INTRODUCTORY LEARNING THEORY

JOHN LANGFORD

Abstract. This is an introduction to basic learning theory in the setting of classical statistics. In addition, we discuss simple practical techniques for tightening general learning theory bounds and show their use. Some notes on computational tricks are included.

1. Formal Model

Let X be the space of the input to a predictor and Y be the space of the output. A labeled example (x, y) consists of an input, x and the desired output, y. Our formal model starts with the assumption that all labeled examples are drawn independenty from a distribution D over the space $X \times Y$. This is strictly more general than the 'target concept' model which assumes that there exists some function $f : X \to Y$ used to generate the label [12]. In particular we can model probabilistic learning problems which do not have a particular Y value for each X value. This generalization is essentially "free" in the sense that it does not add to the complexity of presenting the results.

The set of m independently drawn samples presented to a learning algorithm will be denoted as S. The learning algorithm will output a hypothesis $h : X \to Y$ which has some unobservable true error rate $e_D(h)$ and an observable empirical error rate, $\hat{e}_S(h)$.

Definition 1.1. (True error) The true error $e_D(h)$ of a hypothesis 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.

Definition 1.2. (Empirical Error) Given a sample set S, the *empirical error*, $\hat{e}_S(h)$ is defined as:

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

where $I()$ is a function which maps "true" to 1 and "false" to 0. Here $\Pr_S(...)$ is a probability taken with respect to the uniform distribution over the set of examples, S.

2. Basic observations

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 modeled 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 heads 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.

Definition 2.1. (Binomial Distribution) The Binomial distribution is given by:

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

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:

Definition 2.2. (Binomial Tail)

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

= the probability that m coins with bias p produce k or fewer heads.

For the learning problem, we will always choose $p = e_D(h)$ and $\hat{p} = \hat{e}_S(h)$. With these definitions, we can interpret the Binomial tail as the probability of an empirical error less than or equal to $\frac{k}{m}$.

2.2. **Binomial Tail calculation techniques.** One fundamental question arises: How quickly can we calculate the binomial coefficient, $\binom{m}{k}$? This question is important to answer because we will apply the bounds derived later to real problems, and this application will require calculation of Binomial coefficients and Binomial tails.

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 $O(m)$. Since it takes $O(m)$ time to write the answer, we can not hope to compute the answer in less than $O(m)$ time. The problem is more difficult than this though - even calculating the most significant bits of the appears to take $O(m)$ time.

Our real goal is not merely calculating binomial coefficients but rather calculating the probability of a tail, $\mathrm{Bin}(m,k,p)$. How can we calculate the tail probability quickly? For all approaches, it is necessary to calculate $\log \mathrm{Bin}(m,k,p)$ rather than $\mathrm{Bin}(m,k,p)$ to avoid underflow issues. This is generally possible because we can calculate $\log(a+b)$ given $\log(a)$ and $\log(b)$ without losing precision.

There are several possible approaches of increasing sophistication:

(1) Calculate $\binom{m}{i}$ independently from $i=0$ to $i=k$ and use the results to calculate the sum, $\sum_{i=0}^{k}\binom{m}{i}p^i(1-p)^{m-i}$.
(2) Calculate Pascal's triangle and extract the Binomial coefficients.

(3) Use the fact that

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

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

(4) Calculate $\binom{m}{k}$ directly and then $\binom{m}{i-1}$ given $\binom{m}{i}$ until the added quantity 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 sometimes too expensive for us. Luckily, there exist some quick approximations which can reduce the computation to constant time (or $O(\log m)$ time depending on your computational model).

2.3. **Approximation techniques.** Exact calculation of $\mathrm{Bin}(m,k,p)$ can require computation at least proportional to m, which is often too expensive. For the bounds in this thesis, we will only need to calculate an upper bound on the quantity $\mathrm{Bin}(m,k,p)$. There are several inequalities which are often used. The first of these is the Hoeffding inequality[7]. Assume that $\frac{k}{m} < p$ then we have:

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

Intuitively, this inequality can be seen as fitting a gaussian to the Binomial distribution with $p = \frac{1}{2}$. For any particular m, the variance of the Binomial distribution is maximized when $p = \frac{1}{2}$. Therefore, the Hoeffding inequality is relatively 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 $k = 0$. The realizable bound is:

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

The realizable bound is noticeably tighter with an exponent proportional to $\left|p - \frac{k}{m}\right|$ rather than $\left(p - \frac{k}{m}\right)^2$. 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 calculable bound which achieves the generality of the Hoeffding bound along with the tightness of the realizable bound. We have the relative entropy Chernoff bound [3] for $\frac{k}{m} < p$:

$$\mathrm{Bin}(m,k,p) \leq e^{-m\mathrm{KL}\left(\frac{k}{m}||p\right)} \tag{2.1}$$

Here $\mathrm{KL}(q||p) = q \ln \frac{q}{p} + (1-q)\ln\frac{(1-q)}{(1-p)}$ 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 $k = 0$. In between the extremes, the relative Chernoff bound smoothly interpolates between these possibilities.

We are concerned with the different bounds here because much of the learning theory literature (see [12], [6], [11] for examples) 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 than 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 entropy Chernoff bound and $\mathrm{Bin}(m,k,p)$ are not analytically invertible. Lack of invertibility is a theoretical disadvantage because it means we can not easily parameterize our "precision" paremeter, ϵ in terms of δ. Nonetheless, 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 judgments 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 rate of bound failure and then ask ourselves, "What is a bound on the true error rate that holds with probability 0.9?"

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 monotonically related to each other, this inversion can be defined as:

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

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 $1-\delta$ (over the draws of the examples), the true error rate will be less than $\bar{e}(m,k,\delta)$. This is exactly the kind of quantity that we desire in making decisions about which hypothesis is more desirable. This calculation has been done for several values of m and Binomial tail bounds in figure (2.1).

We will use the process of inversion in many places. The fundamental soundness of inversion rests upon the following lemma.

Lemma 2.3. *(Inversion Lemma) For all predicates, $\phi(X)$*

$$\sup_P \Pr_{X\sim P}(\phi_P(X)) \le \delta \Rightarrow \forall P \ \ \Pr_{X\sim P}(\phi_P(X)) \le \delta$$

This lemma is the trivial statement that a set of objects is less than the sup over the set of objects. Nonetheless, it is a very important step which will appeal to implicitly and explicitly later on.

Proof. By contradiction. Assume there exists P such that the right hand side is not satisfied. Then, the left hand side can not be correct. $\square$

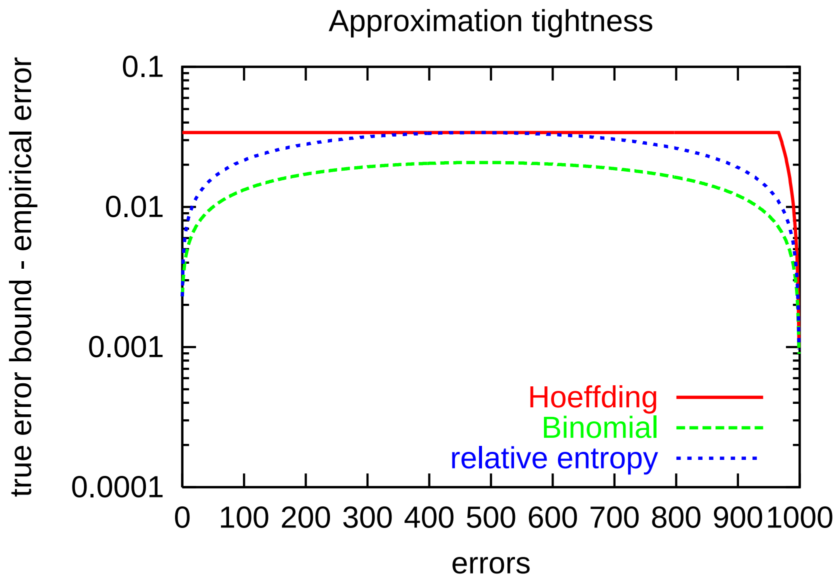

FIGURE 2.1. A plot of the difference between the true error upper bound and the empirical error for various Binomial tail bound approximations. Here $m = 1000$ coins of an unknown bias are used and a confidence of $\delta = 0.1$ and the horizontal axis is the number of errors between $k = 0$ and $k = 1000$. The relative entropy approximation to the Binomial tail is relatively well behaved while the Hoeffding approximation is not. In particular, the Hoeffding approximation does not take into account the decreased variance of low bias Binomials.
Note that the dip at the end of the Hoeffding bound is due to the fact that the true error rate is *always* less than 1.

The Inversion lemma allows us to implicitly parameterize *all* precision parameters $\epsilon(\delta)$ in terms of a fixed probability of failure, δ. This is important for practical application because it means we can choose our probability of failure δ before looking at any examples.

2.5. **Bounding the Union.** One very common technique we will use is the union bound (known as the Bonferroni bound in statistics). Given two coins, each with a bias (probability of heads) 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:

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

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

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

The union bound is the fundamental tool which allows us to reason about multiple hypotheses, each with possibly correllated empirical errors, $X_1 = \hat{e}_S(h_1)$ and $X_2 = \hat{e}_S(h_2)$. The use (and avoidance of the use) of the union bound is a constant issue.

2.6. **Arbitrary Loss functions.** A loss function is *any* function which takes a hypothesis, h, and an example (x, y) as input then outputs a real number. In particular, we could choose $l(h,(x,y)) = I(h(x) \neq y)$ and regard most of the prior discussion as working with a specialized hamming loss function which is 1 when $h(x) \neq y$ and 0 otherwise. Many other possibilities exist. For any bounded loss function, $l(h,(x,y)) \in [0,1]$, we can define:

$$e_D(h) = E_D l(h,(x,y))$$

and

$$\hat{e}_S(h) = E_S l(h,(x,y))$$

All of the bounds reported here will apply for loss functions bounded on the interval $[0,1]$. The fundamental advantage that this gives us is the ability to treat the hypothesis *not* as a black box. Instead, we can derive a bound with a loss function dependent on the structure of the hypothesis (as in margin bounds).

For clarity of presentation, the bounds will all be presented using the hamming loss. However, they will all apply to the more general setting of arbitrary $0-1$ loss functions.

3. Simple Sample Complexity bounds

The observations presented in this chapter are mostly "common knowledge" in the learning theory community. The goal of this chapter is to introduce basic common learning theory which the remainder of the thesis will depend upon.

3.1. **Simple Holdout.** The simplest bound arises for the classical technique of splitting the data set into two pieces: a training set of size m_{train} and a test set of size m_{test}. In this setting, the following simple bound applies:

Theorem 3.1. *(Holdout 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:*

$$\forall h \ \Pr_{D^m}\left(\ e(h) \geq \bar{e}\left(m_{test}, \hat{e}_{test}(h), \delta\right)\right) \leq \delta$$

where $\bar{e}\left(m, \frac{k}{m}, \delta\right) \equiv \max_p\{p: \ Bin(m,k,p) = \delta\}$

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. We can regard this event as a coin flip with bias $e_D(h)$. Since each example is picked independently, 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. $\square$

There are two immediate corollaries of the holdout theorem (3.1) which are mathematically simpler although not as tight. The first corollary applies to the limited "realizable" setting where you happen to observe 0 test errors.

Corollary 3.2. *(Realizable Holdout 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. Specializing theorem 3.1 to the zero empirical error case, we get:

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

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

A second corollary applies to all results, not just those where we observe 0 errors.

Corollary 3.3. *(Agnostic Holdout 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. Loosening theorem 3.1 with the Hoeffding approximation for $\frac{k}{m_{\text{test}}} < \epsilon$, we get:

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

Using the inversion lemma 2.3 we can set this equal to δ, and solve for ϵ to get the result. □

Remark 3.4. 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

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

as our true error bound. We wish to know how much $e_D(h)$ and $\bar{e}(m_{\text{test}}, \hat{e}(h), \delta)$ differ. Applying the Hoeffding approximation, we know that with high probability, $e_D(h) \geq \bar{e}(m_{\text{test}}, \hat{e}(h), \delta) - 2\sqrt{\frac{\ln \frac{1}{\delta}}{2m_{\text{test}}}}$. Thus the region in which $e_D(h)$ is confined with high confidence is of size $2\sqrt{\frac{\ln \frac{1}{\delta}}{2m_{\text{test}}}}$ 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, this can occasionally lead to embarrassing results where error rates such as 0.01 ± 0.02 are reported. The test sample complexity theorem avoids *never* produces an upper bound greater than 1 or lower bound less than 0 because it uses the fundamental Binomial distribution. This approach is the "right" way to report test-set based errors, given the assumption of indepence. Pictorially we can represent this as in figure 3.1.

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—application of the holdout technique requires m_{test} otherwise unused examples. This can strongly degrade the value of the learned hypothesis because an extra m_{test} examples for the training set could reduce the true error of the learned hypothesis from 0.5 to 0.0 on some learning problems. In particular, many learning algorithms implicitly assume that the training error "behaves like" the true error in choosing the hypothesis. With an inadequate number of training examples, there may be very little relationship between the behavior of the training error and the true error.

There are two basic approaches to this difficulty:

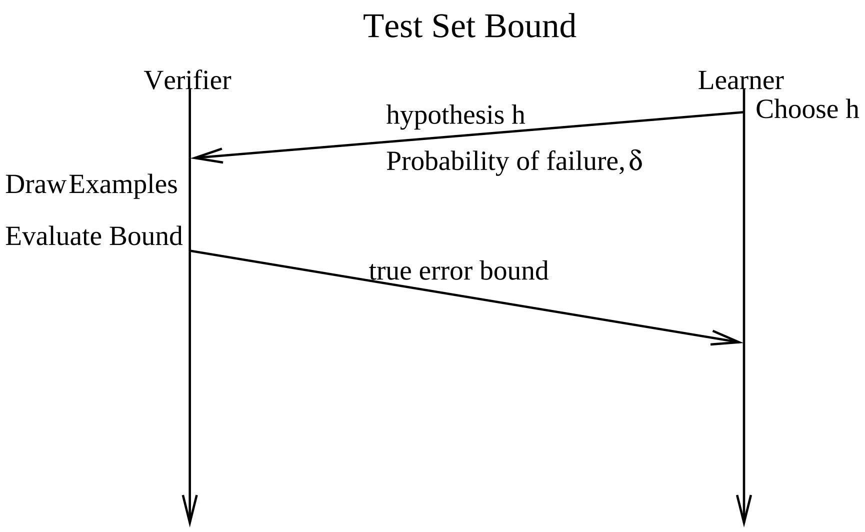

FIGURE 3.1. For this diagram "increasing time" is pointing downwards. The only requirement for applying this bound is that the learner must commit to a hypothesis without knowledge of the test examples. Similar diagrams for other bounds will be presented later (and they are somewhat more complicated). We can think of the bound as a technique by which the "Learner" can convince the "Verifier" that learning has occurred. Each of the proofs in this thesis can be thought of as a communication protocol for an interactive proof of learning by the Learner.

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

Before discussing approach (2) we will make a few comments about approach (1) to suggest the variety of theoretical difficulties which occur when using approach (1).

3.2. **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 [4]. In general, only very weak results are known about bounds on the variance of cross validation for general classifiers. The "general" results include "Sanity check bounds" [9] which state that cross validation is not much worse than a holdout set and some slightly stronger results [2] and [8].

Problem 3.5. (Open) Construct a bound on the deviation of cross validation for arbitrary classifiers which is a quantitative improvement on the results of [2].

3.3. **The basic training set bound.** The most basic of training set based sample complexity bounds is the simple combination of a Binomial tail bound and the union bound. In particular, we have:

Theorem 3.6. *(Discrete Hypothesis Bound) For all hypothesis spaces, H, for all $\delta \in (0,1]$*

$$\Pr_{D^m}\left(\exists h \in H:\ e(h) \geq \bar{e}\left(m, \hat{e}(h), \frac{\delta}{|H|}\right)\right) \leq \delta$$

where $\bar{e}\left(m, \frac{k}{m}, \delta\right) \equiv \max_p\{p:\ Bin(m,k,p) = \delta\}$

Note that this theorem can only be nonvacuous when the hypothesis space, H, has some finite (discrete) size.

Proof. For every individual hypothesis, we know that:

$$\forall h\ \Pr_{D^m}\left(e(h) \geq \bar{e}\left(m, \hat{e}(h), \frac{\delta}{|H|}\right)\right) \leq \frac{\delta}{|H|}$$

Applying the union bound (see section 2.5) for every hypothesis gives us:

$$\Pr_{D^m}\left(\exists h \in H:\ e(h) \geq \bar{e}\left(m, \hat{e}(h), \frac{\delta}{|H|}\right)\right) \leq \delta$$

which is the result. □

Intuitively, this theorem says that as the number of hypotheses grows, we can not guarantee that the empirical error will be near to the true error.

A better understanding can be gained by considering some of the Binomial tail bound approximations. This is also worth mentioning in order to compare this theorem with theorems in more common forms.

Corollary 3.7. *(Relative Entropy Discrete Hypothesis Bound) For all hypothesis spaces, H, for all $\delta \in (0,1]$:*

$$\Pr_{D^m}\left(\exists h \in H:\ KL(\hat{e}(h)||e(h)) \geq \frac{\ln|H| + \ln\frac{1}{\delta}}{m}\right) \leq \delta$$

Proof. Loosen (theorem 3.6) with the relative entropy Chernoff bound (Eqn. 2.1), and use the inversion lemma 2.3. □

The form of this corollary allows us to make two more important observations:

(1) The "cost" of doubling the hypothesis space size is about one extra example. In other words, we can keep a constant bound on the probability of a large deviation with $c\log|H|$ hypotheses (for any c).
(2) The value of δ is not very important as m grows larger.

We can further loosen the last corollary with the Hoeffding approximation to get the following commonly stated bound:

Corollary 3.8. *(Agnostic Discrete Hypothesis Bound) For all hypothesis spaces, H, for all $\delta > 0$,*

$$\Pr_{D^m}\left(\exists h \in H : \, e(h) \geq \hat{e}(h) + \sqrt{\frac{\ln |H| + \ln \frac{1}{\delta}}{2m}}\right) \leq \delta$$

Proof. Loosen corollary (3.7) with the bound $\mathrm{KL}(q||p) \leq 2(q-p)^2$. □

The agnostic form of this bound has the advantage that it explicitly shows that we are forcing the convergence (in high probability) of the empirical error rate to the true error rate. Graphically, we are forcing every empirical error to be near to its true error. This is a picture which represents the connection

$$\hat{e}(h)\text{———-}e(h)$$

Later bounds will have more complicated convergence conditions with more complicated graphs. This is *necessary* because the simple holdout bound does not work in practice.

3.4. **Lower Bounds.** It is important to study lower bounds in order to discover how much room for improvement exists in our upper bounds.

There are two forms of lower bounds: a lower bounds on the true error rate and lower *upper* bounds which lower bound the value our upper bound could return (given available information). For lower bounds essentially the upper bound theorem applies with the bound reversed. This form of lower bound allow us to construct a two-sided confidence interval on the location of the true error rate. Typically, such lower bounds can be formed by simply changing the sign in upper bound arguments.

Theorem 3.9. *(Discrete Hypothesis Lower Bound) For all hypothesis spaces, H, for all $\delta \in (0, 1]$*

$$\Pr_{D^m}\left(\exists h \in H : \; e(h) \leq \bar{e}\left(m, \hat{e}(h), \frac{\delta}{|H|}\right)\right) \leq \delta$$

where $\bar{e}\left(m, \frac{k}{m}, \delta\right) \equiv \min_p\{p : \; Bin(m,k,p) = \delta\}$

Proof. For every individual hypothesis, we know that:

$$\Pr_{D^m}(e(h) \leq \bar{e}(m, \hat{e}(h), \delta)) \leq \delta$$

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

3.5. **Lower Upper Bounds.** The second form of lower bound is a lower bound on the upper bound (similar to the results of [5]). If we can lower bound the upper bound (as a function of its observables), then we can be confident that the upper bound is no looser than the gap between the lower upper bound and the upper bound.

How tight is the discrete hypothesis bound (3.6)? The answer is *sometimes* tight. In particular, we can exhibit a set of learning problem where the discrete hypotheses bound can not be made significantly lower as a function of the observables, m, δ, $|H|$, and $\hat{e}(h)$. Fix the value of these quantities and then we will construct a learning problem for which a lower upper bound can not be stated.

Our learning problem will be defined over the input space $X = \{0,1\}^{|H|}$. The hypotheses will be $h_i(x) = x_i$ where x_i is the ith component of the vector x. This construction allows us to vary the true error rate of each hypothesis independent of the other hypotheses. In fact, we can pick any true error rate for any hypothesis by simply adjusting the probability that $x_i = y$. Our learning problem can therefore generate problems according to the following algorithm:

Algorithm 3.10. *Draw_Sample(float e)*

(1) Pick $y \in \{0,1\}$ uniformly
(2) For i, pick $x_i \neq y$ with probability e.

By construction, the true error rate of each hypothesis will be $e(h_i)$. Now, we can prove the following theorem:

Theorem 3.11. *(Discrete Hypothesis lower upper bound) For all true error rates* $e(h) > \frac{\ln|H| + \ln\frac{1}{\delta}}{m}$ *there exists a learning problem and algorithm such that:*

$$\Pr_{D^m}\left(\exists h \in H:\ e(h) \geq \bar{e}\left(m, \hat{e}(h), \frac{\delta}{|H|}\right)\right) \geq 1 - e^{-\delta}$$

where $\bar{e}\left(m, \frac{k}{m}, \delta\right) \equiv \max_p\{p:\ Bin(m,k,p) = \delta\}$

Intuitively, this theorem implies that we can not improve significantly on the results of theorem 3.6 without using extra information about our learning problem. Some of our later results do exactly this - they use extra information.

Proof. Using the family of learning problems implicitly defined by algorithm 3.10, we know that

$$\forall h, \forall \delta \in [0,1]:\ \Pr_{D^m}\left(e(h) \geq \bar{e}\left(m, \hat{e}(h), \frac{\delta}{|H|}\right)\right) \geq \frac{\delta}{|H|}$$

(negation)

$$\Rightarrow \forall h, \forall \delta \in [0,1]:\ \Pr_{D^m}\left(e(h) < \bar{e}\left(m, \hat{e}(h), \frac{\delta}{|H|}\right)\right) < 1 - \frac{\delta}{|H|}$$

(independence)

$$\Rightarrow \forall \delta \in [0,1]:\ \Pr_{D^m}\left(\forall h\ e(h) < \bar{e}\left(m, \hat{e}(h), \frac{\delta}{|H|}\right)\right) < \left(1 - \frac{\delta}{|H|}\right)^{|H|}$$

(negation)

$$\Rightarrow \forall \delta \in [0,1]:\ \Pr_{D^m}\left(\exists h\ e(h) \geq \bar{e}\left(m, \hat{e}(h), \frac{\delta}{|H|}\right)\right) \geq 1 - \left(1 - \frac{\delta}{|H|}\right)^{|H|}$$

(approximation)

$$\Rightarrow \forall \delta \in [0,1]:\ \Pr_{D^m}\left(\exists h\ e(h) \geq \bar{e}\left(m, \hat{e}(h), \frac{\delta}{|H|}\right)\right) \geq 1 - e^{-\delta}$$

□

For small δ, we get that $1 - e^{-\delta} \simeq \delta$ which implies that, no significantly better true error bound can be stated for all learning algorithms. In particular, the empirical risk minimization algorithm (which chooses the hypothesis with minimal empirical error) will have a significant probability of a large deviation between the empirical and true error.

3.6. **Structural Risk Minimization.** Structural Risk Minimization [13] (SRM) is a technique used in the learning theory community to avoid the difficulties associated with convergence on hypothesis sets that are too "large". SRM works with a sequence of nested hypothesis sets, $H_1 \subset H_2 \subset .... \subset H_l$. For each hypothesis set, a discrete hypothesis bound (3.6) on the difference between empirical and true error exists. For "small" hypothesis sets, this bound may be tight while for large hypothesis sets it may be inherently loose. However, we also expect that the best hypothesis in the hypothesis set improves as the hypothesis set becomes larger. This naturally induces a trade-off: there will be some hypothesis set H_i for which the true error bound is minimized.

We can't simply apply the discrete hypothesis bound to the meta-algorithm which picks the algorithm (and associated hypothesis space) with the smallest true error bound since this meta-algorithm could, potentially, output any hypothesis in H_l. The simplest way to retrofit the bound to include all hypothesis sets is with a simple theorem which essentially states that we can guarantee *nearly* the same bound as would apply on the smallest hypothesis space H_i containing the output hypothesis, h.

Theorem 3.12. *(Structural Risk Minimization) Let $p(i)$ be some measure across the l hypothesis sets with $\sum_{i=1}^{l} p(i) = 1$. Then:*

$$\forall p(i) : \Pr_{D^m}\left(\exists h \in H_i \in \{H_1, ..., H_l\} : e(h) \leq \bar{e}\left(m, \hat{e}(h), \frac{p(i)\delta}{|H_i|}\right)\right) \leq \delta$$

where $\bar{e}\left(m, \frac{k}{m}, \delta\right) \equiv \min_p\{p : \mathit{Bin}(m, k, p) = \delta\}$.

Proof. Apply the union bound to the discrete hypothesis bound (3.6). □

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 (also mentioned in [10]).

4. Incorporating a "Prior"

In constructing the discrete hypothesis bound (3.6), it is notable that an arbitrary choice was made. We decided to give the same error allowance to every hypothesis. This is an arbitrary choice which, in practice, we will wish to make differently. The next theorem is essentially a restatement of the discrete hypothesis bound with this arbitrary choice made explicit. The same bound using the Hoeffding approximation has appeared elsewhere [1][11].

Theorem 4.1. *(Occam's Razor Bound) For all hypothesis spaces, H, for all "priors" $p(h)$ over the hypothesis space, H, for all $\delta \in (0, 1]$:*

$$\forall p(h) \Pr_{D^m}\left(\exists h \in H : e(h) \geq \bar{e}\left(m, \hat{e}(h), \delta p(h)\right)\right) \leq \delta$$

where $\bar{e}\left(m, \frac{k}{m}, \delta\right) \equiv \max_p\{p : \mathit{Bin}(m, k, p) = \delta\}$

It is very important to notice that the "prior" $p(h)$ must be selected *before* seeing the examples.

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

$$\Pr_{D^m}\left(e(h) \geq \bar{e}(m, \hat{e}(h), \delta)\right) \leq \delta$$

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. □

Once again, we can relax the Occam's Razor bound (theorem 4.1) with the relative entropy Chernoff bound (2.1) to get a somewhat more tractable expression.

Corollary 4.2. *(Relative Entropy Occam's Razor Bound) For all hypothesis spaces, H, for all "priors" $p(h)$ over the hypothesis space, H, for all $\delta \in (0, 1]$:*

$$\Pr_{D^m}\left(\exists h \in H : \; KL(\hat{e}(h)||e(h)) \geq \frac{\ln \frac{1}{p(h)} + \ln \frac{1}{\delta}}{m}\right) \leq \delta$$

Proof. approximate the Binomial tail with (2.1) and solve for the minimum. □

The Occam's razor bound is often nonvacuous for discrete learning algorithms such as decision lists and decision trees. The next chapter will discuss a particular motivated choice of the measure $p(h)$ which can lead to much tighter bounds in practice.

The application of the Occam's Razor bound is somewhat more complicated then the application of the test set bound. Pictorially, the protocol for bound application is given in figure 4.1.

References

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

[2] Avrim Blum, Adam Kalai, and John Langford 1999. Beating the Hold-out: Bounds for KFold and Progressive Cross-Validation. COLT99. http://www.cs.cmu.edu/~jcl/papers/progressive_validation/coltfinal.ps

[3] H. Chernoff, "A Measure of the asymptotic efficiency of tests of a hypothesis based on the sum of observations", Annals of Mathematical Statistics, 23:493-507, 1952.

[4] Luc Devroye, Laszlo Gyorfi, Gabor Lugosi, "A Probabilistic Theory of Pattern Recognition" Springer-Verlag New York, 1996.

[5] A. Ehrenfucht, D. Haussler, M. Kearns, and L. Valiant, "A General Lower Bound on the Number of Examples Needed for Learning", Information and Computation 82(3), pp. 247-261, 1989.

[6] David Haussler, "Mathematical perspectives on Neural networks ", Lawrence Erlbaum Associates, 1995.

[7] Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association, 58, 13-30

[8] Adam Kalai, "Probabilistic and On-line methods in Machine Learning".thesis, CMU-CS-01-132.

[9] Michael Kearns and Dana Ron, "Algorithmic Stability and Sanity-Check Bounds for Leave-One-Out Cross-Validation." Neural Computation 11(6), pages 1427-1453, 1999. Also in Proceedings of the Tenth Annual Conference on Computational Learning Theory, ACM Press, 1997, pages 152–162.

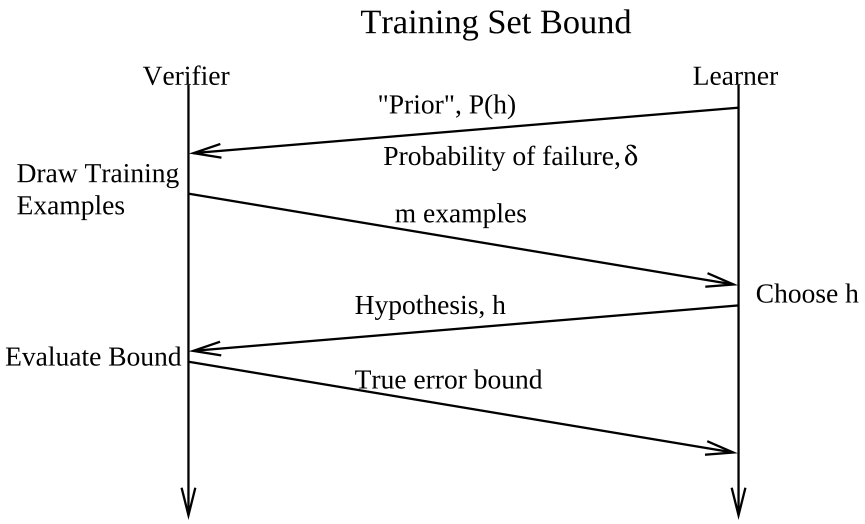

FIGURE 4.1. In order to apply this bound it is necessary that the choice of "Prior" be made before seeing any training examples. Then, the bound is calculated based upon the chosen hypothesis. Note that it *is* "legal" to chose the hypothesis based upon the prior $p(h)$ as well as the empirical error $\hat{e}_S(h)$.

[10] John Langford and Avrim Blum 1999. Microchoice Bounds and Self Bounding learning algorithms. Machine Learning Journal. http://www.cs.cmu.edu/~jcl/papers/microchoice/journal/journal_final.ps
[11] David McAllester, "PAC-Bayesian Model Averaging" COLT 1999.
[12] L.G. Valiant. "A Theory of the Learnable." Communications of the ACM 27(11):1134-1142, November 1984
[13] Vladimir N. Vapnik, "Statistical Learning Theory", Wiley, December 1999.