Journal of Machine Learning Research 9 (2008) 2321-2348 Submitted 6/07; Revised 1/08; Published 10/08

Probabilistic Characterization of Random Decision Trees

Amit Dhurandhar ASD@CISE.UFL.EDU
Alin Dobra ADOBRA@CISE.UFL.EDU
Computer and Information Science and Engineering
University of Florida
Gainesville, FL 32611, USA

Editor: Greg Ridgeway

Abstract

In this paper we use the methodology introduced by Dhurandhar and Dobra (2009) for analyzing the error of classifiers and the model selection measures, to analyze decision tree algorithms. The methodology consists of obtaining parametric expressions for the moments of the generalization error (GE) for the classification model of interest, followed by plotting these expressions for interpretability. The major challenge in applying the methodology to decision trees, the main theme of this work, is customizing the generic expressions for the moments of GE to this particular classification algorithm. The specific contributions we make in this paper are: (a) we primarily characterize a subclass of decision trees namely, Random decision trees, (b) we discuss how the analysis extends to other decision tree algorithms and (c) in order to extend the analysis to certain model selection measures, we generalize the relationships between the moments of GE and moments of the model selection measures given in (Dhurandhar and Dobra, 2009) to randomized classification algorithms. An empirical comparison of the proposed method with Monte Carlo and distribution free bounds obtained using Breiman's formula, depicts the advantages of the method in terms of running time and accuracy. It thus showcases the use of the deployed methodology as an exploratory tool to study learning algorithms.

Keywords: moments, generalization error, decision trees

1. Introduction

Consider the problem of estimating how a given classification algorithm (rather than a particular classifier) performs on a given joint distribution over the input-output space ($X \times Y$). As opposed to the general setup in machine learning where the distribution is unknown and only independent and identically distributed (i.i.d.) samples are available, in this scenario, *in principle*, the behavior of classification algorithm can be accurately studied. If this problem be solved efficiently, it offers an alternative line of study for classification algorithms and potentially unique insights into the *non-asymptotic* behavior of learning algorithms.

While the problem of estimating classification algorithm performance on a given distribution might look simple, solving it efficiently poses significant technical hurdles. The most natural way of studying a classification algorithm would be to sample N datapoints from the given distribution, train the algorithm to produce a classifier, test the classifier on a few sampled test sets and report the average error computed over these test sets. A shortcoming of the above approach is that based on just one single instance of the algorithm (since the algorithm was trained on a single data set of size N) we conclude about its general behavior. A straightforward extension of the above approach to make the results more relevant in studying the algorithm would be to sample multiple data sets of size N, train on each of them to produce different classifiers, compute the test error for each of the classifiers and calculate the average and variance of the obtained test errors. This procedure would be a better indicator of the behavior of the algorithm than the previous case since we study multiple instances of the algorithm than just an isolated instance. Ideally, we would want to study the behavior of the algorithm by training it on all possible data sets of size N producing a variety of classifiers and then evaluating the expected value and variance of the generalization error (GE) of each of these classifiers. The GE of a classifier ζ is given by,

$$\begin{aligned} GE(\zeta) &= E\left[\lambda(\zeta(x),y)\right] \\ &= P\left[\zeta(x) \neq y\right] \end{aligned}$$

where $\lambda(.,.)$ is a 0-1 loss function, x is an input and y is an output and the expectation is over the input-output space $X \times Y$. The expected value and variance of GE over all possible classifiers[1] are denoted by,

$$E_{\mathcal{Z}(N)}\left[GE(\zeta)\right],$$

$$Var(GE(\zeta)) = E_{\mathcal{Z}(N)\times\mathcal{Z}(N)}\left[GE(\zeta)GE(\zeta')\right] - E_{\mathcal{Z}(N)}\left[GE(\zeta)\right]^2$$

where $\mathcal{Z}(N)$ represents the space of all possible classifiers produced by training the classification algorithm on all data sets of size N (denoted by $D(N)$), drawn from the joint distribution. With this we have shown that the moments provide a natural and informative avenue for studying classification algorithms. The question that now arises is, can we compute them efficiently. In our previous work (Dhurandhar and Dobra, 2009), we presented a general framework for computing these quantities for an arbitrary classification algorithm efficiently. By extensive use of the linearity of expectation and change of the order of sums (and integrals), the moments of GE can be expressed in terms of the behavior of the classification algorithm on specific inputs rather than on the whole space, thus reducing the complexity from an exponential in the size of the input space to linear for the computation of the first moment and quadratic for the second moment. As part of this prior work, the generic expressions to compute the moments were customized for the Naive Bayes Classification algorithm. In the present work we customize the generic expressions to compute moments of the generalization error for a more popular classification algorithm: Random decision trees.

The specific contributions we make are: We develop a characterization for a subclass of decision trees. In particular, we characterize Random decision trees which are an interesting variant with respect to three popular stopping criteria namely; fixed height, purity and scarcity (i.e., fewer than some threshold number of points in a portion of the tree). The analysis directly applies to categorical as well as continuous attributes with split points predetermined for each attribute. Moreover, the analysis in Section 3.3 is applicable to even other deterministic attribute selection methods based on information gain, gini gain etc. These and other extensions of the analysis to continuous attributes with dynamically chosen split points is discussed in Section 5. In the experiments that ensue the theory, we compare the accuracy of the derived expressions with direct Monte Carlo (i.e., hold-out-set estimation) and Breiman's strength and correlation based bounds (Breiman, 2001) on synthetic

[1] Expectations over $\mathcal{Z}(N)$ are more general than over $D(N)$ since the classification algorithm can be randomized.

distributions as well as on distributions built on real data. Notice that using the expressions, the moments can be computed without explicitly building the tree. We also extend the relationships between the moments of GE and moments of cross validation error (CE), leave-one-out error (LE) and hold-out-set error (HE) given in Dhurandhar and Dobra (2009) which were applicable only to deterministic classification algorithms, to be made applicable to randomized classification algorithms.

2. Preliminaries

Model selection for classification is one of the major challenges in Machine Learning and Datamining. Given an i.i.d. sample from the underlying probability distribution, the classification model selection problem consists in building a classifier by selecting among competing models. Ideally the model selected minimizes GE. Since GE cannot be directly computed, part of the sample is used to estimate GE through measures such as cross validation, hold-out-set, leave-one-out, etc. Though certain rules of thumb are followed by practitioners w.r.t. training size and other parameters specific to the validation measures in evaluating models through empirical studies (Kohavi, 1995; Blum et al., 1999) and certain asymptotic results exist (Vapnik, 1998; Shao, 1993), the fact remains that most of these models and model selection measures are not well understood in real life (non-asymptotic) scenarios (e.g., what fraction should be test and training, what should be the value k in k-fold cross validation etc.). This lack of deep understanding limits our ability of using the models most effectively and maybe more importantly trusting the models to perform well in a particular application.

Recently, a novel methodology was proposed in Dhurandhar and Dobra (2009) to study the behavior of models and model selection measures. Since the methodology is at the core of the current work, we briefly describe it together with the motivation for using this type of analysis for classification in general and decision trees in particular.

2.1 *What* is the Methodology?

The methodology for studying classification models consists of studying the behavior of the first two central moments of the GE of the classification algorithm studied. The moments are taken over the space of all possible classifiers produced by the classification algorithm, by training it over all possible data sets sampled i.i.d. from some distribution. The first two moments give enough information about the statistical behavior of the classification algorithm to allow interesting observations about the behavior/trends of the classification algorithm w.r.t. any chosen data distribution.

2.2 *Why* have such a Methodology?

The answers to the following questions shed light on why the methodology is necessary if tight statistical characterization is to be provided for classification algorithms.

1. *Why study GE ?* The biggest danger of learning is *overfitting* the training data. The main idea in using GE as a measure of success of learning instead on the empirical error on a given data set is to provide a mechanism to avoid this pitfall. Implicitly, by analyzing GE all the input is considered.

2. *Why study the moments instead of the distribution of GE ?* Ideally, we would study the distribution of GE instead of moments in order to get a complete picture of what is its behavior.

Studying the distribution of discrete random variables, except for very simple cases, turns out to be very hard. The difficulty comes from the fact that even computing the probability of a single point is intractable since all combinations of random choices that result in the same value for GE have to be enumerated. On the other hand, the first two central moments coupled with distribution independent bounds such as Chebychev and Chernoff give guarantees about the worst possible behavior that are not too far from the actual behavior (small constant factor). Interestingly, it is possible to compute the moments of a random variable like GE without ever explicitly writing or making use of the formula for the cumulative distribution function. What makes such an endeavor possible is extensive use of the linearity of expectation.

3. *Why characterize a class of classifiers instead of a single classifier ?* While the use of GE as the success measure is standard practice in Machine Learning, characterizing classes of classifiers instead of the particular classifier produced on a given data set is not. From the point of view of the analysis, without large testing data sets it is not possible to evaluate directly GE for a particular classifier. By considering classes of classifiers to which a classifier belongs, an indirect characterization is obtained for the particular classifier. This is precisely what Statistical Learning Theory (SLT) does; there the class of classifiers consists in all classifiers with the same VC dimension. The main problem with SLT results is that classes based on VC dimension are too large, thus results tend to be pessimistic. In our methodology, the class of classifiers consists only of the classifiers that are produced by the given classification algorithm from data sets of fixed size from the underlying distribution. This is the probabilistic smallest class in which the particular classifier produced on a given data set can be placed in.

2.3 *How* do we Implement the Methodology ?

One way of approximately estimating the moments of GE over all possible classifiers for a particular classification algorithm is by directly using Monte Carlo. If we use Monte Carlo directly, we first need to produce a classifier on a sampled data set then test on a number of test sets sampled from the same distribution acquiring an estimate of the GE of this classifier. Repeating this entire procedure a few times we would acquire estimates of GE for different classifiers. Then by averaging the error of these multiple classifiers we would get an estimate of the first moment of GE. The variance of GE can also be similarly estimated. The problem with this procedure is that the space of all possible data sets can be huge. For instance, if we have d attributes each taking m values then the number of possible data sets of size N is $N^{m^d}-1$. Even for any reasonable assignment to N (say, 100), m (say 2) and d (say 3) the number of experiments that need to be performed to guarantee accurate (if not exact) estimation of the moments seems unreasonable.

Another way of estimating the moments of GE, is by obtaining parametric expressions for them. If this can be accomplished the moments can be computed exactly. Moreover, by dexterously observing the manner in which expressions are derived for a particular classification algorithm, insights can be gained into analyzing other algorithms of interest. Though deriving the expressions may be a tedious task, using them we obtain highly accurate estimates of the moments. In this paper, we propose this second alternative for analyzing a subclass of decision trees. The key to the analysis is focusing on the learning phase of the algorithm. In cases where the parametric expressions are computationally intensive to compute directly, we show that approximating individual terms using Monte Carlo we obtain accurate estimates of the moments when compared to directly using Monte Carlo (first alternative) for the same computational cost.

If the moments are to be studied on synthetic data then the distribution is anyway assumed and the parametric expressions can be directly used. If we have real data an empirical distribution can be built on the data set and then the parametric expressions can be used.

2.4 Applications of the Methodology

It is important to note that the methodology is not aimed towards providing a way of estimating bounds for GE of a classifier on a given data set (i.e., finding distribution free bounds). The primary goal is creating an avenue in which learning algorithms can be studied precisely, that is, studying the statistical behavior of a particular algorithm w.r.t. a chosen/built distribution. Below, we discuss the two most important perspectives in which the methodology can be applied.

2.4.1 Algorithmic Perspective

If a researcher/practitioner designs a new classification algorithm, he/she needs to validate it. Standard practice is to validate the algorithm on a relatively small (5-20) number of data sets and to report the performance. By observing the behavior of only a few instances of the algorithm the designer infers its quality. Moreover, if the algorithm under performs on some data sets, it can be sometimes difficult to pinpoint the precise reason for its failure. If instead he/she is able to derive parametric expressions for the moments of GE, the test results would be more relevant to the particular classification algorithm, since the moments are over all possible data sets of a particular size drawn i.i.d. from some chosen/built distribution. Testing individually on all these data sets is an impossible task. Thus, by computing the moments using the parametric expressions the algorithm would be tested on a plethora of data sets with the results being highly accurate. Moreover, since the testing is done in a controlled environment, that is, all the parameters are known to the designer while testing, he/she can precisely pinpoint the conditions under which the algorithm performs well and the conditions under which the algorithm under performs.

2.4.2 Data Set Perspective

If an algorithm designer validates his/her algorithm by computing moments as mentioned earlier, it can instill greater confidence in the practitioner searching for an appropriate algorithm for his/her data set. The reason for this being, if the practitioner has a data set which has a similar structure or is from a similar source as the test data set on which an empirical distribution was built and favorable results reported by the designer, then this would mean that the results apply not only to that particular test data set, but to other similar type of data sets and since the practitioner's data set belongs to this similar collection, the results would also apply to his. Note that a distribution is just a weighting of different data sets and this perspective is used in the above exposition.

3. Computing Moments

In this section we first provide the necessary technical groundwork, followed by customization of the expressions for decision trees. We now introduce some notation that is used primarily in this section. X is a random vector modeling input whose domain is denoted by $\mathcal{X}$. Y is a random variable modeling output whose domain is denoted by $\mathcal{Y}$ (set of class labels). $Y(x)$ is a random variable modeling output for input x. ζ represents a particular classifier with its GE denoted by

$GE(\zeta)$. $\mathcal{Z}(N)$ denotes a set of classifiers obtained by application of a classification algorithm to different samples of size N.

3.1 Technical Framework

The basic idea in the generic characterization of the moments of GE, is to define a class of classifiers induced by a classification algorithm and an i.i.d. sample of a particular size from an underlying distribution. Each classifier in this class and its GE act as random variables, since the process of obtaining the sample is randomized. Since $GE(\zeta)$ is a random variable, it has a distribution. Quite often though, characterizing a finite subset of moments turns out to be a more viable option than characterizing the entire distribution. Based on these facts, we revisit the expressions for the first two moments around zero of the GE of a classifier,

$$\begin{aligned}&E_{\mathcal{Z}(N)}\left[GE(\zeta)\right] = \\ &\quad \sum_{x\in\mathcal{X}} P\left[X=x\right] \sum_{y\in\mathcal{Y}} P_{\mathcal{Z}(N)}\left[\zeta(x)=y\right] P\left[Y(x)\neq y\right],\end{aligned}$$

$$\begin{aligned}&E_{\mathcal{Z}(N)\times\mathcal{Z}(N)}\left[GE(\zeta)GE(\zeta')\right] = \\ &\quad \sum_{x\in\mathcal{X}}\sum_{x'\in\mathcal{X}} P\left[X=x\right] P\left[X=x'\right] \cdot \\ &\qquad \sum_{y\in\mathcal{Y}}\sum_{y'\in\mathcal{Y}} P_{\mathcal{Z}(N)\times\mathcal{Z}(N)}\left[\zeta(x)=y \wedge \zeta'(x')=y'\right] \cdot \\ &\qquad\quad P\left[Y(x)\neq y\right] P\left[Y(x')\neq y'\right]\end{aligned}$$

From the above equations we observe that for the first moment we have to characterize the behavior of the classifier on each input separately while for the second moment we need to observe its behavior on pairs of inputs. In particular, to derive expressions for the moments of any classification algorithm we need to characterize $P_{\mathcal{Z}(N)}\left[\zeta(x)=y\right]$ for the first moment and $P_{\mathcal{Z}(N)\times\mathcal{Z}(N)}\left[\zeta(x)=y\wedge\zeta'(x')=y'\right]$ for the second moment.[2] The values for the other terms denote the error of the classifier for the first moment and errors of two classifiers for the second moment which are obtained directly from the underlying joint distribution. For example, if we have data with a class prior p for class 1 and *1-p* for class 2. Then the error of a classifier classifying data into class 1 is *1-p* and the error of a classifier classifying data into class 2 is given by p. We now focus our attention on relating the above two probabilities, to probabilities that can be computed using the joint distribution and the classification model viz. Decision Trees.

In the subsections that follow we assume the following setup. We consider the dimensionality of the input space to be d. $A_1, A_2, ..., A_d$ are the corresponding discrete attributes or continuous attributes with predetermined split points. $a_1, a_2, ..., a_d$ are the number of attribute values/the number of splits of the attributes $A_1, A_2, ..., A_d$ respectively. m_{ij} is the i^{th} attribute value/split of the j^{th} attribute, where $i \leq a_j$ and $j \leq d$. Let $C_1, C_2, ..., C_k$ be the class labels representing k classes and N the sample size.

[2] These probabilities and $P\left[Y(x)\neq y\right]$ are conditioned on x. We omit explicitly writing the conditional since it improves readability and is obvious from the context.

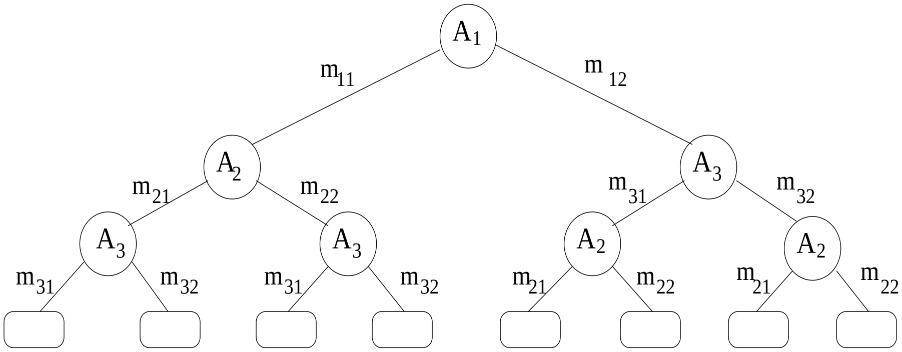

Figure 1: The all attribute tree with 3 attributes $A_1,\ A_2,\ A_3$, each having 2 values.

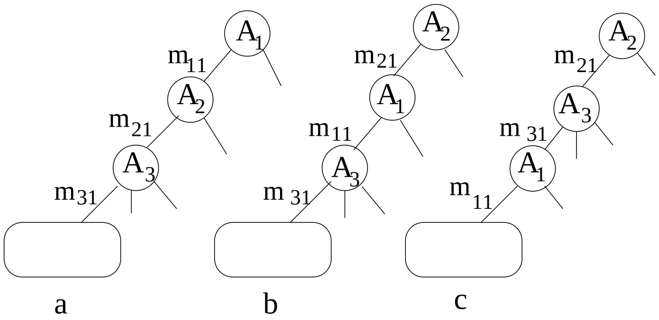

Figure 2: Given 3 attributes $A_1,\ A_2,\ A_3$, the path $m_{11}m_{21}m_{31}$ is formed irrespective of the ordering of the attributes. Three such permutations are shown in the above figure.

3.2 All Attribute Decision Trees (ATT)

Let us consider a decision tree algorithm whose only stopping criterion is that no attributes remain when building any part of the tree. In other words, every path in the tree from root to leaf has all the attributes. An example of such a tree is shown in Figure 1. It can be seen that irrespective of the split attribute selection method (e.g., information gain, gini gain, randomized selection, etc.) the above stopping criteria yields trees with the same leaf nodes. Thus although a particular path in one tree has an ordering of attributes that might be different from a corresponding path in other trees, the leaf nodes will represent the same region in space or the same set of datapoints. This is seen in Figure 2. Moreover, since predictions are made using data in the leaf nodes, any deterministic way of prediction would lead to these trees resulting in the same classifier for a given sample and thus having the same GE. Usually, prediction in the leaves is performed by choosing the most numerous class as the class label for the corresponding datapoint. With this we arrive at the expressions for computing the aforementioned probabilities,

$$\begin{aligned}&P_{\mathcal{Z}(N)}\left[\zeta(x)=C_i\right] = \\ &\quad P_{\mathcal{Z}(N)}[ct(m_{p1}m_{q2}...m_{rd}C_i) > ct(m_{p1}m_{q2}...m_{rd}C_j), \\ &\qquad \forall j \neq i,\ i,j \in [1,...,k]]\end{aligned}$$

where $x = m_{p1}m_{q2}...m_{rd}$ denotes a datapoint which is also a path from root to leaf in the tree. We refer to this path as a cell sometimes since it represents a rectangular region in a d dimensional space. $ct(m_{p1}m_{q2}...m_{rd}C_i)$ is the count of the datapoints in the cell $m_{p1}m_{q2}...m_{rd}C_i$. Henceforth, when using the word "path" we will strictly imply path from root to leaf. By computing the above probability $\forall\ i$ and $\forall\ x$ we can compute the first moment of the GE for this classification algorithm.

Similarly, for the second moment we compute cumulative joint probabilities of the following form:

$$\begin{aligned}&P_{\mathcal{Z}(N)\times\mathcal{Z}(N)}\left[\zeta(x)=C_i \wedge \zeta'(x')=C_v\right] = \\ &\quad P_{\mathcal{Z}(N)\times\mathcal{Z}(N)}[ct(m_{p1}...m_{rd}C_i) > ct(m_{p1}...m_{rd}C_j), \\ &\qquad\qquad ct(m_{f1}...m_{hd}C_v) > ct(m_{f1}...m_{hd}C_w), \\ &\qquad\qquad \forall j \neq i,\ \forall w \neq v,\ i,j,v,w \in [1,...,k]]\end{aligned}$$

where the terms have similar connotation as before. These probabilities can be computed exactly or by using fast approximation techniques proposed in Dhurandhar and Dobra (2009).

3.3 Decision Trees with Non-trivial Stopping Criteria

We just considered decision trees which are grown until all attributes are exhausted. In real life though we seldom build such trees. The main reasons for this could be any of the following: we wish to build small decision trees to save space; certain path counts (i.e., number of datapoints in the leaves) are extremely low and hence we want to avoid splitting further, as the predictions can get arbitrarily bad; we have split on a certain subset of attributes and all the datapoints in that path belong to the same class (purity based criteria); we want to grow trees to a fixed height (or depth). These stopping measures would lead to paths in the tree that contain a subset of the entire set of attributes. Thus from a classification point of view we cannot simply compare the counts in two cells as we did previously. The reason for this being that the corresponding path may not be present in the tree. Hence, we need to check that the path exists and then compare cell counts. Given the classification algorithm, since the $P_{\mathcal{Z}(N)}\left[\zeta(x)=C_i\right]$ is the probability of all possible ways in which an input x can be classified into class C_i for a decision tree it equates to finding the following kind of probability for the first moment,

$$\begin{aligned}&P_{\mathcal{Z}(N)}\left[\zeta(x)=C_i\right] = \\ &\quad \sum_p P_{\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), path_p exists, \\ &\qquad\qquad \forall j \neq i,\ i,j \in [1,...,k]]\end{aligned} \tag{1}$$

where p indexes all allowed paths by the tree algorithm in classifying input x. After the summation, the right hand side term above is the probability that the cell $path_pC_i$ has the greatest count, with the path "$path_p$" being present in the tree. This will become clearer when we discuss different stopping criteria. Notice that the characterization for the ATT is just a special case of this more generic characterization.

The probability that we need to find for the second moment is,

$$\begin{aligned}
&P_{Z(N)\times Z(N)}\left[\zeta(x)=C_i \wedge \zeta'(x')=C_v\right] = \\
&\quad \sum_{p,q} P_{Z(N)\times Z(N)}[ct(path_pC_i) > ct(path_pC_j), path_pexists, \\
&\qquad\qquad ct(path_qC_v) > ct(path_qC_w), path_qexists, \\
&\qquad\qquad \forall j \neq i,\ \forall w \neq v,\ i,j,v,w \in [1,...,k]]
\end{aligned} \tag{2}$$

where p and q index all allowed paths by the tree algorithm in classifying input x and x' respectively. The above two equations are generic in analyzing any decision tree algorithm which classifies inputs into the most numerous class in the corresponding leaf. It is not difficult to generalize it further when the decision in leaves is some other measure than majority. In that case we would just include that measure in the probability in place of the inequality.

3.3.1 CHARACTERIZING *Path Exists* FOR THREE STOPPING CRITERIA

It follows from above that to compute the moments of the GE for a decision tree algorithm we need to characterize conditions under which particular paths are present. This characterization depends on the stopping criteria and split attribute selection method in a decision tree algorithm. We now look at three popular stopping criteria, namely a) Fixed height based, b) Purity (i.e., entropy 0 or gini index 0 etc.) based and c) Scarcity (i.e., too few datapoints) based. We consider conditions under which certain paths are present for each stopping criteria. Similar conditions can be enumerated for any reasonable stopping criteria. We then choose a split attribute selection method, thereby fully characterizing the above two probabilities and hence the moments.

1. **Fixed Height:** This stopping criteria is basically that every path in the tree should be of length exactly h, where $h \in [1,...,d]$. If $h = 1$ we classify based on just one attribute. If $h = d$ then we have the all attribute tree.
   In general, a path $m_{i1}m_{j2}...m_{lh}$ is present in the tree iff the attributes A_1, $A_2,...,$ A_h are chosen in any order to form the path for a tree construction during the split attribute selection phase. Thus, for any path of length h to be present we bi-conditionally imply that the corresponding attributes are chosen.

2. **Purity:** This stopping criteria implies that we stop growing the tree from a particular split of a particular attribute if all datapoints lying in that split belong to the same class. We call such a path pure else we call it impure. In this scenario, we could have paths of length 1 to d depending on when we encounter purity (assuming all datapoints don't lie in 1 class). Thus, we have the following two separate checks for paths of length d and less than d respectively.

   a) Path $m_{i1}m_{j2}...m_{ld}$ present iff the path $m_{i1}m_{j2}...m_{l(d-1)}$ is impure and attributes A_1, $A_2,...,$ A_{d-1} are chosen above A_d, or $m_{i1}m_{j2}...m_{s(d-2)}m_{ld}$ is impure and attributes A_1, $A_2,...,$ A_{d-2}, A_d are chosen above A_{d-1}, or ... or $m_{j2}...m_{ld}$ is impure and attributes $A_2,...,$ A_d are chosen above A_1.
   This means that if a certain set of $d-1$ attributes are present in a path in the tree then we split on the d^{th} attribute iff the current path is not pure, finally resulting in a path of length d.

   b) Path $m_{i1}m_{j2}...m_{lh}$ present where $h < d$ iff the path $m_{i1}m_{j2}...m_{lh}$ is pure and attributes A_1, $A_2,...,$ A_{h-1} are chosen above A_h and $m_{i1}m_{j2}...m_{l(h-1)}$ is impure or the path $m_{i1}m_{j2}...m_{lh}$

is pure and attributes $A_1,\ A_2,...,\ A_{h-2},\ A_h$ are chosen above A_{h-1} and $m_{i1}m_{j2}...m_{l(h-2)}m_{lh}$ is impure or ... or the path $m_{j2}...m_{lh}$ is pure and attributes $A_2,...,\ A_h$ are chosen above A_1 and $m_{j2}...m_{lh}$ is impure.
This means that if a certain set of $h-1$ attributes are present in a path in the tree then we split on some h^{th} attribute iff the current path is not pure and the resulting path is pure.

The above conditions suffice for ”path present” since the purity property is anti-monotone and the impurity property is monotone.

3. **Scarcity:** This stopping criteria implies that we stop growing the tree from a particular split of a certain attribute if its count is less than or equal to some pre-specified pruning bound. Let us denote this number by pb. As before, we have the following two separate checks for paths of length d and less than d respectively.

   a) Path $m_{i1}m_{j2}...m_{ld}$ present iff the attributes $A_1,...,A_{d-1}$ are chosen above A_d and $ct(m_{i1}m_{j2}...m_{l(d-1)}) > pb$ or the attributes $A_1,...,A_{d-2},A_d$ are chosen above A_{d-1} and $ct(m_{i1}m_{j2}...m_{l(d-2)}m_{nd}) > pb$ or ... or the attributes $A_2,...,A_d$ are chosen above A_1 and $ct(m_{i2}m_{j3}...m_{ld}) > pb$.

   b) Path $m_{i1}m_{j2}...m_{lh}$ present where $h < d$ iff the attributes $A_1,...,A_{h-1}$ are chosen above A_h and $ct(m_{i1}m_{j2}...m_{l(h-1)}) > pb$ and $ct(m_{i1}m_{j2}...m_{lh}) \leq pb$ or the attributes $A_1,...,A_{h-2},A_h$ are chosen above A_{h-1} and $ct(m_{i1}m_{j2}...m_{l(h-2)}m_{nh}) > pb$ and $ct(m_{i1}m_{j2}...m_{nh}) \leq pb$ or ... or the attributes $A_2,...,A_h$ are chosen above A_1 and $ct(m_{i2}m_{j3}...m_{lh}) > pb$ and $ct(m_{i1}m_{j2}...m_{lh}) \leq pb$.
   This means that we stop growing the tree under a node once we find that the next chosen attribute produces a path with occupancy $\leq pb$.

   The above conditions suffice for ”path present” since the occupancy property is monotone.

We observe from the above checks that we have two types of conditions that need to be evaluated for a path being present namely, i) those that depend on the sample viz. $m_{i1}m_{j2}...m_{l(d-1)}$ is impure or $ct(m_{i1}m_{j2}...m_{lh}) > pb$ and ii) those that depend split attribute selection method viz. $A_1,\ A_2,...,\ A_h$ are chosen. The former depends on the data distribution which we have specified to be a multinomial. The latter we discuss in the next subsection. Note that checks for a combination of the above stopping criteria can be obtained by appropriately combining the individual checks.

3.4 Split Attribute Selection

In decision tree construction algorithms, at each iteration we have to decide the attribute variable on which the data should be split. Numerous measures have been developed (Hall and Holmes, 2003). Some of the most popular ones aim to increase the purity of a set of datapoints that lie in the region formed by that split. The purer the region, the better the prediction and lower the error of the classifier. Measures such as, i) Information Gain (IG) (Quinlan, 1986), ii) Gini Gain (GG) (Breiman et al., 1984), iii) Gain Ratio (GR) (Quinlan, 1986), iv) Chi-square test (CS) (Shao, 2003) etc. aim at realizing this intuition. Other measures using Principal Component Analysis (Smith, 2002), Correlation-based measures (Hall, 1998) have also been developed. Another interesting yet non-intuitive measure in terms of its utility is the Random attribute selection measure. According to this measure we randomly choose the split attribute from available set. The decision tree that this algorithm produces is called a Random decision tree (RDT). Surprisingly enough, a collection of RDTs quite often outperform their seemingly more powerful counterparts (Liu et al., 2005). In this paper we study this interesting variant. We do this by first presenting a probabilistic characterization in selecting a particular attribute/set of attributes, followed by simulation studies. Characterizations for the other measures can be developed in similar vein by focusing on the working of each measure. As an example, for the deterministic purity based measures mentioned above the split attribute selection is just a function of the sample and thus by appropriately conditioning on the sample we can find the relevant probabilities and hence the moments.

Before presenting the expression for the probability of selecting a split attribute/attributes in constructing a RDT we extend the results in Dhurandhar and Dobra (2009) where relationships were drawn between the moments of HE, CE, LE (just a special case of cross validation) and GE, to be applicable to randomized classification algorithms. The random process is assumed to be independent of the sampling process. This result is required since the results in Dhurandhar and Dobra (2009) are applicable to deterministic classification algorithms and we would be analyzing RDT's. With this we have the following lemma.

Lemma 1 *Let D and T be independent discrete random variables, with some distribution defined on each of them. Let $\mathcal{D}$ and $\mathcal{T}$ denote the domains of the random variables. Let $f(d,t)$ and $g(d,t)$ be two functions such that $\forall t \in \mathcal{T}$ $E_{\mathcal{D}}[f(d,t)] = E_{\mathcal{D}}[g(d,t)]$ and $d \in \mathcal{D}$. Then, $E_{\mathcal{T}\times\mathcal{D}}[f(d,t)] = E_{\mathcal{T}\times\mathcal{D}}[g(d,t)]$*

Proof

$$\begin{aligned}
E_{\mathcal{T}\times\mathcal{D}}[f(d,t)] &= \sum_{t\in\mathcal{T}}\sum_{d\in\mathcal{D}} f(d,t)P[T=t,D=d] \\
&= \sum_{t\in\mathcal{T}}\sum_{d\in\mathcal{D}} f(d,t)P[D=d]P[T=t] \\
&= \sum_{t\in\mathcal{T}} E_{\mathcal{D}}[g(d,t)]P[T=t] \\
&= E_{\mathcal{T}\times\mathcal{D}}[g(d,t)].
\end{aligned}$$

■

The result is valid even when D and T are continuous, but considering the scope of this paper we are mainly interested in the discrete case. This result implies that all the relationships and expressions in Dhurandhar and Dobra (2009) hold with an extra expectation over the $t's$, for randomized classification algorithms where the random process is independent of the sampling process.

3.5 Random Decision Trees

In this subsection we explain the randomized process used for split attribute selection and provide the expression for the probability of choosing an attribute/a set of attributes. The attribute selection method we use is as follows. We assume a uniform probability distribution in selecting the attribute variables, that is, attributes which have already not been chosen in a particular branch, have an equal chance of being chosen for the next level. The random process involved in attribute selection is independent of the sample and hence the lemma 1 applies. We now give the expression for the probability of selecting a subset of attributes from the given set for a path. This expression is required in the computation of the above mentioned probabilities used in computing the moments.

For the first moment we need to find the following probability. Given d attributes $A_1,\ A_2, ...,\ A_d$ the probability of choosing a set of h attributes where $h \in \{1, 2, ..., d\}$ is,

$$P[h\ attributes\ chosen] = \frac{1}{\begin{pmatrix} d \\ h \end{pmatrix}}$$

since choosing without replacement is equivalent to simultaneously choosing a subset of attributes from the given set.

For the second moment when the trees are different (required in the finding of variance of CE since, the training sets in the various runs in cross validation are different, that is, for finding $E_{\mathcal{Z}(N)\times\mathcal{Z}(N)}\left[GE(\zeta)GE(\zeta')\right]$), the probability of choosing l_1 attributes for path in one tree and l_2 attributes for path in another tree where $l_1, l_2 \leq d$ is given by,

$$P[l_1\ attribute\ path\ in\ tree\ 1,\ l_2\ attribute\ path\ in\ tree\ 2] = \frac{1}{\begin{pmatrix} d \\ l_1 \end{pmatrix}\begin{pmatrix} d \\ l_2 \end{pmatrix}}$$

since the process of choosing one set of attributes for a path in one tree is independent of the process of choosing another set of attributes for a path in a different tree.

For the second moment when the tree is the same (required in the finding of variance of GE and HE, that is, for finding $E_{\mathcal{Z}(N)}\left[GE(\zeta)^2\right]$), the probability of choosing two sets of attributes such that the two distinct paths resulting from them co-exist in a single tree is given by the following. Assume we have d attributes $A_1,\ A_2, ...,\ A_d$. Let the lengths of the two paths (or cardinality of the two sets) be l_1 and l_2 respectively, where $l_1, l_2 \leq d$. Without loss of generality assume $l_1 \leq l_2$. Let p be the number of attributes common to both paths. Notice that $p \geq 1$ is one of the necessary conditions for the two paths to co-exist. Let $v \leq p$ be those attributes among the total p that have same values for both paths. Thus $p - v$ attributes are common to both paths but have different values. At one of these attributes in a given tree the two paths will bifurcate. The probability that the two paths co-exist given our randomized attribute selection method is computed by finding out all possible ways in which the two paths can co-exist in a tree and then multiplying the number of each kind of way by the probability of having that way. A detailed proof is given in the Appendix A. The expression for the probability based on the attribute selection method is,

$$P[l_1\ and\ l_2\ length\ paths\ co-exist] =$$
$$\sum_{i=0}^{v} vPr_i(l_1-i-1)!(l_2-i-1)!(p-v)prob_i$$

where $vPr_i = \frac{v!}{(v-i)!}$ denotes permutation and $prob_i = \frac{1}{d(d-1)...(d-i)(d-i-1)^2...(d-l_1+1)^2(d-l_1)...(d-l_2+1)}$ is the probability of the i^{th} possible way. For fixed height trees of height h, $(l_1-i-1)!(l_2-i-1)!$ becomes $(h-i-1)!^2$ and $prob_i = \frac{1}{d(d-1)...(d-i)(d-i-1)^2...(d-h+1)^2}$.

3.6 Putting Things Together

We now have all the ingredients that are required for the computation of the moments of GE. In this subsection we combine the results derived in the previous subsections to obtain expressions for

$P_{Z(N)}[\zeta(x)=C_i]$ and $P_{Z(N)\times Z(N)}[\zeta(x)=C_i \wedge \zeta'(x')=C_v]$ which are vital in the computation of the moments.

Let *s.c.c.s.* be an abbreviation for stopping criteria conditions that are sample dependent. Conversely, *s.c.c.i.* be an abbreviation for stopping criteria conditions that are sample independent or conditions that are dependent on the attribute selection method. We now provide expressions for the above probabilities categorized by the 3 stopping criteria.

3.6.1 FIXED HEIGHT

The conditions for ”path exists” for fixed height trees depend only on the attribute selection method as seen in Section 3.3.1. Hence the probability used in finding the first moment is given by,

$$\begin{aligned}
&P_{Z(N)}[\zeta(x)=C_i]\\
&=\sum_p P_{Z(N)}[ct(path_pC_i) > ct(path_pC_j), path_p exists,\ \forall j \neq i,\ i,j \in [1,...,k]]\\
&=\sum_p P_{Z(N)}[ct(path_pC_i) > ct(path_pC_j), s.c.c.i.,\ \forall j \neq i,\ i,j \in [1,...,k]]\\
&=\sum_p P_{Z(N)}[ct(path_pC_i) > ct(path_pC_j),\ \forall j \neq i,\ i,j \in [1,...,k]] P_{Z(N)}[s.c.c.i.]\\
&=\sum_p \frac{P_{Z(N)}[ct(path_pC_i) > ct(path_pC_j),\ \forall j \neq i,\ i,j \in [1,...,k]]}{\begin{pmatrix} d \\ h \end{pmatrix}}
\end{aligned}$$

where h is the length of the paths or the height of the tree. The probability in the last step of the above derivation can be computed from the underlying joint distribution. The probability for the second moment when the trees are different is given by,

$$\begin{aligned}
&P_{Z(N)\times Z(N)}\left[\zeta(x)=C_i \wedge \zeta'(x')=C_v\right]\\
&=\sum_{p,q} P_{Z(N)\times Z(N)}[ct(path_pC_i) > ct(path_pC_j), path_p exists, ct(path_qC_v) > ct(path_qC_w),\\
&\quad path_q exists, \forall j \neq i,\ \forall w \neq v,\ i,j,v,w \in [1,...,k]]\\
&=\sum_{p,q} P_{Z(N)\times Z(N)}[ct(path_pC_i) > ct(path_pC_j), ct(path_qC_v) > ct(path_qC_w), \forall j \neq i,\\
&\quad \forall w \neq v,\ i,j,v,w \in [1,...,k]] \cdot P_{Z(N)\times Z(N)}[s.c.c.i.]\\
&=\frac{1}{\begin{pmatrix} d \\ h \end{pmatrix}^2}(\sum_{p,q} P_{Z(N)\times Z(N)}[ct(path_pC_i) > ct(path_pC_j), ct(path_qC_v) > ct(path_qC_w),\\
&\qquad \forall j \neq i,\ \forall w \neq v,\ i,j,v,w \in [1,...,k]]).
\end{aligned}$$

The probability for the second moment when the trees are the same is given by,

$$
\begin{aligned}
&P_{\mathcal{Z}(N)}\left[\zeta(x)=C_i \wedge \zeta(x')=C_v\right] \\
&= \sum_{p,q} P_{\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), path_pexists, ct(path_qC_v) > ct(path_qC_w), \\
&\quad path_qexists, \forall j \neq i,\ \forall w \neq v,\ i,j,v,w \in [1,...,k]] \\
&= \sum_{p,q} P_{\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), ct(path_qC_v) > ct(path_qC_w), \forall j \neq i,\ \forall w \neq v,\ i,j, \\
&\quad v,w \in [1,...,k]] \cdot P_{\mathcal{Z}(N)}[s.c.c.i.] \\
&= \sum_{p,q}\sum_{t=0}^{b} bPr_t(h-t-1)!^2(r-v)prob_tP_{\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), \\
&\qquad ct(path_qC_v) > ct(path_qC_w), \forall j \neq i,\ \forall w \neq v,\ i,j,v,w \in [1,...,k]]
\end{aligned}
$$

where r is the number of attributes that are common in the 2 paths, b is the number of attributes that have the same value in the 2 paths, h is the length of the paths and $prob_t = \frac{1}{d(d-1)...(d-t)(d-t-1)^2...(d-h+1)^2}$. As before, the probability comparing counts can be computed from the underlying joint distribution.

3.6.2 PURITY AND SCARCITY

The conditions for "path exists" in the case of purity and scarcity depend on both the sample and the attribute selection method as can be seen in 3.3.1. The probability used in finding the first moment is given by,

$$
\begin{aligned}
&P_{\mathcal{Z}(N)}\left[\zeta(x)=C_i\right] \\
&= \sum_{p} P_{\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), path_pexists,\ \forall j \neq i,\ i,j \in [1,...,k]] \\
&= \sum_{p} P_{\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), s.c.c.i, s.c.c.s.,\ \forall j \neq i,\ i,j \in [1,...,k]] \\
&= \sum_{p} P_{\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), s.c.c.s.,\ \forall j \neq i,\ i,j \in [1,...,k]]P_{\mathcal{Z}(N)}[s.c.c.i.] \\
&= \sum_{p} \frac{P_{\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), s.c.c.s.,\ \forall j \neq i,\ i,j \in [1,...,k]]}{dC_{h_p-1}(d-h_p+1)}
\end{aligned}
$$

where h_p is the length of the path indexed by p. The joint probability of comparing counts and $s.c.c.s.$ can be computed from the underlying joint distribution. The probability for the second moment when the trees are different is given by,

$$
\begin{aligned}
&P_{\mathcal{Z}(N)\times\mathcal{Z}(N)}\left[\zeta(x)=C_i \wedge \zeta'(x')=C_v\right]\\
&=\sum_{p,q} P_{\mathcal{Z}(N)\times\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), path_p exists, ct(path_qC_v) > ct(path_qC_w),\\
&\quad path_q exists, \forall j \neq i,\ \forall w \neq v,\ i,j,v,w \in [1,...,k]]\\
&=\sum_{p,q} P_{\mathcal{Z}(N)\times\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), ct(path_qC_v) > ct(path_qC_w), s.c.c.s., \forall j \neq i,\\
&\quad \forall w \neq v,\ i,j,v,w \in [1,...,k]] \cdot P_{\mathcal{Z}(N)\times\mathcal{Z}(N)}[s.c.c.i.]\\
&=\frac{1}{dC_{h_p-1}dC_{h_q-1}(d-h_p+1)(d-h_q+1)}\left(\sum_{p,q} P_{\mathcal{Z}(N)\times\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j),\right.\\
&\quad \left. ct(path_qC_v) > ct(path_qC_w), s.c.c.s., \forall j \neq i,\ \forall w \neq v,\ i,j,v,w \in [1,...,k]]\right)
\end{aligned}
$$

where h_p and h_q are the lengths of the paths indexed by p and q. The probability for the second moment when the trees are the same is given by,

$$
\begin{aligned}
&P_{\mathcal{Z}(N)}\left[\zeta(x)=C_i \wedge \zeta(x')=C_v\right]\\
&=\sum_{p,q} P_{\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), path_p exists, ct(path_qC_v) > ct(path_qC_w), path_q exists,\\
&\quad \forall j \neq i,\ \forall w \neq v,\ i,j,v,w \in [1,...,k]]\\
&=\sum_{p,q} P_{\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j), ct(path_qC_v) > ct(path_qC_w), s.c.c.s., \forall j \neq i,\ \forall w \neq v,\\
&\quad i,j,v,w \in [1,...,k]] P_{\mathcal{Z}(N)}[s.c.c.i.]\\
&=\sum_{p,q}\sum_{t=0}^{b} \frac{bPr_t(h_p-t-2)!(h_q-t-2)!(r-v)prob_t}{(d-h_p+1)(d-h_q+1)} P_{\mathcal{Z}(N)}[ct(path_pC_i) > ct(path_pC_j),\\
&\quad ct(path_qC_v) > ct(path_qC_w), s.c.c.s., \forall j \neq i,\ \forall w \neq v,\ i,j,v,w \in [1,...,k]]
\end{aligned}
$$

where r is the number of attributes that are common in the 2 paths sparing the attributes chosen as leaves, b is the number of attributes that have the same value, h_p and h_q are the lengths of the 2 paths and without loss of generality assuming $h_p \leq h_q$ $prob_t = \frac{1}{d(d-1)...(d-t)(d-t-1)^2...(d-h_p)^2(d-h_p-1)...(d-h_q)}$. As before, the probability of comparing counts and s.c.c.s. can be computed from the underlying joint distribution.

Using the expressions for the above probabilities the moments of GE can be computed. In next section we perform experiments on synthetic as well as distributions built on real data to portray the efficacy of the derived expressions.

4. Experiments

To exactly compute the probabilities for each path the time complexity for fixed height trees is $O(N^2)$ and for purity and scarcity based trees it is $O(N^3)$. Hence, computing exactly the probabilities and consequently the moments is practical for small values of N. For larger values of N, we propose computing the individual probabilities using Monte Carlo. In the empirical studies we report, we initially set N to small value and compute the error (i.e., expected value + standard deviation) exactly, using the derived expressions (which is thus the golden standard) and compare it

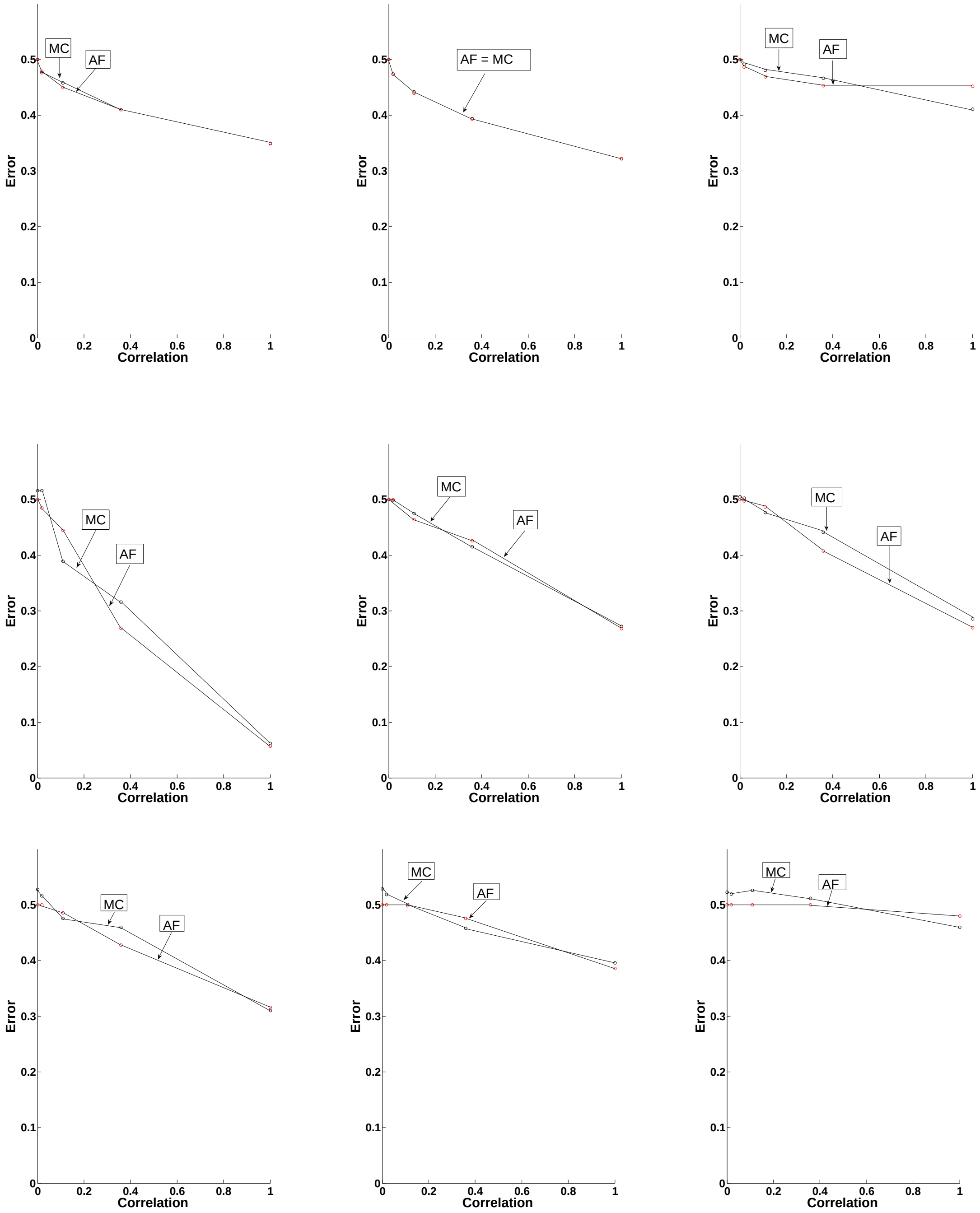

Figure 3: Errors of Fixed height trees (top row figures), Purity trees (center row figures) and Scarcity trees (bottom row figures) with $N = 100$ are shown. The leftmost figures are for $d = 5$ and binary splits, the center figures are for $d = 5$ and ternary splits and the rightmost figures are for $d = 8$ and binary splits. $h = 3$ for Fixed height trees and $pb = \frac{N}{10}$ for Scarcity based trees.

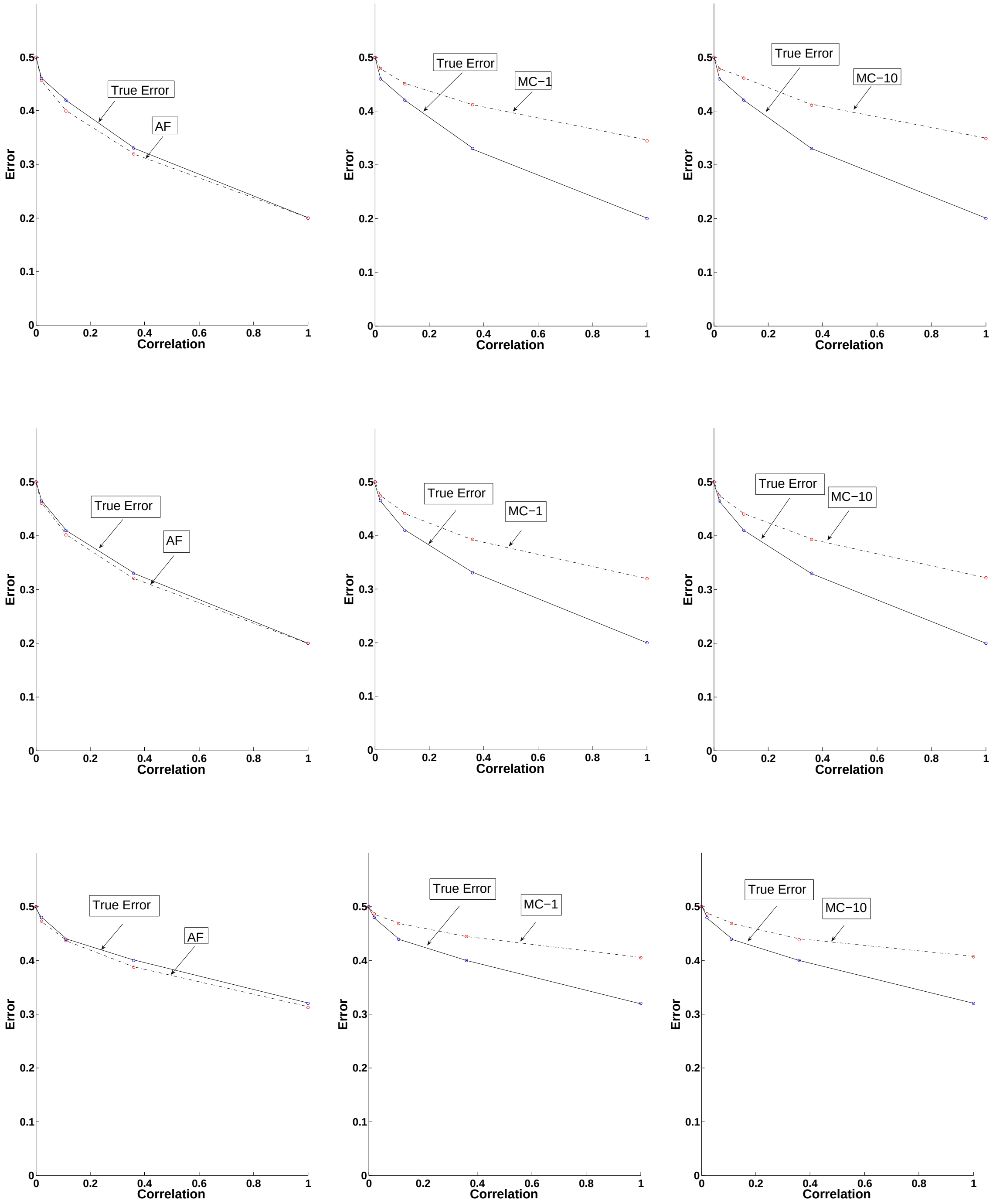

Figure 4: Errors of Fixed height trees with $N = 10000$ and $h = 3$ are shown. In the top row $d = 5$ and splits are binary, in the center row $d = 5$ and splits are ternary and in the last row $d = 8$ and splits are binary.

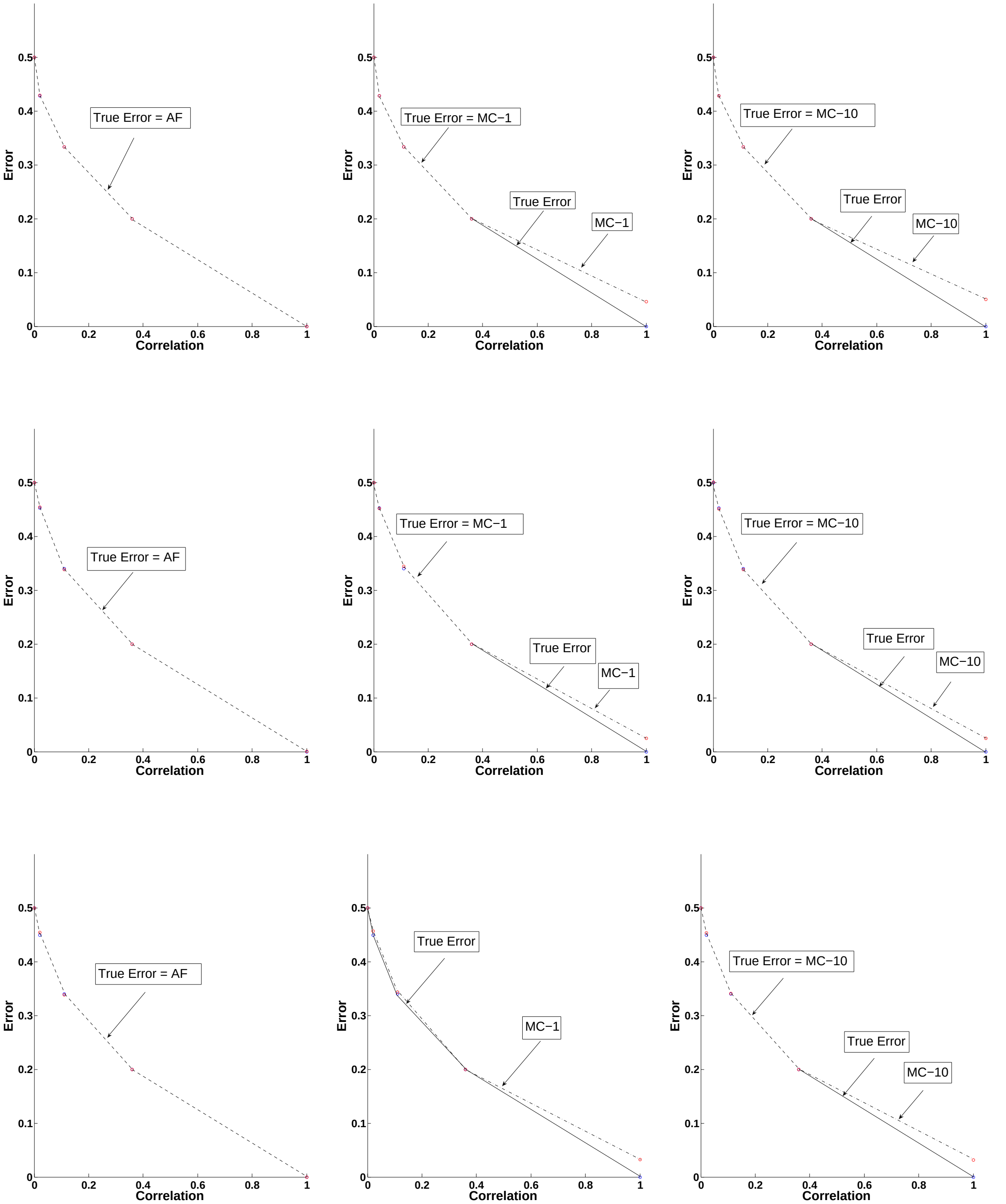

Figure 5: Errors of Purity trees with $N = 10000$ are shown. In the top row $d = 5$ and splits are binary, in the center row $d = 5$ and splits are ternary and in the last row $d = 8$ and splits are binary.

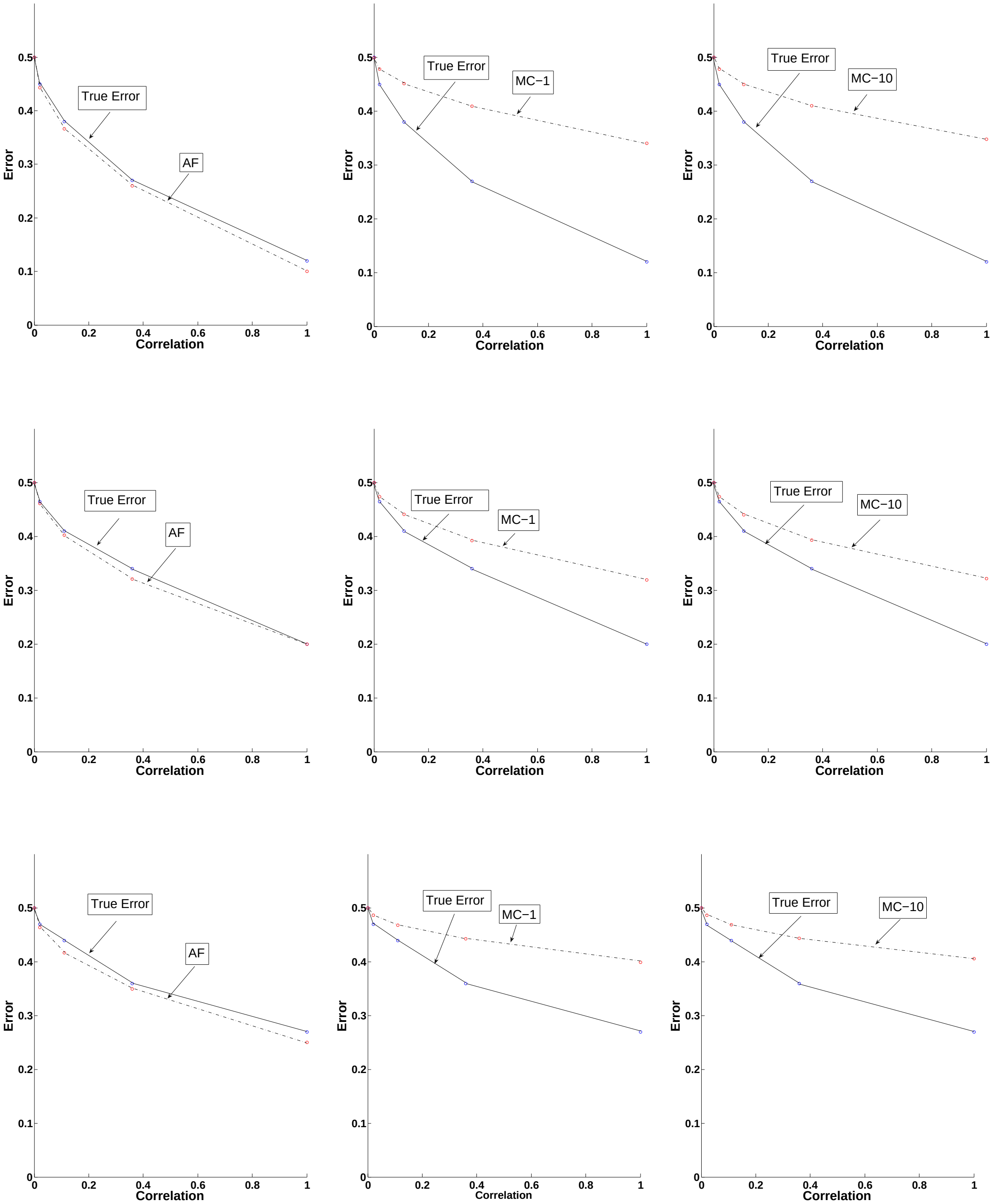

Figure 6: Errors of Scarcity trees with $N = 10000$ and $pb = \frac{N}{10}$ are shown. In the top row $d = 5$ and splits are binary, in the center row $d = 5$ and splits are ternary and in the last row $d = 8$ and splits are binary.

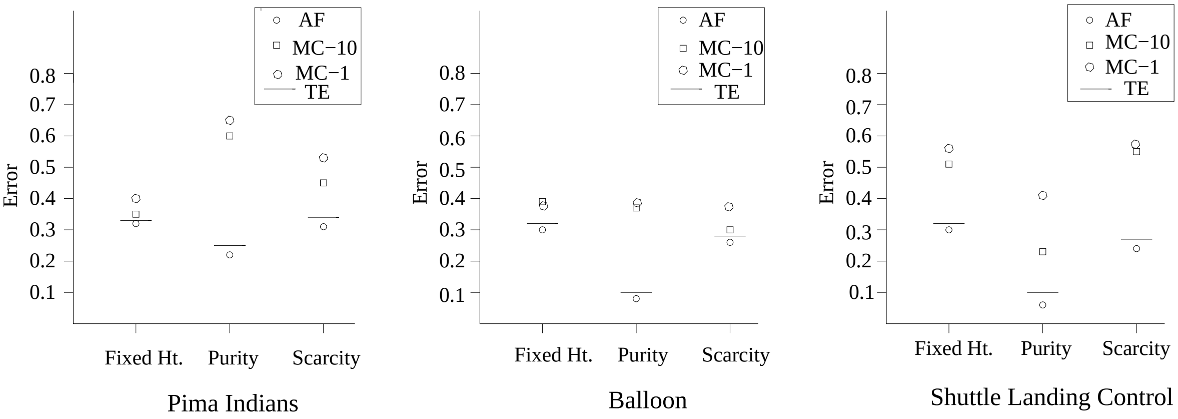

Figure 7: Comparison between AF and MC on three UCI data sets for trees prunned based on fixed height ($h = 3$), purity and scarcity ($pb = \frac{N}{10}$).

Stopping Criteria	Split	$\rho = 1$	$\rho = 0.36$	$\rho = 0.11$	$\rho = 0.02$	$\rho = 0$
Fixed Height						
$N = 100, d = 5, h = 3$	binary	29.67	1.49	0.56	0.34	0.51
$N = 100, d = 5, h = 3$	ternary	277.37	20.49	10.77	7.7	9.23
$N = 100, d = 8, h = 3$	binary	152.21	3.89	2.78	1.33	1.57
$N = 10000, d = 5, h = 3$	binary	41.89	2.99	1.25	0.78	0.71
$N = 10000, d = 5, h = 3$	ternary	575.15	30.9	15.71	11.87	10.8
$N = 10000, d = 8, h = 3$	binary	1813.86	7.21	3.86	2.56	2.3
Purity						
$N = 100, d = 5$	binary	39.67	1154.1	5216.75	10783.19	13750.28
$N = 100, d = 5$	ternary	160.59	181.21	180.5	3281.83	6884.52
$N = 100, d = 8$	binary	2.8	1.9	1035.68	1211.7	1249.32
$N = 10000, d = 5$	binary	40.54	2897.3	11499.57	65581.6	422011.93
$N = 10000, d = 5$	ternary	1386.01	163245.31	675867.31	2662617.25	5781240
$N = 10000, d = 8$	binary	221.98	178913.85	712081.12	3113403.25	6885975
Scarcity						
$N = 100, d = 5$	binary	17.17	17.59	17.5	17.2	17.08
$N = 100, d = 5$	ternary	34.10	33.55	32.88	32.18	31.52
$N = 100, d = 8$	binary	34.42	33.86	33.28	32.59	31.89
$N = 10000, d = 5$	binary	13.04	12.18	11.26	10.32	9.38
$N = 10000, d = 5$	ternary	61.01	60.34	59.51	58.64	57.76
$N = 10000, d = 8$	binary	2643.21	2642.56	2641.75	2640.89	2640.04

Table 1: The above table shows the upper bounds on $E_{\mathcal{Z}(N)}\left[GE(\zeta)\right]$ for different levels of correlation (ρ) between the attributes and class labels obtained using Breiman's formula.

Stopping Criteria	Pima Indians	Balloon	Shuttle Landing Control
Fixed Height	151.58	51.84	1.91
Purity	98.97	50.56	2.74
Scarcity	180.93	41.67	2.32

Table 2: The above table shows the upper bounds on $E_{\mathcal{Z}(N)}[GE(\zeta)]$ for 3 UCI data sets obtained using Breiman's formula.

with MC (i.e., hold-out-set estimation)[3] for the same computational cost. We then choose a larger N and show that the accuracy in estimating the error by using our expressions with Monte Carlo is always greater than by directly using MC for the same computational cost. In fact, the accuracy of using the expressions is never worse than MC even when MC is executed for 10 times the number of iterations as those of our expressions. The true error or the golden standard against which we compare the accuracy of these estimators in this scenario, (since the expressions are also approximated) is MC that is run for around 200 times the number of iterations as those of the expressions. Moreover, in Tables 1 and 2 we depict the upper bounds on the error as computed using Breiman's strength and correlation based upper bound formula (Breiman, 2001).

4.1 Notation

In the experiments, AF refers to the estimates obtained by using the expressions in conjunction with Monte Carlo. MC-i refers to simple Monte Carlo being executed for i times the number of iterations as those of the expressions. Writing just MC denotes MC-1. The term True Error or TE refers to the golden standard against which we compare AF and MC-i. This is relevant only for large N in experiments on synthetic data and experiments on real data, since AF is itself the golden standard for synthetic data experiments with a small N.

4.2 General Setup

We perform empirical studies on synthetic as well as real data. The experimental setup for synthetic data is as follows: In our initial experiments we fix N to a 100 and then increase it to 10000. The number of classes is fixed to two. We observe the behavior of the error for the three kinds of trees with the number of attributes fixed to $d = 5$ and each attribute having 2 attribute values. We then increase the number of attribute values to 3, to observe the effect that increasing the number of split points has on the performance of the estimators. We also increase the number of attributes to $d = 8$ to study the effect that increasing the number of attributes has on the performance. With this we have a $d+1$ dimensional contingency table whose d dimensions are the attributes and the $(d+1)^{th}$ dimension represents the class labels. When each attribute has two values the total number of cells in the table is $c = 2^{d+1}$ and with three values the total number of cells is $c = 3^d \times 2$. If we fix the probability of observing a datapoint in cell i to be p_i such that $\sum_{i=1}^{c} p_i = 1$ and the sample size to N the distribution that perfectly models this scenario is a multinomial distribution

[3] In hold-out set we build a tree, find the test error by averaging over multiple test sets. Perform this procedure multiple times to obtain multiple test errors and find the average and variance of these test errors.

with parameters N and the set $\{p_1, p_2, ..., p_c\}$. In fact, irrespective of the value of d and the number of attribute values for each attribute the scenario can be modeled by a multinomial distribution. In the studies that follow the p_i's are varied and the amount of dependence between the attributes and the class labels is computed for each set of p_i's using the Chi-square test (Connor-Linton, 2003). More precisely, we sum over all i the squares of the difference of each p_i with the product of its corresponding marginals, with each squared difference being divided by this product, that is, correlation $= \sum_i \frac{(p_i - p_{im})^2}{p_{im}}$, where p_{im} is the product of the marginals for the i^{th} cell. The behavior of the error for trees with the three aforementioned stopping criteria is seen for different correlation values and for a class prior of 0.5.

In case of real data, we perform experiments on distributions built on three UCI data sets. We split the continuous attributes at the mean of the given data. We thus can form a contingency table representing each of the data sets. The counts in the individual cells divided by the data set size provide us with empirical estimates for the individual cell probabilities (p_i's). Thus, with the knowledge of N (data set size) and the individual p_i's we have a multinomial distribution. Using this distribution we observe the behavior of the error for the three kinds of trees with results being applicable to other data sets that are similar to the original.

In Tables 1 and 2 we see the upper bounds computed using Breiman's formula (Breiman, 2001): $\kappa\frac{(1-s^2)}{s^2}$ where κ is the correlation between the random decision trees in an ensemble and s is the strength of the resultant classifier.[4] Since, we consider only single random decision trees in this paper and not random forests $\kappa = 1$. To compute s we build a tree and calculate the necessary probabilities. Knowing κ and s we find the upper bound on the GE for the particular classifier. Since, we need an estimate of $E_{\mathcal{Z}(N)}[GE(\zeta)]$, we perform the above procedure multiple times thus building multiple trees and computing an upper bound on GE for each. We then average the upper bounds that we have computed and report the result as an estimate of the upper bound on $E_{\mathcal{Z}(N)}[GE(\zeta)]$.

4.3 Observations

In Figure 3 we observe the behavior of MC vs AF (the golden standard) for $N = 100$. We observe that the estimates provided by MC are reasonable but not as accurate as AF for the same computational cost. The behavior of MC becomes worse as we increase the data set size (N) to 10000 as we discuss now. Figure 4 depicts the error of Fixed height trees for different dimensionalities (5 and 8) and for different number of splits (binary and ternary). We observe here that AF is significantly more accurate than both MC-1 and MC-10. In fact the performance of the 3 estimators namely, AF, MC-1 and MC-10 remains more or less unaltered even with changes in the number of attributes and in the number of splits per attribute. A similar trend is seen for both purity based trees Figure 5 as well as scarcity based trees 6. Though in the case of purity based trees the performance of both MC-1 and MC-10 is much superior as compared with their performance on the other two kinds of trees, especially at low correlations. The reason for this being that, at low correlations the probability in each cell of the multinomial is non-negligible and with $N = 10000$ the event that every cell contains at least a single datapoint is highly likely. Hence, the trees we obtain with high probability using the purity based stopping criteria are all ATT's. Since in an ATT all the leaves are identical irrespective of the ordering of the attributes in any path, the randomness in the classifiers produced, is only due to the randomness in the data generation process and not because of the random attribute selection method. Thus, the space of classifiers over which the error is computed reduces and MC performs

[4] For further details refer to Breiman (2001) and Buttrey and Kobayashi (2003).

well even for a relatively fewer number of iterations. At higher correlations and for the other two kinds of trees the probability of smaller trees is reasonable and hence MC has to account for a larger space of classifiers induced by not only the randomness in the data but also by the randomness in the attribute selection method.

In case of real data too Figure 7, the performance of the expressions is significantly superior as compared with MC-1 and MC-10. The performance of MC-1 and MC-10 for the purity based trees is not as impressive here since the data set sizes are much smaller (in the tens or hundreds) compared to 10000 and hence the probability of having an empty cell are not particularly low. Moreover, the correlations are reasonably high (above 0.6).

By inspecting Tables 1 and 2 it is immediately apparent that the bound in Breiman (2001) when applied to a single tree is ineffective in most situations—the prediction for the GE is larger than 1. For this formula to provide reasonable predictions, a large number of mostly uncorrelated trees needs to be used so that the constant κ balances the influence of s.

4.4 Reasons for Superior Performance of Expressions

With simple MC, trees have to be built while performing the experiments. Since, the expectations are over all possible classifiers, that is, over all possible data sets and all possible randomizations in the attribute selection phase, the exhaustive space over which direct MC has to run is huge. No tree has to be explicitly built when using the expressions. Moreover, the probabilities for each path can be computed parallelly. Another reason as to why calculating the moments using expressions works better is that the portion of the probabilities for each path that depend on the attribute selection method are computed *exactly* (i.e., with no error) by the given expressions and the inaccuracies in the estimates only occur due to the sample dependent portion in the probabilities.

5. Discussion

In the previous sections we derived the analytical expressions for the moments of the GE of decision trees and depicted interesting behavior of RDT's built under the 3 stopping criteria. It is clear that using the expressions we obtain highly accurate estimates of the moments of errors for situations of interest. In this section we discuss issues related to extension of the analysis to other attribute selection methods and issues related to computational complexity of algorithm.

5.1 Extension

The conditions presented for the 3 stopping criteria namely, fixed height, purity and scarcity are applicable irrespective of the attribute selection method. Commonly used deterministic attribute selection methods include those based on Information Gain (IG), Gini Gain (GG), Gain ratio (GR) etc. Given a sample the above metrics can be computed for each attribute. Hence, the above metrics can be implemented as corresponding functions of the sample. For example, in the case of IG we compute the loss in entropy ($qlogq$ where the q's are computed from the sample) by the addition of an attribute as we build the tree. We then compare the loss in entropy of all attributes not already chosen in the path and choose the attribute for which the loss in entropy is maximum. Following this procedure we build the path and hence the tree. To compute the probability of *path exists*, we add these sample dependent conditions in the corresponding probabilities. These conditions account for a particular set of attributes being chosen, in the 3 stopping criteria. In other words, these conditions quantify the conditions in the 3 stopping criteria that are attribute selection method dependent. Similar conditions can be derived for the other attribute selection methods (attribute with maximum gini gain for GG, attribute with maximum gain ratio for GR) from which the relevant probabilities and hence the moments can be computed. Thus, while computing the probabilities given in Equations 1 and 2 the conditions for *path exists* for these attribute selection methods depend totally on the sample. This is unlike what we observed for the randomized attribute selection criterion where the conditions for *path exists* depending on this randomized criterion, were sample independent while the other conditions in purity and scarcity were sample dependent. Characterizing these probabilities enables us in computing the moments of GE for these other attribute selection methods.

In the analysis that we presented, we assumed that the split points for continuous attributes were determined apriori to tree construction. If the split point selection algorithm is dynamic, that is, the split points are selected while building the tree, then in the *path exists* conditions of the 3 stopping criteria we would have to append an extra condition namely, the split occurs at ”this” particular attribute value. In reality, the value of ”this” is determined by the values that the samples attain for the specific attribute in the particular data set, which is finite (since data set is finite). Hence, while analyzing we can choose a set of allowed values for ”this” for each continuous attribute. Using these updated set of conditions for the 3 stopping criteria the moments of GE can be computed.

Another interesting extension to the current work, in which we customized expressions for RDT’s is to extend the analysis to Random Forests. Random Forests are essentially an ensemble of RDT’s and the decision to classify a datapoint is based on a majority vote taken from this ensemble. Hence, in the analysis to compute $P_{\mathcal{Z}(N)}\left[\zeta(x)=y\right]$ (which is the key ingredient in finding the moments), we would have to compute the probability of the event that more than half of the trees classify the input x into class y. The precise details as to how this might be accomplished efficiently is a part of future research.

5.2 Scalability

The time complexity of implementing the analysis is proportional to the product of the size of the input/output space[5] and the number of paths that are possible in the tree while classifying a particular input. To this end, it should be noted that if a stopping criterion is not carefully chosen and applied, then the number of possible trees and hence the number of allowed paths can become exponential in the dimensionality. In such scenarios, studying small or at best medium size trees is feasible. For studying larger trees the practitioner should combine stopping criteria (e.g., pruning bound and fixed height or scarcity and fixed height), that is, combine the conditions given for each individual stopping criteria or choose a stopping criterion that limits the number of paths (e.g., fixed height). Keeping these simple facts in mind and on appropriate usage, the expressions can assist in delving into the statistical behavior of the errors for decision tree classifiers. Further speedup w/o compromising much on accuracy is a challenge for the future.

5.3 Strengths and Limitations of the Applied Methodology

We now discuss the primary advantage and weakness of the approach taken by Statistical Learning Theory (SLT) and our methodology from the point of view of studying classification algorithms. SLT categorizes classification algorithms (actually the more general learning algorithms) into dif-

[5] In case of continuous attributes the size of the input/output space is the size after discretization.

ferent classes called Concept Classes. The concept class of a classification algorithm is determined by its Vapnik-Chervonenkis (VC) dimension which is related to the shattering capability of the algorithm. Distribution free bounds on the generalization error of a classifier built using a particular classification algorithm belonging to a concept class are derived in SLT. The bounds are functions of the VC dimension, the sample size and the training error. The strength of this technique is that by finding the VC dimension of an algorithm we can derive error bounds for the classifiers built using this algorithm without ever referring to the underlying distribution. A consequence of the fact that the characterization is general is that the bounds are usually loose (Boucheron et al., 2005; Williamson, 2001) which in turn results in making statements about any particular classifier and hence classification algorithm weak.

The idea behind the methodology pursued in this paper was to define a class of classifiers induced by a given learning algorithm and i.i.d. data of a given size. As a consequence, this class of classifiers is much smaller than the classes considered in SLT. Hence, the characterization of this class is strongly connected to the behavior of the classifiers and hence the classification algorithm (as seen in this paper for RDT's). The downside of our method is the fact that we loose the strength to make generalized statements to the extent that SLT makes, that is, bounds that are distribution independent. While the process of characterizing classification algorithms employing the deployed methodology might be tedious, we believe that it leads to a more precise study of individual learning algorithms.

6. Conclusion

In this paper we have developed a general characterization for computing the moments of the GE for decision trees. In particular we have specifically characterized RDT's for three stopping criteria namely, fixed height, purity and scarcity. Being able to compute moments of GE, allows us to compute the moments of the various validation measures and observe their relative behavior. Using the general characterization, characterizations for specific attribute selection measures (e.g., IG, GG etc.) other than randomized can be developed as described before. As a technical result, we have extended the theory in Dhurandhar and Dobra (2009) to be applicable to randomized classification algorithms; this is necessary if the theory is to be applied to random decisions trees as we did in this paper. The experiments reported in Section 4 had two purposes: (a) portray the manner in which the expressions can be used as an exploratory tool to gain a better understanding of decision tree classifiers, and (b) show that the methodology in Dhurandhar and Dobra (2009) together with the developments in this paper provide can prove to be a superior analysis tool when compared with other techniques such as Monte Carlo and distribution free bounds.

More work needs to be done to explore the possibilities and test the limits of the kind of analysis that we have performed. However, if learning algorithms are analyzed in the manner that we have shown, it would aid us in studying them more precisely, leading to better understanding and improved decision-making in the practice of model selection.

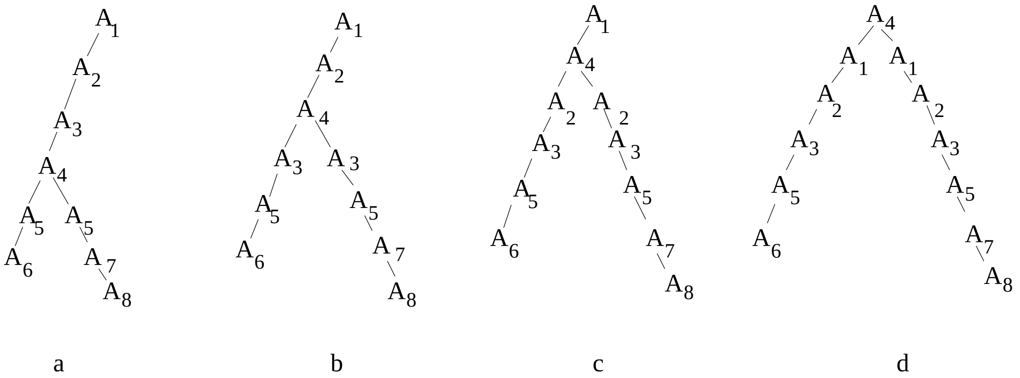

Figure 8: Instances of possible arrangements.

Acknowledgments

We would like to thank Dr. Arunava Banerjee for his comments regarding the introduction. We would also like to thank Dr. Greg Ridgeway and the other anonymous reviewers for their constructive comments. We are grateful to Dr. Medha Dhurandhar for proof reading the paper. This work is supported by the National Science Foundation Grant, NSF-CAREER-IIS-0448264.

Appendix A.

The probability that two paths of lengths l_1 and l_2 ($l_2 \geq l_1$) co-exist in a tree based on the randomized attribute selection method is given by,

$$P[l_1 \; and \; l_2 \; length \; paths \; co-exist] =$$
$$\sum_{i=0}^{v} vPr_i (l_1 - i - 1)!(l_2 - i - 1)!(r - v)prob_i$$

where r is the number of attributes common in the two paths, v is the number attributes with the same values in the two paths, $vPr_i = \frac{v!}{(v-i)!}$ denotes permutation and
$prob_i = \frac{1}{d(d-1)...(d-i)(d-i-1)^2...(d-l_1+1)^2(d-l_1)...(d-l_2+1)}$.

We now prove the above result. The derivation of the above result will become clearer through the following example. Consider the total number of attributes to be d as usual. Let A_1, A_2 and A_3 be three attributes that are common to both paths and also having the same attribute values. Let A_4 and A_5 be common to both paths but have different attribute values for each of them. Let A_6 belong to only the first path and A_7, A_8 to only the second path. Thus, in our example $l_1 = 6$, $l_2 = 7$, $r = 5$ and $v = 3$. For the two paths to co-exist notice that atleast one of A_4 or A_5 has to be at a lower depth than the non-common attributes A_6, A_7, A_8. This has to be true since, if a non-common attribute say A_6 is higher than A_4 and A_5 in a path of the tree then the other path cannot exist. Hence, in all the possible ways that the two paths can co-exist, one of the attributes A_4 or A_5 has to occur at a maximum depth of $v + 1$, that is, 4 in this example. Figure 8a depicts this case. In the successive tree structures, that is, Figure 8b, Figure 8c the common attribute with distinct attribute values (A_4) rises higher up in the tree (to lower depths) until in Figure 8d it becomes the root. To find the probability that the two paths co-exist we sum up the probabilities of such arrangements/tree structures. The probability of the subtree shown in Figure 8a is $\frac{1}{d(d-1)(d-2)(d-3)(d-4)^2(d-5)^2(d-6)}$ considering that we choose attributes w/o replacement for a particular path. Thus the probability of choosing the root is $\frac{1}{d}$, the next attribute is $\frac{1}{d-1}$ and so on till the subtree splits into two paths at depth 5. After the split at depth 5 the probability of choosing the respective attributes for the two paths is $\frac{1}{(d-4)^2}$, since repetitions are allowed in two separate paths. Finally, the first path ends at depth 6 and only one attribute has to be chosen at depth 7 for the second path which is chosen with a probability of $\frac{1}{d-6}$. We now find the total number of subtrees with such an arrangement where the highest common attribute with different values is at depth of 4. We observe that A_1, A_2 and A_3 can be permuted in whichever way w/o altering the tree structure. The total number of ways of doing this is 3!, that is, $3Pr_3$. The attributes below A_4 can also be permuted in 2!3! w/o changing the tree structure. Moreover, A_4 can be replaced by A_5. Thus, the total number of ways the two paths can co-exist with this arrangement is $3Pr_3 2!3!2$. The probability of the arrangement is hence given by, $\frac{3Pr_3 2!3!2}{d(d-1)(d-2)(d-3)(d-4)^2(d-5)^2(d-6)}$. Similarly, we find the probability of the arrangement in Figure 8b where the common attribute with different values is at depth 3 then at depth 2 and finally at the root. The probabilities for the successive arrangements are $\frac{3Pr_2 3!4!2}{d(d-1)(d-2)(d-3)^2(d-4)^2(d-5)^2(d-6)}$, $\frac{3Pr_1 4!5!2}{d(d-1)(d-2)^2(d-3)^2(d-4)^2(d-5)^2(d-6)}$ and $\frac{3Pr_0 5!6!2}{d(d-1)^2(d-2)^2(d-3)^2(d-4)^2(d-5)^2(d-6)}$ respectively. The total probability for the paths to co-exist is given by the sum of the probabilities of these individual arrangements.

In the general case, where we have v attributes with the same values the number of arrangements possible is $v+1$. This is because the depth at which the two paths separate out lowers from $v+1$ to 1. When the bifurcation occurs at $v+1$ the total number of subtrees is $vPr_v(l_1-v-1)!(l_2-v-1)!(r-v)$ with this arrangement. vPr_v is the permutations of the common attributes with same values. $(l_1-v-1)!$ and $(l_2-v-1)!$ are the total permutations of the attributes in path 1 and 2 respectively after the split. $r-v$ is the number of choices for the split attribute. The probability of any one of the subtrees is $\frac{1}{d(d-1)...(d-v)(d-v-1)^2...(d-l_1+1)^2(d-l_1)...(d-l_2+1)}$ since until a depth of $v+1$ the two paths are the same and then from $v+2$ the two paths separate out. The probability of the first arrangement is thus, $\frac{vPr_v(l_1-v-1)!(l_2-v-1)!(r-v)}{d(d-1)...(d-v)(d-v-1)^2...(d-l_1+1)^2(d-l_1)...(d-l_2+1)}$. For the second arrangement with the bifurcation occurring at a depth of v, the number of subtrees is $vPr_{v-1}(l_1-v)!(l_2-v)!(r-v)$ and the probability of any one of them is $\frac{1}{d(d-1)...(d-v+1)(d-v)^2...(d-l_1+1)^2(d-l_1)...(d-l_2+1)}$. The probability of the arrangement is thus $\frac{vPr_{v-1}(l_1-v)!(l_2-v)!(r-v)}{d(d-1)...(d-v+1)(d-v)^2...(d-l_1+1)^2(d-l_1)...(d-l_2+1)}$. Similarly, the probabilities of the other arrangements can be derived. Hence the total probability for the two paths to co-exist which is the sum of the probabilities of the individual arrangements is given by,

$$
\begin{aligned}
&P[l_1 \ and \ l_2 \ length \ paths \ co-exist] = \\
&\sum_{i=0}^{v} \frac{vPr_i(l_1-i-1)!(l_2-i-1)!(r-v)}{d(d-1)...(d-i)(d-i-1)^2...(d-l_1+1)^2(d-l_1)...(d-l_2+1)}.
\end{aligned}
$$

References

A. Blum, A. Kalai, and J. Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Computational Learing Theory*, 1999.

S. Boucheron, O. Bousquet, and G. Lugosi. Introduction to statistical learning theory. http://www.kyb.mpg.de/publications/pdfs/pdf2819.pdf, 2005.

L. Breiman. Random forests. http://oz.berkeley.edu/users/breiman/randomforest2001.pdf, 2001.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, 1984.

S. Buttrey and I. Kobayashi. On strength and correlation in random forests. In *Proceedings of the 2003 Joint Statistical Meetings, Section on Statistical Computing*, 2003.

J. Connor-Linton. Chi square tutorial. http://www.georgetown.edu/faculty/ballc/webtools/web_chi_tut.html, 2003.

A. Dhurandhar and A. Dobra. Semi-analytical method for analyzing models and model selection measures based on moment analysis. *ACM Transactions on Knowledge Discovery and Data Mining*, 2009.

P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006. ISSN 0885-6125. doi: http://dx.doi.org/10.1007/s10994-006-6226-1.

M. Hall. Correlation-based feature selection for machine learning. Ph.D diss. Hamilton, NZ: Waikato University, Department of Computer Science, 1998.

M. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE TRANSACTIONS ON KDE*, 2003.

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

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *In Proceedings of the Fourteenth IJCAI.*, 1995.

F. Liu, K. Ting, and W. Fan. Maximizing tree diversity by building complete-random decision trees. In *PAKDD*, pages 605–610, 2005.

J. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

J. Shao. Linear model selection by cross validation. *JASA*, 88, 1993.

J. Shao. *Mathematical Statistics*. Springer-Verlag, 2003.

L. Smith. A tutorial on principal components analysis. www.csnet.otago.ac.nz/cosc453/ student_tutorials/principal_components.pdf, 2002.

V. Vapnik. *Statistical Learning Theory*. Wiley & Sons, 1998.

R. Williamson. Srm and vc theory (statistical learning theory). http://axiom.anu.edu.au / williams/papers/P151.pdf, 2001.

K. Zhang, W. Fan, B. Buckles, X. Yuan, and Z. Xu. Discovering unrevealed properties of probability estimation trees: On algorithm selection and performance explanation. *ICDM*, 0:741–752, 2006. ISSN 1550-4786. doi: http://doi.ieeecomputersociety.org/10.1109/ICDM.2006.58.