

# Performance of Cross Validation in Tree-Based Models

Seoung Bum Kim, Xiaoming Huo, Kwok-Leung Tsui

School of Industrial and Systems Engineering  
Georgia Institute of Technology  
Atlanta, Georgia 30332

{*sbkim, xiaoming, ktsui*}@*isye.gatech.edu*

July 11, 2005

## Abstract

Cross Validation (CV) is widely used to measure the performance of a classifier. The main purpose of this study is to explore the behavior of CV in tree-based models. We report experimental studies that compare a cross-validated tree classifier with an oracle classifier that is ideally derived on the knowledge of underlying distributions. The main observation of this study indicates that the difference between the testing and training error from a cross-validated tree classifier and an oracle classifier empirically has a linear regression relation. The “slope” and the “ $R^2$ ” of regression models are employed as the performance measures of a cross-validated tree classifier. Moreover, simulation reveals that the performance of a cross-validated tree classifier depends on the geometry, the parameters of the underlying distributions, and sample size. Such observations can explain and justify the behavior of CV in tree-based models.

KEY WORDS: Cross validation; Data mining; Oracle property; Trees-based models

## 1. Introduction

Cross Validation (CV) was described as early as Stone (1974). It has been of tremendous interest to characterize why and how a CV method works. With the statistical point of view, most of the theoretical work on CV concentrates on *regression* applications rather than *classification*. Some well cited works include Efron (1983, 1986), Shao (1993, 1996, 1998), and Zhang (1992, 1993a,

1993b). Of special value is Zhang’s description of a distributional property of CV 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 the regression problem, the risk function is continuous. Hence it is relatively easy to study the behavior of CV. However, in classification problems, nonlinearity related to categorical response makes it hard to establish an equivalence between CV and some existing criteria. Despite this difficulty, a number of research have been done for exploring the performance of CV, in particular, leave-one-out CV. In leave-one-out CV, we reserve one data point and utilizes the remaining  $m - 1$  points to train the algorithm, where  $m$  is the number of data points. This process is repeated for  $m$  times 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  $k$ -nearest neighbor algorithms within the *Probably Approximately Correct* (PAC) framework. More precisely, they provided the bound of the probability that the leave-one-out CV error departs from the estimate of true error. Holden (1996) also derived upper bounds of CV estimates for two specific algorithms: the closure algorithm and the deterministic 1-inclusion graph prediction strategy. Kearns and Ron (1999) derived sanity-check bounds for the leave-one-out CV estimate showing that the bounds from the leave-one-out CV estimate are not worse than that of the training error estimate.

Despite its popularity by many practitioners, the leave-one-out CV has some shortcomings. Most obvious disadvantage is high computational time except the several situations since process requires the same number of operations as the data points. Moreover, leave-one-out CV estimate has high variance in spite of its low bias mainly due to the use of similar training set in each CV steps. This implies that the leave-one-out CV is not recommended when the learning algorithm is instable. As described in Hastie *et al.* (2001), the CV estimate in tree-based models can underestimate the true error considerably since the reserved testing set strongly affects determining of the optimal tree. Hastie *et al.*, (2001) recommended five or ten-fold CV as a good compromise between variance and bias. Kohavi (1995), Zhang (1992), and Brieman and Spector (1989) also showed that 10-fold CV

produces smaller variance than 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 CV estimate (10-fold) in tree-based models. The main questions addressed in our paper are 1) how well does CV in tree-based models estimate test error? 2) how the CV performance varies under different situations? We answer those questions using an experimental approach (rather than theoretical approach.) The following is a synopsis of the approach.

1. *Oracle classifier.* Given the distribution of point clouds derived from likelihood ratio of the Neyman-Pearson, an optimal classification rule is derived. Since one needs to know the underlying distribution, such a classifier is called the *oracle classifier*.
2. *Cross-validated classifier.* Given a training set, a classifier can be trained by a minimized average error rate given in the form of CV. The description of the CV error rate is presented in the following sections. This classifier is called the *cross-validated classifier*.
3. *Training and testing errors.* Both classifiers (oracle and cross-validated) can be applied to the training and testing sets. In general, 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 oracle 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.

$$(\text{testing error} - \text{training error})_{\text{oracle}} \approx 0$$

On the other hand, if the testing-to-training error difference is huge, the randomly sampled data does not reflect the underlying distribution. This suggests that the classifier selected is inappropriate.

4. *Methodology evaluation.* Based on previous analysis, the following method can be used to analyze 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  $e_{2,A}$  respectively denote the *training* error of the oracle and the cross-validated classifiers, where

- “1” stands for oracle 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 testing set.

We consider the differences:

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

5. *Main observation.* The main observation is that the above two quantities have a roughly statistically linear relationship. This is more evident in Figure 6. Let  $D_1 = e_{1,B} - e_{1,A}$  and  $D_2 = e_{2,B} - e_{2,A}$ , we have

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

where the constant  $C$ ,  $|C| \leq 1$ , depends on the underlying distribution, and the random variable  $\varepsilon$  has zero mean and seemingly normal distribution.

In our simulations, the data are generated according to known distributions. Based on the distributions, two classifiers are considered: the oracle classifier and the cross-validated tree classifier. In most cases, we observe the phenomenon that is depicted in (1). The influence of the geometry of decision boundary, the parameter of the underlying distributions, and sample size are studied in the simulations.

The rest of the paper is organized as follows. Section 2 reviews the cross-validation principle. Section 3 describes the tree-based models and their links with CV. Section 4 presents some initial theoretical analysis. Section 5 describes the simulation results. The final section, Section 6, ends the paper with a few concluding remarks and suggestions for future study.

## 2. The Cross-Validation Principle

Suppose we have two disjoint sets: a training and a testing set. The former set is used to learn the model and the latter to evaluate the performance of the trained model. The framework of a generic validation process is illustrated in Figure 1 and can be summarized the following 5 steps.

*Step 1.* Divide the data into training and testing sets.

*Step 2.* Train the model using the training set.

*Step 3.* Select the parameter(s) of the model using the training set via CV.

*Step 4.* Select the best model from steps 2 and 3.

*Step 5.* Assess the final model using the testing set.

Let  $A$  denote the training set of size  $N_1$  and  $B$  the testing set of size  $N_2$ . Let  $F$  denote the common underlying rule for both sets. We 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 by

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

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

$$\alpha_0 = \underset{\alpha}{\operatorname{argmin}} CV(A; \alpha)$$

Let  $T_{\alpha_0}(A)$  denote the model that is built by using  $\alpha = \alpha_0$  and the training sample  $A$ . We then have two different errors: training error based on CV and testing error. The former can be expressed as  $e_{CV}(A; \alpha_0)$ , which the model  $T_{\alpha_0}(A)$  produces. The testing error can be represented as  $e_T(T_{\alpha_0}(A), B)$ , which denotes the error rate when the model  $T_{\alpha_0}(A)$  is applied to the data  $B$ .

The quantities described here can be summarized as follows:

$A \Rightarrow \alpha_0 = \underset{\alpha}{\operatorname{argmin}} \frac{1}{K} \sum_{i=1}^K e_\alpha^{(-i)} = \underset{\alpha}{\operatorname{argmin}} CV(A; \alpha)$	CV
$\Rightarrow T_{\alpha_0}(A)$	optimal model
$\Rightarrow e_{CV}(A; \alpha_0)$	training error based on CV
$A, B \Rightarrow e_T(T_{\alpha_0}(A), B)$	testing error

### 3. Cross Validation in Tree-Based Models

#### 3.1 Complexity-Penalized Loss Function

Tree-based models have been very popular in various fields because of their interpretability and flexibility. Tree modeling involves two major steps: tree growing and tree pruning. Tree growing

searches over the whole data set to find the splitting point that leads to the greatest improvement in a specified score function. Once the trees become fully grown state where further improvement is no longer useful, we prune back the tree (or reducing the number of terminal nodes) to pursue the right sized tree that provides the minimum error when the tree is applied to unseen data.

On the whole, the process of CV in tree-based model is required in tree-pruning step. Cost-Complexity tree-Pruning (CCP) (Breiman *et al.*, 1984) and Frontier-Based tree-Pruning (FBP) (Huo *et al.*, 2004) algorithms utilize the Complexity-Penalized Loss Function (CPLF) (Equation (2)) containing an algorithmic parameter, which controls the size of trees. The main idea of CCP and FBP algorithms consider CPLF and search the possible set of a penalizing parameter to find the optimal tree using CV. The difference between CCP and FBP is described in Huo *et al.* (2004). For more precise explanation of CPLF in Equation (2),  $L(T_b)$  is the loss function associated with tree  $T_b$  and  $|T_b|$  is the size of the tree  $T_b$ , which is defined as the number of terminal nodes.

$$L(T_b) + \alpha|T_b|, \quad (2)$$

where  $\alpha$  is a penalizing parameter. The principle of minimizing CPLF is to choose a subtree  $T_b$  that minimizes the CPLF. In other words, the principle of minimizing CPLF is to solve

$$T_{b_0} = \underset{T_b}{\operatorname{argmin}} \quad L(T_b) + \alpha|T_b|. \quad (3)$$

### 3.2 Integration with Cross Validation

In this study we employ the FBP algorithm (Huo *et al.* 2004) that provides an efficient graphical illustration to prune the tree. Suppose the observations are

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

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:

$$S_1 \cup S_2 \cup \dots \cup S_k.$$

At each step, we reserve one subset (i.e.,  $S_i$ ) for testing and use the remaining subsets to grow a tree and then prune back the tree. The core idea of the FBP algorithm is illustrated in Figure

2. 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 intercept.) The first step of FBP is to list CPLF in each node of the tree using bottom-up tree-pruning algorithm. Then all the information is summarized at the root node as a list of CPLF. 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 (Figure 2.) 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 subset, 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 the value of the average of  $e_{-i}$ 's

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

is minimized. The tree size corresponding to the optimal  $\alpha$  is the final tree. Figure 3 shows the error rates ( $e_{CV}(A; \alpha)$ ) vary depending on  $\alpha$ . The lowest part of the step function indicates the optimal  $\alpha$ .

## 4. Analysis

In this section we describe some distributional analysis. Suppose the data is again divided into a training set and a testing set. If there is an oracle, who knows the underlying distribution, he/she can derive a classifier, which works statistically optimal – having the minimum testing error overall, following the principle of Neymann-Pearson. We call such a classifier an Oracle Classifier (OC.) Note that this classifier does not depend on the sampled data. Let  $e_{1,*}$  denote the error rate by applying the OC to data \*. Then we can obtain  $e_{1,A}$  and  $e_{1,B}$  using the following equations:

$$e_{1,A} = \frac{1}{N_A} \sum_{Y_A} I(\hat{Y}_{OC(X_A)}, Y_A), \quad (4)$$

$$e_{1,B} = \frac{1}{N_B} \sum_{Y_B} I(\hat{Y}_{OC(X_B)}, Y_B), \quad (5)$$

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

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

Also, we can compute their difference:

$$D_1 = e_{1,B} - e_{1,A}. \quad (6)$$

**Proposition 4.1** *When errors  $e_{1,A}$  and  $e_{1,B}$  are defined as in equations (4) and (5), we have  $e_{1,A} \sim \mathcal{N}(p, \sigma_{e_{1,A}}^2)$  and  $e_{1,B} \sim \mathcal{N}(p, \sigma_{e_{1,B}}^2)$  where  $p$  is the true risk. Therefore,  $D_1 = e_{1,B} - e_{1,A} \sim \mathcal{N}(0, \sigma_{D_1}^2)$ .*

**Proof.**  $I(\hat{Y}_{OC(X_A)}, Y_A)$  can be described as an independent and identical Bernoulli distribution with  $p$ , where  $p$  is the true risk. Independency and identity are hold since the decision boundaries of an oracle classifier are fixed in each experiment. Therefore,  $\sum_{Y_A} I(\hat{Y}_{OC(X_A)}, Y_A)$  follows a Binomial distribution with  $N$  and  $p$ , where  $N$  is the number of experiments. This can be approximated by  $\mathcal{N}(Np, Np(1-p))$ . Thus,  $e_{1,A}$  can be described as  $\mathcal{N}\left(p, \left(\sqrt{\frac{p(1-p)}{N_A}}\right)^2\right)$ . Similarly, we have  $e_{1,B} \sim \mathcal{N}\left(p, \left(\sqrt{\frac{p(1-p)}{N_B}}\right)^2\right)$ . Furthermore, because the difference of two Normal distributions is also a Normal distribution,  $D_1 = e_{1,B} - e_{1,A}$  will also follow a approximate Normal distribution with mean 0 and variance equals to  $\frac{p(1-p)}{N_A} + \frac{p(1-p)}{N_B}$ .  $\square$

Note that depending on the sampled data,  $D_1$  is not necessarily zero. However, since it only depends on the sampling errors, its expectation should be around zero and its variance ( $\sigma_{D_1}^2$ ) is in some sense the minimum. The following is an observation.

**Observation 4.2** *Suppose  $D_\xi$  is the difference between testing and training errors in any other classifier. We have  $\sigma_{D_1}^2 \leq \sigma_{D_\xi}^2$ .*

In the following paragraph, we briefly review some known general principles. The proof of the above observation can be derived by following general principles, with more technical detail.

The target space ( $T$ ) can be defined as the space of the functions, containing the ideal classifier that minimize the risk. And the hypothesis space ( $H$ ) can be defined as the space of functions that a learning algorithm is allowed to search. Several risks (from  $T$  and  $H$ ) can be defined as follows:

- $E_{f_T}$ : The true risk of the best function in  $T$ ,

- $E_{f_H}$ : The true risk of the best function in  $H$ , and
- $E_{f_S}$ : The empirical risk of the function in  $H$  we actually find.

Sampling error, which depicts the difference between the best function in  $H$  and the function in  $H$  we actually find can be represented as  $SE = E_{f_S} - E_{f_H}$ . The sampling errors occur because our finite sample does not give us enough information to choose the best function in  $H$ . Approximation error is the difference between the true risk in  $H$  and  $T$ , which can be represented as  $AE = E_{f_H} - E_{f_T}$ . This error occurs because  $H$  is smaller than  $T$ . Based on the relations described above, we can formulate our empirical risk as the sum of sampling error, approximation error, and the true risk in  $T$ :  $E_{f_S} = SE + AE + E_{f_T}$ . In an oracle classifier,  $AE$  is always “in some sense” equal to zero because the target space and the hypothesis space are the same. In any other classifier, however,  $AE$  can be greater than or equal to zero. Hence, the variance of the error difference in an oracle classifier should always be less than the corresponding variance in any other classifier.

Incorporating CV in a tree-based model yields a classifier, called a cross-validated tree classifier (CVT). Let  $e_{2,A}$  denote the training error based on CV, which is the error rate by applying CVT to the training data (Equation (7).) Let  $e_{2,B}$  denote the testing error when the CVT is applied to the testing data (Equation (8).)

$$e_{2,A} = \frac{1}{N} \sum_{Y_A} I(\hat{Y}_{CVT(X_A)}, Y_A), \quad (7)$$

$$e_{2,B} = \frac{1}{N} \sum_{Y_B} I(\hat{Y}_{CVT(X_A)}, Y_B). \quad (8)$$

A quantity similar to the one in Equation (6) is

$$D_2 = e_{2,B} - e_{2,A}. \quad (9)$$

One can still argue that the variance  $D_2$  can be decomposed into two components which are sampling and approximate errors described above. We have the following observation.

**Observation 4.3** *For error difference  $D_2$  that is described in (9), we have*

$$D_2 \sim \mathcal{N}(0, \sigma_{D_2}^2).$$

The distribution of  $e_{2,A}$ ,  $e_{2,B}$ , and  $D_2$  cannot be derived directly because of the correlation between iterations of a CV procedure. Since distributions of each iteration in CV (e.g., 10 iterations in 10-fold CV) are correlated: one can not simply apply asymptotic approximation in this case. Instead of a theoretical proof, we analyze their distributions empirically.

Figure 4 illustrates the normal probability plots of three types of errors, i.e.,  $e_{2,A}$ ,  $e_{2,B}$ , and  $D_2$ . The errors are generated under the rectangular decision boundary with parameter  $p = 0.1$  and sample size 800: training 400, testing 400 (more details are provided in the next section.) It suggests that all three quantities follows Normal distributions with corresponding means and variances.

Now we consider a statistical relation between  $D_1$  and  $D_2$ , using a linear regression method. Our main observation is the following.

**Observation 4.4** *Given the error differences that are defined in (6) and (9), we have*

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

where  $C$  is a constant and its range should be between 0 and 1, and  $\varepsilon$  is a random error satisfying  $\mathcal{N}(0, \sigma^2)$ .

In this paper we perform simulations to justify the observations. The interpretation of such a result is that the equality of errors between testing and training set from a CVT is comparable with that of an OC up to a constant  $C$ .

## 5. Simulations

### 5.1 Setup

We consider three different decision boundaries (denoted by  $\mathcal{B}$ ) inside a unit square (denoted by  $\mathcal{S}$ ) and a underlying rule  $F$ .

#### Decision Boundaries, $\mathcal{B}$

- *Case 1:*  $X \in \mathcal{B}$  where  $\mathcal{B}$  is a rectangular decision boundary, which is  $0.2 \leq X_1 \leq 0.8$  and  $0.3 \leq X_2 \leq 0.8$ .
- *Case 2:*  $X \in \mathcal{B}$  where  $\mathcal{B}$  is a circular decision boundary, which is  $(x_1 - 0.5)^2 + (x_2 - 0.5)^2 < 0.2^2$ .

- *Case 3:*  $X \in \mathcal{B}$  where  $\mathcal{B}$  is a triangular decision boundary, which is  $X_2 > 0.2$ ,  $X_2 < 2X_1 - 0.2$ , and  $x_2 < -2X_1 + 1.8$ .

**Rule,  $F(\mathcal{B}, p)$**

- If input  $X \in \mathcal{B}$ , then response

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

- If input  $X \notin \mathcal{B}$ , then response

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

Figure 5 illustrates the three different decision boundaries and data points generated by the underlying rule. Two hundred simulated data (training:100, testing:100) with  $p = 0.1$  is utilized. For each randomly generated data, based on a underlying rule  $F$ , we can obtain a series of error rates ( $e_{1,A}$ ,  $e_{2,A}$ ,  $e_{1,B}$ , and  $e_{2,B}$ ) defined in the previous section. Note that we employ 10-fold CV to compute  $e_{2,A}$  and  $e_{2,B}$ .

## 5.2 Relation Between $D_1$ and $D_2$

To identify a statistical relation between  $D_1$  and  $D_2$ , we utilize linear regression analysis.  $D_1$  is taken as the response variable and  $D_2$  as the predictor variable.

Figure 6 represents linear regression between  $D_1$  and  $D_2$  with three different decision boundaries. We consider  $p = 0.1$  and the 200 sample size. It suggests that there is a statistical relation between two variables. Slopes and intercepts of regression lines are shown in Tables 1 and 2. Note that the number of experiments do not significantly affect the slope. Moreover, Table 2 shows that intercepts are not significant in most cases (see  $p$ -value). Thus, each case of the regression function can be presented as follows using the average of the slopes (indicated in the 9th column of Table 1),

$$E_\alpha\{D_1\} = 0.766 \cdot D_2,$$

$$E_\beta\{D_1\} = 0.511 \cdot D_2,$$

$$E_\gamma\{D_1\} = 0.459 \cdot D_2,$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  respectively represent rectangular, circular, and triangular decision boundaries.

The slope parameter 0.766 in a rectangular decision boundary indicates that the expected difference

between testing and training error from an OC is 0.766 times that of a CVT. Similar interpretations can be made with respect to the circular and triangular decision boundaries.

Table 1: Slopes in a regression line with differently shaped decision boundaries and number of experiments. The values in the parentheses indicate the slopes in a regression line through the origin

# of exp.	20	50	100	200	300	400	500	Mean	Stdev
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)	0.759 (0.766)	0.066 (0.048)
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)	0.522 (0.511)	0.017 (0.020)
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)	0.466 (0.459)	0.054 (0.053)

Table 2: Intercepts in regression lines and their significance with different decision boundaries and the number of experiments. The values in the parentheses indicate the  $p$ -values of intercepts

# of exp.	20	50	100	200	300	400	500
Rectangle	0.00233 (0.682)	0.0094 (0.624)	0.00094 (0.856)	-0.00174 (0.108)	0.00121 (0.802)	-0.000586 (0.462)	0.00044 (0.573)
Circle	0.00230 (0.682)	0.00218 (0.551)	0.00534 (0.011)	0.00084 (0.656)	0.00162 (0.195)	0.00420 (0)	0.00224 (0.024)
Triangle	0.00705 (0.164)	0.00478 (0.254)	0.00446 (0.064)	-0.00009 (0.959)	0.00114 (0.410)	-0.00018 (0.534)	0.00108 (0.644)

### 5.3 Effects of the Geometry of Decision Boundaries

Figure 8 shows that the relationship of the difference between testing and training error of the OC and CVT is affected by the geometry of the decision boundaries. The larger value of the slope, the less difference between  $D_1$  and  $D_2$ . It is not hard to imagine why a rectangular decision boundary has a larger value of the slope than other boundaries. This is due to the characteristic of the recursively binary splitting of the feature space in tree-based methods. Furthermore, Table 3 shows the  $R^2$  (coefficient of determination) of each boundary. It also shows that the rectangular boundary has larger  $R^2$  than the others. This result suggests that a strong degree of linear association between  $D_1$  and  $D_2$  exists within a rectangular decision boundary. In other words, a cross-validated tree classifier based on rectangular decision boundary behaves more like the oracle classifier than for the other geometries. Note that in Tables 1 and 3 the number of experiments do not significantly affect the slope and  $R^2$ .

Table 3:  $R^2$  (Coefficient of Determination) with different decision boundaries and number of experiments

# of exp.	20	50	100	200	300	400	500	Mean	Stdev
Rectangle	0.878	0.705	0.733	0.718	0.643	0.729	0.674	0.725	0.074
Circle	0.339	0.441	0.459	0.430	0.456	0.431	0.467	0.432	0.043
Triangle	0.449	0.282	0.316	0.320	0.370	0.388	0.414	0.362	0.059

#### 5.4 The Effect of the Parameters in an Underlying Distribution

Recall that an underlying distribution in our simulation is Bernoulli and its parameter is the probability of any particular points being inside the decision boundary. Table 4 describes the slopes of regression lines with different parameters based on a rectangular decision boundary. The other geometries of decision boundaries give similar results. Figure 10 is the box plot of the slopes in different parameter values. It shows that parameter values between 0.1 and 0.2 produce a strong linear relationship between the OC and the CVT but this relationship becomes weaker as the parameter value becomes either extremely small or close to 0.5. It's 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 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 produce very small error rates. This high sensitivity results in a relationship between the two classifiers that is relatively weak.

Table 5 and Figure 11 show the  $R^2$  for the above regression analysis. These results show that when the slope is large, there is a stronger case for the existence of a regression model.

#### 5.5 The Effect of the Sample Size

In this section we study the relationship of the equality between testing and training errors from both classifiers with different sample sizes. First we consider five different sample sizes (training + 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 6 shows the slopes in a regression line

Table 4: Slopes in a regression line with different parameters. The values in the parentheses are the slopes in a regression line through the origin

	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)	
S.D.	0.042 (0.041)	0.087 (0.052)	0.062 (0.062)	0.083 (0.073)	0.126 (0.133)	0.080 (0.101)	

Table 5:  $R^2$  in a regression line with different parameters

	0.01	0.05	0.1	0.2	0.3	0.4	0.5
1	0.206	0.609	0.733	0.627	0.443	0.165	0.008
2	0.390	0.569	0.804	0.706	0.218	0.048	0.006
3	0.241	0.330	0.743	0.673	0.424	0.112	0.024
4	0.309	0.520	0.741	0.529	0.229	0.014	0.012
5	0.342	0.684	0.637	0.716	0.362	0.125	0.006
Average	0.307	0.595	0.770	0.758	0.575	0.093	0.011
S.D.	0.042	0.087	0.062	0.083	0.126	0.054	0.007

from different ratios of the training and the testing sample sizes. Again, since the intercepts in a regression line are not statistically significant, we consider the slopes with zero intercept shown in the parentheses in Table 6. Figure 12 illustrates a three-dimensional contour plot. The x and y-axes respectively represent the sample size and the ratio of testing to training samples. For instance, if the values on the x and y-axes are 300 and 2, the experiment has a training sample size of 100 and 200 for the testing sample. The z-axis (the values on the contour plot) indicates the slopes of each regression line. This plot provides a nice guideline for determining the ratio of testing to the training sample size for achieving the targeted performance. For instance, if we want our cross-validated tree classifier to be  $\frac{1}{0.84677}$  of the oracle classifier, the corresponding values on the x-axis give the ratio corresponding to the different total sample sizes. In addition, we observe

that the slopes change a lot when the size of sample is less than 300 but stabilize when sample size becomes larger than 300. This implies that in this example, the sample size 300 is sufficient for good performance of the tree classifier compared to the oracle classifiers with respect to testing and training error.

Table 6: Slopes in a regression line with different sizes and ratio of training and testing sets. The values in the parentheses indicate the slopes in a regression line through origin

	100	200	300	400	500	Average	S.D.
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)		
S.D.	0.156 (0.161)	0.138 (0.141)	0.080 (0.078)	0.113 (0.118)	0.0986 (0.098)		

## 6. Conclusions

We present an experimental way to measure the performance of CV in tree-based models. We compares a CVT with an OC based on the knowledge of an underlying distribution. Main observation provides the linear statistical relationship of the difference between testing and training errors from two CVT and OC. Simulation results indicate that the intercept of the regression line is zero. These results suggest that the difference between testing and training errors from a CVT is a constant factor of that of an oracle classifier. Various simulations appear to justify the authors' observations. Regression slopes and  $R^2$  are employed to measure of the degree of the relationship. Both the slope and  $R^2$  being equal to 1 suggest a strong relationship between two classifiers. Additionally, we demonstrate that the above relationship is influenced by other factors such as the geometry of the decision boundaries, the probabilistic parameter of an underlying distribution, and sample size. There are two interesting directions for future research. On the theoretical front, all

the observations are waiting to be proven. For more numerical studies, one can extend our study to other machine learning algorithms, such as support vector machines, neural networks, and so on. Also our study can be used to identify a statistical relationship between the sizes of training and testing samples. This relationship can surely help improve classification accuracy.

## **Acknowledgement**

## References

- [1] Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth International Group.
- [2] Breiman, L., & Spector, P. (1992). Submodel selection and evaluation in regression: the X-random case. *International Statistical Review*, 60(3), 291–319.
- [3] Devroye, L.P., & Wagner, T.J. (1979a). Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions on Information Theory*, 25(2), 202–207.
- [4] Devroye, L.P., & Wagner, T.J. (1979b). Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5), 601–604.
- [5] Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement of cross-validation. *Journal of the American Statistical Association*, 78(382), 316–331.
- [6] Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81(394), 461–470.
- [7] Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer.
- [8] Holden, S. B. (1996). Proceedings of the Ninth Annual Conference on Computational Learning Theory (pp.41 – 50), New York.
- [9] Huo, X., Kim, S.B., Tsui, K- L., & Wang, S. (2004). A frontier-based tree-pruning algorithm (FBP). *INFORMS Journal on Computing*, Accepted.
- [10] Kearns, M. & Ron, D. (1999). Algorithmic stability and sanity check bounds for leave-one-out cross validation bounds. *Neural Computation*, 11(6), 1427–1453.
- [11] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *International Joint Conference of Artificial Intelligence (IJCAI)*, 1137–1145.
- [12] Lachenbruch, P.A., & Mickey M. R. (1968). Estimation of Error Rates in Discriminant Analysis. *Technometrics*, 10(1), 1–11.

- [13] Racine, J. 2000. Consistent cross-validators for dependent data: hv-block cross-validation. *Journal of Econometrics*, 99(1): 39–61.
- [14] Rogers, W.H., & Wagner, T.J. (1978). A finite sample distribution-free performance bound for local discrimination rule. *Annals of Statistics*, 6, 506–514.
- [15] Shao, J. (1993). Linear-model selection by cross-validation. *Journal of the American Statistical Association*, 88(422), 486–494.
- [16] Shao, J. (1996). Bootstrap model selection. *Journal of the American Statistical Association*, 91(434), 655–665.
- [17] Shao, J. (1998). Convergence rates of the generalization information criterion. *Journal of Nonparametric Statistics*, 9(3), 217–225.
- [18] Stone, M. (1974). Cross-validators choice and assessment of statistical prediction. *Journal of the Royal Statistical Society, Ser.B(36)*, 111–133.
- [19] Vehtari, A., & Lampinen, J. (2002). Bayesian model assessment and comparison using cross-validation predictive densities. *Neural Computation*, 14(10), 2439–2468.
- [20] Zhang, P. (1992). On the distributional properties of model selection criteria. *Journal of the American Statistical Association*, 87(419), 732–737.
- [21] Zhang, P. (1993a). Model selection via multifold cross validation. *The Annals of Statistics*, 21(1), 299–313.
- [22] Zhang, P. (1993b). On the convergence rate of model selection criteria. *Communications in statistics: Theory and Methods*, 22(10), 2765–2775.

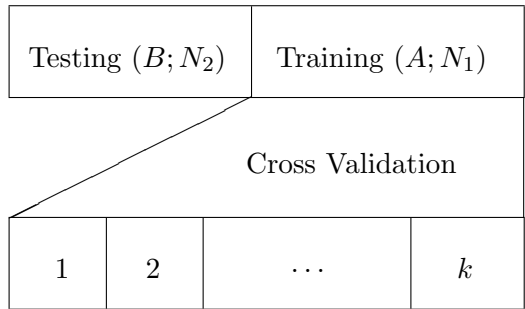


Figure 1: A structure of the CV process.

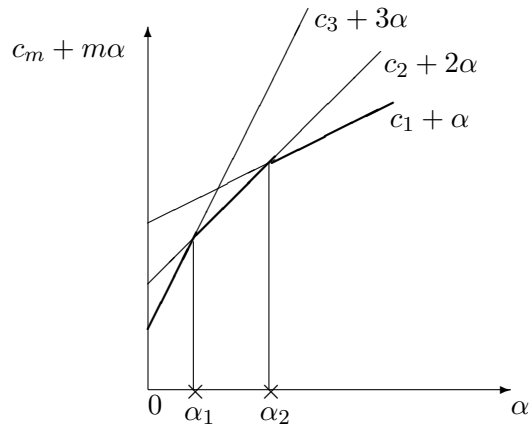


Figure 2: An illustration of the frontier-based tree-pruning algorithm.

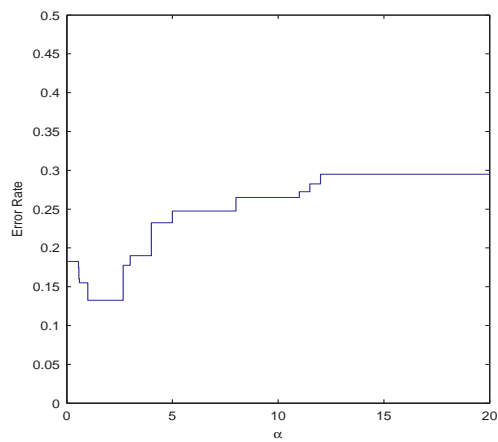


Figure 3: The range of the optimal  $\alpha$  that produces the smallest error rate.

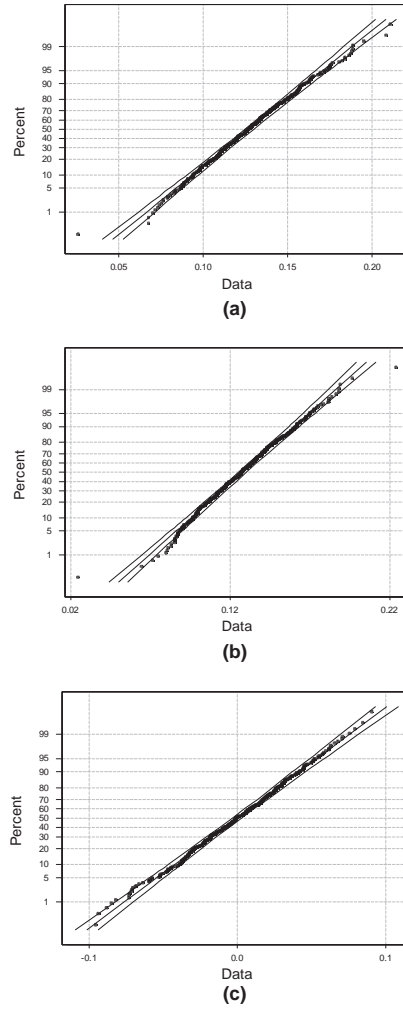


Figure 4: Normal probability plot for the errors in a CVT. (a) training error for the CVT ( $e_{2,A} \sim \mathcal{N}(0.127, 0.03)$ ), (b) testing error for the CVT ( $e_{2,B} \sim \mathcal{N}(0.127, 0.03^2)$ ), and (c) error for the difference ( $D_2 = e_{2,B} - e_{2,A} \sim \mathcal{N}(0.000, 0.03^2)$ ).

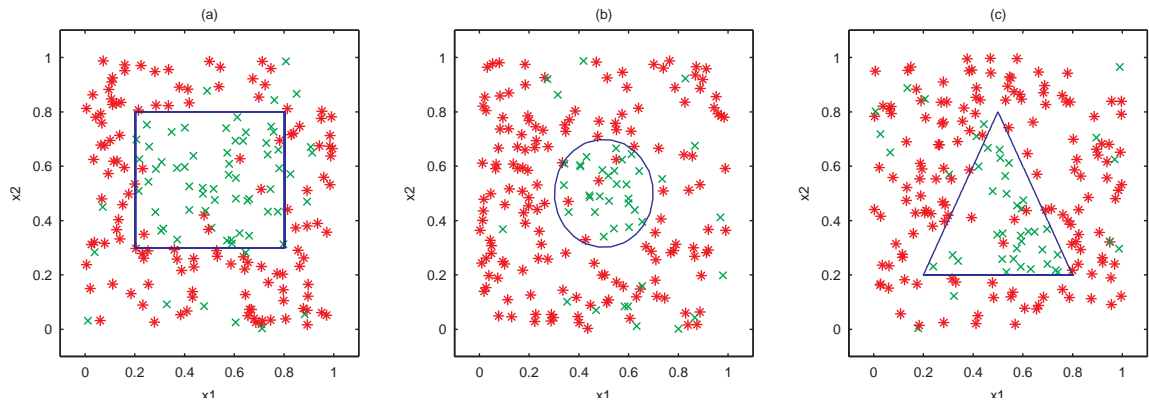


Figure 5: Illustration of 200 simulated data sets with three different decision boundaries. (a) Rectangular decision boundary, (b) Circular decision boundary, and (c) Triangular decision boundary.

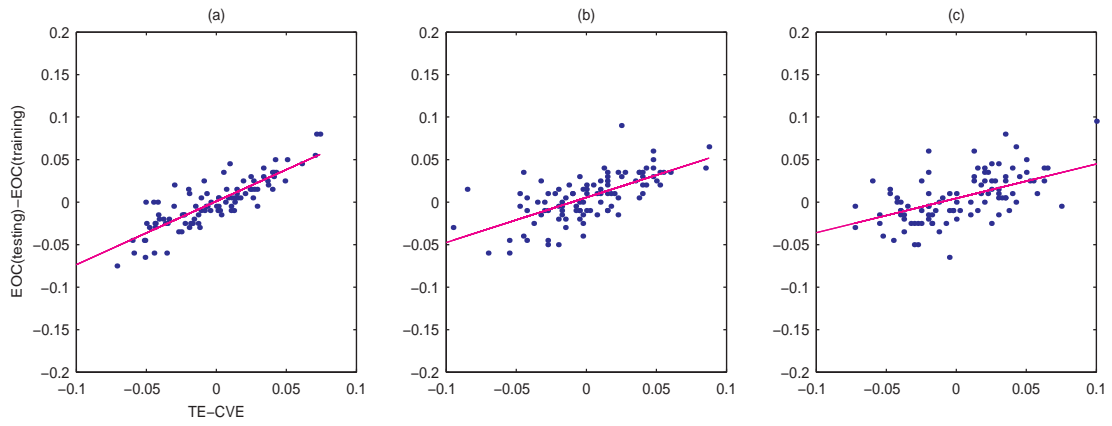


Figure 6: Regression plots between  $D_2$  and  $D_1$ . (a) Rectangular decision boundary, (b) Circular decision boundary, and (c) Triangular decision boundary.

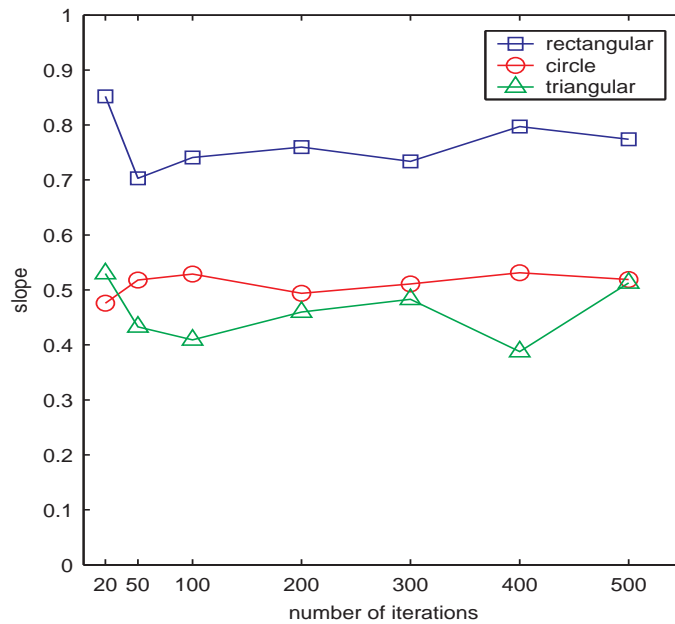


Figure 7: Slopes in a regression line with different decision boundaries and sample sizes.

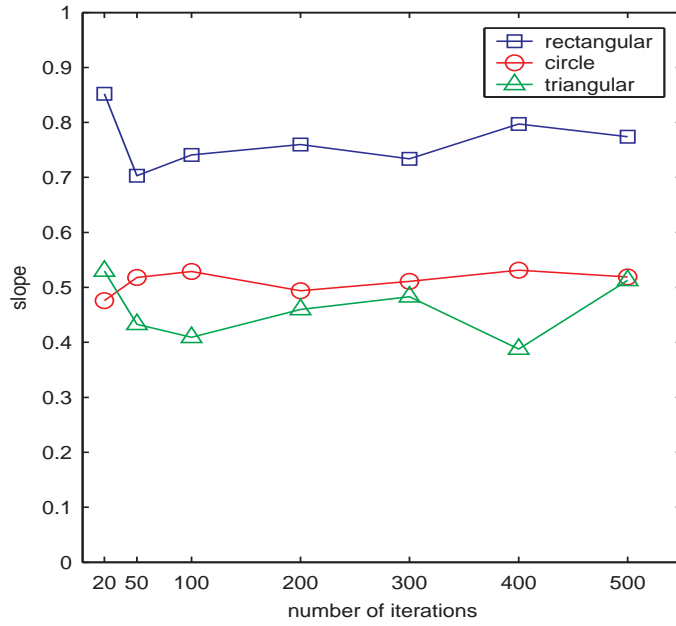


Figure 8: Slopes in a regression line with different decision boundaries and sample sizes.

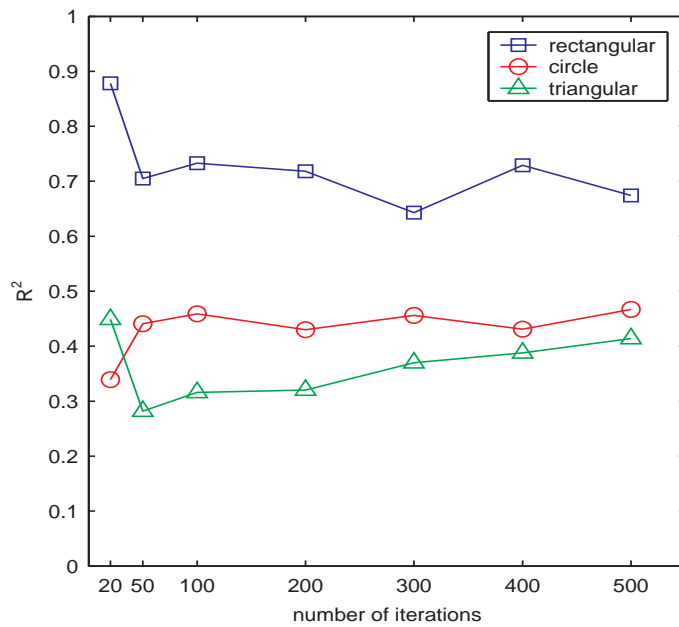


Figure 9:  $R^2$  (Coefficient of Determination) in a regression line with different decision boundaries and number of experiments.

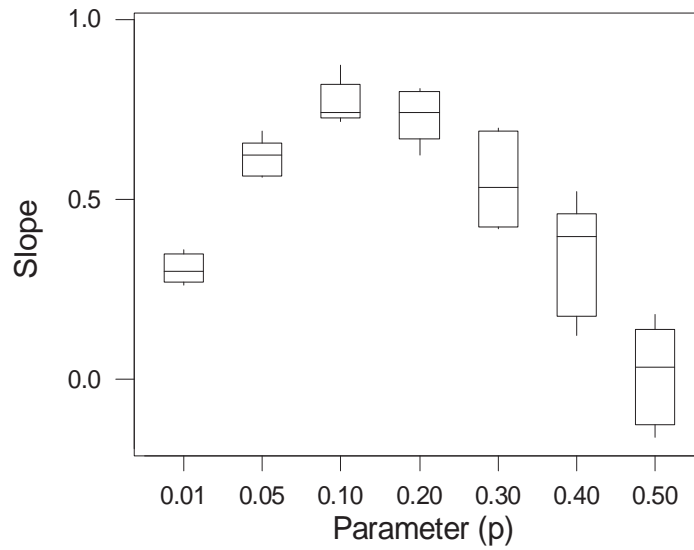


Figure 10: Slopes in a regression line with different parameters.

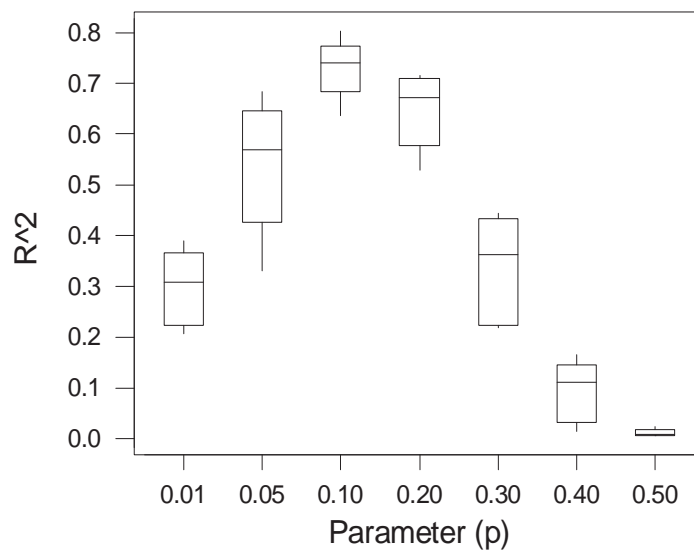


Figure 11:  $R^2$  in a regression line with different parameters.

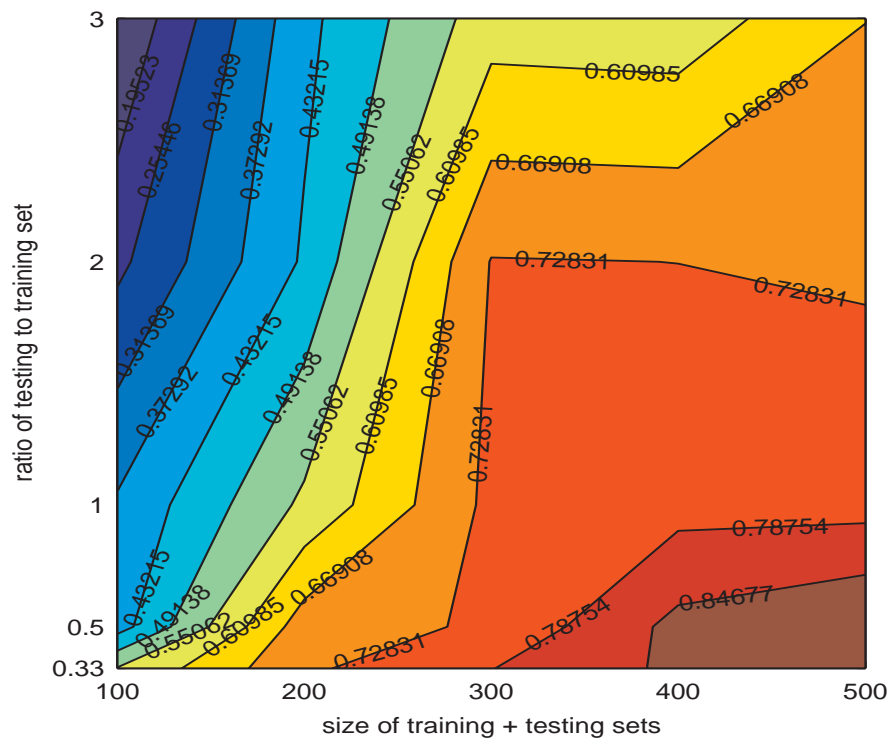


Figure 12: Contour plot of slopes with different sizes and ratios of training and testing set.