Combining Train Set and Test Set Bounds

John Langford JCL@CS.CMU.EDU
Computer Science Department, Carnegie-Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15217

Abstract

This paper is about bounds on future error rates. We present a theorem for combining an arbitrary test set based bound with an arbitrary training set based bound. Appropriate use of this theorem results in a combined bound with two properties: 1) the combined bound is never much worse than either the training set based bound or the test set based bound and 2) the combined bound is sometimes better than either bound individually. Empirical validation is presented showing the effectiveness of the combined bound.

1 Introduction

Traditionally, there are two broad classes of techniques for bounding the future error rate of a learned classifier under an assumption that all examples are drawn iid from an unknown distribution. These techniques are "training set" based techniques and "test set" based techniques. Each approach has both disadvantages and advantages relative to the other. The message of this paper is that the techniques are *not* exclusive. The two approaches can be freely blended to construct a bound which is "better" (in some sense) than either bound individually.

Test set based techniques include bagging, cross-validation, and the simple holdout set. The primary advantage of test set based techniques is that they work: typically they can be used to give a tight upper bound on the future error rate of some hypothesis. There are, however, several drawbacks to the test set approach. The largest drawback is that data used for testing can not be used for training. This can be a very serious issue when learning problems exhibit "phase transitions" where a few extra examples suddenly make the chosen classifier much more accurate. If these extra examples are in the holdout set, our learning algorithm will produce a poor classifier. Another, drawback of test set based techniques is that they are not always well-analyzed. Of the above approaches, only the behavior of the holdout set is well understood on arbitrary learning algorithms.

Training set based techniques include the famous VC analysis [7], and is the focus of much of the work in computational learning theory. The biggest advantage of training set based techniques is that all examples can be used for both learning and bound construction. The drawback of training set based techniques is that they are often too loose to provide any useful information. Sometimes training set based techniques are actually tighter than test set based techniques. However this has not been used in practice, principally because it is difficult to predict in advance whether or not a training set based technique will be tight enough to yield interesting information.

The combination theorem proved in this paper makes *all* computable training set based true error bound techniques practically useful. The prescription for use is straightforward:

(1) Choose a training set bound, a holdout set, and a combined train set/test set bound.
(2) Train on the training set to choose a classifier.
(3) Evaluate errors on the train and the test sets and use the combined train set/test set bound to report a high-probability bound on the future error rate.

A judicious choice of combined bound can be used with the simple guarantee that the combined bound is never much worse than the lowest of the training set and test set bounds. In addition, we can also guarantee that the combined bound is sometimes better than either individual bound.

The remainder of this paper first develops the combined bound theorem, and then presents empirical results showing how this technique can be useful.

2 How to Combine Training and Testing Bounds

2.1 Setup and Motivation

We will define a classification problem as a distribution, D, on a space $X \times \{0,1\}$. Given a set S of $|S| = m$ draws of labeled examples (x, y) from the distribution D, the goal of any learning algorithm is to choose a hypothesis, $h : X \to \{0,1\}$, with a low true error rate, $e_D(h) = \Pr_{x,y\sim D}(h(x) \neq y)$. Unfortunately, the distribution D is unknown and so the true error rate, $e_D(h)$ is not evaluatable. Nonetheless, it is often possible to bound the true error rate in terms of observable quantities such as the empirical (or training) error $\hat{e}_S(h) = \Pr_{x,y\sim U(S)}(h(x) \neq y)$. Here $U(S)$ is the uniform distribution on the set S. It will be important to make a distinction between train and test sets. Test sets, S_{test} will always have a subscript "test" and the number of test set examples will be denoted by m_{test}. For brevity, we will denote the empirical error on a test set as $\hat{e}_{\text{test}}(h)$.

Given a fixed classifier and a classification problem, the distribution of a test error on m labeled examples is simple. The probability of an error for each example is $e_D(h)$ and independence implies the empirical error will be Binomially distributed. In particular, the Binomial distribution is given by:

$$\mathrm{Bin}(m,k,p) \quad \equiv \quad \sum_{j=0}^{k} \binom{m}{j} p^j (1-p)^{m-j}$$

and note that:

$$\mathrm{Bin}(m,k,p) = \Pr_{S_{\text{test}}\sim D^{m_{\text{test}}}} \left(\hat{e}_{\text{test}}(h) \leq \frac{k}{m_{\text{test}}} | e_D(h) \right)$$

Given a fixed learning algorithm and learning problem, the training error will have a considerably more complicated distribution. We can nonetheless, regard the training error as a fixed random variable which has some cumulative distribution parameterized by many parameters, one of which is the true error rate of the output hypothesis (which is itself a random variable).

How can we construct a confidence interval based upon information from both the training and testing sets? There are several possibilities.

(1) Construct an interval based upon the probability that *both* true error upper bounds are violated.

(2) Construct an interval based upon the probability that at least one of the true error upper bounds is violated.

(3) Use a more general technique.

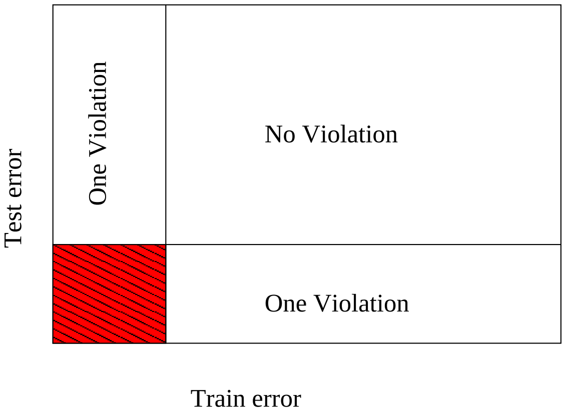

Figure 2.1. This is a graph representing the rejection region for a combined bound based upon both bounds being violated. The marked region is the low probability events disallowed.

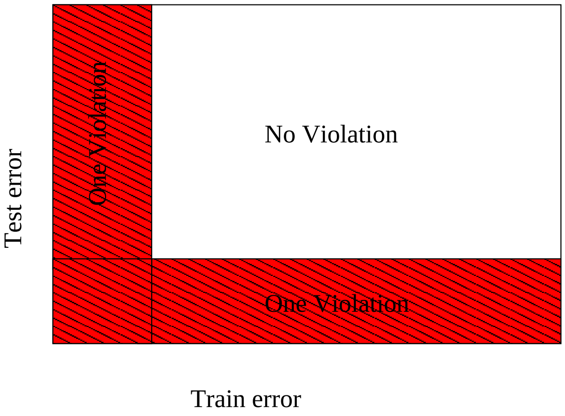

Figure 2.2. This is a graph representing the rejection region for a combined bound based upon either bound being violated. Note that the "width" of the one violation area is smaller than in figure 2.1. This occurs because the probability mass of the rejection region stays constant.

Figure 2.1 represents Technique (1) visually.

The essential problem with technique (1) is that the resulting true error bound is the *maximum* (minus a small amount) of the bounds based upon both the test set and the training set. Given that we don't trust the training set based bound to always be tight, we expect this combination not behave well.

Technique (2) can be seen visually in Figure 2.2.

Technique (2) works moderately well. Mathematically, we can calculate the minimum of the two error bounds and add a small amount. This approach is nearly equivalent to taking a union bound. While this approach allows us to combine the bounds, it does not let us achieve an improvement over either bound individually. This property is intuitively possible. Certainly,

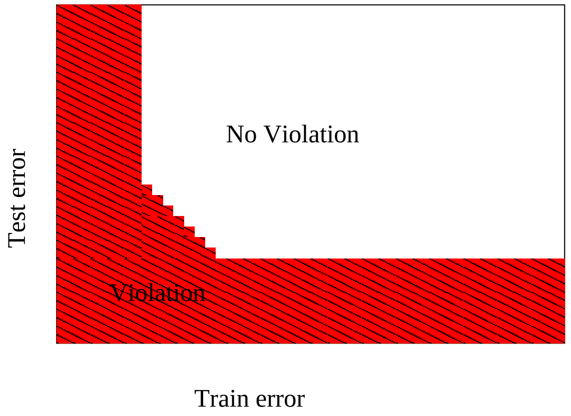

Figure 2.3. This is a graph representing a possible rejection region for some other combined bound.

if we use two test sets, the confidence interval should improve.

A better approach is possible. We would like to construct a combined bound with the form given in Figure 2.3.

Such a rejection region has two important properties:

(1) If one bound is loose, it does not greatly harm the final true error bound.
(2) The final true error bound can be tighter than either individual true error bound.

Showing that technique (2) works is just an application of the union bound. Given any two bounds on the true error rate, we can apportion $\frac{\delta}{2}$ confidence to each bound. Then both bounds will hold with probability δ which implies that the minimum of the two true error bounds (worsened by the substitution $\delta \to \frac{\delta}{2}$) holds.

One interesting possibility to consider is the rejection region of two test set based bound. The *standard* rejection region for the combination of two test set bounds will be a diagonal orthogonal to the identity as in Figure 2.4. A bound based upon a rejection region of this form is desirable because it can result in significant improvements. The combined bound we develop mixes the possible improvement of Figure 2.4 with the soft minimum of 2.2. The soft minimum is necessary in order to protect against the (sometimes large) pessimism of training set based bounds.

2.2 General Approaches for Combined Bounds

Showing that a more general technique works must start with a discussion of confidence intervals. Fundamentally, a bound can be viewed as a set of outcomes. Let X be a space of outcomes, then a bound $\phi \subseteq X$ is a subset. The probability that this generalized bound is violated for some distribution P is given by:

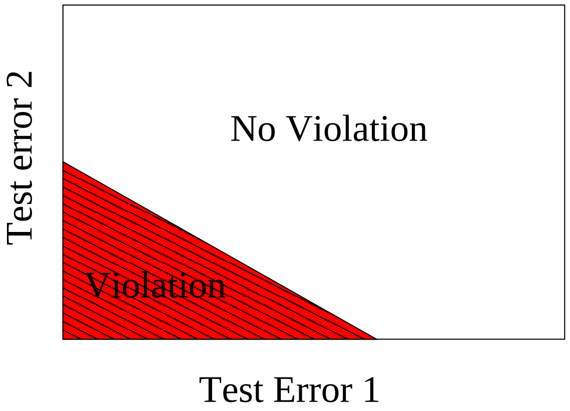

Figure 2.4. This is a graph showing the bound constructed by combining two (asymmetrically sized) test sets. Since the total number of errors is a sufficient statistic for bound calculations, the boundary of the bound violation region is at a constant number of errors which is perpendicular to the identity.

$$\Pr_{x\sim P}(x \in \phi)$$

Note that the rejection region ϕ can be parameterized by both the distribution P and confidence δ to get:

$$\Pr_{x\sim P}(x \in \phi_P(\delta)) \leq \delta$$

We can expand the definition of a bound to include a *randomized* rejection region. In particular, let $\phi_P(w,\delta)$ satisfy:

$$\forall w : \Pr_{x\sim P}(x \in \phi_P(w,\delta)) \leq \delta$$

Then, the following statement holds:

$$\Pr_{w\sim Q, x\sim P}(x \in \phi_P(w,\delta)) \leq \delta$$

In fact, we can make a stronger statement. If

$$E_{w\sim Q} \Pr_{x\sim P}(x \in \phi_P(w,\delta)) \leq \delta \tag{2.1}$$

then

$$\Pr_{w\sim Q, x\sim P}(x \in \phi_P(w,\delta)) \leq \delta \tag{2.2}$$

.

Randomized confidence intervals are useful here because we can regard the draw of the “test” set as constructing a randomized interval for the “train” set. As long as constraint 2.1 is obeyed, the bound will hold with probability at least δ. We can then take a supremum over all classification problems D and the bound will then yield the following theorem:

Theorem 2.1. *(Exact test and train bound) Let $\phi_D(\delta)$ be any bound satisfying:*

$$\forall D\ \forall \delta \geq 0\ \Pr_{S\sim D^m}(S \in \phi_D(\delta)) \leq \delta$$

Let $f(S_{test})$ be any function satisfying:

$$E_{S_{test}\sim D^{m_{test}}} \Pr_{S\sim D^m}(S \in \phi_D(f(S_{test}) * \delta)) \leq \delta$$

then:

$$\Pr_{S,S_{test}\sim D^{m+m_{test}}}(S \in \phi_D(f(S_{test}) * \delta)) \leq \delta$$

Proof. This is a simple application of the definition of probability. □

At this point we are (in some sense) "done"—we can flexibly combine any train set based bound with the holdout bound and verify that it holds using an exact calculation of the expectation,

$$E_{S_{\text{test}}\sim D^{m_{\text{test}}}} \Pr_{S\sim D^m}(S \in \phi_D(f(S_{\text{test}}) * \delta)) \leq \delta$$

There are many possible choices of the function $f(S_{\text{test}})$, each of which leads to a different combined training and testing bound. In particular, there exists an f for the conjunction and disjunction methods—cases (1) and (2) in the previous section. We can also design an f which satisfies our goals of "never much worse and sometimes better".

Nonetheless, some more effort is required because in practice it is sometimes desirable to use a bound for the holdout set rather than an exact calculation of the expectation.

2.3 Approximations in Combinations

The inexact nature of bounds forces us to impose a monotonic structure on the function $f(S_{\text{test}})$. For simplicity, we will restrict to functions of the form $f(\hat{e}_{\text{test}}(h))$ where $\hat{e}_{\text{test}}(h)$ is the test error on hypothesis h. This simplification is not necessary and this technique can be extended to arbitrary test set based techniques.

We can consider any upper bound $\theta_D(\delta)$ on the true error rate, $e_D(h)$, as inducing a cumulative distribution on the test set events.

This cumulative distribution is *not* the cumulative distribution of the underlying (Binomial) probability. To construct this distribution, let:

$$F_\theta(\hat{e}_{\text{test}}(h)) = \inf\{\delta : \hat{e}_{\text{test}}(h) \in \theta_D(\delta)\}$$

Intuitively, $F_\theta(\hat{e}_{\text{test}}(h))$ is the smallest δ such that the test error $\hat{e}_{\text{test}}(h)$ is rejected.

Lemma 2.2. *The function $F_\theta(\hat{e}_{test}(h))$ is a cumulative distribution function.*

Proof. In order to show that the function is a cumulative distribution function, we must show that it varies between 0 and 1 for all values of $\hat{e}_{\text{test}}(h)$. Since $\theta_D(\delta)$ is an upper bound, the following inequality holds:

$$\forall S_{\text{test}}\ F_\theta(\hat{e}_{\text{test}}(h)) \geq \text{Bin}(m, m * \hat{e}_{\text{test}}(h), e_D(h))$$

This inequality implies the value of $F_\theta(\hat{e}_{\text{test}}(h))$ is always at least as large as the CDF of the underlying Binomial distribution. Note, that different S_{test} are implicitly aliased under this technique. We also have the inequality $F_\theta(\hat{e}_{\text{test}}(h)) \leq 1$ because all true error rate upper bounds are vacuous above a true error rate bound of 1. □

We have shown that F_θ is a cumulative distribution function over the value of the empirical error. Given the upper bound cumulative, F_θ, we can look at distributions satisfying:

$$E_{\hat{e}_{\text{test}}(h)\sim F_\theta} \Pr_{S\sim D^m}(S \in \phi(f(\hat{e}_{\text{test}}(h)) * \delta)) \leq \delta$$

If we are guaranteed that $f(\hat{e}_{\text{test}}(h))$ decreases monotonically then equation 2.2 will hold. This is the essence of our theorem.

Theorem 2.3. *(Approximate test and train bound) Let $\phi_D(\delta)$ be any bound satisfying*

$$\forall D\ \forall\delta > 0\ \Pr_{S\sim D^m}(S \in \phi_D(\delta)) \leq \delta$$

. Let $f(\hat{e}_{test}(h))$ be any monotonic decreasing function satisfying:

$$E_{\hat{e}_{test}(h)\sim F_\theta} \Pr_{S\sim D^m}(S \in \phi_D(f(\hat{e}_{test}(h)) * \delta)) \leq \delta$$

, then:

$$\Pr_{S,S_{test}\sim D^{m+m_{test}}}(S \in \phi_D(f(\hat{e}_{test}(h)) * \delta)) \leq \delta$$

Proof. Note that $\Pr_{S\sim D^m}(S \in \phi_D(f(\hat{e}_{\text{test}}(h)) * \delta))$ is a monotonic decreasing function of $\hat{e}_{\text{test}}(h)$. For any monotonic decreasing function $g(x)$ and any two cumulative distribution functions $F_1(x)$ and $F_2(x)$ satisfying $\forall x\ F_1(x) \leq F_2(x)$ we have:

$$E_{x\sim F_1(x)} g(x) \leq E_{x\sim F_2(x)} g(x)$$

Let $F(x)$ be the cumulative distribution of the Binomial and note that the definition of a bound implies: $\forall x\ F_\theta(x) \geq F(x)$. Applying these inequalities, we get:

$$\delta \geq E_{\hat{e}_{\text{test}}\sim F_\theta} \Pr_{S\sim D^m}(S \in \phi_D(f(\hat{e}_{\text{test}}(h)) * \delta))$$

$$\geq E_{\hat{e}_{\text{test}}\sim F} \Pr_{S\sim D^m}(S \in \phi_D(f(\hat{e}_{\text{test}}(h)) * \delta))$$

Given this, an application of theorem 2.1 completes the proof. □

The only constraint that we must check in applying a combined train and test bound is the monotonicity constraint. Heuristically, this is satisfied for the functions graphed in figures 2.1, 2.2, 2.3, and 2.4 since the set of excluded events increases monotonically along the x axis as it decreases along the y axis.

It is worth noticing that this approach applies to *any* training set based bound which holds for all $\delta > 0$ and all learning problems D. In particular, it holds for VC-dimension based bound [7] although we do not use that approach here.

2.4 The train and test bound

An explicit mathematical form for a combined train and test bound can be given by considering the bound-based cumulative distributions, F_θ, and a similar distribution for the training set, F_ϕ. In particular, we can define the rejection region to be

$$\{S_{\text{test}}, S : F_\theta(S_{test})F_\phi(S) \leq t(\delta)\}$$

where $t(\delta)$ is a function satisfying

$$\Pr_{S_{\text{test}},S\sim F_\theta,F_\phi}(F_\theta(\hat{e}_{\text{test}}(h))F_\phi(S)) \leq t(\delta)) \leq \delta$$

. The monotonic constraint is satisfied by this construction because F_θ is implicitly monotonic decreasing with $F_\phi(S)$ given a constant t. We will use this bound for combining train and test sets in the experiments.

Calculation of this bound is straightforward. Essentially, we do a binary search for the true error rate $e(h)$ which places our observed test error and training error on the boundary of the rejection region. Since $e(h)$ is in the interval $[0, 1]$, each tested true error rate will increase the precision of the true error bound calculation by 1 bit. The computation can be halted at the machine precision or some other negligible size.

By observation, this combined bound will satisfy our desirable properties (1) “never hurt much” and (2) “sometimes help”.

3 Experimental Results: Decision Trees

3.1 The Approach and Bounds used

We will test and compare various bounds on an ID3 based decision tree algorithm using discrete datasets from the UCI database of machine learning problems. The exact details of the decision tree and bound implementations are not discussed here although the details are quite important in order to replicate these results. For full details see [2] and note that a true error bound calculation program is available [3].

For every dataset, our goal is to learn a hypothesis h and a high-probability upper bound on the future error rate of that hypothesis. We wish to find a hypothesis with the smallest possible high-probability upper bound. For our purposes, there are 3 varieties of bounds: Training set bounds, test set bounds, and combined training and testing set based bounds. We will compare these bounds in 3 ways:

(1) Test set bounds (figure 3.2): Train on 80% of the data then use the remaining 20% for the test set bound.
(2) Training set bounds (figure 3.3): Train on *all* the data and then calculate the training set based bound.
(3) Train and Test bounds (figure 3.4): Train on 80% of the data then calculate a bound using performance on both the training and the testing sets.

We choose this comparison because it is realistic. Anyone attempting to apply machine learning techniques faces a choice of train set/test set example allocation and the choices made here are not uncommon.

The training set based bounds are the following:

(1) The discrete hypothesis bound such as appears in [6] formed by simply counting the number of decision trees.
(2) The Microchoice bound (first introduced in [4] but see [2] for improvements). The Microchoice bound can (almost) be thought of as a particular choice of description language for the Occam's Razor bound [1].
(3) The Shell bound [5] ([2] for improvements) and the Sampled Shell bound [2]. The shell bound is a functional improvement on the Occam's Razor style bounds which requires significant computation in order to evaluate.

We only use the holdout bound (see [2] chapter 2 for details) for test sets because other test set based bounds lack a solid analysis on decision trees. We also compare two combined bounds:

(1) A combined training and testing bound using the holdout bound and microchoice bound.
(2) A combined training and testing bound using the holdout bound and the (stochastic) shell bound.

The datasets (see figure 3.1 for sizes) consist of 13 UCI database discrete datasets which appeared easiest to use with a decision tree. It is important to note that the problems were not chosen because they optimized particular bounds well, nor were any of the bounds “tuned” to do better on any of these datasets.

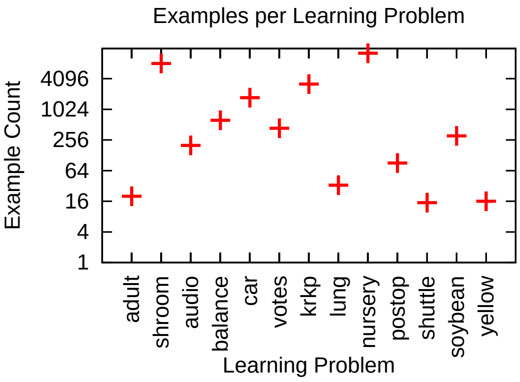

Figure 3.1. A graph showing the number of examples in each problem. Note that holdout sets use only 20% of the available data.

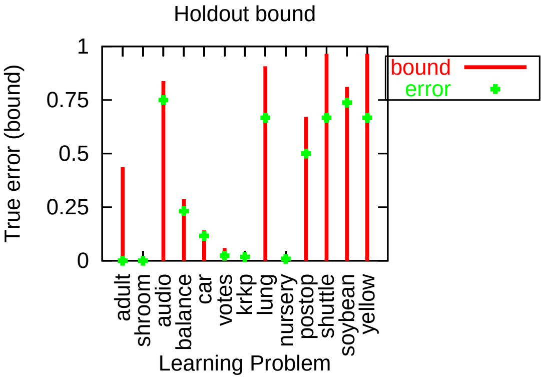

Figure 3.2. This is a graph of the confidence intervals implied by the holdout bound (See the introduction of [2] for details). In this figure (and all others) we use $\delta = 0.1$ bounds for each tail so the confidence intervals hold with probability 0.8 over the data set. Note that an estimate of the size of a the holdout set can be inferred by the width of the confidence interval.

For all bounds, we use a probability of failure of $\delta = 0.1$. The exact data generating Figure 3.3 and Figure 3.4 are listed in the appendix.

3.2 Discussion

It is difficult to answer the question "which bound is tighter?" in a theoretical way. In fact, all of the training set based bounds we use *could* be the "best" depending on the exact learning problem (and algorithm). For example, the Microchoice bound is worse than the Simple bound when the hypothesis chosen happens to be one of the "last" hypotheses with a long description length. The results in Figure 3.3 show there is no total ordering amongst the bounds although there is a noticeable rough ordering:

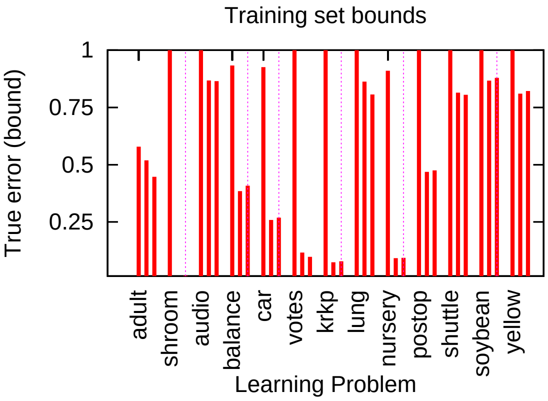

Figure 3.3. This graph compares upper bounds based upon the training set. The left column is a discrete hypothesis bound, the middle column is a microchoice bound, and the right column is a shell bound. The existence of a dashed line implies an approximate calculation of the stochastic shell bound.

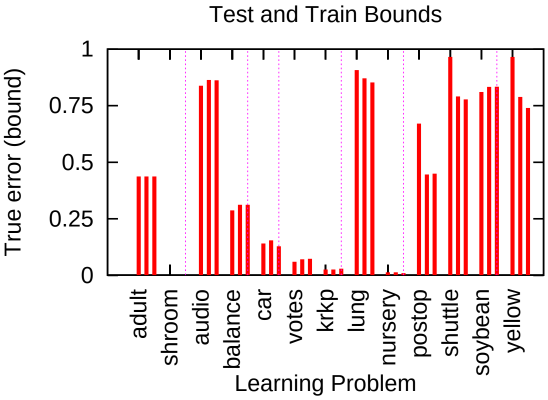

Figure 3.4. This graph compares the true error upper bound for test set (left column) and two combined bounds based upon Microchoice+holdout (middle column) and Shell+holdout (right column) bounds. Once again, a dashed line implies an approximate calculation.

$$\text{Simple} > \text{Microchoice} > \text{Shell} \simeq \text{Holdout}$$

This ordering is approximately as expected based on theoretical considerations. The Simple bound can never be much better than Microchoice bound and the Microchoice bound can be arbitrarily tighter than the Simple bound. A similar statement holds for the Microchoice Bound and the Shell bound. The Shell bound is not always the best, but it does behave well in comparison to the more standard holdout approach.

Empirically, we can observe a very noticeable behavior. For problems with less than 100 examples the training set based bounds are superior to the holdout bound. Between 100 and 1000 examples, the behavior changes with the holdout bound generally winning, although not necessarily by much. Above 1000 examples, the holdout bound is significantly and consistently tighter than the training set based bounds. This behavior is expected because the training set based bounds are typically loose. In particular, the problem of correlated hypotheses has yet to be solved in a convincing manner on discrete hypothesis spaces. Once the holdout set becomes large enough to achieve statistical certainty, the training set based bounds can not compete.

The combined bounds are compared with the holdout bound in 3.4 and the results are the real import of this paper. By comparison, we have the following rough ordering:

$$\text{Holdout} > \text{Holdout} + \text{Micro} > \text{Holdout} + \text{Shell}$$

These improvements are typically most significant when the bound for the test set is weak due to few test examples, but improvement can and does occur even when this is not the case. For example, this occurs with the “shroom” problem.

The combined approach appears to have the best behavior in practice on both large and small datasets. By examination, we can also see that the two guarantees stated in the abstract hold as well:

(1) The combined bound is never much worse than the best bound.
(2) The combined bound is is sometimes (a little) better than either bound.

4 Conclusion

We proved a theorem allowing us to combine train and test set based true error bounds in a general manner. Then, we picked a particular technique for combining the train and test bounds and applied them to a decision trees learned on discrete UCI database problems. The results show that the combined approach is generally better on these experiments.

One significant implication of the combined bound approach is that it makes theoretical improvements in training set based bounds much more relevant for practical use. This is important because it connects significant theoretical work with practical use.

There are several directions of future investigation which could further strengthen any of these approaches. For the training set based bounds, finding a quantitatively useful bound which takes into account correlation between hypotheses remains an open problem. Note that VC-bound and associated covering number approaches [7] address this, but not in a manner that results in a bound which is satisfying for practical use.

We tested the simplest of holdout techniques so another natural extension is to test other holdout techniques. This was not done here, because the theory of these other techniques is lacking and contributions there could be of great import.

The particular method we used to combine training and testing set bounds was a simple choice. It is easy to imagine that other combinations (other choices of the function f) can be beneficial.

References

[1] A. Blumer, A. Ehrenfeucht, D. Haussler, M. Warmuth. “Occam’s Razor.” Information Processing Letters 24: 377-380, 1987.

[2] John Langford, Thesis “Quantitatively Tight Sample Complexity Bounds”, Forthcoming, draft available at: http://www.cs.cmu.edu/~jcl/research/thesis/thesis.ps

[3] John Langford, bound calculation programs, http://www.cs.cmu.edu/~jcl/programs/bound/bound.html

[4] John Langford and Avrim Blum 1999. Microchoice Bounds and Self Bounding learning algorithms. COLT99. http://www.cs.cmu.edu/~jcl/papers/microchoice/mc.ps

[5] John Langford and David McAllester, “Computable Shell Decomposition Bounds” COLT 2000.

[6] Tom Mitchell, “Machine Learning” McGraw Hill 1997.

[7] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probab. and its Applications, 16(2):264-280, 1971.

5 Appendix: Exact results

Here we present the exact results behind the graphs. This first table lists the upper bounds associated training set bounds and the holdout set bound.

Problem	Number	Simple	Micro	Shell	Holdout
adult	20	0.579	0.605	0.447	0.438
shroom	8124	1.0	0.083	0.016	0.0014
audio	200	1.0	0.922	0.865	0.838
balance	625	0.933	0.405	0.409	0.287
car	1728	0.926	0.303	0.269	0.141
votes	435	1.0	0.180	0.097	0.06
krkp	3196	1.0	0.171	0.079	0.026
lung	32	1.0	0.963	0.051	0.907
nursery	12960	0.91	0.181	0.093	0.013
postop	90	1.0	0.498	0.475	0.671
shuttle	15	1.0	0.908	0.805	0.965
soybean	307	1.0	0.945	0.879	0.811
yellow	16	1.0	0.902	0.822	0.965

The next table presents the results of holdout and combined bounds.

Problem	Holdout	Holdout+Micro	Holdout+Shell
adult	0.438	0.438	0.438
shroom	0.0014	0.0014	0.0014
audio	0.838	0.864	0.862
balance	0.287	0.313	0.313
car	0.141	0.155	0.129
votes	0.06	0.072	0.074
krkp	0.026	0.026	0.030
lung	0.907	0.871	0.853
nursery	0.013	0.013	0.0100
postop	0.671	0.446	0.450
shuttle	0.965	0.791	0.778
soybean	0.811	0.833	0.834
yellow	0.965	0.789	0.740