set $X_A$ is large, the classifier $T(X_{-k}, \alpha)$ for $1 \leq k \leq 10$ and $T(X_A; \alpha)$ lead to similar classification rule for similar values of α. This is to ensure that optimal classifier is achieved simultaneously (with respect to value of α) for $T(X_{-k}; \alpha)$ with different k.*

The *reasonableness* in the above property means that sets $X_A$ as well as $X_{-1}, \ldots, X{-}10$ have similar statistical behavior, such that tree classifiers that are built according to them are similar as well. Note that the tree classifier, under Property 1, would converge to the Bayes classifier. This property is to ensure that the convergence occurs for a similar $\alpha$ value. The reasonableness can be examined by noticing that $X_A$ as well as $X_{-1}, \ldots, X{-}10$ are samples from an identical source with about the same sizes.

The actual proof of the above two properties will be tedious, and perhaps more suitable for a mathematically oriented paper. However, they are true at least intuitively. In this paper, we choose to sacrifice the mathematical rigor in order to focus on the simulation study that come later. We now show that the aforementioned two properties will lead to the convergence of the CVT to the Bayes classifier. According to Property 1, there exists an $\alpha$ such that $T(X_{-k}; \alpha)$ becomes a classifier that purely depends on the membership of the predictor $X$ in the underlying set $\Omega$. Suppose this classifier is $T$. From Property 2, such a classifier can be achieved simultaneously for all the $k$, $1 \leq k \leq 10$. Note that when the sample size goes to infinity, the value of the expression in (9) satisfies:

$$\sum_{k=1}^{10} \sum_{i \in X_{(k)}} I[T(X_i) \neq Y_i]/|X_A| \Rightarrow EI[T(X)],$$

where "⇒" stands for "converging to," and the right hand side is minimized by the Bayse classifier. From the definition of the Bayes classifier, the minimizer $T$ should be the Bayes classifier. This demonstrates that the CVT, when the sample size converges to infinity, converges to the Bayes classifier.

*Remark 1* Property 1 is to ensure that the geometric region $\Omega$ is manageable by a binary tree model.

*Remark 2* Property 2 says that when the sample is large, the sets $X_A, X_{-1}, \ldots, X{-}10$ have similar statistical behavior, so they lead to the similar tree classifier for a similar and moderate $\alpha$.

*Remark 3* The above argument can be generalized to any $K$-fold cross validation as long as $K$ is finite. The above argument cannot be utilized for leave-one-out cross validation because the number of folders then goes to infinity along with the sample size.

### 3.4 $D_1$ versus $D_2$

As mentioned above, in an asymptotic sense, the CVT should converge to the Bayes classifier. Hence we should have $D_1 = D_2$ for a large sample size. In the small sample case, the aforementioned may not be true. It is interesting to observe that the two normally distributed random variables $D_1$ and $D_2$ seem to fit nicely into a simple linear regression model based on an ordinary least squares estimation method. In particular, we can conjecture that the following is approximately true in finite-sample situations:

$$D_1 = a + c \cdot D_2 + \varepsilon, \tag{10}$$

where the error $\varepsilon$ has zero mean and a constant variance and the $y$ intercept $a$ is constant value. Moreover, we have that $0 < c < 1$ and the value of $c$ together with the variance of $\varepsilon$ depend on the underlying geometry, the sample size, and the ratio between the numbers of training and testing samples. A direct application is as follows: a combination of larger $c$ and smaller variance of $\varepsilon$ indicates an amenable scenario to adopt the CV approach. Such an observation can be utilized to develop guidelines on when to apply CV when the data size is small. Evidently, such a guideline is important in applications. This paper focuses on the framework, instead of deriving the exact guideline for specific models.

## 4 Simulations

### 4.1 The setting

Data points $X$ are uniformly generated in a unit square. Figure 4 illustrates 200 data points. Decision boundaries are as follows.

- Case 1: X $\in B_r$ where $B_r$ is a rectangular decision boundary for $0.2 \le X_1 \le 0.8$ and $0.3 \le X_2 \le 0.8$.
- Case 2: X $\in B_c$ where $B_c$ is a circular decision boundary for $(X_1-0.5)^2 + (X_2-0.5)^2 < 0.2^2$.
- Case 3: X $\in B_t$ where $B_t$ is a triangular decision boundary for $X_2 > 0.2$, $X_2 < 2X_1 - 0.2$, and $x_2 < -2X_1 + 1.8$.

Success probability $p$ plays the role as in the following:

- If input $X \in B_x$ for $x \in \{r, c, t\}$, then we have response

$$Y = \begin{cases} 1 & \text{with probability (w.p.)} \quad p, \\ 0 & \text{w.p.} \quad 1-p. \end{cases}$$

- If input $X \notin B_x$, then we have response

$$Y = \begin{cases} 1 & \text{w.p.} \quad p-1, \\ 0 & \text{w.p.} \quad p. \end{cases}$$

For each generated data, we compute a series of aforementioned error rates ($e_{1,A}$, $e_{2,A}$, $e_{1,B}$, and $e_{2,B}$). Recall that 10-fold CV is used to obtain $e_{2,A}$ and $e_{2,B}$.

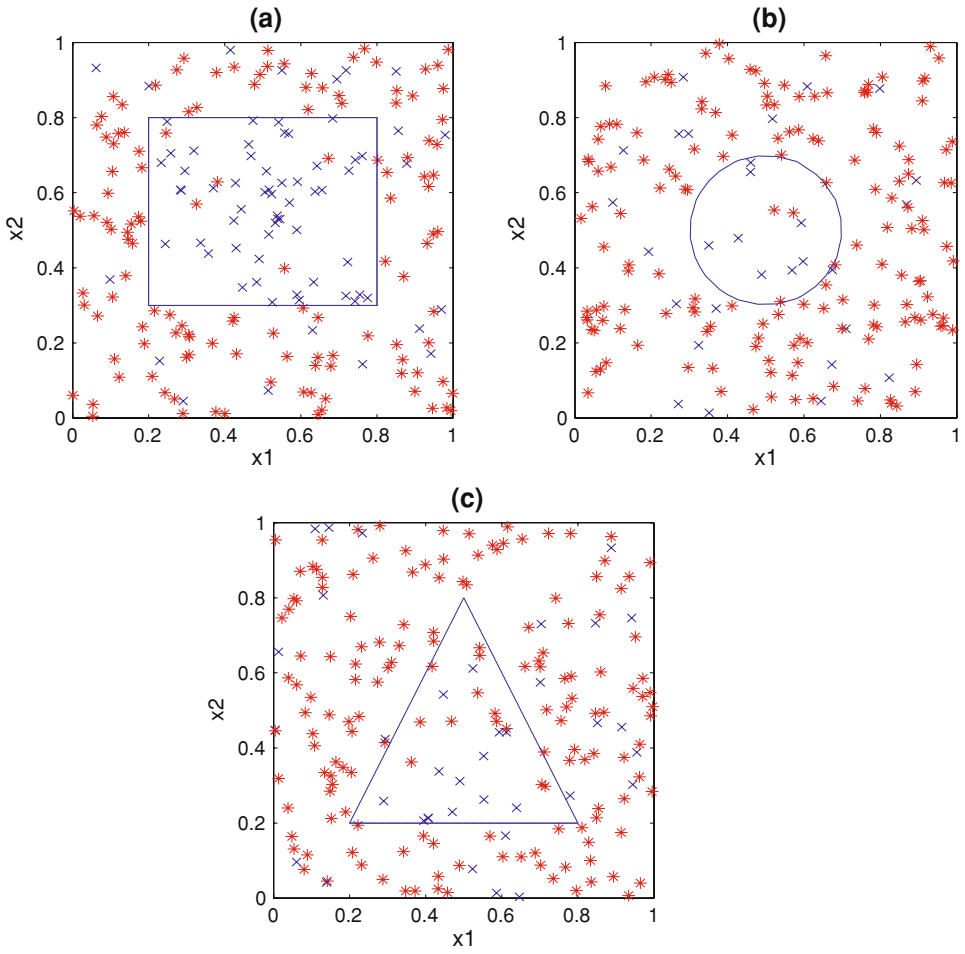


**Fig. 4** Illustration of 200 simulated data with three different decision boundaries. **a** rectangular decision boundary, **b** circular decision boundary, and **c** triangular decision boundary

### 4.2 Relation between $D_1$ and $D_2$ and effects of the geometry of decision boundaries

To identify a statistical relation between $D_1$ and $D_2$, we utilize a linear regression analysis. $D_1$ is taken as the response variable and $D_2$ as the predictor variable. Figure 5 represents linear regression lines between $D_1$ and $D_2$ with three different types of decision boundaries. Our experimental results with the finite sample size show that differences between testing and training errors from the Bayes classifiers ($D_1$) and the CVT ($D_2$) are not zero but can be modeled by a simple linear regression model.

Slopes of regression lines are shown in Table 1. Note that the sample size does not significantly affect the slope. It can be also seen that the slopes of the regression models through the origin (values in the parentheses in Table 1) are not significantly different from the ones of the regression models with the intercepts. This small difference, however, does not indicate that the $y$ intercepts are statistically zero.

Table 1 shows that the relationship of the difference between testing and training errors of the BC and CVT is affected by the geometry of the decision boundaries. The closer the slope is to one, the less the difference between

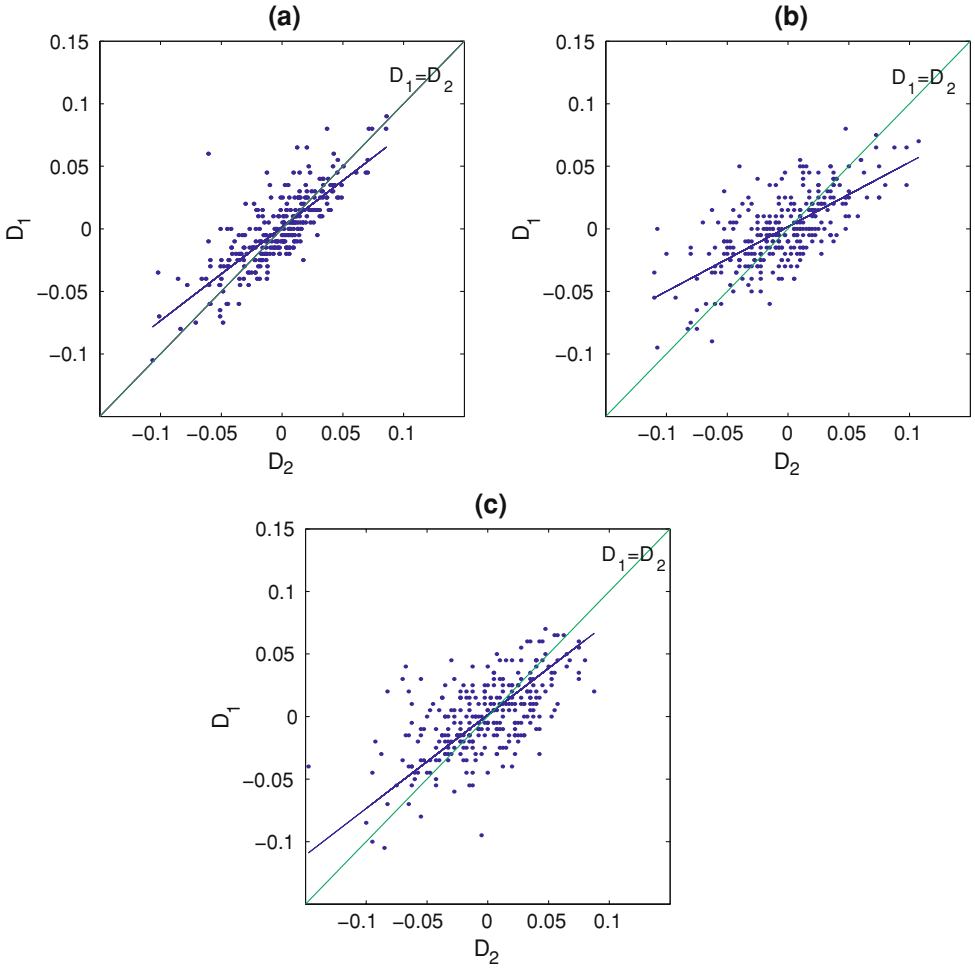


**Fig. 5** Regression plots between $D_2$ and $D_1$ with *rectangular*, *circular*, and *triangular* boundaries. Regression lines are generated with $p = 0.1$ and sample size 300. The *diagonal lines* ($D_1 = D_2$) are plotted as references

**Table 1** Slopes of the regression models with different decision boundaries and dataset sizes

| Sample size | 20 | 50 | 100 | 200 | 300 | 400 | 500 |
|---|---|---|---|---|---|---|---|
| Rectangle | 0.852 (0.852) | 0.635 (0.703) | 0.745 (0.741) | 0.764 (0.760) | 0.747 (0.734) | 0.797 (0.797) | 0.775 (0.774) |
| Circle | 0.501 (0.476) | 0.529 (0.518) | 0.528 (0.529) | 0.502 (0.494) | 0.516 (0.511) | 0.548 (0.531) | 0.528 (0.519) |
| Triangle | 0.525 (0.530) | 0.489 (0.433) | 0.403 (0.409) | 0.459 (0.460) | 0.485 (0.483) | 0.387 (0.388) | 0.516 (0.513) |

The values in the parentheses indicate the slopes in the regression models through the origin

**Table 2** Coefficient of determination ($R^2$) with different decision boundaries and sample sizes

| Sample size | 20 | 50 | 100 | 200 | 300 | 400 | 500 |
|---|---|---|---|---|---|---|---|
| Rectangle | 0.878 | 0.705 | 0.733 | 0.718 | 0.643 | 0.729 | 0.674 |
| Circle | 0.339 | 0.441 | 0.459 | 0.430 | 0.456 | 0.431 | 0.467 |
| Triangle | 0.449 | 0.282 | 0.316 | 0.320 | 0.370 | 0.388 | 0.414 |

$D_1$ and $D_2$. It is not hard to imagine why a rectangular decision boundary has a larger value of the slope than other decision boundaries. This is due to the characteristic of the recursively binary splitting of the feature space in tree-based models. Furthermore, Table 2 shows the coefficient of determination ($R^2$) of each boundary. It shows that a rectangular boundary yields larger $R^2$ than the others, which suggests that a strong degree of linear association between $D_1$ and $D_2$. In other words, a CVT based on a rectangular decision boundary behaves more like the Bayes classifier compared to circular and triangular boundaries.

### 4.3 The effect of the parameters in an underlying distribution

Recall that the underlying distribution in our simulation is Bernoulli($p$), where $p$ is the constant misclassification rate applying to the entire state space (see Sect. 1). Table 3 describes the slopes of regression lines with different values of $p$ with a rectangular decision boundary.

The other geometries of decision boundaries give similar results. Figure 6 displays the boxplots of the slopes for different parameter values. It shows that parameter values between 0.1 and 0.2 produce a strong linear relationship between the BC and the CVT, however this relation becomes weaker as the parameter value becomes either extremely small or close to 0.5. It is not difficult to explain why $D_1$ and $D_2$ have a weak linear relationship as $p$ approaches to 0.5. If $p$ equals 0.5, we have the same probability for each class being inside or outside of a decision boundary. In this case, classification processes are mostly affected by random effects instead of the decision

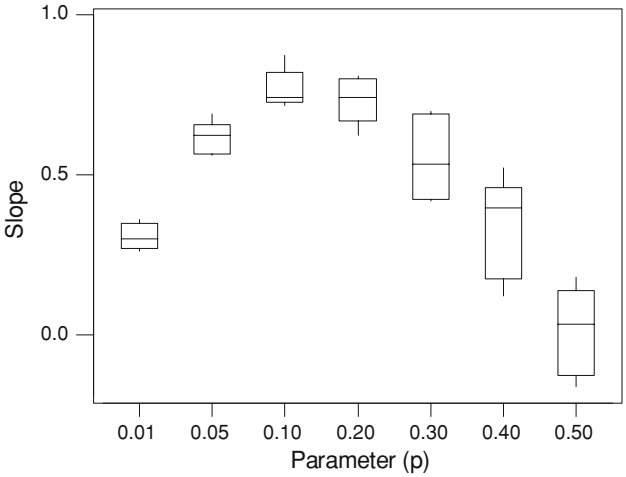


**Fig. 6** Boxplots of slopes in a regression line with different parameters. Plots are generated with rectangular decision boundary

rule. This randomness causes a weak relationship between the two classifiers. For small $p$ (e.g., $p = 0.01$), the relationship of two classifiers is very sensitive to the changes of error rates because both classifiers render very small error rates. Such a high sensitivity results in a weak relationship between the two classifiers. $R^2$s for the above regression analysis show similar patterns with slopes in regression models (this result is not reported).

### 4.4 The effect of the sample size

We study the relationship of the equality between testing and training errors from both classifiers with different sample sizes. First we consider five different total sample sizes (testing & training): 100, 200, 300, 400, 500. For each sample size, we consider five different ratios of testing to the training samples: 1:3, 1:2, 1:1, 2:1, 3:1. Table 4 shows the slopes in a regression line from different ratios of the training and the testing sample sizes. Again, because the intercepts in a regression line are not statistically significant, we consider the slopes with zero intercept shown in the parentheses in Table 4. Figure 7 illustrates a

**Table 3** Slopes in a regression line with different parameters

| | 0.01 | 0.05 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
|---|---|---|---|---|---|---|---|
| 1 | 0.261 (0.261) | 0.631 (0.626) | 0.747 (0.741) | 0.747 (0.741) | 0.700 (0.700) | 0.506 (0.522) | 0.120 (0.099) |
| 2 | 0.352 (0.339) | 0.624 (0.623) | 0.875 (0.874) | 0.810 (0.810) | 0.458 (0.432) | 0.275 (0.228) | 0.002 (0.033) |
| 3 | 0.272 (0.279) | 0.460 (0.561) | 0.743 (0.743) | 0.790 (0.789) | 0.714 (0.680) | 0.496 (0.400) | −0.181 (−0.162) |
| 4 | 0.348 (0.359) | 0.570 (0.570) | 0.770 (0.767) | 0.619 (0.624) | 0.458 (0.419) | 0.157 (0.120) | 0.158 (0.179) |
| 5 | 0.301 (0.301) | 0.690 (0.690) | 0.714 (0.716) | 0.822 (0.716) | 0.542 (0.535) | 0.481 (0.397) | −0.089 (−0.101) |
| Average | 0.307 (0.301) | 0.595 (0.614) | 0.770 (0.768) | 0.758 (0.736) | 0.575 (0.553) | 0.082 (0.107) | |
| SD | 0.042 (0.041) | 0.087 (0.052) | 0.062 (0.062) | 0.083 (0.073) | 0.126 (0.133) | 0.080 (0.101) | |

The values in the parentheses are the slopes in a regression line through the origin

**Table 4** Slopes in a regression line with different sizes and ratio of training and testing sets

| | 100 | 200 | 300 | 400 | 500 | Average | SD |
|---|---|---|---|---|---|---|---|
| 3:1 | 0.557 (0.553) | 0.710 (0.718) | 0.790 (0.786) | 0.858 (0.859) | 0.861 (0.861) | 0.755 (0.755) | 0.127 (0.127) |
| 2:1 | 0.401 (0.407) | 0.696 (0.700) | 0.740 (0.740) | 0.859 (0.864) | 0.906 (0.906) | 0.720 (0.722) | 0.198 (0.196) |
| 1:1 | 0.388 (0.381) | 0.568 (0.563) | 0.761 (0.744) | 0.770 (0.767) | 0.774 (0.767) | 0.652 (0.644) | 0.171 (0.170) |
| 1:2 | 0.254 (0.240) | 0.432 (0.440) | 0.730 (0.731) | 0.736 (0.736) | 0.721 (0.720) | 0.576 (0.572) | 0.219 (0.223) |
| 1:3 | 0.148 (0.136) | 0.416 (0.416) | 0.583 (0.582) | 0.584 (0.575) | 0.666 (0.668) | 0.479 (0.475) | 0.206 (0.210) |
| Average | 0.349 (0.343) | 0.566 (0.576) | 0.721 (0.716) | 0.761 (0.758) | 0.786 (0.784) | | |
| SD | 0.156 (0.161) | 0.138 (0.141) | 0.080 (0.078) | 0.113 (0.118) | 0.0986 (0.098) | | |

The values in the parentheses indicate the slopes in a regression line through origin

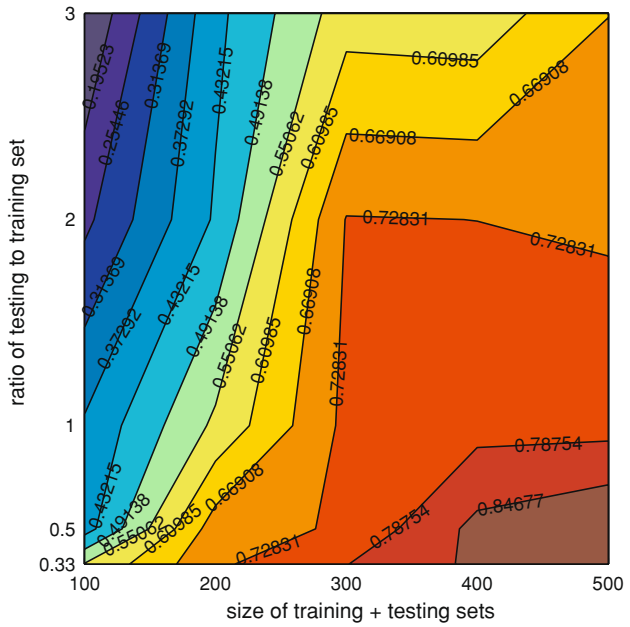


**Fig. 7** Contour plot of slopes with different sizes and ratios of training and testing set. Plot is generated with rectangular decision boundary

three-dimensional contour plot in which $x$ and $y$-axes represent the sample size and the ratio of testing to training samples. For example, if the values on the $x$ and $y$-axes are 300 and 2, this indicates that the experiment is performed using 100 training samples and 200 testing samples. The $z$-axis (the values on the contour plot) indicates the slopes of each regression line. This plot can be used to determine the ratio of testing to the training sample size for achieving the targeted performance. For instance, if we want our CVT classifier to be $\frac{1}{0.84677}$ of the Bayes classifier, the corresponding values on the $x$-axis suggest the ratio corresponding to the different total sample sizes. The result also indicates that changes of slopes become stabilized when sample size is larger than 300. This may indicate the border line between the large sample (asymptotic) size and the finite sample size.

## 5 Conclusions

An experimental study is presented to measure the performance of CV in tree-based models. We compares a CVT with a Bayes classifier, which is derived from the knowledge of the underlying distribution. We focus on the finite-sample case. Main observation is that the differences between testing and training errors from both the CVT and the Bayes classifiers follow a simple linear regression model. The slope of the regression line and the variance of the random error can be served as a measure on how well CV may work in that particular situation for tree-based models. $R^2$ are employed to validate the relation. Both the slope and $R^2$ being equal to one suggests a strong relationship between two classifiers. In addition, it is demonstrated that the above relation is influenced by other factors such as the shape of the decision boundaries, the probabilistic parameter of the underlying distribution, and the sample size. It should be noted that because our current study was conducted based solely upon 10-fold CV, the results may not be generally applicable to CV on different number of folds. There are interesting directions for future research: one can extend our study to other learning algorithms, such as support vector machines, neural networks, and so on. Authors believe that the finite sample behavior of the cross validation error is a fascinating research topic.

**Acknowledgements** The authors would like to thank the editors and two anonymous reviewers whose comments helped significantly to improve the quality of this paper.

## References

M. Anthony, S.B. Holden, Cross-validation for binary classification by real-valued functions: theoretical analysis, in *Proceedings of the 11th Annual Conference on Computational Learning Theory*, (1998) pp. 218–229

L. Breiman, P. Spector, Submodel selection and evaluation in regression: the X-random case. Int. Stat. Rev. **60**(3), 291–319 (1992)

L. Breiman, J. Friedman, R. Olshen, C. Stone, *Classification and Regression Trees* (Wadsworth International Group, Belmont, 1984)

L.P. Devroye, T.J. Wagner, Distribution-free inqualities for the deleted and holdout error estimates. IEEE Trans. Inf. Theory **25**(2), 202–207 (1979a)

L.P. Devroye, T.J. Wagner, Distribution-free performance bounds for potential function rules. IEEE Trans. Inf. Theory **25**(5), 601–604 (1979b)

B. Efron, Estimating the error rate of a prediction rule: improvement of cross-validation. J. Am. Stat. Assoc. **78**(382), 316–331 (1983).

B. Efron, How biased is the apparent error rate of a prediction rule? J. Am. Stat. Assoc. **81**(394), 461–470 (1986)

T. Hastie, R. Tibshirani, J.H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, New York, 2001)

X. Huo, S.B. Kim, K.-L. Tsui, S. Wang, FBP: a frontier-based tree-pruning algorithm. INFORMS J. Comput. **18**(4), 494–505 (2006)

M. Kearns, D. Ron, Algorithmic stability and sanity check bounds for leave-one-out cross validation bounds. Nueral Comput. **11**(6), 1427–1453 (1999)

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection, in *International Joint Conference of Artificial Intelligence (IJCAI)* (1995), pp. 1137–1145.

W.H. Rogers, T.J. Wagner, A finite sample distribution-free performance bound for local discimination rule. Ann. Stat. **6**, 506–514 (1978)

J. Shao, Linear-model selection by cross-validation. J. Am. Stat. Assoc. **88**(422), 486–494 (1993)

J. Shao, Bootstrap model selection. J. Am. Stat. Assoc. **91**(434), 655–665 (1996)

J. Shao, Convergence rates of the generalization information criterion. J. Nonparametr. Stat. **9**(3), 217–225 (1998)

M. Stone, Cross-validatory choice and assessment of statistical prediction. J. Roy. Stat. Soc. B **36**(2), 111–133 (1974)

P. Zhang, On the distributional properties of model selection criteria. J. Am. Stati. Assoc. **87**(419), 732–737 (1992)

P. Zhang, Model selection via multifold cross validation. Ann. Stat. **21**(1), 299–313 (1993a)

P. Zhang, On the convergence rate of model selection criteria. Commun. Stat. Theory Methods **22**(10), 2765–2775 (1993b)