Quantitatively Tight Sample Complexity Bounds

John Langford

I present many new results on sample complexity bounds (bounds on the future error rate of learning algorithms). Of theoretical interest are qualitative and quantitative improvements in sample complexity bounds as well as some techniques and criteria for judging the tightness of sample complexity bounds.

On the practical side, I show quantitatively interesting results applying these sample complexity bounds to real world problems and present a technique for using both sample complexity bounds and (more traditional) holdout techniques. Code for calculating these bounds is provided.

CHAPTER 1

Introduction

What is a sample complexity bound? Informally, it is a bound on the number of examples required to learn a function. Therefore, in order to motivate the use of a sample complexity bound, we must first motivate the learning problem.

1.1. The learning problem

What is learning? Learning is the process of discovering relationships between events. Knowledge of the relationships between events is of great importance in coping with the environment. Naturally, humans are incredibly good at the process of learning—so good that we sometimes do not even realize when it is hard.

The learning problem which we focus on is learning for computers. In particular, "How can a computer learn?" A few examples are illustrative:

1. How can we make a computer with a microphone output a text version of what is being spoken?
2. How can we make a computer with a camera recognize John?
3. How can we make a computer controlling a robot arrive at some location?

1.2. The problem with the learning problem

The learning problem, as stated, is somewhat ill-posed. There are some very obvious ways for a computer to learn—for example by memorization. The difficulty arises when memorization is too expensive. Essentially, the real learning problem becomes, "How can a computer learn given incomplete information?" This formulation of the learning problem gives rise to a new problem - quantifying the amount of information required to learn. Sample complexity bounds address this second question: "When can a computer learn?"

At this point, it is worth mentioning that in some cases learning is essentially hopeless. Even when relations exist they can be extremely difficult to learn. The prime example of this is provided by cryptography where a common task is to work out functions for which it is not feasible to predict the input given the output.

1.3. A plethora of learning models

There are several possible learning models which can be divided along several axis. Our first axis is the sort of information given to the learning algorithm. There are several possibilities:

1. Examples: Vectors of observations.
2. Partial relations: partial relations between events.

We will assume that just examples are available as a lowest common denominator amongst learning problems. It is worth noting that this does not preclude the use of other forms of information which could be much more powerful then mere examples.

Another important axis is the level of helpfulness in the world. Is the world making it hard to learn? Is the world making it easy to learn? Or is the world oblivious?

(1) Teacher: The teacher model is a “best case” model. Here, we assume that the world is providing the best examples possible in order to learn a relationship.
(2) Oblivious: The oblivious model is an “in between” model where we assume that the world doesn’t care whether or not we learn. Examples are picked in some unspecified manner and our goal is to do an ’average case’ analysis.
(3) Opponent: The opponent model is a “worst case” model. Here, we assume that world is choosing examples in way which minimize our chance of learning.

Clearly, the strongest form of learning is learning in the opponent model, because if something is learnable in the opponent model, then it is learnable in the oblivious model. The same relationship also holds for oblivious and teacher models. We will work in an oblivious model where examples are chosen by uncaring nature. Why the oblivious model? Aside from the intractability of analysis in an opponent model, we expect that most learning problems actually are oblivious: we have neither an active teacher nor an active opponent. Thus an analysis in the oblivious model will be directly applicable to many learning problems.

We have committed to an oblivious model with examples as our source of information. With these two questions decided all the remaining questions will essentially be decided in favor of simplicity. There are two more very important questions to decide. The first is: does our algorithm get to pick the examples or are the examples picked for us?

(1) Active learning: The learning algorithm chooses a partial example and the remainder is filled in by nature.
(2) Inactive learning: The learning algorithm is simply given examples.

Active learning (aka experimental science) is inherently more powerful then inactive learning. Despite this, we will choose to work with an inactive learning model, principally because analyzing the active learning setting in a generic manner appears very difficult.

Our plan is to focus on an oblivious model with examples chosen by the world. The remaining question is: Do we know which relation we want to learn? The two possibilities are:

(1) Supervised learning: We want to learn to model an output in terms of an input.
(2) Unsupervised learning: We want to learn to model an arbitrary portion of the features in terms of other features.

Again, we will choose the simple thing: supervised learning.

The question we want to answer then is, “When is supervised learning in an oblivious model with examples chosen by the world feasible?”. It is not time to quantify this question.

1.4. The oblivious inactive supervised learning model

Oblivious will be modeled by an unknown distribution D over examples. The fundamental assumption we will make in all of our sample complexity bounds is that all examples are drawn independently from the unknown distribution D. This assumption must be stated explicitly and always kept in mind when considering the relevance of sample complexity bounds.

AXIOM 1.4.1. *All examples are drawn independently from an unknown distribution D.*

With the exception of this assumption, all of the other parameters in our bounds will be verifiable at the time the bound is applied.

Since this is a supervised learning model, all of our examples will split into two parts, (x, y) where x is the "input" and y is the "output" (the thing we wish to predict). For simplicity, we will typically work with theorems for binary valued y. This choice is not critical to the idea of a sample complexity bound—it is just made for simplicity of presentation.

The number of samples, m, required for learning is the fundamental quantity we will be concerned with. In particular, we will not be concerned with the time complexity or the space complexity of learning algorithms. This choice is made for the purposes of simplicity and implies the relationship between sample complexity bounds and learning algorithms will be similar to the difference between information theory and coding information for transmission across a noisy channel.

Any learning algorithm must output some hypothesis, h, for predicting the output given the input. This hypothesis is essentially a program which, given the input, predicts the output. The hypothesis may or may not be randomized.

The next item to quantify is learning. When has learning occurred? We will say that learning has occurred when the *true error* is less then a random hypothesis would give us. The true error $e_D(h)$ is defined in the following way:

$$e_D(h) = \Pr_D(h(x) \neq y)$$

Unfortunately, the true error is not an observable quantity in our model because the distribution, D, is unknown. However, there is a related quantity which is observable. Given a sample set S, the *empirical error*, $\hat{e}_S(h)$ is defined similarly as:

$$\hat{e}_S(h) = \Pr_S(h(x) \neq y) = \frac{1}{m}\sum_{i=1}^{m} I(h(x_i) \neq y_i)$$

Here $\Pr_S(...)$ is a probability taken with respect to the uniform distribution over the set of examples.

1.5. Questions we can answer

Our real goal in learning theory is to answer the question "When can we learn?" Unfortunately, there is no good answer to this question in the model we have chosen. There exist learning problems for which it is essentially impossible to learn. The simplest example of such a learning problem is the case of a distribtion D which always flips a coin in deciding the value of the output y. The bias of the coin is the same irrespective of the input x. Since the value of the input is explictly independent of the output we surely can not hope to learn a useful relation between the input and output.

We will instead focus on a different question: “Have we learned?” This question *is* answerable in a probabilistic manner. In particular we can make a statement such as “With high probability over samples drawn from D we have learned if the empirical error is less then some value.” In practice, we will want to know how *much* we have learned which we can do by providing a high confidence bound on the true error rate of the learned hypothesis.

The question answered here differs significantly from the learning theory results of Vapnik [] and many others which address the question “When is learning possible?” We instead address “Have we learned?” because this question appears to be more mathematically elegant to answer. The most significant difference in answer “Have we learned?” is that it is unnecessary to make assumption about the learning algorithm. Our results will apply to all possible learning algorithms and thus are more similar to informatiuon theory than complexity theory.

1.6. Overview of the document

This document is primarily about the theory of sample complexity for answering the question “Have we learned?”. However, we do not neglect the experimental side. In particular, following the theory we will present results for application of sample complexity bounds to machine learning problems. These results are the ’best known results’ in terms of bound tightness and should be considered as a guide and challenge to others working on sample complexity bounds.

All of the sample complexity bounds presented here will fall within the paradigm of classical (non-bayesian) statistics. This is not meant to scare away bayesians or bayesian analysis. In particular, it’s worth noting that we will consider the use of a ’prior’ and a ’posterior’ (in a *classical* manner) within these bounds.

In order to make this thesis more coherent, previous work (of which there is quite a bit) will be integrated into the presentation rather then seperated into a section of it’s own. Credit will be given at the time the work is introduced.

Our first section covers the theory of sample complexity bounds. This section will be split into three basic parts:

1. Simple observations on the problem at hand.
2. Sample complexity bounds where all the data is used for training.
3. Holdout bound techniques where only some of the data is used for training.

CHAPTER 2

Basic Observations

The observable quantity here is the empirical error rate ($\hat{e}_S(h)$) of a hypothesis. What is the distribution of the empirical error rate for a fixed hypothesis? For each example, we know that the probability that the hypothesis will err is given by true error rate, $e_D(h)$. This can be modelled by a biased coin flip: heads if you are wrong and tails if you are right.

Let us call the bias of the coin $p = e_D(h)$. What then is the probability of observing k errors out of m coin flips? This is a very familiar distribution in statistics called the Binomial distribution. Let $\hat{p}$ be the observed rate of heads. The Binomial distribution is given by:

$$\Pr_{\{0,1\}^m}(\hat{p} = \frac{k}{m}|p) = \binom{m}{k} p^k (1-p)^{m-k}$$

Here we use 'choose' notation defined by $\binom{m}{k} = \frac{m!}{(m-k)!k!}$.

2.1. The Basic Building Block

Our real interest will be captured by Binomial tails because we wish to bound the probability of observing a misleadingly small event. The probability of a Binomial tail is just the cumulative distribution function:

$$\Pr_{\{0,1\}^m}(\hat{p} \leq \frac{k}{m}|p) = \sum_{j=1}^{k} \binom{m}{j} p^j (1-p)^{m-j} \equiv \text{Bin}(m,k,p)$$

2.2. Binomial Tail calculation techniques

One fundamental question arises: How quickly can we calculate the binomial coefficient, $\binom{m}{k}$? The answer is: not very fast. It is an open problem to calculate $\binom{m}{k}$ in time not exponential in $\log m$ and $\log k$ (the representation length of m and k). In general, this problem is intractable. For example, we note that

$$\binom{m}{\frac{m}{2}} \simeq \frac{2^m}{\sqrt{m}}$$

which has an asymptotic representation length of $O(m)$ for any p.

Our real goal is not merely calculating binomial coefficients but rather calculating the probability of a tail, $\text{Bin}(m,k,p)$. How can we calculate the tail probability quickly? For all approaches, it is necessary to calculate $\log \text{Bin}(m,k,p)$ rather then $\text{Bin}(m,k,p)$ to avoid underflow issues.

There are several possible approaches of increasing sophistication:

(1) Calculate $\log \binom{m}{i}$ independently from $i = 0$ to $i = k$ and use the results in the sum, $\log \sum_{i=0}^{k} \binom{m}{i} p^i (1-p)^{m-i}$

(2) Calculate Pascal's triangle and extract the Binomial coefficients.

(3) Use the fact that

$$\binom{m}{i+1} = \frac{m!}{(m-i-1)!(i+1)!}$$

$$= \frac{m!}{(m-i)!i!}(m-i)(i+1) = (m-i)(i+1)\binom{m}{i}$$

to calculate each of the coefficients in sequence.

(4) Calculate $\binom{m}{k}$ directly and then $\binom{m}{i-1}$ given $\binom{m}{i}$ until the quantity $\binom{m}{i}p^i(1-p)^{m-i}$ falls below the machine precision.

Approaches (1) and (2) both require $O(m^2)$ work while approaches (3) and (4) require merely $O(m)$ work. We will use approach (4) here. Yet, as noted in the beginning of this section, $O(m)$ is still too expensive for us. Luckily, there exist some quick approximations which can reduce the computation to constant time.

2.3. Approximation techniques

Exact calculation of $\text{Bin}(m, k, p)$ can require computation at least proportional to m, which is often too expensive. In practice we will only need to calculate an upper bound on the quantity $\text{Bin}(m, k, p)$. There are several inequalities which are often used. The first of these is the Hoeffding inequality. Assume that $\frac{k}{m} < p$ then we have:

$$\text{Bin}(m, k, p) \leq e^{-2m(p-\frac{k}{m})^2}$$

This inequality essentially arises from fitting a gaussian to the Binomial distribution with maximum variance. For any particular m, the variance of the Binomial distribution is maximised when $p = \frac{1}{2}$. Therefore, the Hoeffding inequality is tight when $p = \frac{1}{2}$. Unfortunately, the Hoeffding approximation is not tight enough for our purposes. In machine learning, our goal is to find a hypothesis with a true error rate far away from $\frac{1}{2}$ where the Hoeffding inequality becomes loose.

There is another bound known as the "realizable bound" which applies only when the $k = 0$. The realizable bound is:

$$\text{Bin}(m, 0, p) = (1-p)^m \leq e^{-mp}$$

The realizable bound is noticeably tighter with an exponent proportional to $(p-\frac{k}{m})$ rather then $(p-\frac{k}{m})^2$. The disadvantage of the realizable bound is that it only applies to a very limited setting - when our empirical error rate happens to be 0.

Luckily, there exists a quickly calculatable bound which achieves the generality of the Hoeffding bound along with the tightness of the realizable bound. We have the relative Chernoff bound for $\frac{k}{m} < p$:

$$\text{Bin}(m, k, p) \leq e^{-m\text{KL}(\frac{k}{m}||p)}$$

Here $\text{KL}(q||p) = q \ln \frac{q}{p} + (1-q) \ln \frac{(1-q)}{(1-p)}$ is the KL-divergence between a coin of bias q and another coin of bias p. The relative Chernoff bound is as tight as the Hoeffding bound when p is near $\frac{1}{2}$ and as tight as the realizable bound when $k = 0$.

(show pretty graph)

We are concerned with the different bounds here because much of the learning theory literature works with either the realizable bound or the Hoeffding bound, or both. In contrast, we will work with either the relative Chernoff bound or the exact tail probability, $\mathrm{Bin}\,(m,k,p)$. There are several advantages to this approach:

(1) Sometimes, a different approach to producing a bound will appear better then previous approaches, but the apparent benefit can simply be traced to the use of a tighter bound on $\mathrm{Bin}\,(m,k,p)$.
(2) The bounds presented here will all be immediately applicable to direct calculation.
(3) We avoid the need to state two versions of the same theorem: once for the realizable (0 empirical error) case and once for the agnostic (arbitrary empirical error) case.

The principle *dis*advantage of this approach is that both the relative Chernoff bound and $\mathrm{Bin}(m,k,p)$ are not analytically invertible. In particular, we can not discover $\max_p \mathrm{Bin}(m,k,p)] \leq \delta$ via an analytic calculation. This is not a severe computational disadvantage because the quantity $\mathrm{Bin}\,(m,k,p)$ is convex in p implying that a binary search is capable of solving the inequality. The process of (and need for) inversion is discussed next.

2.4. Converting to a P-value approach

When making judgements about which hypothesis to choose, the relevant quantity is *not* the probability of error as we calculate above. Instead, it is a bound on the true error rate which holds with high probability over draws of the sample set. We might decide that $\delta = 0.05$ was an acceptable error rate and then ask ourselves, "What is a bound on the true error rate that holds with probability 0.05?"

Functionally, instead of calculating:

$$\mathrm{Bin}(m,k,p) = \delta$$

we want to invert the output, δ, with respect to the input, p. Since p and δ are simply related to each other, this inversion can be defined as:

$$\bar{e}(m,k,\delta) \equiv \max_p p : \ \mathrm{Bin}(m,k,p) \geq \delta$$

What is the interpretation of $\bar{e}$? The inversion $\bar{e}(m,k,\delta)$ is a high confidence bound on the true error rate. With probability at least $1-\delta$, the true error rate will be less then $\bar{e}(m,k,\delta)$. This is exactly the kind of quantity that we desire in making decisions about which hypothesis is more desirable.

2.5. Bounding the Union

One last very common technique we will use is the union bound. Given two coins, each with a bias of p, what is the probability that if we flip each coin m times, one of the coins will have k or fewer heads?

Let $X_1 =$ the proportion of heads in the first coin flip and $X_2 =$ the proportion of heads in the second coin flip. Then we get:

$$\Pr(X_1 \leq \frac{k}{m} \text{ or } X_2 \leq \frac{k}{m})$$

$$\leq \Pr(X_1 \leq \frac{k}{m}) + \Pr(X_2 \leq \frac{k}{m})$$

where the inequality is known as the union bound. This approach is very powerful because it works even when the values of X_1 and X_2 are correllated in arbitrary ways.

CHAPTER 3

Simple Sample Complexity bounds

In order to discuss the improved shell bounds, we must first present some basic bounds. The most basic of sample complexity bounds is the simple combination of a Binomial tail bound and the union bound. In particular, we have:

THEOREM 3.0.1. *(Discrete Hypothesis Sample Complexity Bound) For all hypothesis spaces, H, for all $\delta > 0$,*

$$\Pr_D(\exists h \in H: \; e(h) > \bar{e}(m, \hat{e}(h), \frac{\delta}{|H|})) \leq \delta$$

where $\bar{e}(m, \frac{k}{m}, \delta) \equiv \max_p\{p: \; Bin(m,k,p)\} \geq \delta$

PROOF. For every individual hypothesis, we know that:

$$\Pr_D(e(h) > \bar{e}(m, \hat{e}(h), \delta)) \leq \delta$$

Applying the union bound for every hypothesis gives us the result. □

The Discrete Hypothesis bound is seldom interesting in practice because it does not take into account the fact that we care about deviations for some hypotheses more than for others. One way of formalizing this is with the "Occam's Razor Bound".

THEOREM 3.0.2. *(Occam's Razor Bound, []) For all hypothesis spaces, H, for all "priors" p over the hypothesis space, H, for all $\delta > 0$,*

$$\Pr_D(\exists h \in H: \; e(h) > \bar{e}(m, \hat{e}(h), \delta p(h))) \leq \delta$$

where $\bar{e}(m, \frac{k}{m}, \delta) \equiv \max_p\{p: \; Bin(m,k,p)\} \geq \delta$

PROOF. The proof again starts with the basic observation that:

$$\Pr_D(e(h) > \bar{e}(m, \hat{e}(h), \delta)) \leq \delta$$

then, we apply the union bound in a *nonuniform* manner. In particular, we allocate confidence $\delta p(h)$ to hypothesis h. Since p is normalized, we know that

$$\sum_h \delta p(h) = \delta$$

which implies that the union bound completes the proof. □

The Occam's razor bound can often be interesting for discrete learning algorithms which include decision lists and decision trees. The next bound will discuss an improved version of the Occam's razor bound.

CHAPTER 4

Microchoice Bounds (the algebra of choices)

4.1. Introduction

Sample complexity bounds tend to be too loose for application to real world learning problems using common learning algorithms. They require more examples than experimentally necessary to guarantee a reasonable accuracy with a reasonable probability. Chief amongst the reasons for this failure is that typical sample complexity bounds are worst-case in nature and do not take into account what may turn out to be a fortunate target function and/or data distribution. A tighter bound can perhaps be derived by taking properties of the observed data into account.

Many learning algorithms work by an iterative process in which they take a sequence of steps each from a small set of choices (small in comparison to the overall hypothesis set size). Local optimization algorithms such as hill-climbing or simulated annealing, for example, work in this manner. Each step in a local optimization algorithm can be viewed as making a choice from a small set of possible steps to take. If we take into account the number of choices made and the size of each choice set, can we produce a tighter bound?

We develop a bound we call the Microchoice bound, which is a natural approach to analyzing algorithms of this form and lends itself to the construction of Self bounding Learning algorithms in the style of Freund [**?**] in a straightforward way. The goal is to allow algorithms to bound the difference between the true and empirical errors of the output hypothesis based on the choices actually taken and the choice sets actually explored without needing to consider sets that "might have been" visited for different learning problems. By keeping track of the sizes of the choice sets visited, the bound can be calculated and output along with the answer. The simpler version of the Microchoice bound, which we describe first, can be thought of as a direct application of the Occam's razor connection developed by [**?**]. In particular, the idea is to use the learning algorithm itself to define a description language for hypotheses, so that the description length of the hypothesis actually produced gives a bound on the estimation error. In the second half of this paper, we show how the Microchoice bound can be combined with the query-tree approach of Freund to produce a variant of the query tree that can be used much more efficiently. Our procedures work without altering the final hypothesis produced by the learning algorithm. It is also possible to modify the learning algorithm to use a current bound in order to implement early stopping in a manner similar to that of dynamic Structural Risk Minimization [**?**].

Self bounding Learning algorithms were first suggested by Freund [**?**]. Work developing approximately Self Bounding Learning algorithms was also done by Domingos [**?**].

4.2. A Motivating Observation

For a given learning algorithm A, a distribution P on labeled examples induces a distribution $p(h)$ over the possible hypotheses $h \in H$ produced by algorithm A after N examples[1]. If we knew P, then we could apply the Realizable Folk Theorem with the measure $m(h) = p(h)$. This choice of $m(h)$ is optimal for minimizing the expected value of our true error bound, $\epsilon(h)$. In particular, notice that

$$\inf_m \sum_{h\in H} p(h)\frac{1}{N}\left(\ln\frac{1}{\delta} + \ln\frac{1}{m(h)}\right)$$

$$= \frac{1}{N}\left(\ln\frac{1}{\delta} + \inf_m \sum_{h\in H} p(h)\ln\frac{1}{m(h)}\right)$$

This corresponds to minimizing the Kullback-Liebler divergence which provides a solution of $m(h) = p(h)$.

Interestingly, for the agnostic bound this choice of measure, $m(h) = p(h)$, does not minimize the expected value of $\epsilon(h)$, as can be seen from a quick analysis of the two hypothesis case. Instead, this choice of measure minimizes the expected value of $\epsilon(h)^2$.

$$\inf_m \sum_{h\in H} p(h)\epsilon(h)^2 = \inf_m \sum_{h\in H} p(h)\frac{1}{2N}(\ln\frac{1}{\delta} + \ln\frac{1}{m(h)})$$

From here, the analysis is the same as for the realizable case.

The point of these observations is to notice that if we can use the structure of the learning algorithm to produce a choice of $m(h)$ that approximates $p(h)$, this should result in better estimation bounds.

4.3. The Microchoice Bound

The Microchoice bound is essentially a compelling and easy way to compute the selection of the measure $m(h)$ for learning algorithms that operate by making a series of small choices. In particular, consider a learning algorithm that works by making a sequence of choices, $c_1, ..., c_d$, from a sequence of choice sets, $C_1, ..., C_d$, finally producing a hypothesis, $h \in H$. Specifically, the algorithm first looks at the choice set C_1 and the data z^N to produce choice $c_1 \in C_1$. The choice c_1 then determines the next choice set C_2 (different initial choices produce different choice sets for the second level). The algorithm again looks at the data to make some choice $c_2 \in C_2$. This choice then determines the next choice set C_3, and so on. These choice sets 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 containing some choice set C has branching factor $|C|$. Depending on the learning algorithm, subtrees of the overall tree may be identical. We address optimization of the bound for this case later. Eventually there is a final choice leading to a leaf, and a single hypothesis is output.

For example, the decision list algorithm of Rivest [**?**] applied to a set of n features uses the data to choose one of $4n + 2$ rules (e.g., "if $\bar{x}_3$ then $-$") to put at the top. Based on the choice made, it moves to a choice set of $4n - 2$ possible rules

[1] $p(h)$ is the probability over draws of examples and any internal randomization of the algorithm.

to put at the next level, then a choice set of size $4n-6$, and so on, until eventually it chooses a rule such as "else $+$" leading to a leaf.

The way we determine $m(h)$ (equivalently, $\delta(h)$) is as follows. We take our "supply" of δ and give it to the root of the choice tree. The root takes its supply and splits it equally among all its children. Recursively, each child then does the same: it takes the supply it is given and splits it evenly among its children, until all of the supplied confidence parameter is allocated among the leaves. If we examine some leaf containing a hypothesis h, we see that this method gives it $\delta(h) = \delta \prod_{i=1}^{d(h)} \frac{1}{|C_i(h)|}$, where $d(h)$ is the depth of h in the choice tree and $C_1(h), C_2(h), \ldots, C_{d(h)}(h)$ is the sequence of choice sets traversed on the path to h. Equivalently, we allocate $m(h) = \prod_{i=1}^{d(h)} \frac{1}{|C_i(h)|}$ measure to each h. Note it is possible that several leaves will contain the same hypothesis h, and in that case one should really add the allocated measures together. However, we will neglect this issue, implying that the Microchoice bound will be unnecessarily loose for learning algorithms which can arrive at the same hypothesis in multiple ways. The reason for neglecting this is that now, $m(h)$ is something the learning algorithm itself can calculate by simply keeping track of the sizes of the choice sets it has encountered so far. It is important to notice that this construction is defined before observing any data. Consequently, every hypothesis has some bound associated with it before the data is used to pick a particular hypothesis and its corresponding bound.

Another way to view this process is that we cannot know in advance which choice sequence the algorithm will make. However, a distribution P on labeled examples induces a probability distribution over choice sequences, inducing a probability distribution $p(h)$ over hypotheses. Ideally we would like to use $m(h) = p(h)$ in our bounds as noted above. However, we cannot calculate $p(h)$, so instead, our choice of $m(h)$ will be just an estimate. In particular, our estimate $m(h)$ is the probability distribution resulting from picking each choice uniformly at random from the current choice set at each level (note: this is different from picking a final hypothesis uniformly at random). I.e., it can be viewed as the measure associated with the assumption that at each step, all choices are equally likely.

We immediately find the following theorems.

THEOREM 4.3.1. *(Realizable Microchoice Theorem) For any* $\delta \in (0,1)$,

$$\Pr_{z^N}\left[\exists h \in H : E(h, z^N) = 0 \textit{ and } E_P(h) > \epsilon(h)\right] < \delta$$

$$\textit{for} \quad \epsilon(h) = \frac{1}{N}\left(\ln\frac{1}{\delta} + \sum_{i=1}^{d(h)} \ln|C_i(h)|\right)$$

Proof. Specialization of the Realizable Folk Theorem.

THEOREM 4.3.2. *(Agnostic Microchoice Theorem) For any* $\delta \in (0,1)$,

$$\Pr_{z^N}\left[\exists h \in H : E_P(h) > E(h, z^N) + \epsilon(h)\right] < \delta$$

$$\textit{for} \quad \epsilon(h) = \sqrt{\frac{1}{2N}\left(\ln\frac{1}{\delta} + \sum_{i=1}^{d(h)} \ln|C_i(h)|\right)}$$

Proof. Specialization of the Agnostic Folk Theorem.

The point of these Microchoice bounds is that the quantity $\epsilon(h)$ is something the algorithm can calculate as it goes along, based on the sizes of the choice sets encountered. Furthermore, in many natural cases, a "fortuitous distribution and target concept" corresponds to a shallow leaf or a part of the tree with low branching, resulting in a better bound. For instance, in the Decision List case, $\sum_{i=1}^{d(h)} \ln |C_i(h)|$ is roughly $d \log F$ where d is the length of the list produced and F is the number of features. Notice that $d \log F$ is also the description length of the final hypothesis produced in the natural encoding, thus in this case these theorems yield similar bounds to Occam's razor or SRM.

More generally, the Microchoice bound is similar to Occam's razor or SRM bounds when each k-ary choice in the tree corresponds to $\log k$ bits in the natural encoding of the final hypothesis h. However, sometimes this may not be the case. Consider, for instance, a local optimization algorithm in which there are n parameters and each step adds or subtracts 1 from one of the parameters. Suppose in addition the algorithm knows certain constraints that these parameters must satisfy (perhaps a set of linear inequalities) and the algorithm restricts itself to choices in the legal region. In this case, the branching factor, at most $2n$, might become much smaller if we are "lucky" and head towards a highly constrained portion of the solution space. One could always reverse-engineer an encoding of hypotheses based on the choice tree, but the Microchoice approach is much more natural.

There is also an opportunity to use *a priori* knowledge in the choice of $m(h)$. In particular, instead of splitting our confidence equally at each node of the tree, we could split it unevenly, according to some heuristic function g. If g is "good" it may produce error bounds similar to the bounds when $m(h) = p(h)$. In fact, the method of section 4.5 where we combine these results with Freund's query-tree approach can be thought of along these lines.

4.3.1. Examples. The Microchoice Bound is not always better than the simple sample complexity bound (**??**). To develop some understanding of how they compare we consider several cases.

4.3.1.1. *Greedy Set Cover.* Consider a greedy set cover algorithm for learning an OR function over F boolean features. The algorithms begins with a choice space of size $F+1$ (one per feature or halt) and chooses the feature that covers the most positive examples while covering no negative ones. It then moves to a choice space of size F (one per feature remaining or halt) and chooses the best remaining feature and so on until it halts. If the number of features chosen is k then the Microchoice bound is:

$$\epsilon(h) = \frac{1}{N}\left(\ln\frac{1}{\delta} + \sum_{i=1}^{k} \ln(F-i+2)\right) \leq \frac{1}{N}\left(\ln\frac{1}{\delta} + k\ln(F+1)\right)$$

The bound of (**??**) is:

$$\epsilon = \frac{1}{N}\left(\ln\frac{1}{\delta} + F\ln 2\right).$$

If k is small, then the Microchoice bound is a lot better, but if $k = O(F)$ then the Microchoice bound is slightly worse. Notice that in this case the Microchoice bound is essentially the same as the standard Occam's razor analysis when one uses $O(\ln F)$ bits per feature to describe the hypothesis.

4.3.1.2. *Decision Trees.* Decision trees over discrete sets (say, $\{0,1\}^F$) are another natural setting for application of the Microchoice bound.

A decision tree differs from a decision list in that the size of the available choice set is larger due to the fact that there are multiple nodes where a new test may be applied. In particular, for a decision tree with K leaves at an average depth of d, the choice set size is $K(F-d)$, giving a bound noticeably worse than the bound for the decision list. This motivates a slightly different decision algorithm which considers only one leaf node at a time. The algorithm adds a new test or decides to never add a new test at this node. In this case, there are $(F-d(v)+1)$ choices for a node v at depth $d(v)$, implying the bound:

$$E_P(h) \le E(h,z^N) + \sqrt{\frac{1}{2N}(\ln\frac{1}{\delta} + \sum_v \ln(F-d(v)+1))} \tag{4.3.1}$$

where v ranges over the nodes of the decision tree.

4.3.2. Pruning. Decision tree algorithms for real-world learning problems often have some form of "pruning" as in [**?**] and [**?**]. The tree is first grown to full size producing a hypothesis with minimum empirical error. Then the tree is "pruned" starting at the leaves and progressing up through the tree towards the root node using some test for the significance of an internal node. An internal node is not significant if the reduction in total error is small in comparison to the complexity of its children. Insignificant internal nodes are replaced with a leaf resulting in a smaller tree.

Microchoice bounds have the property that they incidentally prove a bound for every decision tree which can be found by pruning internal nodes. In particular, one of the choices available when constructing a node is to make the node a leaf. Therefore, if we begin with the tree T and then prune to the smaller tree T', we can apply the bound (4.3.1) to T' *as if* the algorithm had constructed T' directly rather than having gone first through the tree T. This suggests another possible pruning criterion: prune a node if the pruning would result in an improved Microchoice bound. That is, prune if the increase in empirical error is less than the decrease in $\epsilon(h)$. The similarities to SRM are discussed below.

4.4. Relationship with Structural Risk Minimization

Structural Risk Minimization works with a sequence of nested hypothesis sets, $H_1 \subset H_2 \subset \ldots. \subset H_l$. For each hypothesis set, a PAC bound on the difference between empirical and true error exists.

Theorem 4.4.1. *(Standard Sample Complexity Bound)*

$$\Pr_{z^N}\left[\exists h \in H_i : E_P(h) > E(h,z^N) + \epsilon(h)\right] < \delta$$

$$\textit{for} \quad \epsilon(h) = \sqrt{\frac{1}{2N}\left(\ln\frac{1}{\delta} + \ln|H_i|\right)}$$

The simplest way to improve the bound to include all hypothesis sets is with the following simple theorem.

THEOREM 4.4.2. *(Structural Risk Minimization)Let $m(i)$ be some measure across the l hypothesis sets with $\sum_{i=1}^{l} m(i) = 1$. Then:*

$$\Pr_{z^N}\left[\exists h \in H_i \in H_1, ..., H_l : E_P(h) > E(h, z^N) + \epsilon(h)\right] < \delta$$

$$\textit{for} \quad \epsilon(h) = \sqrt{\frac{1}{2N}\left(\ln\frac{1}{\delta} + \ln|H_i| + \ln\frac{1}{m(i)}\right)}$$

The proof is simple — just apply the union bound to the standard PAC theorem.

The SRM bound is slightly inefficient in the sense that the bound for all hypotheses in H_2 includes a bound for every hypothesis in H_1. This effect is typically small because the size of the hypothesis sets usually grows exponentially, implying that the extra confidence given to a hypothesis h in H_1 by the bounds used on hypothesis set $H_2, H_3, ...$ is small relative to the confidence given by the bound for H_1. One can remove this slack in Structural Risk Minimization bound by "cutting out" the nested portion of each hypothesis set in the formulation of $H_1, ..., H_l$. We will call this Disjoint Structural Risk Minimization.

The above Microchoice bound is essentially a compelling application of the Disjoint SRM bound where the description language for a hypothesis is the sequence of data-dependent choices which the algorithm makes in the process of deciding upon the hypothesis. The hypothesis set H_i is all hypotheses with the same description length in this language.

As an example, consider a binary decision tree with F boolean features and a boolean label. The first hypothesis set, H_1 will consist of 2 hypotheses; always false and always true. In general, we will have one hypothesis set for every legal configuration of internal nodes. The size of a hypothesis set where every tree contains k internal nodes will be 2^{k+1} because there are $k+1$ leaves each of which can take 2 values. The weighting $m(i)$ across the different hypothesis sets is defined by the Microchoice allocation of confidence.

The principle disadvantage of the Microchoice bound is that the sequence of data-dependent choices may contain redundancy. A different SRM bound with a different set of disjoint hypothesis sets might be able to better avoid redundancy. As an example, assume that we are working with a decision tree on F binary features. There are $F+2$ choices (any of F features or 2 labels) at the top node. At the next node down there will be $F+1$ choices in both the left and right children. Repeat until a maximal decision tree is constructed. There will be $\prod_{i=0}^{F}(F-i+2)^{2^i}$ possible trees. This number is somewhat larger than the number of boolean functions on F features: 2^{2^F}.

4.5. Combining Microchoice with Freund's Query Tree approach

The remainder of the paper is devoted to an improvement of the Microchoice bound called Adaptive Microchoice which arises from synthesizing Freund's query trees with the Microchoice bound. This improvement is not easily expressed as a simplification of Structural Risk Minimization. First we require some background material in order to state and understand Freund's bound.

4.5.1. Preliminaries and Definitions. The statistical query framework introduced by Kearns [**?**] is the same as the PAC framework, except that the learning algorithm can only access its data using statistical queries. A statistical query takes as input a binary predicate, χ, mapping examples [2] to a binary output: $(X, Y) \rightarrow \{0, 1\}$. The output of the statistical query is the average of χ over the examples seen,

$$\hat{P}_\chi = \frac{1}{N} \sum_{i=1}^{N} \chi(z_i)$$

The output is an empirical estimate of the true value $P_\chi = \mathbf{E}_P[\chi(z)]$ of the query under the distribution P. It is convenient to define

$$\alpha(\delta) = \sqrt{\frac{1}{2N} \ln \frac{2}{\delta}}.$$

Then, by Hoeffding's inequality,

$$\Pr\left[|\hat{P}_\chi - P_\chi| > \alpha(\delta)\right] < \delta.$$

4.5.2. Background and Summary. Freund [**?**] considers choice algorithms that at each step perform a Statistical Query on the sample, using the result to determine which choice to take. For an algorithm A, tolerance α, and distribution P, Freund defines the query tree $T_A(P, \alpha)$ as the choice tree created by considering only those choices resulting from answers $\hat{P}_\chi$ to queries χ such that $|\hat{P}_\chi - P_\chi| \leq \alpha$. The idea is that if a particular predicate, χ, is true with probability .9 on a random example it is very unlikely that the empirical result of the query will be .1. More generally, the chance the answer to a given query is off by more than α is at most $2e^{-2N\alpha^2}$ by Hoeffding's inequality. So, if the entire tree contains a total of $|Q(T_A(P, \alpha))|$ queries in it, the probability *any* of these queries is off by more than α is at most $2 \cdot |Q(T_A(P, \alpha))| \cdot e^{-2N\alpha^2}$. In other words, this is an upper bound on the probability the algorithm ever "falls off the tree" and makes a low probability choice. The point of this is that we can allocate half (say) of the confidence parameter δ to the event that the algorithm ever falls off the tree, and then spread the remaining half evenly on the hypotheses in the tree (which hopefully is a much smaller set than the entire hypothesis set).

Unfortunately, the query tree suffers from the same problem as the $p(h)$ distribution considered in Section 4.2, namely that to compute it, one needs to know P. So, Freund proposes an algorithmic method to find a superset approximation of the tree. The idea is that by analyzing the results of queries, it is possible to determine which outcomes were unlikely given that the query is close to the desired outcome. In particular, each time a query χ is asked and a response $\hat{P}_\chi$ is received, if it is true that $|\hat{P}_\chi - P_\chi| \leq \alpha$, then the range $[\hat{P}_\chi - 2\alpha, \hat{P}_\chi + 2\alpha]$ contains the range $[P_\chi - \alpha, P_\chi + \alpha]$. Thus, under the assumption that no query in the *correct* tree is answered badly, a superset of the correct tree can be produced by exploring all choices resulting from responses within 2α of the response actually received. By applying this method to every node in the query tree we can generate an empirically

[2] In the real SQ model there is no set of examples. The algorithm asks a query χ and is given a response $\hat{P}_\chi$ that is guaranteed to be near to the true value P_χ. That is, the true SQ model is an abstraction of the scenario described here where $\hat{P}_\chi$ is computed from an observed sample.

observable superset of the query tree: that is, the original query tree is a pruning of the empirically constructed tree.

A drawback of this method is that it can easily take exponential time to produce the approximate tree, because even the smaller correct tree can have a size exponential in the running time of the learning algorithm. Instead, we would much rather simply keep track of the choices actually made and the sizes of the nodes actually followed, which is what the Microchoice approach allows us to do. As a secondary point, given δ, computing a good value of α for Freund's approach is not trivial, see [?]; we will be able to finesse that issue.

In order to apply the Microchoice approach, we modify Freund's query tree so that different nodes in the tree receive different values of α, much in the same way that different hypotheses h in our choice tree receive different values of $\epsilon(h)$.

4.5.3. Microchoice Bounds for Query Trees. The manipulations of the choice tree are now reasonably straightforward. We begin by describing the *true* microchoice query tree and then give the algorithmic approximation. As with the choice tree in Section 4.3, one should think of each node in the tree as representing the current internal state of the algorithm.

We incorporate Freund's approach into the choice tree construction by having each internal node allocate a portion δ' of its "supply" $\tilde{\delta}$ of confidence parameter to the event that the output of the statistical query being asked is within $\alpha(\delta')$ of the correct answer. The node then splits the remainder of its supply, $\tilde{\delta} - \delta'$, evenly among the children corresponding to choices that result from answers $\hat{P}_\chi$ with $|\hat{P}_\chi - P_\chi| \leq \alpha(\delta')$. Choices that would result from "bad" answers to the query are *pruned away* from the tree and get nothing. This continues down the tree to the leaves.

How should δ' be chosen? Smaller values of δ' result in larger values of $\alpha(\delta')$ leading to more children in the pruned tree and less confidence given to each. Larger values of δ' result in less left over to divide among the children. Unfortunately, our algorithmic approximation (which only sees the empirical answers $\hat{P}_\chi$ and needs to be efficient) will not be able to make this optimization. Therefore, we define δ' in the *true* Microchoice query tree to be $\frac{\tilde{\delta}}{d+1}$ where d is the depth of the current node.

The algorithmic approximation uses the idea of [?] of including all choices within 2α of the observed value $\hat{P}_\chi$. Unlike [?], however, we do not actually create the tree; instead we just follow the path taken by the learning algorithm, and argue that the "supply" $\delta(h)$ of confidence remaining at the leaf is no greater than the amount that would have been there in the original tree. Finally, the algorithm outputs $\epsilon(h)$ corresponding to $\delta(h)$.

Specifically, the algorithm is as follows. Suppose we are at a node of the tree containing statistical query χ at depth $d(\chi)$ and we have a $\tilde{\delta}$ supply of confidence parameter. (If the current node is the root, then $\tilde{\delta} = \delta$ and $d(\chi) = 1$). We choose $\delta' = \tilde{\delta}/(d+1)$, ask the query χ, and receive $\hat{P}_\chi$. We now let k be the number of children of our node corresponding to answers in the range $[\hat{P}_\chi - 2\alpha(\delta'), \hat{P}_\chi + 2\alpha(\delta')]$. We then go to the child corresponding to the answer $\hat{P}_\chi$ that we received, giving this child a confidence parameter supply of $(\tilde{\delta} - \delta')/k$. This is the same as we would have given it had we allocated $\tilde{\delta} - \delta'$ to the children equally. We then continue from that child. Finally, when we reach a leaf, we output the current hypothesis h along with the error estimate $E(h, z^N) + \epsilon(\tilde{\delta})$, where $\epsilon(\tilde{\delta})$ is the error term corresponding to the remaining confidence supply $\tilde{\delta}$.

We now prove that with probability $1-\delta$, our error estimate $E(h, z^N) + \epsilon(h)$ is correct. By design, with probability $1-\delta$ all queries in the true Microchoice query tree receive good answers, *and* all hypotheses in that tree have their true errors within their estimates. We argue that in this case, the confidence assigned to the final hypothesis in the algorithmic construction is no greater than the confidence assigned in the true query tree. In particular, assume inductively that at the current node of our empirical path the supply $\tilde{\delta}_{\text{emp}}$ is no greater than the supply $\tilde{\delta}_{\text{true}}$ given to that node in the true tree. Now, under the assumption that the response $\hat{P}_\chi$ is "good", it must be the case that the interval $[\hat{P}_\chi - 2\alpha(\tilde{\delta}_{\text{emp}}/(d+1)), \hat{P}_\chi + 2\alpha(\tilde{\delta}_{\text{emp}}/(d+1))]$ contains the interval $[P_\chi - \alpha(\tilde{\delta}_{\text{true}}/(d+1)), P_\chi + \alpha(\tilde{\delta}_{\text{true}}/(d+1))]$. Therefore, the supply given to our child in the empirical path is no greater than the supply given in the true tree.

Let $d(h)$ be the depth of some hypothesis h in the empirical path and $\hat{C}_1(h)$, $\hat{C}_2(h)$, $\ldots$, $\hat{C}_d(h)$ be the sequence of choice sets resulting in h in the algorithmic construction; i.e., $\hat{C}_i(h)$ is the number of unpruned children of the i-th node. Then, the confidence placed on h will be:

$$\delta'(h) = \delta \prod_{i=1}^{d(h)} \left(\frac{i}{i+1} \frac{1}{|\hat{C}_i(h)|} \right) = \delta \frac{1}{d(h)+1} \prod_{i=1}^{d(h)} \frac{1}{|\hat{C}_i(h)|} \tag{4.5.1}$$

Let $A(z^N) = h$ be the output of the learning algorithm.

We now have the following two theorems for the realizable and agnostic cases:

THEOREM 4.5.1. *(Realizable Adaptive Microchoice Theorem) For any* $0 < \delta < 1$*, under the assumption that* A *produces only hypotheses with zero empirical error,*

$$\Pr_{z^N}\left[A(z^N) = h : E_P(h) > \epsilon(h)\right] < \delta$$

$$\textit{for } \epsilon(h) = \frac{1}{N}\left(\ln\frac{1}{\delta} + \ln(d(h)+1) + \sum_{i=1}^{d(h)} \ln(|\hat{C}_i(h)|) \right).$$

THEOREM 4.5.2. *(Agnostic Adaptive Microchoice Theorem) For any* $0 < \delta < 1$,

$$\Pr_{z^N}\left[A(z^N) = h : E_P(h) > E(h, z^N) + \epsilon(h)\right] < \delta$$

$$\textit{for } \epsilon(h) = \sqrt{\frac{1}{2N}(\ln\frac{1}{\delta} + \ln(d(h)+1) + \sum_{i=1}^{d(h)} \ln(|\hat{C}_i(h)|))}.$$

The bounds in theorems 4.5.1 and 4.5.2 are very similar to 4.3.1 and 4.3.2 except that the choice complexity is slightly worsened with the $\ln(d(h)+1)$ term but improved by replacing $C_i(h)$ with the smaller $\hat{C}_i(h)$.

4.5.4. Allowing batch queries. Most natural Statistical Query algorithms make each choice based on responses to a *set* of queries, not just one. For instance, to decide what variable to put at the top of a decision tree, we ask F queries, one for each feature; we then choose the feature whose answer was most "interesting". This suggests generalizing the query tree model to allow each tree node to contain a set of queries, executed in batch. Requiring each node in the query tree to contain just a single query as in the above construction would result in an unfortunately high branching factor just for the purpose of "remembering" the answers received so far. [3]

Extending the algorithmic construction to allow for batch queries is easily done. If a node has q queries $\chi_1, \ldots, \chi_q$, we choose the query confidence $\delta' = \tilde{\delta}/(d+1)$ as before, but we now split the δ' evenly among all q queries. We then let k be the number of children corresponding to answers to the queries $\chi_1, \ldots, \chi_q$ in the ranges $[\hat{P}_{\chi_1} - 2\alpha(\delta'/q), \hat{P}_{\chi_1} + 2\alpha(\delta'/q)], \ldots, [\hat{P}_{\chi_q} - 2\alpha(\delta'/q), \hat{P}_{\chi_q} + 2\alpha(\delta'/q)]$ respectively. We then go to the child corresponding to the answers we actually received, and as before give the child a confidence supply of $(\tilde{\delta} - \delta')/k$. Theorems 4.5.1 and 4.5.2 hold exactly as before; the only change is that $|\hat{C}_i(h)|$ means the size of the i-th choice set in the batch tree rather than the size in the single-query-per-node tree.

4.5.4.1. *Example: Batch Queries for Decision trees.* When growing a decision tree, it is natural to make a batch of queries and then make a decision about which feature to place in a node. The process is then repeated to grow the full tree structure. As in the decision tree example described in the simple Microchoice section, if we have F features and are considering adding a node at depth $d(v)$, there are $F - d(v) + 1$ possible features that could be chosen for placement in a particular node. The decision of which feature to use is made by comparing the results of $F - d(v) + 1$ queries to pick the best feature according to some criteria, such as information gain. We can choose $\delta' = \tilde{\delta}/(d+1)$, then further divide δ' into confidences of size $\delta'/(F - d(v) + 1)$, placing each divided confidence on one of the $F - d(v) + 1$ statistical queries. We now may be able to eliminate some of the $F - d(v) + 1$ choices from consideration, allowing the remaining confidence, $\tilde{\delta} - \delta$ to be apportioned evenly amongst the remaining choices. Depending on the underlying distribution this could substantially reduce the size of the choice set. The best case occurs when one feature partitions all examples reaching the node perfectly and all other features are independent of the target. In this case the choice set will have size 1 if there are enough examples.

4.5.5. Adaptive Microchoice vs. Basic Microchoice.

The Adaptive Microchoice bound is a significant improvement over the simple Microchoice bound when the distribution is such that each choice is clear. For example, consider F boolean features and $N = O(F)$ examples. Suppose that one feature is identical to the label and all the rest of the features are determined with a coin flip independent of the label.

When we apply a decision tree to a dataset generated with this distribution, what will be the resulting bound? Given enough examples, with high probability there will only be one significant choice for the first batch query: the feature identical to the label. The second and third batch queries, corresponding to the children of the root feature, will also have a choice space of size 1 with very high probability.

[3] Consider a decision tree algorithm attempting to find the right feature for a node. If the first query returns a value of $\hat{P}_\chi$ with a confidence of δ then the branching factor would be approximately $m \cdot \left[\hat{P}_\chi - 2\alpha(\delta) - (\hat{P}_\chi - 2\alpha(\delta))\right] = 4m \cdot \alpha(\delta)$. This branching factor would be approximately the same for further queries required by the algorithm to make a decision about what feature to use. This results in a total multiplied choice space size of approximately $[4m \cdot \alpha(\delta)]^F$. This can be reduced to F or less using a batch query.

The "right" choice will be the label value. Each choice set has size 1 resulting in a complexity of $\ln 4$ due to allocation of confidence to the statistical queries necessary for learning the decision tree. $\ln 4$ is considerably better than $\ln(F+2)+2\ln(F+1)$ which the simple version of the Microchoice bound provides. Note that the complexity reduction only occurs with a large enough number of examples N implying that the value of $\epsilon(h)$ calculated can improve faster than (inverse) linearly in the number of examples.

The Adaptive Microchoice bound is never much looser than the simple Microchoice bound because under the assumption that choice sets are of size at least 2, the penalty for using the adaptive version, $\ln d$, is always small compared to the complexity term for the simple Microchoice bound, $\sum_{i=1}^{d(h)} \ln |C_i(h)|$.

4.5.6. Alternative Adaptive Microchoice. The Adaptive Microchoice bound provides a simple scheme for dividing confidence between choices and queries. There are other choices which may be useful in some settings. Any scheme which *a priori* divides the confidence between queries and choices at every node will generally work. Here are two schemes which may be useful:

- Assign a constant proportion of confidence to the query. This scheme is more aggressive than the one used in the Adaptive Microchoice bounds and may result in a lower complexity when many choices are eliminatable. The drawback is we no longer get the telescoping in equation (4.5.1) and so the term logarithmic in $d(h)$ in theorems 4.5.1 and 4.5.2 becomes linear in $d(h)$.
- For a decision tree, assign a portion dependent on the depth of the node in the decision tree that the choice set is over. It is unlikely that choices are eliminatable from nodes not near the root because the number of examples available at a node typically decays exponentially with the (decision tree) depth. A progressive scheme which allocates less confidence to queries for deep nodes will probably behave better in practice.

4.5.7. Comparison with Freund's Work. Freund's approach for self-bounding learning algorithms can require exponentially more computation then the Microchoice approach. In its basic form, it requires explicit construction of every path in the state space of the algorithm not pruned in the tree. There exist some learning algorithms where this process can be done implicitly making the computation feasible. However, in general this does not appear to be possible.

The Adaptive Microchoice bound only requires explicit construction of the size of each subset from which a choice is made. Because many common learning algorithms work by a process of making choices from small subsets, this is often computationally easy. The Adaptive Microchoice bound does poorly, however, when Freund's query tree has a high degree of sharing; for example, when many nodes of the tree correspond to the same query, or many leaves of the tree have the same final hypothesis. Allowing batch queries alleviates the most egregious examples of this. It is also possible to interpolate between the Adaptive Microchoice bound and Freund's bound by a process of conglomerating the subsets of the Microchoice bound.

4.5.8. Choice Set Conglomeration. The mechanism of choice set conglomeration is a similar to the batch query technique. It allows you to trade increased computation for a tighter bound. When starting with the simple Microchoice bound, this technique can smoothly interpolate with the simple sample complexity bound (**??**). When starting with the Adaptive Microchoice bound, we can interpolate with Freund's bound.

Consider a particular choice set, $\hat{C}_i$, with elements c_i. Each c_i indexes another choice set, $\hat{C}_{i+1}(c_i)$. If the computational resources exist to calculate the union $\hat{C}_{i,i+1} = \bigcup_{c_i \in \hat{C}_i} \hat{C}_{i+1}(c_i)$, then $|\hat{C}_{i,i+1}|$ can be used in place of $|\hat{C}_i| \cdot |\hat{C}_{i+1}|$ in the adaptive Microchoice bound. The conglomeration can be done repeatedly to build large choice sets and also applies to the simple Microchoice bound 4.3.1 and 4.3.2. Conglomeration can be useful for tightening the bound when there are multiple choice sequences leading to the same hypothesis. However, choice set conglomeration is not always helpful because it trades away the fine granularity of the Microchoice bound. The extreme case where all choice sets are conglomerated into one choice set and every hypothesis and query have the same weight is equivalent to Freund's bound.

When the choices of the attached choice sets are all different, conglomeration will have little use because the size of the union of the choice sets is the sum of the sizes of each choice set $|\hat{C}_{i,i+1}| = \sum_{c_i \in \hat{C}_i} |\hat{C}_{i+1}(c_i)|$. If the child sets each have the same size $|\hat{C}_{i+1}|$ then this simplifies to $|\hat{C}_{i,i+1}| = |\hat{C}_i| \cdot |\hat{C}_{i+1}|$ which results in the same confidence applied to each choice whether conglomerating or not. The best case for conglomeration is equivalent to the batch query case: every subchoice set contains the same elements. Then we have $|\hat{C}_{i,i+1}| = |\hat{C}_{i+1}|$ and can pay no cost for the choice set $|\hat{C}_i|$.

4.5.9. Conclusion. The goal of this work is to produce tighter bounds on the future error rates of a learned hypothesis. We have presented the Microchoice bound which is a kind of Occam's razor bound using the structure of the learning algorithm itself. The Microchoice bound was then synthesized with Freund's query tree approach for Self-Bounding learning algorithms to produce the Adaptive Microchoice bound. We also presented some simple techniques for improving these bounds. One technique is a method for interpolating between the Microchoice bound and Freund's query tree approach using conglomeration of choice sets. Another technique is the batch query which is especially useful for decision trees. Some examples of the application of these bounds to decision trees were given along with a new pruning technique suggested by the Microchoice bounds.

The Microchoice bound can be applied to many common learning algorithms unintrusively — the path through state space that the algorithm takes need not be affected. Some extra computation may be necessary to assess the size of the choice set from which the algorithm makes its choices. In return, the algorithm can state a bound along with its hypothesis. The Microchoice approach also shows a very natural way of performing SRM.

In a given situation, the Microchoice bound may or may not be better than another PAC bound. In practice, it may be worthwhile to output a bound which is the minimum of that given by the Microchoice bound and a more general PAC bound. This can be done in a theoretically sound manner by first mapping $\delta \to \frac{\delta}{2}$ in each bound which slightly worsens the result. The resulting $\epsilon = \min(PAC(\delta/2), Microchoice(\delta/2))$ bound will then hold with probability $1 - \delta$.

There are several remaining open questions. Is there a satisfying, natural bound for the continuous case? Can this approach be useful for commonly used learning algorithms?

4.6. Improved PAC-Bayes bound

To improve on this bound, we employ a PAC-Bayes bound from McAllester [?]. In the PAC-Bayes setting, a classifier is also defined by a distribution Q over the hypothesis space. However, each classification [4] is carried out according to a hypothesis sampled from Q rather than by the averaging classifier c defined by Q. We are interested in the gap between the *expected* generalization error and the *expected* empirical error, where both expectations are taken with respect to Q. We need to introduce the *relative entropy* (or *Kullback-Leibler divergence*; e.g. [?]):

$$\mathrm{KL}(Q||P) = E_{h\sim Q} \ln \frac{q(h)}{p(h)} \tag{4.6.1}$$

where q, p denote the probability densities of the distributions Q, P. If the support is finite, we have

$$\mathrm{KL}(Q||P) = \sum_h q(h) \ln \frac{q(h)}{p(h)} \tag{4.6.2}$$

The relative entropy is an asymmetric distance measure between probability distributions, with $\mathrm{KL}(Q||P) = 0 \Leftrightarrow Q = P$ almost everywhere.

\begin{theo}[PAC-Bayes \cite{McAllester}]\label{th-mcall} Let \(l(h,(x,y)) \)be a binary loss function, \(P \)any prior distribution over \({\cal H} \)and \(\delta\in (0,1) \). With probability at least \(1-\delta \)over random samples \(S \)from \(D \)we have that for all distributions \(Q \)over the hypothesis space \({\cal H} \):\begin{equation} \pr{{D,Q}}[l(h,(x,y))=1] \leq \pr{{S,Q}}[l(h,(x,y))=1] +\sqrt{ \frac{\KL(Q\|P) +\ln\delta^{-1} +\ln m+2}{2m-1} } \end{equation} \end{theo}

An example of a loss function is the well-known *zero-one loss* \(l(h,(x,y)) = I_{\{h(x)\ne y\}} \).

We can tighten this bound by employing a more accurate tail bound on the Binomial distribution, which leads us to the following theorem.

\begin{theo} (PAC-Bayes Relative Entropy bound)\label{th-pbr} Let \(P \)be any prior distribution over \({\cal H} \)and \(\delta\in (0,1) \). With probability at least \(1-\delta \)over random samples \(S \)from \(D \)we have that for all distributions \(Q \)over the hypothesis space \({\cal H} \):

\[\KL(Ber_{S,Q}\|Ber_{D,Q})\leq \frac{\KL(Q\|P)+\ln \frac{2m}{\delta }}{m-1}\] where \(Ber_{S,Q} = 1 \)with probability \(\pr{S,Q}(l(h,(x,y))=1) \)and \(0 \)otherwise, and \(Ber_{D,Q} = 1 \)with probability \(\pr{D,Q}(l(h,(x,y))=1) \)and \(0 \)otherwise.

\end{theo}

This theorem gives a constraint on the KL divergence between the average empirical and average true errors rather then the standard \(l_1 \)distance. This bound is always at least as tight as the original PAC-Bayes bound \cite{McAllester} and sometimes much tighter, such as when the average empirical error is near \(0 \). A proof is given in section \ref{sec-improve}.

[4] Such classifiers, also called “discriminants”, are called *Gibbs classifiers* (e.g. [?]).

This theorem holds for finite and infinite hypothesis spaces. The PAC-Bayes theorem guarantees a tighter bound (except at low order) than earlier results such as the following Occam's razor theorem.

\section{Improving the PAC-Bayes bound}\label{sec-improve}

In order to fully benefit from the improved Chernoff bound we need to prove \ref{th-pbr} using the Chernoff relative entropy bound. The retrofit of the PAC-Bayes bound is not a simple substitution of the Hoeffding inequality with the Chernoff relative entropy bound so a proof is given.

The proof of the improved PAC-bayes theorem (\ref{th-pbr}) relies upon two lemmas. The first is Lemma 22 from \cite{McAllester} which is given by:

\begin{lemma} \label{lem:helper2}For \(\beta >0,K>0 \) and \(Q,P,y\in R^{n} \) satisfying \(P_{i}>0,Q_{i}>0, \) and \(\sum _{i}Q_{i}=1 \), if \[\sum _{i=1}^{n}P_{i}e^{\beta y_{i}}\leq K\] then \[\sum _{i=1}^{n}Q_{i}y_{i}\leq \frac{\KL(Q\|P)+\ln K}{\beta }\] \end{lemma}

The second lemma we will need to prove ourselves. It is basically an improved version of Lemma 17 from \cite{McAllester}.

\begin{lemma} \label{lem:helper1} \[\forall \delta >0\; \forall ^{\delta }S\; \; \; E_{h\sim P}e^{(m-1)\KL(\hat{e}(h)\|e(h))}\leq \frac{2m}{\delta }\] \end{lemma}

First, for any given hypothesis \(h \) we prove the following. \begin{equation} \label{eqn:expectation} E_{S}[e^{(m-1)\KL(\hat{e}(h)\|e(h))}]\leq 2m \end{equation} Lemma~\ref{lem:helper1} follows from (\ref{eqn:expectation}) by taking an expectation over selecting \(h \) according to any distribution \(P \) over \(h \), reversing the two expectations, and applying Markov's inequality. We now show that (\ref{eqn:expectation}) follows from (\ref{eqn:relent-prelim}) and (\ref{eqn:relent-prelim_2}). More specifically, we maximize \(\int _{0}^{1}e^{(m-1)\KL(x\|p)}f(x)dx \) over all functions \(f(x) \) satisfying the following for all \(q_{1}\geq e(h) \) and \(q_{2}\leq e(h) \). \[\int _{q_{1}}^{1}f(x)dx\leq e^{-m\KL(q_{1}\|e(h))}\] \[\int _{0}^{q_{2}}f(x)dx\leq e^{-m\KL(q_{2}\|e(h))}\] The value of \(E_{S}[e^{(m-1)\KL(\hat{e}(h)\|e(h))}] \) must be less than this maximum. The integral \(\int _{0}^{1}e^{(m-1)\KL(x\|e(h))}f(x)dx \) is maximized when \(f(x) \) is as "spread out" as possible, i.e., when the above inequalities are replaced by equalities. This gives the following. \[f(x)=\left\{ \begin{array}{ll} m\frac{\partial \KL(x\|p)}{\partial x}e^{-m\KL(x\|p)} & \mbox {for}\; x\geq p\\ & \\ -m\frac{\partial \KL(x\|p)}{\partial x}e^{-m\KL(x\|p)} & \mbox {for}\; x\leq p \end{array}\right. \] Which, in turn, gives the following. \begin{eqnarray*} E_{S}[e^{(m-1)\KL(\hat{e}(h)\|e(h))}] & \leq & \int _{0}^{1}e^{(m-1)\KL(x\|e(h))}f(x)dx\\ & & \\ & = & \int _{0}^{e(h)}-m\frac{\partial \KL(x\|e(h))}{\partial x}e^{-\KL(x\|e(h))}dx \\ & + & \int _{e(h)}^{1}m\frac{\partial \KL(x\|e(h))}{\partial x}e^{-\KL(x\|e(h))}dx\\ & \leq & 2m \end{eqnarray*}

Now we have the necessary lemmas to finish the proof of \ref{th-pbr}.

By Jensen's inequality, we have:

\[\KL(E_{h\sim Q}\hat{e}(h)\|E_{h\sim Q}e(h))\leq E_{h\sim Q}\KL(\hat{e}(h)\|e(h)) \] Furthermore, according to Lemma~\ref{lem:helper1} we can apply Lemma~\ref{lem:helper2} with \(K=\frac{2m}{\delta } \) and \(\beta =m-1 \) and \(y_{i}=\KL(\hat{e}(h)\|\hat{e}(h)+\epsilon) \) to get: \[E_{h\sim Q}\KL(\hat{e}(h)\|e(h))=\sum _{i=1}^{n}Q_{i}\KL(\hat{e}(h)\|\hat{e}(h)+\epsilo)\leq \frac{\KL(Q\|P)+\ln \frac{2m}{\delta }}{m-1}\] and we are done.

4.7. Averaging Bounds (Improved margin)

\section{Introduction}\label{sec-intro}

Averaging is a standard technique in applied machine learning for combining multiple classifiers to achieve greater accuracy. % via averaging. Examples include {\em Bayesian classification} \cite{Cheeseman}, {\em boosting} \cite{Freund}, {\em bagging} \cite{Breiman}, {\em Winnow} \cite{Littlestone}, {\em Maximum Entropy discrimination } \cite{Jaakkola}, and {\em Bayes point machines} \cite{Herbrich}. Despite the prevalence of this technique there is only weak theoretical justification so far for the practice. This paper provides a new stronger theoretical justification for the practice of averaging. In particular, we state and prove a bound on the gap between the training set error rate and the predictive error rate which improves as more hypotheses are averaged over.

Until 1998, theoretical bounds such as the Occam's razor bound \cite{Blumer} suggested that averaging was \emph{wrong} because it increased the description length of the resulting hypothesis.\footnote{We note, however, that it is the {\em minimum} description length that should be used in the bound.} The Occam's razor bound \emph{only suggests} that averaging \emph{may} be bad since there is no corresponding lower bound. Schapire, Freund, Bartlett and Lee \cite{Schapire} showed a great improvement on the naive bound for an average-of-classifiers hypothesis. Loosely speaking, their margin bound states that if the average has a small empirical error rate (i.e., it is accurate on most training examples) and has a large "margin" (defined in \ref{sec-setting}), then its true error rate is also small. The proof itself works in a very intuitive manner by showing that the accuracy of a large margin classifier is close to the accuracy of a simple classifier, for which standard bounds are tight.

The problem with this result is that the value of the bound depends only on the empirical margin which does not necessarily improve with an average over a larger number of hypotheses. This bound suggests using the simple criteria: choose the average to maximize the margin. However, empirical results \cite{Grove} indicate that this procedure is not optimal.

In this paper, we prove a new bound on the true error rate, which suggests a new optimization criterion, namely, optimize for a large margin \emph{and} for a uniform average over as many hypotheses as possible.

% BEGIN OLD The layout of this paper is as follows:

\begin{enumerate} \item Discussion of the relationship with prior relevant results. \item Development of a simple improved theoretical bound. \item A proof of the bound. \item An example of the benefit of the new bound on a toy problem. \item Discussion of implications of the new bound on prior work. \end{enumerate} % END OLD % BEGIN SUGGESTION: Saves space! % We aren't short -John %The layout of the paper is as follows. First, we discuss the relationship %with prior results. We then develop and discuss the improved theoretical bound %with an emphasis on simplicity, followed by a self-contained proof. In %section \ref{sec-implic}, we give an example showing the benefits of the %new bound on a toy problem, and we discuss implications of our result to %prior work, namely to a range of frequently used machine learning algorithms. % END SUGGESTION

\section{The setting and important earlier results}\label{sec-sett-earlier}

\subsection{The setting}\label{sec-setting}

We first explain the setting, which is the same as the one used in \cite{Schapire}.

An input space $\cal X$ is given, where the members of $\cal X$ are also referred to as {\em examples}. The set ${\cal X} \times \{-1,1\}$ is the space of {\em labeled} examples. A {\em base hypothesis} h is a mapping from the input space $\cal X$ into $\{-1,1\}$. A (possibly infinite) space $\cal H$ (the hypothesis space) is given and the goal is to construct an {\em averaging classifier} $c:{\cal X} \rightarrow \{-1,1\}$ as a weighted average of base hypotheses: \[c(x)=\mathop{\rm sign} \sum_{j=1}^{k}q_{j}h_{j}(x) \quad (x \in \cal X)\ , \] where $q_j\ge 0$, $j=1,\ldots,k$, and $\sum_{j=1}^k q_i = 1$. The fundamental assumption here is that labeled training examples are drawn independently, with replacement, from some probability distribution D over ${\cal X} \times \{-1,1\}$. In all the theorems we discuss, D is assumed to be unknown to the procedure which constructs the classifier, and the results hold for all D. Probabilities and expectations over D will be denoted by the subscript D; for example, the true error of an averaging classifier is denoted by: \begin{equation} e_D(c) = E_D[I(c(x) \neq y)] = E_{(x,y)\sim D}[I(c(x) \neq y)]. \end{equation} Here, $I(\cdot)$ is the indicator function, which is 1 if its argument evaluates to \emph{true} and 0 otherwise. Probabilities with respect to D are written as $\pr{D}$.

With S, we denote a sample $\{(x_i,y_i)\, |\, i=1,\dots,m\}$ drawn independently and identically distributed (i.i.d.) from $D(x,y)$. The i.i.d assumption is the one fundamental assumption we make in this work. The subscript S denotes empirical expectation or probability over S, for example the empirical error of an averaging classifier is given by: \begin{equation} e_S(c) = E_S[I(c(x) \neq y)] = \frac{1}m \sum_{i=1}^m I(c(x_i) \neq y) \end{equation} Probabilities with respect to S are written as $\pr{S}$.

\subsection{Quantities used in the bound}\label{sec-quantities} The basic learning model needs to be augmented with a few definitions for the analysis.

Given a subset $\{h_1,\ldots,h_k\} \subseteq \cal H$, the set $\{q_1,\ldots,q_k\}$ can be interpreted as a probability distribution over the set $\cal H$.\footnote{ In the case of finite or countably infinite ${\cal H}$, this is achieved by assigning all hypotheses outside the subset the weight zero. If ${\cal H}$ is finite, we will usually work with ${\cal H} = \{h_1,\ldots, h_k\}$ for simplicity. For uncountable spaces, we define Q as $\sum_j q_j \delta(h,h_j)$, where $\delta(h,h_j)$ is the delta distribution centered on h_j.} This distribution will be denoted by \(Q \). We will also often use the unsigned version of the classifier: \[f(x)= E_{h\sim Q}[h(x)] = \sum_{j=1}^{k}q_{j}h_{j}(x)\ . \] It is important to note that we make no assumption about how the weights $q_1,\ldots,q_k$ are obtained, so our results are applicable to many algorithms.

The derived bounds depend on the powerful concept of the {\em margin}, $t (x,y)$, of a labeled example with respect to a classifier, namely, \[t(x,y) = y \sum _{j=1}^{k}q_{j}h_{j}(x) = y f(x)\ .\] The margin is a quantitative measure of how decided the average is. Obviously, $-1\le t(x,y) \le 1$. If $t(x,y)=1$ (resp.\ $t(x,y)=-1$), then all the base hypotheses classify correctly (resp.\ incorrectly). When $t(x,y)$ is close to zero, the classifier is, in some sense, undecided. Note that $c(x) = y$ iff $t(x,y) > 0$.

\subsection{Earlier results}\label{sec-earlier}

The new averaging bound arises from improving one critical step in the proof of the original margin bound, which we state here for reference.

\begin{theo}[Margin Bound \cite{Schapire}]\label{th-schapire} Let $\delta\in (0,1)$. With probability at least $1-\delta$ over random samples S from D we have that for all distributions $Q=(q_1,\dots,q_k)$ over the finite hypothesis space ${\cal H}$ and all margin thresholds $\theta\in (0,1]$: \begin{equation} \eqalign{ \pr{D}[y f(x)\le 0] \le &\ \pr{S}[y f(x)\le \theta] \cr + &\ O\left(\sqrt{\frac{\theta^{-2} \ln |{\cal H}| \log m + \ln\delta^{-1} } {m}} \right), \cr } \end{equation} where $f(x) = E_{h\sim Q}[h(x)] = \sum_j q_j h_j(x)$. \end{theo} Here, the notation $b(m)=O(a(m))$ means there exists a constant C such that $b(m)\leq C\cdot a(m)$ for all m. This margin bound implies that if most training examples have a large margin θ (i.e. $t(x,y) > \theta$ for most $(x,y)\in S$) and the hypothesis space is not too large, then the generalization error cannot be large.

We will improve on this bound in Section \ref{sec-main-proof} by employing the PAC-Bayes bound from McAllester \cite{McAllester}. In the PAC-Bayes setting, a classifier is also defined by a distribution Q over the hypothesis space. However, each classification\footnote{Such classifiers are called {\em Gibbs classifiers} (e.g. \cite{Haussler}).} is carried out according to a hypothesis sampled from Q rather than by the averaging classifier c defined by Q. We are interested in the gap between the {\em expected} generalization error and the {\em expected} empirical error, where both expectations are taken with respect to Q. We need to introduce the {\em relative entropy} (or {\em Kullback-Leibler (KL) divergence}; e.g., \cite{Cover}): \begin{equation}\label{eq-relent} \KL(Q\|P) = E_{h\sim Q} \left[\ln \frac{q(h)}{p(h)} \right]\ , \end{equation} where q, p denote the probability densities of the distributions Q, P. If ${\cal H}$ is finite, we have\footnote{ Here and elsewhere, we agree on the definition $0 \log 0 = \lim_{t\to 0+} t \log t = 0$.} \begin{equation} \KL(Q\|P) = \sum_{j=1}^k q_j \ln \frac{q_j}{p_j}, \end{equation} where $Q=(q_1,\dots,q_k),\; P=(p_1,\dots,p_k)$. The relative entropy is an asymmetric distance measure between probability distributions, with $\KL(Q\|P)=0$ if and only if $Q=P$ almost everywhere.

\begin{theo}[PAC-Bayes \cite{McAllester}]\label{th-mcall} % Let ℓ be a loss function mapping R into $\{0,1\}$. Let P be any prior distribution over ${\cal H}$ and $\delta\in (0,1)$. With probability at least $1-\delta$ over random samples S from D we have that for all distributions Q over the hypothesis space ${\cal H}$: \[\eqalign{% \begin{split} \pr{{D,Q}}[h(x)\neq y] \leq &\ \pr{{S,Q}}[h(x)\neq y] \cr +&\sqrt{ \frac{\KL(Q||P) +\ln\delta^{-1} +\ln m+2}{2m-1} } \cr } %\end{split} \] \end{theo} Here, $\pr{{D,Q}}[\cdot]$ is short for $E_{h\sim Q}[\pr{D}[\cdot]]$, and $\pr{{S,Q}}[\cdot]$ stands for $E_{h\sim Q}[\pr{S}[\cdot]]$. This theorem holds for finite and infinite hypothesis spaces. The PAC-Bayes theorem guarantees a tighter bound (except at low order) than earlier results such as the following Occam's razor theorem.

\begin{theo}[Occam's Razor \cite{Blumer}]\label{th-occam} Let P be a distribution over a hypothesis space ${\cal H}$ and $\delta\in (0,1)$. With probability at least $1-\delta$ over random samples S from D, for all hypotheses $h\in {\cal H}$: \begin{equation} \eqalign{ %\begin{split} \pr{D}[h(x) \ne y] \le &\ \pr{S}[h(x)\ne y] \cr +&\ \sqrt{ \frac{ \ln(1/p(h)) + \ln\delta^{-1} } {2 m} }\ . \cr} %\end{split} \end{equation} \end{theo}

Note that, for finite ${\cal H}$ and up to low order terms, theorem \ref{th-occam} is a special case of theorem \ref{th-mcall}, where we choose delta distributions $Q=(0,\dots,0,1,0,\dots,0)$ in theorem \ref{th-mcall}. The essence of our improvement of the standard margin bound comes from the application of the PAC-Bayes bound instead of the Occam's razor bound within the standard proof of the margin bound.

\section{An improved averaging bound}\label{sec-main}

In this section, we state and prove our main result, a PAC-Bayes generalization error bound for averaging classifiers. Before we do this, we provide a discussion of a special case in order to put across an intuition of how our bound can improve upon theorem \ref{th-schapire}. In this discussion, we limit ourselves to a finite ${\cal H} = \{h_1,\dots,h_k\}$, while the main result will be stated for arbitrary ${\cal H}$. Our bound relies on a “posterior” distribution $Q=(q_1,\ldots,q_k)$ over ${\cal H}$, from which the average $f(x)$ is defined as $f(x) = \sum_k q_k h_k(x)$. The “posterior” Q may depend on the training sample S in an arbitrary way.\footnote{ An example is, in Bayesian classification, the {\em posterior distribution} over ${\cal H}$. The corresponding averaging classifier $f(x)$ is called {\em Bayes classifier} or {\em Bayes-optimal classifier} (e.g. \cite{Haussler}).}

The {\em entropy} $H(Q)$ (e.g. \cite{Cover}) of Q is defined as $H(Q) = -\sum_i q_i \ln q_i$. It measures the “uncertainty” in Q, in that delta distributions $Q=(0,\dots,0,1,0,\dots,0)$ have minimum entropy 0 and the uniform distribution has maximum entropy $\ln k$. We can state a special case of our main result as follows. \begin{theo}[Special Case]\label{th-special} Let $\delta\in (0,1)$. With probability at least $1-\delta$ over random samples S from D, for all distributions Q over the hypothesis space ${\cal H}$ and for all margin thresholds $\theta \in (0,1]$: \begin{equation} \eqalign{ % \begin{split} \pr{D} & [y f(x)\le 0] \le \pr{S}[y f(x) \le \theta] \cr + O & \left(\sqrt{\frac{ \theta^{-2}(\ln |{\cal H}| - H(Q)) \ln m + \ln m + \ln\delta^{-1} } {m} } \right). \cr }% \end{split} \end{equation} \end{theo} This theorem is just a simplification of theorem \ref{th-main} to finite hypothesis spaces with a uniform prior P.

How much can the improvement help us? About the best case we could hope for is a uniform average over half the hypothesis space\footnote{ An average over a subset of the hypothesis space ${\cal H}$ includes only those hypotheses h_j with coefficients q_j significantly different from zero.}. In that case, the complexity term \((\ln |H|-H(Q)) \ln m \) is quite small: \(\ln m \ln 2 \). In the worst case, when the average is over only a number of hypotheses \(k \) similar to the number of examples \(m \), there is no significant improvement over the original margin bound.

It is easy to generalize the improved averaging bound to continuous spaces with arbitrary priors by carefully applying the PAC-Bayes bound.

\begin{theo}[Main Theorem]\label{th-main} Let P be any continuous probability distribution over ${\cal H}$ and let $\delta\in (0,1)$. With probability at least $1-\delta$ over random samples S of D, for all margin thresholds $\theta>0$ and for every %(data dependent) This is meaningless distribution Q over ${\cal H}$: \begin{equation} \eqalign{% \begin{split} \pr{D}& \left[y f(x)\le 0 \right] \le \pr{S}\left[y f(x)\le \theta \right] \cr &+ O \left(\sqrt{ \frac{\theta^{-2} \KL(Q\|P) \ln m + \ln m + \ln \delta^{-1} }{m} } \right) \cr

}%\end{split} \end{equation} where $f(x)=E_{h\sim Q}[h(x)]$. \end{theo} The proof is given in \ref{sec-main-proof}.

Theorem \ref{th-main} holds also for the case of bounded real-valued hypotheses, without any loss in the tightness of the bound. The theorem can also be tightened in several quantitatively important ways. Details can be found in \cite{TR}.

There exists an alternative approach for deriving a bound similar to Theorem \ref{th-main} which needs to be mentioned. Essentially, starting with the covering number based approach of \cite{Bartlett} we can use the covering number results from theorem 3.6 of \cite{Zhang} to arrive at a similar bound. The principle advantage of our approach over this one is simplicity of argumentation combined with quantitatively tighter results.

The continuous form of the improved averaging bound applies to arbitrary averages over continuous hypothesis spaces, the {\em finite} averages defined in subsection \ref{sec-quantities} are special cases. Note that in this setting, the average needs to be an integral over an uncountably infinite set of hypotheses, otherwise the KL-divergence does not converge. In practice, this is not a significant problem because machine learning algorithms over large hypothesis spaces typically have some parameter stability. In other words, a small shift in the parameters of the learned model produces a small change in the prediction of the hypothesis. With hypothesis stability, we can convert any average over a finite set of hypotheses into an average over an infinite set of hypotheses without significantly altering the predictions of the average.

\subsection{Proof of main theorem}\label{sec-main-proof}

The proof has the same structure as the original margin bound proof \ref{th-schapire} with one step replaced by the application of the PAC-Bayes theorem \ref{th-mcall}.

Our averaging classifier is specified by \[c(x) = \mathop{\rm sign}\, E_{h\sim Q}[h(x)] \ .\] Let N be any natural number; later, the choice of N will be optimized. For every distribution Q, we construct a random function $g=g_Q$ as follows. Draw N hypotheses i.i.d. from Q and define \begin{equation}\label{eq-sample} g(x) = \frac{1}N \sum_{j=1}^N h_j(x). \end{equation} The set of all possible g's is denoted \begin{equation}\label{eq-cnset} {\cal H}_N = \biggl\{ \frac{1}N \sum_{j=1}^N h_j(x)\; \biggl|\; h_j\in {\cal H} \biggr\}, \end{equation} and we denote the distribution of g (i.e., over the set ${\cal H}_N$) by Q^N. Note that for a fixed pair (x,y), the quantities $h_j(x)$ in the expression for $g(x)$ (see (\ref{eq-sample})) are i.i.d.\ Bernoulli variables (over $\{-1,1\}$) with mean \begin{equation} y E_{h_j\sim Q}[h_j(x)] = y f(x) \ . \end{equation} Therefore, $y E_{g\sim Q^N}[g(x)] = y f(x)$. Since $g(x)$ is the average over N i.i.d.\ Bernoulli variables, Hoeffding's bound (see \cite{Devroye}, p.122) applies. Thus, for every $x \in {\cal X},\; y\in\{-1,+1\}$, the probabilities with respect to the sampling of $h_1,\ldots,h_N$ satisfy \begin{equation}\label{eq-purehoeff} \pr{g\sim Q^N} \left[y (g(x)- f(x)) > \epsilon \right] \leq e^{-\frac{1}{2} N \epsilon^2} \end{equation} For every $\theta>0$ and for every (fixed) $g\in {\cal H}_N$, the following simple inequality holds: \begin{equation}\label{eq-simple} \begin{split} & \pr{D}[y f(x)\le 0] \\ & = \pr{D}[y g(x)\le \halftheta,\, y f(x)\le 0] \\ & + \pr{D}[y g(x) > \halftheta,\, y f(x)\le 0] \\ % & \le \pr{D}[y g(x)\le \halftheta] + \pr{D}[y g(x) > \halftheta\, |\, y f(x)\le 0] % \pr{D}[y f(x)\le 0] \\ & \le \pr{D}[y g(x)\le \halftheta] + \pr{D}[y g(x) >

\halftheta\, |\, y f(x)\le 0]. \end{split} \end{equation} Note that the left-hand side does not depend on g. By taking the expectation over $g\sim Q^N$ (and exchanging the order of expectations in the second term on the right-hand side), we arrive at \begin{equation} \begin{split} \pr{D}[y f(x)\le 0] & \le E_{g\sim Q^N}\left[\pr{D}[y g(x)\le \halftheta] \right] \\ & + E_D\left[P_{g\sim Q^N}[y g(x)>\halftheta\, |\, y f(x)\le 0] \right]. \end{split} \end{equation} As discussed above, we are now ready to apply \mbox{Hoeffding's} inequality (\ref{eq-purehoeff}) with $\epsilon=\theta/2$. For any fixed (x,y) we have \begin{equation}\label{eq-prefirsthoeff} Pr_{g\sim Q^N}[y g(x)>\halftheta\, |\, y f(x)\le 0] \le e^{-\frac{1}{8} N \theta^2}, \end{equation} so \begin{equation}\label{eq-firsthoeff} \pr{D}[y f(x)\le 0] \le E_{g\sim Q^N}\left[\pr{D}[y g(x)\le \halftheta] \right] + e^{-\frac{1}{8} N \theta^2}. \end{equation} We would like to apply the PAC-Bayes theorem \ref{th-mcall} to the right-hand side. For simplicity we stated theorem \ref{th-mcall} for the common {\em zero-one loss} $I(h(x)\neq y)$, but it holds more generally for arbitrary binary loss functions (see \cite{McAllester}). Here we use the loss function $I(y g(x)\leq \theta/2)$. Recall that theorem \ref{th-mcall} applies for any fixed hypothesis space and "prior" distribution. The hypothesis space here will be ${\cal H}_N$. We use as the "prior" the distribution P^N over ${\cal H}_N$, which is constructed from the prior P over ${\cal H}$ exactly as Q^N is constructed from Q (see (\ref{eq-sample})). It is easy to see that $\KL(Q^N\| P^N) = N \KL(Q\|P)$.\footnote{Note that this reveals a tradeoff between N and $\KL(Q\| P)$. Namely, for large N, $g\sim Q^N$ will be a close approximation to the averaging classifier f, which keeps (\ref{eq-prefirsthoeff}) small, but if $\KL(Q\| P)$ is not very small, Q^N will be rather far from P^N in terms of relative entropy, as a consequence of the strict factorized forms of the two distributions (they are constructed using i.i.d.\ samples of size N).}

It follows from Theorem \ref{th-mcall} that with probability at least $1-\delta$ over random choices of S, for every Q, \begin{equation}\label{eq-usemcall} \begin{split} & E_{g\sim Q^N}\left[\pr{D}[y g(x)\le \halftheta] \right] \\ & \le E_{g\sim Q^N}\left[\pr{S}[y g(x)\le \halftheta] \right] \\ & + \sqrt{\frac{N \KL(Q\|P) + \ln m + \ln(1/\delta) + 2} {2m-1}}. \end{split} \end{equation} By the same argument as in (\ref{eq-simple}), for every $g\in {\cal H}_N$: \begin{equation} \begin{split} & \pr{S}[y g(x)\le \halftheta] \\ & \le \pr{S}[y g(x)\le \halftheta,\; y f(x) > \theta] + \pr{S}[y f(x) \le \theta] \\ & \le \pr{S}[y g(x)\le \halftheta\, |\, y f(x) > \theta] + \pr{S}[y f(x) \le \theta]. \end{split} \end{equation} Again, we take expectations over $g\sim Q^N$ on both sides, interchange the order of the expectations and apply Hoeffding's inequality (\ref{eq-purehoeff}) with $\epsilon=\theta/2$: \begin{equation} E_S\left[P_{g\sim Q^N}\left[y g(x)\le \halftheta\, |\, y f(x) > \theta \right] \right] \le e^{-\frac{1}{8}N\theta^2}, \end{equation} to arrive at \begin{equation}\label{eq-secondhoeff} E_S\left[P_{g\sim Q^N}[y g(x)\le \halftheta] \right] \le e^{-\frac{1}{8}N\theta^2} + \pr{S}\left[y f(x)\le \theta \right]. \end{equation} Combining (\ref{eq-firsthoeff}), (\ref{eq-usemcall}) and (\ref{eq-secondhoeff}), we conclude that with probability at least $1-\delta$, for every Q \begin{equation} \begin{split} \pr{D}\left[y f(x)\le 0 \right] - \pr{S}\left[y f(x)\le \theta \right] \le 2 e^{-\frac{1}{8}N\theta^2} \\ + \sqrt{\frac{N \KL(Q\|P) + \ln m + \ln(1/\delta) + 2}{2m-1}}. \end{split} \end{equation} This bound holds for any fixed N and θ, which is not yet what we need here, since we want to allow these to depend on the data S. We apply a standard technique to resolve this problem. In essence, the bound we proved so far is a statement about certain events, parameterized by N and θ, namely the probability of each event is smaller than δ. However, we need to prove that the probability of the {\em union} of all these events is smaller than δ. To this end, we first observe that this union is contained in the union of a {\em countable} number of events. Note that if $g\in {\cal H}_N$ (see (\ref{eq-sample})), then $g(x)\in \{(2k-N)/N | k=0,1,\ldots,N\}$. Thus, even with all the possible (positive) values of θ, there are no more than $N+1$ events of the form $\{y g(x)\le \theta/2\}$. Denote by $k(\theta,N)$ the largest integer $k\le N$ such that $k/N \le \theta/2$. We observe that for every $\theta>0$, every $g\in {\cal H}_N$ and every distribution over (x,y): \begin{equation} Pr\left[y g(x)\le \theta/2 \right] = Pr\left[y g(x)\le k(\theta,N)/N \right]. \end{equation} This means that the middle step in the proof above, i.e.\ the application of theorem \ref{th-mcall}, depends on (N,θ) only through (N,k). Since the other steps, i.e. the applications of Hoeffding's inequality, are true with probability one, we see that we can restrict ourselves to the union of countably many events, indexed by (N,k). Now, we “allocate” parts of the confidence quantity δ to each of these events, namely (N,k) receives $\delta_{N,k}=\delta/(N(N+1)^2),\; N=1,2, \dots;\, k=0,\dots,N$. It follows easily that the union of all these events has probability at most $\sum_{N,k} \delta_{N,k} = \delta$. Therefore we have proved that with probability at least $1-\delta$ over random choices of S, for {\em all} N and {\em all} $\theta>0$, \begin{equation}\label{eq-finalbound} \begin{split} & \pr{D}\left[y f(x)\le 0 \right] - \pr{S}\left[y f(x)\le \theta \right] \\ & \le 2 e^{-\frac{1}{8}N\theta^2} + \sqrt{\frac{N \KL(Q\|P) + \ln m + \ln(1/\delta_{N,k}) + 2}{2m-1}} \\ & \le 2 e^{-\frac{1}{8}N\theta^2} \\ & + \sqrt{\frac{N \KL(Q\|P) + \ln m + \ln(1/\delta) + 3 \ln(N+1) + 2}{2m-1}} \end{split} \end{equation} where $k=k(\theta,N)$. The asymptotic bound stated in the theorem can be derived by choosing N (with respect to θ and Q) so as to approximately minimize the bound we have derived above. We can choose \[N=\left\lceil 4 \theta^{-2} \ln \frac{m}{\KL(Q||P) + 1}\right\rceil. \]

\section{Implications}\label{sec-implic}

We wish to apply the preceding theory to two general learning methods: Maximum Entropy discrimination\cite{Jaakkola} and Bayes as well as Bayes Point Classifiers \cite{Minka} \cite{Herbrich}. We choose these two learning methods because the average in these cases is over many hypotheses, so that the low order terms in the bound are not very significant. We begin with a simple toy example that illustrates the bound application.

\subsection{Example}\label{sec-toyex} A quick example will illustrate the advantage of the improved bound. Suppose the input space is ${\cal X} = \{-1,1\}^n$, and let ${\cal H}= \{h_1,\ldots,h_n\}$, where for every $x=(x_1,\ldots,x_n)\in {\cal X}$, $h_i(x)=x_i$, $i=1,\ldots,n$. Fix a parameter $0< \theta< 1$.

The setting falls within the naive Bayes probability model. The probability distribution D can be described as follows: First, the value of y is 1 with probability 0.5, and -1 with probability 0.5. Given y, the entries of an instance $(x_1,\ldots,x_n,y)\in{\cal X}\times\{-1,1\}$ are (conditionally) independent. For every i, x_i equals y with probability $\half+\half \theta$, and $-y$ with probability $\half-\half\theta$.

It follows, that for every i, $y h_i(x)=1$ (i.e., h_i predicts correctly) with probability $\half+\half\theta$, so the expected value of $y h_i(x)$ is $\half+\half\theta-(\half-\half\theta)= \theta$. Thus, the expected value of $t(x,y) = \sum_{i=1}^n q_i (yh_i(x))$ is also θ. For a large number of independent hypotheses, with uniform weights q_i, the value of $t(x,y)$ is probably approximately θ.

What will the old margin bound suggest using? The old margin bound depends purely on the proportion of examples at some margin so it suggests averaging over the few hypotheses which happen to do better than expected on this particular sample set.

What does the improved averaging bound suggest? The improved averaging bound will include many more hypotheses because it becomes tighter with a more uniform average over the hypothesis space.

We implemented two quick learning algorithms to explore the implications of this bound to this problem.

The first algorithm, which motivated the development of the original margin bound, is Adaboost \cite{Freund}. Our Adaboost implementation uses a weak learning algorithm which simply selects the hypotheses with smallest error under the distribution over examples and we set the number of rounds to \(100 \).

The second algorithm is a Gibbs averaging algorithm. The Gibbs averaging algorithm picks a weight for each hypothesis proportional to \[e^{e_S(h)/T}\ , \] where \(T \) is a "temperature" parameter. Motivated by the variance of a binomial distribution, we set \(T = 1/\sqrt{n} \). After finding the weights of all hypotheses we create an averaging classifier by taking the sign of the expectation with respect to the Gibbs distribution.

In real world examples, the number of hypotheses is typically much larger than the number of examples so we use examples with 10000 hypotheses/features and 10 to 150 examples. For all experiments we set the true margin in data generation to 0.5.

{\par\centering \resizebox*{0.9\columnwidth}{!}{\rotatebox{270}{\includegraphics{plot.ps}}} \par}

This is a plot of the complexity \(\KL(Q\|P)\) of the averaging hypotheses returned by Adaboost and Gibbs-averaging versus the number of examples. All error bars are at one standard deviation.

Adaboost just picks one hypothesis (the one which happens to get every example correct) when the number of examples is small and eventually limits to a near uniform distribution over as many hypotheses as it has iterations. The Gibbs-averaging hypothesis instead always controls complexity well. The error bars are very small everywhere except for 30 and 40 examples where Adaboost suddenly starts using more then one hypothesis.

This example is arranged so the "right" answer is to use every hypothesis with the same weight. In general, the goals of complexity control and error minimization are often opposed and the averaging bound suggests how to trade off between these goals.

\subsection{Maximum Entropy Discrimination}

Maximum Entropy discrimination (MED) is tailor-made to take advantage of our new bound. This is especially interesting because it was proposed before the averaging bound was developed. Consequently, the application of the improved averaging bound to the MED framework provides an additional motivation for its use.

Maximum Entropy discrimination (MED) is founded on Minimum Relative Entropy discrimination (MRED) which is equivalent to MED when the prior happens to be uniform. In the MRED paradigm, one starts with some prior distribution P over the hypothesis space and the goal is to find a distribution Q which minimizes the KL-divergence $\KL(Q||P)$ to the prior subject to classification constraints (further explained below). The latter are stated in terms of the expectation over \(Q \) of the so-called discriminant function (see below).

For each hypothesis h, the discriminant function $L(x) = L(x|h)$ assigns real numbers to examples $x\in \cal X$. The value of $L(x|h)$ can be interpreted as a “confidence-rated” classification. Thus, if $L(x|h)$ is large, the hypothesis h places great confidence on the classification of x as positive.

For every h, the discriminant function $L(\cdot|h)$ is determined by a parameter triple $\Theta^h=\{\theta^h_{+},\theta^h_{-},b^h\}$. It is derived from some parameterized family $P(\cdot|\theta)$ of probability distributions over the set $\cal X$. The discriminant function is: \[L(x|h)=\ln \frac{P(x|\theta^h_{+})} {P(x|\theta^h_{-})} +b^h \ . \] Intuitively, $P(\cdot|\theta^h_{+})$ (resp.\ $P(\cdot|\theta^h_{-})$) is the posterior distribution over $\cal X$, given a “positive” (resp.\ “negative") classification. Thus, the discriminant is a “biased” (by b^h) log-likelihood ratio with respect to the distributions $P(\cdot|\theta^h_{+})$ and $P(\cdot|\theta^h_{-})$. Every distribution Q over $\cal H$ induces an average over the discriminant functions of the individual hypotheses. The constraints imposed on \(Q \) guarantee a desired margin θ: \[\forall x,y\in S:\,\,y\int_{\cal H}L(x|h)\, dQ(h)\geq \theta \ .\] Classification is then done in the MED framework by calculating the expected value of the discriminant function under the distribution Q: \[c(x)=\mathop{\rm sign} \int_{\cal H}L(x|h)\, dQ(h) \ . \] For details on how to find \(Q \) subject to these constraints see \cite{Jaakkola}.

How does our theoretical result apply to MRED? The latter averages over real-valued discriminant functions $L(x|h)$ instead of binary hypotheses h, as in our setting. As already mentioned above, our main theorem \ref{th-main} holds also for spaces ${\cal H}$ of {\em bounded real-valued} hypotheses, without further loss. If the $L(x|h)$ in an MRED application are bounded, our result therefore applies, using the hypothesis space $\{ L(x|h) | h\in {\cal H} \}$ instead of ${\cal H}$ directly. However, in most MRED applications, the discriminant functions are not bounded, and an extension of our result to this case is subject to future work. %In all applications of MRED we know of, the class of discriminant %functions {\em is} uniformly bounded if we restrict ourselves to a %bounded region in ${\cal X}$ where all the data lies with high %probability. Therefore, even our current result motivates MRED in a %strong theoretical sense.

The algorithm directly motivated by the averaging bound would be “Minimum Relative Entropy Classification” (MREC) which is identical to the MRED framework except that instead of averaging over real-valued discriminants the average is done over binary valued classifiers. It is unclear whether the MREC criteria is actually better then the MRED framework in practice for either accuracy or ease of solution.

\subsection{Bayes and Bayes Point Classifiers}

In the Bayesian approach to classification, given an i.i.d.\ training sample S, a posterior distribution \(Q \) over the hypotheses is derived according to Bayes law: \[dQ(h) = d\Pr(h | S) = \frac{\Pr(S | h)\, d\Pr(h)}{\int \Pr(S | h)\, d\Pr(h)} \ . \] Here, $\Pr(S | h)$ is the likelihood of the sample S, given the hypothesis h, and $\Pr(h)$ is a prior distribution over ${\cal H}$. The classification for a new example x is done by calculating the expectation over \(Q \), as in the MRED classifier above: \[c(x)=\mathop{\rm sign} \int_{\cal H} h(x)\, dQ(h)\ . \] The Bayes (optimal) classifier is an average over many classifiers, and so our improved averaging bound applies with a prior P, given by $P(h) = \Pr(h)$, and a "posterior" \(Q \), which in this case is the Bayesian posterior distribution.

One significant drawback of this technique is that it is very often intractable. Bayes Point classifiers attempt to address this intractability by finding a {\em single} hypothesis \(h_{BP}(x)\in {\cal H} \), which is close to the Bayes (optimal) classifier \(c(x) \). Thus, if \(h_{BP}(x) \) is a good approximation to \(c(x) \), then the improved averaging bound will approximately apply to the Bayes Point classifier as well.

\section{Conclusion and Future Work}

We have presented a simple qualitative improvement to the margin bound, which motivates the techniques of several learning algorithms and validates the intuition that "averaging is good". There are many directions for interesting future work including the following:

\begin{enumerate} \item The improved averaging bound has some messy low order constants, which are probably removable with an improved argument. \item Can we give a stronger theoretical motivation of the Maximum Entropy discrimination framework with unbounded discriminant functions? \item Empirical application of the bound. When applying this bound in the boosting framework, can we get quantitatively interesting results on real world problems? \end{enumerate}

4.8. Computable Shell bounds (exceeding the speed of light)

We are concerned with the problem of bounding the true error rate of a classifier learned via training on m examples. Shell bounds [**?**] are the tightest known distribution free bounds on the true error rate of a hypothesis chosen from a discrete hypothesis set. We improve and generalize shell bounds.

The improvements result in a qualitative change in the understanding of the Shell Bound. The original Shell Bound divided all hypotheses according to their true error into m shells where every shell consisted of hypotheses with true error rate in $[\frac{i}{m}, \frac{i+1}{m})$. Then, it was proved that with high probability most hypotheses will have an empirical error "near" to their true error. Given this locality, the size of the set of all hypotheses with true error near 0.5 (for example) can be overestimated by measurement of the empirical errors of "nearby" hypotheses. Given this overestimate of the distribution of true errors, the Shell Bound proves that the probability of a high true error hypothesis producing a misleadingly small empirical error is small.

The shell bound presented here, in some sense, is not a shell bound. In particular, we can avoid the initial discretization of the true errors into "shells" and instead only derive our notion of a "shell" from the observed empirical errors. Start with the distribution of empirical errors over hypotheses and subtract a small amount from the empirical error rates to create a pessimistic distribution. With high probability, the cumulative of the pessimistic distribution will lower bound the cumulative distribution of hypothesis true error rates. Given this, we can directly calculate a bound on the probability that a "large" hypothesis will produce a misleadingly small error.

The Shell bound improvements are relatively small—merely $O(\frac{\ln m}{m})$, but important because they reduce the gap between the upper and lower bounds to a difference of just $\frac{\ln m}{m}$. Luckily, this gap can be reduced by lowering the upper bound rather than raising the lower bound. This lowering of the upper bound is done with more refined argumentation which may be easier for some to follow.

A significant drawback to the shell bounds is that they were stated only in terms of a discrete hypothesis space. Here, we will state a generalization of the shell bounds which allows for nontrivial bounds on continuous hypothesis spaces. The style of this extension is similar to the PAC-Bayes [**?**] approach and results in a bound on a stochastic classifier where the randomization in the classifier is over a sufficiently large subset of the hypothesis space. The results presented here show that the PAC-Bayes approach can be mixed with the Shell bound approach to generate an improved bound for continuous hypothesis spaces.

The Shell bound generalization has a nice property: When applied to a discrete hypothesis space, it produces essentially the same bound as the discrete Shell bound on the same discrete space. This property is not shared by most covering number (and VC) bounds for continuous spaces. The tight extension property can be considered evidence that the argument for the bound does not introduce too much slack between the lower and upper bounds.

The remainder of this paper first introduces the setting and then is organized in two sections which:

(1) Present the more refined shell bound argument.
(2) Present the generalization of the shell bound to continuous hypothesis spaces.

4.8.1. The Learning Setting. We will work in the standard agnostic distribution free setting. In particular, assume that we have m examples of *(input,ouput)* pairs (x, y) drawn from some unknown distribution D. The set of m examples will be denoted by S. The goal of machine learning is to find a hypothesis, h, which will have a small true error rate. The true error rate is the expected future error rate and is defined as:

$$e(h) = \Pr_D(h(x) \neq y)$$

In general, it is not possible to guarantee learning for arbitrary problems, but it is possible to guarantee convergence of the empirical error to the true error. The empirical error is the error rate on the m examples and is defined as:

$$\hat{e}(h) = \Pr_S(h(x) \neq y)$$

The empirical error rate is a Binomially distributed random variable with bias equal to the true error rate. Our complications arise when we wish to bound the true error rate of *many* hypotheses simultaneously.

Of central importance in this work are bounds the cumulative distribution of a Binomial. Let $B(\leq K, e(h), m)$ be the probability that a Binomial distributed random variable of m coins with bias $e(h)$ produces K or fewer heads (or "errors" in our setting). Also, let $B(>K, e(h), m) = 1 - B(\leq K, e(h), m)$ be the probability of producing more than K heads.

We will often wish to approximate the Binomial distribution using the Relative Entropy Chernoff bound. Let

$$\mathrm{KL}(\hat{e}(h)||e(h)) = \begin{cases} (1-\hat{e}(h)) \ln \frac{1-\hat{e}(h)}{1-e(h)} + \hat{e}(h) \ln \frac{\hat{e}(h)}{e(h)} & \text{if } \hat{e}(h) > e(h) \\ 0 & \text{otherwise} \end{cases}$$

Note that this is not quite the standard definition of KL divergence—the value decreases monotonically in $\hat{e}(h)$. The Relative Entropy Chernoff bound is:

$$\forall \delta > 0 \ \forall^{\delta} S \ \mathrm{KL}(\hat{e}(h)||e(h)) \leq \frac{\ln \frac{1}{\delta}}{m}$$

Here, the notation "$\forall \delta > 0 \ \forall^{\delta} S$" is as in [**?**] and reads as "For all $\delta > 0$ and for all but δ sample sets S, ..." For more discussion of this approximation, look in [**?**].

4.8.2. Tightening the Discrete Shell Bound. There are actually several independent technical improvements.

(1) Using a one-sided version of lemma 3.4 [**?**] with a Chernoff Relative Entropy style bound rather than a Hoeffding style bound.
(2) Tightening lemma 3.2 [**?**] by insisting on the convergence of only one random variable rather than m random variables.
(3) Bounding the "cumulative distribution of hypotheses" rather than the "probability distribution of hypotheses" to avoid overcounting the number of hypotheses by about a factor of $\sqrt{m}$.

We start with a replacement of lemma 3.1 in [**?**]. In this replacement, we avoid the need to prove the convergence of m distinct random variables by working directly with the probability that a high true error hypothesis produces a small empirical error. This gives us an $O(\frac{\ln m}{m})$ improvement in tightness since we do not need to apply the union bound m times.

For the next lemma, we define:

$$P(\epsilon, K) \equiv \sum_{h: e(h) > K + \epsilon} B(\leq K, e(h), m)$$

Intuitively, $P(\epsilon, K)$ is a bound on the probability that a hypotheses with a true error larger than $K + \epsilon$ will have an empirical error of K. The contribution to the sum will fall off exponentially as the true error, $e(h)$, increases.

Theorem 4.8.1. *(Full knowledge)*

$$\forall \delta > 0 \ \forall^{\delta} S \ \forall h \in H \ e(h) \leq \hat{e}(h) + \epsilon(\delta, \hat{e}(h))$$

where $\epsilon(\delta, K) = \min \left\{ \epsilon : \ P(\epsilon, K) \leq \frac{\delta}{m} \right\}$

Proof. For every hypothesis with a true error rate of

$$e(h) > K + \epsilon(\delta, K)$$

the probability of producing an empirical error of

$$\hat{e}(h) \leq K$$

is $B(\leq K, e(h), m)$. Applying the union bound over all hypotheses and all m possible nontrivial values of K completes the proof. □

Note McDiarmid's inequality [**?**] implies that the range of hypotheses with minimum empirical error is of size $O(\sqrt{m})$ with high probability. Therefore, if we are only concerned with the learning algorithm which chooses the hypotheses with minimum empirical error, the last $\frac{\ln m}{m}$ term could be reduced to $\frac{\frac{1}{2}\ln m}{m}$ by noting that we need only apply the union bound to $O(\sqrt{m})$ possible minimum empirical error rates.

We are ready to state and prove the improved Shell bound. First, we need a couple of definitions.

$$\bar{e}(\epsilon, K, \delta, \frac{i}{m}) = \max \left\{ K+\epsilon,\ \min \left\{ p:\ \mathrm{KL}\left(\frac{i}{m}||p\right) \leq \frac{\ln \frac{2m}{\delta}}{m-1} \right\} \right\}$$

$$\hat{P}(\epsilon, K, \delta) \equiv 2 \sum_{h} B(\leq K, \bar{e}(\epsilon, K, \delta, \hat{e}(h)), m)$$

Noting that there are only $m+1$ possible empirical errors, we can first let

$$c\left(\frac{i}{m}\right) = \left| \left\{ h:\ \hat{e}(h) = \frac{i}{m} \right\} \right|$$

Then we can redefine:

$$\hat{P}(\epsilon, K, \delta) \equiv 2 \sum_{i=0}^{m} c\left(\frac{i}{m}\right) B(\leq K, \bar{e}(\epsilon, K, \delta, \frac{i}{m}), m)$$

Later, we will prove that with high probability, $\hat{P}(\epsilon, K, \delta) \geq P(\epsilon, K)$. Given that this is so, we can prove a theorem which *only* relies on observable quantities.

THEOREM 4.8.2. *(Improved Observable Shell Bound)*

$$\forall \delta > 0\ \forall^{\delta} S\ \forall h \in H\ e(h) - \hat{e}(h) \leq \epsilon(\delta, \hat{e}(h))$$

where

$$\epsilon(\delta, K) = \min \left\{ \epsilon:\ \hat{P}\left(\epsilon, K, \frac{\delta}{2}\right) \leq \frac{\delta}{2m} \right\}$$

It is interesting to observe the implications of this bound. The standard bound can be written in the following form:

THEOREM 4.8.3. *(Discrete Hypothesis Bound)*

$$\forall \delta > 0\ \forall^{\delta} S\ \forall h \in H\ e(h) - \hat{e}(h) \leq \epsilon(\delta, \hat{e}(h))$$

where

$$\epsilon(\delta, \hat{e}(h)) = \min \left\{ \epsilon:\ |H| B(\leq K, K+\epsilon, m) \leq \delta \right\}$$

By inspection, we lose at most factor of $\frac{\ln m + \ln 2}{m}$ when using the Improved Shell Bound. When most of the true error rates are "far" from the empirical error rate (and m is large enough), we expect to make large (functional) improvements on the standard Discrete Hypothesis Bound. Since the Discrete Hypothesis Bound has a matching lower bound on some learning problems, we note that the Improved Observable Shell Bound is close to a lower bound on these same problems.

The proof of the theorem rests upon several lemmas which will be presented next. First, we will make a direct improvement of Lemma 3.4 from[**?**]. The improved version of Lemma 3.4 states:

Lemma 4.8.4. *(Upper Tail Distribution Convergence)*

$$\forall \alpha > 0 \; \forall \delta > 0 \; \forall^\delta S \; \forall Q \sim h \; E_Q e^{(1-\alpha) m KL(\hat{e}(h)||e(h))} < \frac{1}{\alpha\delta}$$

Proof. detailed in [?]. The technique is similar to Lemma 3.4 of [?] although applied in only a one-sided way and using the Relative Entropy Chernoff bound rather than a Hoeffding style bound. □

This lemma will allow us to make two distinct improvements. First, we will be able to find a worst-case *cumulative* distribution function rather than a worst-case *probability* distribution function. Given the worst-case cumulative distribution function, we will be able to calculate a worst-case probability distribution which avoids overcounting, in contrast with the previous approach. Overcounting typically occurred about $\sqrt{m}$ times implying another $O\left(\frac{\ln m}{m}\right)$ improvement in tightness. In addition, we use the relative Chernoff bound rather than the looser Hoeffding style bound: $B(> C, e(h), m) \leq e^{-m(C-e(h))^2}$ which gives us a functional improvement when $e(h)$ is extreme (near 0 or 1). In learning, we are particularly concerned with exreme true error rates.

Our next step is the analog of Lemma 3.2 from [?]. Here, we prove that our estimate, $\hat{P}$, is usually larger than a bound on the true error rate given by the unobservable quantity, P.

Lemma 4.8.5. *(Unobservable bound)* $\forall K \; \forall \delta > 0 \; \forall^\delta S \; \forall Q \sim h$

$$2 \sum_h Q(h) B(\leq K, \bar{e}(\epsilon, K, \delta, \hat{e}(h)), m) \geq \sum_{h: e(h) > K+\epsilon} Q(h) B(\leq K, e(h), m)$$

This lemma is powerful because it bounds the unobservable right hand side in terms of the observable left hand side.

Proof. Choose $P(h) \propto Q(h)B(\leq K, e(h), m)$ for $e(h) > K+\epsilon$ and 0 otherwise. Also choose $\alpha = \frac{1}{m}$ and apply the upper tail distribution convergence lemma 4.8.4 to get:

$$E_{h\sim P} e^{(m-1)\mathrm{KL}(\hat{e}(h)||e(h))} \leq \frac{m}{\delta}$$

$$\Rightarrow \Pr_{h\sim P}\left(e^{(m-1)\mathrm{KL}(\hat{e}(h)||e(h))} \geq \frac{2m}{\delta}\right) \leq \frac{1}{2}$$

$$\Rightarrow \Pr_{h\sim P}\left(e^{(m-1)\mathrm{KL}(\hat{e}(h)||e(h))} \leq \frac{2m}{\delta}\right) \geq \frac{1}{2}$$

$$\Rightarrow \Pr_{h\sim P}\left(\mathrm{KL}(\hat{e}(h)||e(h)) \leq \frac{\ln \frac{2m}{\delta}}{m-1}\right) \geq \frac{1}{2}$$

$$\Rightarrow \frac{\sum_{h: e(h) > K+\epsilon \wedge \mathrm{KL}(\hat{e}(h)||e(h) \leq \frac{\ln \frac{2m}{\delta}}{m-1}} Q(h) B(\leq K, e(h), m)}{\sum_{h: e(h) > K+\epsilon} Q(h) B(\leq K, e(h), m)} \geq \frac{1}{2}$$

$$\Rightarrow 2 \sum_h Q(h) B(\leq K, \bar{e}(\epsilon, K, \delta, \hat{e}(h)), m) \geq \sum_{h: e(h) > K+\epsilon} Q(h) B(\leq K, e(h), m)$$

Applying the definitions gives us the Lemma. □

The unobservable bound is improved by a factor of $O\left(\frac{\ln m}{m}\right)$ over the analog lemma 3.2 of [?] because we do not need convergence for every shell individually. Instead, convergence for the entire sum at once is required.

We now have all the tools required to prove the theorem.

PROOF. (of theorem 4.8.2) Choose $Q(h)$ = the uniform distribution on a our hypothesis space. Then, we know that with probability $1-\delta$, $\hat{P}(\epsilon, K, \delta) \geq P(\epsilon, K)$. Therefore, we can allocate a $\frac{\delta}{2}$ probability of failure to the unobservable bound 4.8.5 and a $\frac{\delta}{2}$ probability of failure to the full knowledge bound 4.8.1. Assuming $\hat{P}(\epsilon, K, \delta) \geq P(\epsilon, K)$, the observable bound will be more pessimistic than the full knowledge bound. □

4.8.3. Large Space Shell. Now, we want to derive a theorem for large spaces. In order to do this, we will first assume that there is some measure Q over the space H. Suppose that we choose all of the hypotheses of some empirical error $\hat{e}$. Let $Q(\hat{e})$ be the measure of all of the hypotheses with empirical error $\hat{e}$. In the stochastic hypothesis setting, we will need a different definition of P. Let:

$$P_s(\epsilon,\ K) \equiv \int_{h:e(h)>K+\epsilon} Q(h) B(\leq K, e(h), m) dh$$

We will also need the concept of rounding. Suppose $\frac{1}{m^i} \geq Q(\hat{e}) \geq \frac{1}{m^{i+1}}$ then, define:

$$\lfloor Q(\hat{e}) \rfloor = \frac{1}{m^{i+1}}$$

Now, the following theorem holds:

THEOREM 4.8.6. *(Stochastic Full knowledge)*

$$\forall Q\ \forall \delta > 0\ \forall^{\delta} S\ \forall \hat{e}\ \forall h \in H\ e_Q(h) \leq \hat{e}_Q(h) + \frac{1}{m} + \epsilon(\delta, \hat{e})$$

where $\epsilon(\delta, K) = \min\left\{\epsilon:\ P_s(\epsilon, K) \leq \frac{\delta \lfloor Q(K) \rfloor}{m^3}\right\}$

PROOF. Call a hypothesis with a large true error ($e(h) > K + \epsilon$) and small empirical error ($\hat{e}(h) \leq K$) a "bad" hypothesis. $P_s(\epsilon, K)$ is the expected measure of the bad hypotheses. We will use Markov's inequality to bound the actual measure of bad hypotheses. Then, given that the quantity is bounded, we can bound the expected true error by assuming that we included every bad hypothesis in our posterior.

Let

$$\epsilon_{Ki} = \min\left\{\epsilon:\ P_s(\epsilon, K) \leq \frac{\delta}{m^{3+i}}\right\}$$

Intuitively, ϵ_{Ki} is the value we will use when $\hat{e} = K$ and $\lfloor Q(\hat{e}) \rfloor = \frac{1}{m^i}$. Let

$$\hat{P}_s(\epsilon, K) = \int_{h:e(h)>K+\epsilon \wedge \hat{e}(h) \leq K} p(h) dh$$

be the actual measure of bad hypotheses. Then, Markov's inequality, tells us:

$$\forall \delta > 0\ \Pr_D(\hat{P}_s(\epsilon_{Ki}, K) \geq \frac{m^2}{\delta} P_s(\epsilon_{Ki}, K)) \leq \frac{\delta}{m^2}$$

Taking the union bound over all values of ϵ_{Ki}, we get:

$$\forall \delta > 0 \ \Pr_D(\forall K, i \ \hat{P}_s(\epsilon_{Ki}, K) \geq \frac{m^2}{\delta} P_s(\epsilon_{Ki}, K)) \leq \delta$$

This implies that with probability $1-\delta$ for all values of K and i, we have: $\hat{P}_s(\epsilon_{Ki}, K) \leq \frac{1}{m^{1+i}}$. Therefore, if $Q(\hat{e}) \geq \frac{1}{m^i}$, we know that $\frac{Q(\hat{e})}{\hat{P}_s(\epsilon_{Ki}, K)} \geq \frac{1}{m}$. Assuming that all of the bad hypotheses have true error 1 and all the rest have true error at most $K + \epsilon_{Ki}$, we get the following true error bound:

$$e_Q(h) \leq \hat{e} + \frac{1}{m} + \epsilon_{\hat{e}i}$$

□

The stochastic full knowledge bound is loose when applied to the full knowledge setting by $\delta \rightarrow \frac{\delta}{m^2}$. Typically, this results in only a $\frac{2 \ln m}{m}$ increase in the size of ϵ, although the increase can sometimes be much larger near phase transitions. These factors of $\frac{\ln m}{m}$ may be removeable with improved argumentation. Naturally, stochastic full knowledge bound can be used to prove a stochastic observable bound.

The next theorem is the observable analog of the stochastic full knowledge bound. Here, we eliminate the unobservable quantities to produce a stochastic observable shell bound. The observable quantity we will use is:

$$\hat{P}_s(\epsilon, K, \delta) \equiv 2 \sum_h Q(h) B(\leq K, \bar{e}(\epsilon, K, \delta, \hat{e}(h)), m)$$

THEOREM 4.8.7. *(Stochastic Observable Shell Bound)*

$$\forall Q \ \forall \delta > 0 \ \forall^\delta S \ \forall h \in H \ e_Q(h) \leq \hat{e}_Q(h) + \frac{1}{m} + \epsilon(\delta, \hat{e})$$

where $\epsilon(\delta, K) = \min \left\{ \epsilon : \ \hat{P}_s(\epsilon, K, \frac{\delta}{2}) \leq \frac{\delta \lfloor Q(K) \rfloor}{2m^3} \right\}$

PROOF. First note that the unobservable bound lemma 4.8.5 implies that with probability $1 - \frac{\delta}{2}$, we have $\hat{P}_s(\epsilon, K, \frac{\delta}{2}) \geq P_s(\epsilon, K)$. Given that this is the case, our choice of ϵ will be at least as pessimistic as the choice defined by the Stochastic Full Knowledge bound 4.8.6. We thus have two sources of failure: the unobservable bound lemma will fail with probability at most $\frac{\delta}{2}$ and the stochastic full knowledge bound will fail with probability $\frac{\delta}{2}$. The union bound then implies that the Stochastic Observable Shell Bound holds with probability $1 - \frac{\delta}{2}$. □

4.8.4. Conclusion.

We reduced the gap between the upper and lower bounds for shell bounds to one factor of $\frac{\ln m}{m}$. We have also established a stochastic shell bound which holds (nontrivially) on continuous hypotheses spaces.

There remain several important open questions:

(1) Can we remove the remaining factor of $\frac{\ln m}{m}$?
(2) For the Stochastic Shell bounds, two additional factors of $\frac{\ln m}{m}$ were introduced. Is it possible to remove these factors with an improved argument?
(3) Our extension to the continuous case was done in the style of PAC-Bayes bounds, but a more common technique for extending to the continuous case is via the use of covering numbers. Is there a natural way to extend Shell bounds to the continuous case using the concept of a covering number?

4.9. Tight covering number bounds?

CHAPTER 5

Holdout bounds

Holdout bounds are the simplest setting in which to apply sample complexity results and are therefore presented first. The simplest holdout bound arises with the classical technique of splitting the data set into two pieces: a training set, m_{train} and a test set m_{test}. In this setting, the following simple bound applies:

THEOREM 5.0.1. *(Test sample complexity) Let $\hat{e}_{test}(h)$ be the empirical error on the test set and $e_D(h)$ be the true error rate of the hypothesis, then we have:*

$$\forall h \ \Pr_D(e_D(h) > \max_p p : \ Bin(m_{test}, m_{test}\hat{e}_{test}(h), p) \geq \delta) \leq \delta$$

PROOF. The proof is just a simple identification with the Binomial. For any distribution over (x, y) pairs and any hypothesis, h, there exists some probability, $e_D(h)$, that the hypothesis predicts incorrectly. The distribution of the empirical error rate will then be a Binomial distribution. Given that the distribution is Binomial we calculate an upper bound which holds with high probability. □

There are two immediate corollaries of the holdout theorem which are mathematically simpler although not as tight. The first corollary applies to the limited "realizable" setting where our test error is 0.

COROLLARY 5.0.2. *(Realizable Test Sample Complexity)*

$$\forall h \ \Pr_D(\hat{e}_{test}(h) = 0 | e_D(h) \geq \frac{\ln \frac{1}{\delta}}{m_{test}}) \leq \delta$$

PROOF.

$$\text{Bin}(m_{\text{test}}, 0, \epsilon) = (1 - \epsilon)^{m_{\text{test}}} \leq e^{-\epsilon m_{\text{test}}}$$

Setting this equalt to δ and solving for ϵ gives us the result. □

The second corollary applies to all results.

COROLLARY 5.0.3. *(Agnostic Test Sample Complexity)*

$$\forall h \ \Pr_D(e_D(h) - \hat{e}_{test}(h) \geq \sqrt{\frac{\ln \frac{1}{\delta}}{2m_{test}}}) \leq \delta$$

PROOF. Use the Hoeffding approximation which for $\frac{k}{m_{\text{test}}} < \epsilon$, says:

$$\text{Bin}(m_{\text{test}}, k, \epsilon) \leq e^{-2m_{\text{test}}(\epsilon - \frac{k}{m_{\text{test}}})^2}$$

Now, set this equal to δ, and solve for ϵ to get the result. □

REMARK 5.0.4. Similar theorems apply to bound $\hat{e}_{\text{test}}(h) - e_D(h)$.

How tight is the test sample complexity theorem? The answer is very tight. Let us define

$$\bar{e}(m, \hat{e}(h), \delta) \equiv \max_{p} p : \ \mathrm{Bin}(m_{\mathrm{test}}, m_{\mathrm{test}} \hat{e}_S(h), p) \geq \delta$$

as our true error bound. Then, we wish to know how much $e_D(h)$ and $\bar{e}(m_{\mathrm{test}}, \hat{e}(h), \delta)$ differ. Applying the hoeffding approximation, we know that with high probability, $e_D(h) \geq \bar{e}(m_{\mathrm{test}}, \hat{e}(h), \delta) - 2\sqrt{\frac{\ln \frac{1}{\delta}}{2m_{\mathrm{test}}}}$. Thus the region in which $e_D(h)$ is confined with high confidence is of size $2\sqrt{\frac{\ln \frac{1}{\delta}}{2m_{\mathrm{test}}}}$ or smaller.

It is common practice in the field of machine learning to use the gaussian approximation in reporting error bars. The practice is reasonably safe because it is usually pessimistic. However, it can occasionally lead to embarrasing results where you report error rates such as 0.01 ± 0.02. The test sample complexity theorem is the "right" way to report error bars because it works with the actual distribution over errors. In order to promote such good practices, two programs: upper_bound_test and lower_bound_test are provided.

Given that the bounds for the simple holdout technique are so tight, why do we need to engage in further work? There is one serious drawback to the holdout technique - every application of the holdout technique requires m_{test} otherwise unused examples. This can become remarkably expensive. For example, testing k functions with the holdout bound will require $m_{\mathrm{test}} k$ examples. There are two basic approaches to this difficulty:

(1) Try to reduce m_{test} using more sophisticated holdout techniques.
(2) Don't use a holdout set. Instead, train and test on the same set of examples using a more sophisticated bound.

The next few subsections deal with approach (1) while the next section deals with approach (2).

5.0.1. Cross validation. One of the standard techniques for attempting to improve on the holdout bound is cross validation. K-fold cross validation divides the data into K folds of size $\frac{m}{K}$ (assume m is divisible by K for simplicity). Then, for every fold i, holdout fold i, train on the remainder of the data and test on fold i. Let the hypotheses we found by training be known as $h_1, ..., h_K$ and their respective holdout errors as $\hat{e}_1, ..., \hat{e}_K$. Also let $\hat{e}_{cv} = \frac{1}{K} \sum_{i=1}^{K} \hat{e}_i$

There are several variations of cross validation. If $K = m$, the procedure is often called "leave one out cross validation". In one variant, you train on all of the data to learn a new hypothesis, h, and assume a true error rate near $\hat{e}_{CV}$. In another variant, you predict according to $h_{cv} = Uniform(h_1, ..., h_K)$. The latter variant is simpler to analyze because linearity of expectation implies that $\hat{e}_{cv}$ is an unbiased estimate of $\bar{e}_{cv}$.

There are strong results known for cross validation on nearest-neighbor, kernel, and histogram classifiers []. In general, only very weak results are known about bounds on the variance of cross validation for general classifiers.

Open problem: bound the variance of cross validation

5.0.2. Progressive validation. Progressive validation is a technique which allows you to use almost $\frac{1}{2}$ of the data in a holdout set for training purposes while still providing the same guarantee as the holdout bound. It first appeared in [] and is discussed in a more refined and detailed form here.

Suppose that you have a training set of size m_{train} and test set of size m_{test}. Progressive validation starts by first learning a hypothesis on the training set and then testing on the first example of the test set. Then, we train on training set plus the first example of the test set and test on the second example of the test set. The process continues m_{test} iterations. Let m abbreviate m_{test}. Then, we have m hypotheses, $h_1, ..., h_m$ and m error observations, $\hat{e}_1, ..., \hat{e}_m$. The hypothesis output by progressive validation is the randomized hypothesis which chooses uniformly from $h_1, ..., h_m$ and evaluates to get an estimated output.

Since we are randomizing over hypotheses trained on m_{train} to $m_{\text{train}} + m_{\text{test}} - 1$ examples, the expected number of examples used by any hypothesis is $m_{\text{train}} + \frac{m_{\text{test}}-1}{2}$. Given that training can exhibit phase transitions, the extra few examples can greatly improve the accuracy of the trained example.

The true error rate of this randomized hypothesis will be:

$$e_{\text{pv}} = \frac{1}{m}\sum_{i=1}^{m} e(h_i)$$

and the empirical error estimate of this randomized hypothesis will be:

$$\hat{e}_{\text{pv}} = \frac{1}{m}\sum_{i=1}^{m} \hat{e}_i$$

Bounding the deviation of $\hat{e}_{\text{pv}}$ is more difficult than bounding the deviation of a holdout error. To understand this, we can think of two games, the holdout game and the progressive validation game.

In the holdout game, your opponent chooses a bias and then nature flips m coins with that bias. If the deviation of the average number of heads is larger than ϵ, then you lose. Otherwise, you win.

In the progressive validation game, the opponent chooses the bias of *each* coin just before it is flipped. The goal of the opponent remains the same, and the opponent wins if a large deviation is observed.

The progressive validation opponent is stronger than the holdout opponent since the progressive validation opponent could choose the same bias for every coin. Nonetheless, we will see that the progressive validation opponent is *not* much stronger.

There are two ways in which we can show that the progressive validation opponent is not much stronger. The first technique will show that the variance of the progresssive validation estimate is smaller than might be expected.

THEOREM 5.0.5. *Suppose we test the progressive validation hypothesis on m additional examples. Let $\hat{e}_{test}$ be the empirical error on these m examples. Then, we have:*

$$E(\hat{e}_{test} - e_{pv})^2 \geq E(\hat{e}_{pv} - e_{pv})^2$$

PROOF. Every example on the left hand side can be though of as a coin with biase e_{pv}. The variance of the LHS is then $m * e_{\text{pv}}(1 - e_{\text{pv}})$. The right hand side is:

$$E(\hat{e}_{\text{pv}} - e_{\text{pv}})^2$$

$$= E\left[\sum_{i=1}^{m} \hat{e}_i - e_i\right]^2$$

$$= E\left[\sum_{i\neq j}(\hat{e}_i - e_i)(\hat{e}_j - e_j) + \sum_{i=1}^{m}(\hat{e}_i - e_i)^2\right]$$

$$= \left[\sum_{i\neq j}E(\hat{e}_i - e_i)(\hat{e}_j - e_j) + \sum_{i=1}^{m}E(\hat{e}_i - e_i)^2\right]$$

The cross product terms have expectation zero because $E(\hat{e}_i - e_i) = 0$ regardless of the value of $\hat{e}_j$. This implies:

$$= \sum_{i=1}^{m} E\left[(\hat{e}_i - e_i)^2\right]$$

$$= \sum_{i=1}^{m} e_i(1 - e_i)$$

So, all that we must show is:

$$me_{\text{pv}}(1 - e_{\text{pv}}) \geq \sum_{i=1}^{m} e_i(1 - e_i)$$

which follows from Jensen's inequality and the convexity of $x(1-x)$ on the interval $[0, 1]$. □

The second way in which we can show that the progressive validation opponent is not so strong is by bounding the deviation directly. Surprisingly, we can prove exactly the same bound as for Hoeffding inequality.

THEOREM 5.0.6.

$$\Pr_{D^m}(e_{pv} \geq \hat{e}_{pv} + \epsilon) \leq e^{-2m\epsilon^2}$$

PROOF. (A variant of Chernoff)

$$\Pr_{D^m}(e_{\text{pv}} \geq \hat{e}_{\text{pv}} + \epsilon)$$

$$\Leftrightarrow \forall\lambda \ \Pr_{D^m}(e^{\lambda m e_{\text{pv}}} \geq e^{\lambda m(\hat{e}_{\text{pv}}+\epsilon)})$$

$$\Leftrightarrow \Pr_{D^m}(e^{\lambda m(e_{\text{pv}} - \hat{e}_{\text{pv}} - \epsilon)} \geq 1)$$

$$\leq E_{D^m} e^{\lambda m(e_{\text{pv}} - \hat{e}_{\text{pv}} - \epsilon)}$$

$$= E_{D^m} e^{\lambda(\sum_{i=1}^{m} e_i - \hat{e}_i - \epsilon)}$$

$$= \prod_{i=1}^{m} E_D\left[e^{\lambda(e_i - \hat{e}_i - \epsilon)}|\hat{e}_1, ..., \hat{e}_{i-1}\right]$$

The value of $e^{\lambda(e_i - \hat{e}_i - \epsilon)}$ is only dependent on $\hat{e}_1, ..., \hat{e}_{i-1}$ through the value of e_i. For all possible hypotheses, h_i, we know that $\hat{e}_i$ is a Bernoulli random variable with bias $e_i = \Pr_D(h_i(x) \neq y)$. This is true regardless of how we choose h_i. Consequently, we have:

$$= \prod_{i=1}^{m} E_D\left[e^{\lambda(e_i - \hat{e}_i - \epsilon)}\right] \tag{5.0.1}$$

Since this is true for all λ, we can choose the optimal λ. In general, this is an optimization problem which can be worst-case simplified with the inequality:

$$E_D e^{\lambda(e_i - \hat{e}_i)} \leq e^{\frac{\lambda^2}{8}}$$

which implies:

$$\forall\lambda \ \leq \prod_{i=1}^{m} e^{\frac{\lambda^2}{8} - \lambda\epsilon)}$$

This is optimized for $\frac{2\lambda}{8} = \epsilon$ which implies $\lambda = 4\epsilon$ giving:

$$\leq \prod_{i=1}^{m} e^{-2\epsilon^2}$$

$$= e^{-2m\epsilon^2}$$

□

Open Problem: Starting with 5.0.1, derive a bound similar to the relative entropy chernoff bound which holds for progressive validation

5.0.3. Combining sample complexity and holdout bounds. include write up

5.1. Experimental results

5.1.1. decision trees. do more experiments.

5.1.2. neural networks. include paper

5.2. Open questions and Challenges

summary

5.3. Technical definitions

Term	Definition
x	The “input” which we can predict with.
y	The “output” which we want to predict.
(x, y)	An “example” or “sample” which is an <input,output> pair
S	A set of samples.
m	The number of samples.
m_{test}	The number of samples in the testing set.
m_{train}	The number of samples in the training set.
h	A hypothesis = a function from x to y
D	An (unknown) distribution over (x, y) pairs.
$\text{Bin}(m, k, p)$	The probability that a Binomial with m coins and bias p has k or fewer heads
$\bar{e}(m, k, \delta)$	An upper bound on the bias of a coin flipped m times with k heads that holds with probability