A Tighter Error Bound for Decision Tree Learning Using PAC Learnability

Chaithanya Pichuka, Raju S. Bapi, Chakravarthy Bhagvati,
Arun K. Pujari, B. L. Deekshatulu
Department of Computer and Information Sciences
University of Hyderabad
Gachibowli, Hyderabad, 500046, India
pssgrk438@yahoo.com; {bapics, chakcs, akpcs, bldcs}@uohyd.ernet.in

Abstract

Error bounds for decision trees are generally based on depth or breadth of the tree. In this paper, we propose a bound for error rate that depends both on the depth and the breadth of a specific decision tree constructed from the training samples. This bound is derived from sample complexity estimate based on PAC learnability. The proposed bound is compared with other traditional error bounds on several machine learning benchmark data sets as well as on an image data set used in Content Based Image Retrieval (CBIR). Experimental results demonstrate that the proposed bound gives tighter estimation of the empirical error.

1 Introduction

Computational learning theory (CoLT) deals with the characterization of the difficulty of learning problems and the capabilities of machine learning algorithms [Vapnik, 1998; Vidyasagar, 1997]. The probably approximately correct (PAC) framework is formulated to characterize classes of hypotheses that can be reliably learned from a reasonable (polynomial) number of randomly drawn training examples with a reasonable amount of computation. *Sample complexity* specifies the number of training examples needed for a classifier to converge, with a high probability, to a successful hypothesis. Within the PAC framework, it is possible to derive bounds on sample complexity by combining the expression for the size of the hypothesis space and the empirical error [Mitchell, 1997]. Thus characterization of the hypothesis space is an important issue in CoLT.

In order to understand the representational capability of various classifiers, we ran several algorithms such as perceptron, linear and polynomial support vector machines (SVM), Bayes maximum likelihood (ML), decision tree (DT), oblique decision tree, and k-nearest neighbor (kNN) classifiers on the standard Iris benchmark dataset. Iris is a four-dimensional dataset comprising three classes, of which two classes are not linearly separable. Figure 1 shows a two-dimensional projection of the decision regions described by the trained classifiers on the benchmark dataset. Perceptron, linear SVM and DT define elementary decision regions comprising linear and axis-parallel rectangular surfaces. Oblique DT, polynomial SVM, Bayesian ML and kNN describe increasingly more complex decision regions and hence have richer hypothesis representation capability. It is to be noted that although limited empirical characterization of classifiers is possible as depicted in Figure 1, theoretical characterization of classifiers, in general, is difficult. In this paper we shall consider DTs for further theoretical and empirical analysis.

Several techniques have been proposed for estimating the future error rate of a decision tree classifier [Kaariainen and Langford, 2005; Mansour, 2000]. There are two primary classes of error estimation methods available. The first class of methods utilize the empirical error on the training samples in predicting the future error. Empirical error could be based on the training set or on the test set or on both. For example, we can use k-fold cross-validation on a dataset and then transform the cross-validation estimate into an estimate or heuristic confidence interval of the error of the final decision tree learned from all examples. Another approach is the sample compression bound which considers the test set after labelling it, that is, it estimates the next label based on the training set and the labelled previous test set [Langford, 2005]. There are several other methods such as Microchoice bound [Langford and Blum, 1999], Test set bound and Occam bounds [Langford, 2005]. Microchoice bound inherently depends on the structure of the DT by calculating the choice spaces at every node of the DT. Test set bound is a test set-based bound and is entirely characterized by the errors on the labelled test set. As the test set bound incorporates test set error directly, its estimate is usually good. Occam bound assumes the underlying distribution to be Binomial and computes an estimate based on the empirical errors observed on the training dataset.

The second class of methods utilize structural aspects of the classifier in order to arrive at an estimate. For example, in the case of DTs, one could consider the depth or breadth of the DT constructed on the training samples in the estimation of future error.

In this paper we propose a structure-based error bound for DTs using the PAC learning framework. The new error bound considers both depth and breadth of the DT learned from training examples. We conducted several experiments on benchmark datasets to compare the results of various error bound estimation methods and found that the proposed estimation method works well. The rest of the paper is organized as follows. Firstly, we describe both the structure-based and

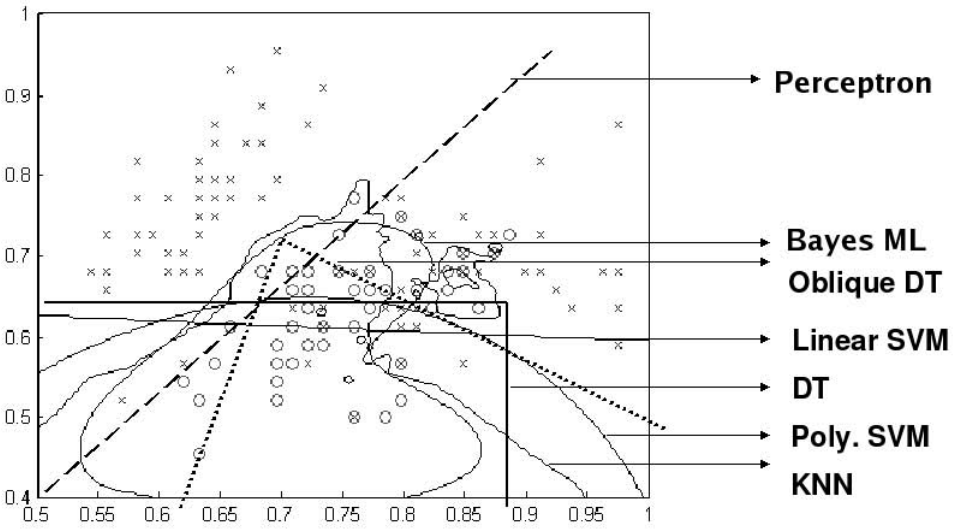

Figure 1: **Decision Regions of Classifiers:** Comparison of the hypotheses learned by various classification methods on the Iris machine learning benchmark dataset.

empirical error-based estimation methods. Experimental results are presented subsequently. Discussion of the results is followed by conclusions.

2 Classifier Structure-Based Error Bounds

In this section, before introducing classifier structure based bounds, we will describe PAC learning in detail and explore how generalization bounds could be formulated based on PAC learning framework.

2.1 PAC Learning and Error Bounds

In this section we review the relation between Probably Approximately Correct (PAC) learning framework and generalization error bounds [Valiant, 1984; Mitchell, 1997]. PAC tackles the questions related to the number of examples and the amount of computation required to learn various classes of target functions. PAC learning framework has two major assumptions. One that assumes that the classification error is bounded by some constant, ε, that can be made arbitrarily small. The second assumption requires that the classifier's probability of failure is bounded by some constant, δ, that can be made arbitrarily small. In short, we require that the classifier *probably* learns a hypothesis that is *approximately* correct – hence it is called the Probably Approximately Correct (PAC) learning framework. Defining the error on entire instance distribution (T) of data as *true error* $(error_T)$ and the error on training set (D) as *empirical error* $(error_D)$, PAC learnability can be defined formally as follows.

Definition [Mitchell, 1997]: Consider a concept class C defined over a set of instances X of length n and a learner L using hypothesis space H. C is **PAC-Learnable** by L using H if for all $c \,\epsilon\, C$, distribution T over X, ε such that $0 < \varepsilon < \frac{1}{2}$, and δ such that $0 < \delta < \frac{1}{2}$, learner L will with probability at least $(1-\delta)$ output a hypothesis $h \in H$ such that $error_{\boldsymbol{T}}(h) \leq \varepsilon$, in time that is polynomial in $\frac{1}{\varepsilon}$, $\frac{1}{\delta}$, n, and $size(c)$. This can also be represented mathematically as follows.

$$P[error_{\boldsymbol{T}}(h) > \varepsilon] \leq \delta \quad (or) \quad P[error_{\boldsymbol{T}}(h) < \varepsilon] \geq (1-\delta) \tag{1}$$

PAC-Learnability concept can be used to derive a general lower bound on the size m of the training set required by a learner. There are two possible scenarios. In the first scenario, we assume that the learner is *consistent*, that is, the learner outputs hypotheses that are consistent with the target concept on the training set. The probability that a single hypothesis having true error greater than ε and is consistent with the training set is at most $(1-\varepsilon)$. And the probability that this hypothesis will be consistent with m individuals is at most $(1-\varepsilon)^m$. If, instead of one, there are k such hypotheses whose true error is greater than ε and are consistent with the training set, then the upper bound on the probability that at least one of these k hypotheses will be consistent with all the m randomly drawn individuals is, $k*(1-\varepsilon)^m$. Using the fact that $k \leq |H|$, we can write down the following inequality in Equation 2.

$$P[error_T(h) > \varepsilon] \leq H*(1-\varepsilon)^m \leq \delta \tag{2}$$

We can rewrite this as shown in Equation 3 and solving for m (the number of training examples needed by the learning algorithm) results in the inequality shown in Equation 4.

$$p[error_T(h) > \varepsilon] \leq H * e^{(-m\varepsilon)} \leq \delta \quad (3)$$

$$m \geq \frac{1}{\varepsilon}(ln|H| + ln(\frac{1}{\delta})) \quad (4)$$

In the second scenario when the learner is *agnostic*, that is, the true error on the training set is not necessarily zero, then we can use Chernoff approximation to estimate the error bound in the PAC learning framework [Mitchell, 1997]. Analogous to Equation 4, the error bound for agnostic learner can be derived and is shown in Equation 5.

$$m \geq \frac{1}{2\varepsilon^2}(ln|H| + ln(\frac{1}{\delta})) \quad (5)$$

Now, we can use Equation 5 to derive the generalization error bound as shown in Equation 6.

$$\varepsilon \geq \sqrt{\frac{1}{2m}(ln|H| + ln(\frac{1}{\delta}))} \quad (6)$$

We can write this in terms of true error $(error_{\boldsymbol{T}}(h))$ and training error $(error_D(h))$ as shown in Equation 7.

$$error_{\boldsymbol{T}}(h) \leq error_D(h) + \sqrt{\frac{1}{2m}(ln|H| + ln(\frac{1}{\delta}))} \quad (7)$$

Coming back to the original problem of deriving error bounds for DTs, Equation 7 can be used if only we know the size of the hypothesis space, $|H|$. Thus for PAC based error bounds, we need an estimate of $|H|$. There are three ways of estimating $|H|$ for DTs. The first approach is to estimate based on the depth of DT, the second method is to estimate based on the breadth of DT, and the third possibility is to use both of these measures. We propose a bound based on the third approach wherein recursive estimation of $|H|$ is done. In the subsequent sub-sections, we describe the three structure-based error bounds for DTs.

2.2 Depth-Based Error Bound for DT (PAC I)

Computation of $|H|$ is based on depth of the tree and the approach is akin to the one shown in the lecture notes of Guestrin [Guestrin, 2005]. We assume that the DTs are binary trees and use the idea that every k-depth decision tree will have $(k-1)$-depth decision trees as its children.

Approximation of $|H|$ for PAC I

Given n attributes,

$$\begin{array}{lcl} H_k & = & Number\ of\ decision\ trees\ of\ depth\ k \\ H_0 & = & 2 \\ H_{k+1} & = & (choices\ of\ root\ attribute)* \\ & & (possible\ left\ subtrees)* \\ & & (possible\ right\ subtrees) \\ H_{k+1} & = & n * H_k * H_k \\ H_k & = & 2^{2^k} * n^{2^k-1} \\ if\ L_k & = & log_2 H_k\ and\ L_0 = 1 \\ L_{k+1} & = & log_2 n + 2L_k \\ So,\ L_k & = & (2^k - 1)(1 + log_2 n) + 1 \end{array} \quad (8)$$

If we substitute this into Equation 7, then we get the expression for error bound for decision tree using the depth feature as shown in Equation 9.

$$error_{\boldsymbol{T}}(h) \leq error_D(h) + \sqrt{\frac{ln2}{2m}((2^k - 1)(1 + log_2 n) + 1 + ln(\frac{1}{\delta}))} \quad (9)$$

2.3 Breadth-Based Error Bound for DT (PAC II)

In this method, computation of $|H|$ is based on breadth of the tree and the approach is akin to the one shown in the lecture notes of Guestrin [Guestrin, 2005]. We assume that the DTs are binary trees and use the idea that every k-leaves decision tree will have $(k-1)$-leaves decision trees in it.

Approximation of $|H|$ for PAC II

Given n attributes,

$$\begin{array}{lcl} H_k & = & Number\ of\ decision\ trees\ of\ leaves\ k \\ H_0 & = & 2 \\ H_k & = & n \sum_{i=1}^{k-1} H_i H_{k-i} \\ H_k & = & n^{k-1}(k+1)^{2k-1} \end{array} \quad (10)$$

The last step in Equation 10 is a rough (crude) approximation to H_k from the previous step. If we substitute this into Equation 7, then we get the expression for error bound for decision tree using the breadth feature as shown in Equation 11. The crude approximation will be replaced by a better approximation obtained by recursive estimation in PAC(III).

$$error_{\boldsymbol{T}}(h) \leq error_D(h) + \sqrt{\frac{1}{2m}((n-1)ln(n) + (2k-1)ln(k+1) + ln(\frac{1}{\delta}))} \quad (11)$$

2.4 Depth and Breadth-Based Error Bound for DT (PAC III)

We propose this new approach for calculating error bounds for DTs. This approximation is very much similar to PAC II although the resulting bound will be tighter as we will show empirically. The simple idea behind this approach is that it is important to consider several structural features of the DT to get a closer approximation of $|H|$ which in turn leads to a tighter estimate for the error bound.

Approximation of $|H|$ for PAC III

Given n attributes,

$$\begin{aligned} H_k &= Number\ of\ decision\ trees\ of\ leaves\ k \\ H_0 &= 2 \\ H_1 &= n * H_0 * H_0 \\ H_2 &= n * H_1 * H_1 \\ &\vdots \\ H_k &= n \sum_{i=1}^{k-1} H_i H_{k-i} = n^{k-1} * F_k * H_1^k \\ Where, & \\ F_k &= \sum_{i=1}^{k-1} F_i F_{k-i} \\ F_1 &= 1\ (Initial\ condition) \end{aligned} \tag{12}$$

We have to substitute the expression for H into Equation 7 to get error bound for DT using this approximation.

3 Empirical Error-Based Bounds

In this section, we will describe generalization bounds formulated based on the empirical error observed on the dataset. Unlike the structure-based error bounds discussed in the previous section, bounds reviewed in this section explicitly incorporate empirical errors observed on the datasets — training and test sets.

3.1 Microchoice Bound

Microchoice approach tries to estimate an *a priori* bound on the true error based on the sequential trace of the algorithm [Langford and Blum, 1999]. When a learning algorithm is applied on training set of n instances, the algorithm successively makes a sequence of choices $c_1, \ldots, c_n$ from a sequence of choice spaces, $C_1, \ldots, C_n$ finally producing a hypothesis, $h \in H$. Specifically, the algorithm looks at the choice space C_1 to produce choice c_1. The choice c_1 in turn determines the next choice space C_2. It is to be noted here that at every stage the previous choices are eliminated from consideration at the next stage. The algorithm again looks at the data to make the next choice $c_2 \in C_2$. This choice then determines the next choice space C_3, and so on. These choice spaces can be thought of as nodes in a choice tree, where each node in the tree corresponds to some internal state of the learning algorithm, and a node belonging to some choice space C_i.

We can apply this Microchoice bound to decision tree by taking into account the choice spaces available at every node of the decision tree and then using PAC bound technique to get the error bound as shown in Equation 13.

$$error_{\boldsymbol{T}}(h) \leq error_D(h) + \sqrt{\frac{1}{2m}\Big(\sum_{i \in Nodes(DT)} C_i + ln(\frac{1}{\delta})\Big)} \tag{13}$$

3.2 Test Set Bound

In this bound assumptions about the error distribution are made. In particular, classification error distribution is modelled as coin flips distribution or the Binomial distribution as shown in Equation 16.

$$Bin(\frac{k}{m}, c_D) = \sum_{i=1}^{k} \binom{m}{j} c_D^j (1 - c_D)^{m-j} \tag{14}$$

The expression in Equation 14 computes the probability that m examples (coins) with error rate c_D produce k or fewer errors. We can interpret the Binomial tail as the probability of an empirical error greater than or equal to $\frac{k}{m}$. But we are interested in error bound for a classifier given the probability of error δ and m, so we define Binomial tail inversion as given in Equation 15 which gives the largest true error such that the probability of observing $\frac{k}{m}$ or more errors is at least δ.

$$\overline{Bin}(\frac{k}{m}, error_{\boldsymbol{T}}(h)) = max_p\{p : Bin(\frac{k}{m}, p) \geq \delta\} \tag{15}$$

Now, given the number of test instances m, test set bound can be formulated as shown in Equation 16.

Test Set Bound :

$$P[error_{\boldsymbol{T}}(h) \leq \overline{Bin}(error_{test}, \delta)] \geq 1 - \delta \tag{16}$$

One serious drawback of the test set bound is that it is possible that the test set and training sets are incompatible and thereby introduce inaccuracies in the error bound estimation [Kaariainen and Langford, 2005].

3.3 Occam's Razor Bound

This is can be termed as a training set-based bound as it takes the training set performance into consideration. This is reasonable as many learning algorithms implicitly assume that the train set accuracy behaves like the true error. Additionally, in this bound we need to know the prior probability of the hypotheses.

Occam's Razor Bound:

For all priors $P(h)$, over all classifiers h, for all $\delta \in (0, 1)$,

$$P[error_{\boldsymbol{T}}(h) \leq \overline{Bin}(error_D(h), \delta P(h))] \geq 1 - \delta \tag{17}$$

We obtain the Occam's razor bound by negating the above Equation 17.

$$P[error_{\boldsymbol{T}}(h) \geq \overline{Bin}(error_D(h), \delta P(h))] < \delta \tag{18}$$

It is very important to notice that the prior $P(h)$ must be selected before looking at the instances. We can relax the Occam's Razor bound with the entropy Chernoff bound to get a somewhat more tractable expression [Langford, 2005].

Chernoff Occam's Razor Bound:

For all priors P(h), over all classifiers h, for all $\delta \in (0, 1)$,

$$P[err_{\boldsymbol{T}}(h) \geq err_D(h) + \sqrt{\frac{1}{2m}(ln(\frac{1}{P(h)}) + ln(\frac{1}{\delta}))}] < \delta \tag{19}$$

The application of the Occam's Razor bound is somewhat more complicated than the application of the test set bound.

Table 1: Results of experiments for error bound calculating on 15 different data sets.

Data Set	S	P(I)	P(II)	P(III)	MC	Occ	Test	Emp	ATT	$\overline{B}$	$\overline{D}$	$\overline{E}$
Weather	14	0.489	**0.356**	**0.356**	0.490	0.953	0.485	0.25	4	1	1	3.5
Yellow-small-	16	0.521	**0.409**	**0.409**	0.473	0.889	0.263	0.263	5	1	1	4.2
Adult-	20	0.462	**0.362**	**0.362**	0.425	0.793	0.427	0.05	5	1	1	1
Adult+	20	0.462	**0.362**	**0.362**	0.425	0.793	0.427	0.05	5	1	1	1
Yellow-small+	20	0.462	**0.362**	**0.362**	0.4294	0.793	0.388	0.04	5	1	1	0.8
Contact Lenses	24	0.532	0.354	**0.347**	0.466	0.734	0.394	0.096	5	1.1	1.7	2.3
Labor	57	0.368	**0.220**	**0.220**	0.394	0.424	0.294	0.071	17	0.7	1.5	4.1
Monk1	432	0.374	0.211	**0.167**	0.296	0.288	0.190	0.138	7	5	4.9	59.9
Monk2	432	0.423	0.271	0.206	0.325	0.288	**0.188**	0.131	7	7.2	5.2	56.6
Monk3	432	0.374	0.212	**0.168**	0.276	0.274	0.255	0.106	7	4.5	4.9	46.2
Voting-records	435	0.32	0.18	**0.149**	0.219	0.215	0.158	0.056	17	3.5	4.6	24.5
CRX	690	0.419	0.129	**0.107**	0.150	0.150	0.139	0.104	16	3.5	5.5	72.3
Tic-tac-toe	958	0.958	0.247	**0.201**	0.237	0.230	0.221	0.087	10	12.5	4.1	84
Segment	1500	0.168	0.154	0.123	0.184	0.113	**0.077**	0.025	20	8.6	9.4	38.8
CBIR	696	0.812	0.296	0.219	0.321	0.271	**0.190**	0.184	15	11	11	128

S: Size; P(I): PAC(I); P(II): PAC(II); P(III): PAC(III); MC: Microchoice; Occ: Occam's Razor; Test: Test set; Emp: Empirical Error; ATT: Number of Attributes; $\overline{B}$: Average Breadth of DTs; $\overline{D}$: Average Depth of DTs; $\overline{E}$: Average count of misclassification errors

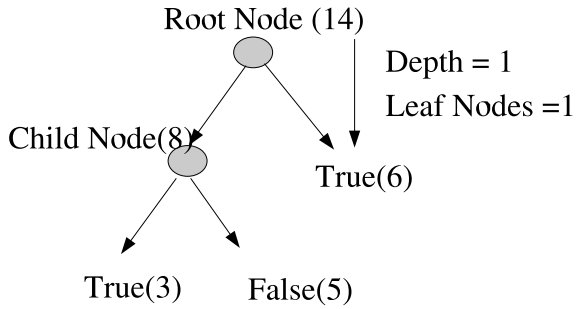

Figure 2: **Example DT**: corresponding to *Adult-* dataset where breadth and depth are 1 and the quantity in parenthesis denotes the size of the choice space at every node.

4 Illustrative Example

$$
\begin{array}{lrcl}
PAC-I: & H_k & = & 5^k * n^{2^k-1}\ [Eq\ 9] \\
 & H_1 & = & 5^1 * 4^{2^1-1} = 20 \\
PAC-II: & H_k & = & n^{k-1}(k+1)^{2k-1}\ [Eq\ 11] \\
 & H_1 & = & 5^0(2)^{2*1-1} = 1*2 = 2 \\
PAC-III: & H_k & = & n\sum_{i=1}^{k-1} H_i H_{k-i}\ [Eq\ 12] \\
 & H_1 & = & 2\ [k=1,\ Init.Cond.] \\
Microchoice: & H & = & \sum_{i \in Nodes(DT)} C_i \\
 & H & = & 14+8
\end{array}
\tag{20}
$$

Here we consider an example experimental dataset, namely, the *Adult-* dataset. The DT structure obtained from one of the ten experiments on *Adult-* dataset is chosen to illustrate how various error bounds are computed. In this case the breadth and depth are one each and the choice space size at each node is also indicated in Figure 2. The number of attributes, n, is 5.

The H value computed for various error bounds in Equation 20 are now substituted in Equation 7 to obtain the final theoretical estimates of the error bound under different schemes. Taking $m = 14$ and $\delta = 0.05$, we obtain the following values for error bounds: PAC(I) = 0.462; PAC(II) = 0.362; and PAC(III) = 0.362. For the Microchoice bound, we need to take Equation 13 and substitute the choice-space sizes from the DT in Figure 2. Then the Microchoice error works out to be 0.524.

5 Empirical Results

We have done experiments on 14 different machine learning benchmark datasets from the UCI repository. The results reported are averaged over 10 experiments conducted on randomly chosen subsets of the datasets. Training was done on 66% of the data and testing on the remaining 33%. In the case of image dataset from content based image retrieval (CBIR) system, due to the complexity of the experiment only one run was conducted. The results are summarized in Table 1. Table reports the observed error rates (on training set and test set) as empirical error. The goal of the experimentation is to see if the theoretical bounds estimated from various methods come close to the observed empirical error. The bold-faced entries in Table 1 correspond to the best estimate of the true error. From the table, as expected we can see that PAC(III) always gives better estimate than PAC(I) and PAC(II). Depth based measures overestimate the size of the tree since they assume a complete binary tree and in reality, the DT may be

far from complete. Thus PAC(I) estimates will be inferior to those of PAC(III). This can be clearly observed in the table where the PAC(I) estimates become worse with increasing depth. In PAC(II), breadth of DT is considered. However, since a crude approximation of the actual value of H is considered in PAC(II) as discussed in Section 2.3, PAC(II) does not give a better estimate compared to PAC(III).

From Table 1 we can also observe that PAC(III) always outperforms the microchoice and Occam's razor bounds. Microchoice bound is an estimate of H based on choice spaces available at every node of DT and the choice space sizes can lead to overestimation of H. Test set and Occam's bounds rely on binomial distribution based measures which take total number of instances and observed errors into consideration. Conceptually, it is not straight forward to compare PAC(III) and binomial theory based bounds. However it can be observed that whenever empirical error is low, we tend to get good estimation from Occam and Test set bounds. Test set bound performed well in five out of the fifteen experiments whereas PAC(III) fared well in the remaining 10 experiments. It appears that in the majority of these five cases, empirical (test) error was low and the average breadth of the DT was high. These may be the possible reasons for poorer estimation by the PAC(III) approach. Further, it is easy to see that these empirical results seem reasonable from theoretical considerations also when different expressions for $|H|$ are compared for the three PAC-based bounds.

Results in Table 1 suggest a possible subset relation among the various error bounds studied in this paper.

$PAC(I) \geq PAC(II) \geq PAC(III)$
$Micro\ choice \geq PAC(III)$
$Occam \geq PAC(III)$

6 Conclusion

In this paper, we proposed a bound for error rate that depends both on the depth and the breadth of a specific decision tree constructed from the training samples. PAC leaning framework is used to derive this bound. The proposed bound is compared with other traditional error bounds on several machine learning benchmark data sets and on an image data set. Experimental results demonstrate that the proposed bound gives tighter estimation of the empirical error. The bound we have obtained here is considerably tighter than previous bounds for Decision Tree classifiers. We arrived at a possible subset relations among various structure-based and empirical error-based bounds.

References

[Guestrin, 2005] C. Guestrin. Lecture notes, Carnegie Mellon University (ML course No: 10701/15781). February 2005.

[Kaariainen and Langford, 2005] M. Kaariainen and J. Langford. A comparison of tight generalization error bounds. In *International conference on Machine Learning*, volume 119, pages 409–416, August 2005.

[Langford and Blum, 1999] J. Langford and A. Blum. Microchoice bounds and self bounding learning algorithms. In *COLT*, pages 209–214, 1999.

[Langford, 2005] J. Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6:273–306, September 2005.

[Mansour, 2000] Y. Mansour. Generaliztion bounds for decision tree. In *COLT*, pages 69–74, 2000.

[Mitchell, 1997] T. M. Mitchell. *Machine Learning*. The McGraw-Hill Companies, Singapore, 1997.

[Valiant, 1984] L. Valiant. A theory of the leranable. *Communications of the ACM*, 27:1134–1142, 1984.

[Vapnik, 1998] V. N. Vapnik. *Statistical Learning Theory*. Wiley & Sons, Inc., 1998.

[Vidyasagar, 1997] M. Vidyasagar. *A Theory of Learning and Generalization*. Springer-Verlag, New York, 1997.