On the Boosting Ability of Top-Down Decision Tree Learning Algorithms

Michael Kearns
AT&T Research
mkearns@research.att.com

Yishay Mansour
Tel-Aviv University
mansour@math.tau.ac.il

Abstract

We analyze the performance of top-down algorithms for decision tree learning, such as those employed by the widely used C4.5 and CART software packages. Our main result is a proof that such algorithms are *boosting* algorithms. By this we mean that if the functions that label the internal nodes of the decision tree can weakly approximate the unknown target function, then the top-down algorithms we study will amplify this weak advantage to build a tree achieving any desired level of accuracy. The bounds we obtain for this amplification show an interesting dependence on the *splitting criterion* used by the top-down algorithm. More precisely, if the functions used to label the internal nodes have error $1/2 - \gamma$ as approximations to the target function, then for the splitting criteria used by CART and C4.5, trees of size $(1/\epsilon)^{O(1/\gamma^2\epsilon^2)}$ and $(1/\epsilon)^{O(\log(1/\epsilon)/\gamma^2)}$ (respectively) suffice to drive the error below ϵ. Thus (for example), small constant advantage over random guessing is amplified to constant error with trees of constant size. For a new splitting criterion suggested by our analysis, the much stronger bound of $(1/\epsilon)^{O(1/\gamma^2)}$ (which is polynomial in $1/\epsilon$) is obtained. The differing bounds have a natural explanation in terms of concavity properties of the splitting criterion.

The primary contribution of this work is in proving that some popular and empirically successful heuristics that are based on first principles meet the criteria of an independently motivated theoretical model.

1 Introduction

In experimental and applied machine learning work, it is difficult to exaggerate the influence of top-down heuristics for building a decision tree from labeled sample data. Such algorithms grow a tree from the root to the leaves by repeatedly replacing an existing leaf with an internal node, thus "splitting" the data reaching this leaf into two new leaves, and reducing the empirical error on the given sample. The tremendous popularity of such programs (which include the C4.5 and CART software packages [15, 3]) is due to their efficiency and simplicity, the potential advantages of using decision trees (such as interpretability for humans), and of course, to their success in generating trees with good generalization ability. Dozens of papers describing experiments and applications involving top-down decision tree learning algorithms appear in the machine learning literature each year [1].

There has been difficulty in finding natural theoretical models in which to analyze the performance of top-down heuristics, to compare variants of these heuristics, and to compare them to learning algorithms that use entirely different representations and approaches. For example, even if we make the rather favorable assumption that there is a small decision tree labeling the data, and that the inputs are distributed uniformly (thus, we are in the well-known Probably Approximately Correct or PAC model [17], with the additional restriction to the uniform distribution), the problem of finding *any* efficient algorithm with provably nontrivial performance remains an elusive open problem in computational learning theory. Furthermore, superpolynomial lower bounds for this same problem have been proven for a wide class of algorithms that includes the top-down decision tree approach (and also all variants of this approach that have been proposed to date) [2]. The positive results for efficient decision tree learning in computational learning theory all make extensive use of *membership queries* [13, 5, 4, 11], which provide the learning algorithm with black-box access to the target function (experimentation), rather than only an oracle for *random* examples. Clearly, the need for membership queries severely limits the potential application of such algorithms, and they seem unlikely to encroach on the popularity of the top-down decision tree algorithms without significant new ideas. In summary, it seem fair to say that despite their other successes, the models of computational learning theory have not yet provided significant insight into the apparent empirical success of programs like C4.5 and CART.

In this paper, we remedy this state of affairs by analyzing top-down decision tree learning algorithms in the model of *weak learning* [16, 10, 9]. In the language of weak learning, we prove here that the standard top-down decision tree algorithms are in fact *boosting* algorithms. By this we mean that if we make a favorable (and apparently necessary) assumption about the relationship between the class of functions labeling the decision tree nodes and the unknown target function — namely, that these node functions perform slightly better than random guessing as approximations to the target function — then the top-down algorithms will eventually drive the error below any desired value. In other words, the top-down algorithms will amplify slight advantages over random guessing into arbitrarily accurate hypotheses. Our results imply, for instance, that if there is always a node function whose error with respect to the target function (on an appropriate distribution) is a non-zero constant less than $1/2$ (say, 49% error), then the top-down algorithms will drive the error below any desired constant (say, 1% error) using only constant tree

size. For certain top-down algorithms, the error can be driven below ϵ using a tree whose size is polynomial in $1/\epsilon$. (See Theorem 4.1 for a precise statement.)

Thus, even though the top-down heuristics and the notion of weak learning were developed independently, we prove that these heuristics do in fact achieve nontrivial performance in the weak learning model. This is perhaps the most significant aspect of our results, and one that eluded previous theoretical studies: the proof that some popular and empirically successful algorithms that are based on first principles meet the criteria of an independently motivated theoretical model.

A prime benefit of our ability to analyze the top-down algorithms in a standard theoretical model is that it allows comparisons between variants. In particular, the choice of the *splitting criterion* function G of the top-down algorithm has a profound effect on the resulting bound. We define the splitting criterion shortly, but intuitively its role is to assign values to potential splits, and thus implicitly determines which leaf will be split, and which function will be used to label the split. Our analysis yields radically different bounds for the choices made for G by C4.5 and CART, and indicates that both may be inferior to a new choice for G suggested by our analysis [1].

In addition to providing a nontrivial analysis of the performance of top-down decision tree learning, the proof of our results gives a number of specific technical insights into the success and limitations of these heuristics. The weak learning framework also allows comparison between the top-down heuristics and the previous boosting algorithms designed especially for the weak learning model. Perhaps not surprisingly, the bounds we obtain for the top-down algorithms are worse than those obtained for other boosting algorithms. However, as we discuss in Section 3, this is unavoidable consequence of the fact that we are using decision trees, and is not due to any inherent algorithmic limitations of the top-down approach. In Section 3 we also briefly discuss why the performance of the top-down algorithms may be better in practice than our bounds indicate.

2 Top-Down Decision Tree Learning Algorithms

We study a class of learning algorithms that is parameterized by two quantities: the *node function class* F and the *splitting criterion* G. Here F is a class of boolean functions with input domain X, and $G : [0,1] \to [0,1]$ is a function having three properties:

- G is symmetric about $1/2$. Thus, $G(x) = G(1-x)$ for any $x \in [0,1]$.
- $G(1/2) = 1$ and $G(0) = G(1) = 0$.
- G is concave.

We will call such a function G a *permissible* splitting criterion. The binary entropy $G(q) = H(q) = -q\log(q) - (1-q)\log(1-q)$ is a typical example of a permissible splitting criterion. We will soon describe the roles of F and G precisely, but intuitively the learning algorithms will build decision trees in which the internal nodes are labeled by functions in F, and the splitting criterion G will be used by the learning algorithm to determine which leaf should be split next, and which function $h \in F$ to use for the split.

Throughout the paper, we assume a fixed *target function* f (*not* assumed to lie in F) with domain X, and a fixed *target distribution* P over X. Together f and P induce a distribution on *labeled examples* $\langle x, f(x)\rangle$. We will assume that the algorithms we study can *exactly* compute some rather simple probabilities with respect to this distribution on labeled examples. If this distribution can instead only be sampled, then our algorithms will approximate the probabilities accurately using only a polynomial amount of sampling, and via standard arguments the analysis will still hold.

[1] A recent paper [7] provides some preliminary empirical evaluation for this suggestion.

Let T be any decision tree whose internal nodes are labeled by functions in F, and whose leaves are labeled by values in $\{0,1\}$. We will use $leaves(T)$ to denote the leaves of T. For each $\ell \in leaves(T)$, we define $w(\ell)$ (the *weight* of ℓ) to be the probability that a random x (drawn according to P) reaches leaf ℓ in T, and we define $q(\ell)$ to be the probability that $f(x) = 1$ *given* that x reaches ℓ. Let us assume that each leaf ℓ in T is labeled 0 if $q(\ell) \leq 1/2$, and is labeled 1 otherwise (we refer to this labeling as the *majority* labeling). We define

$$\epsilon(T) = \sum_{\ell \in leaves(T)} w(\ell)\min(q(\ell), 1-q(\ell)). \quad (1)$$

Note that $\epsilon(T) = \Pr_P[T(x) \neq f(x)]$, so $\epsilon(T)$ is just the usual measure of the error of T with respect to f and P. Now if G is the splitting criterion, we define

$$G(T) = \sum_{\ell \in leaves(T)} w(\ell)G(q(\ell)). \quad (2)$$

It will be helpful to think of $G(q(\ell))$ as an approximation to $\min(q(\ell), 1-q(\ell))$, and thus of $G(T)$ as an approximation to $\epsilon(T)$. In particular, if G is a permissible splitting criterion, then we have $G(q) \geq \min(q, 1-q)$ for all $q \in [0,1]$, and thus $G(T) \geq \epsilon(T)$ for all T [2]. The top-down algorithms we examine make local modifications to the current tree T in an effort to reduce $G(T)$, and therefore hopefully reduce $\epsilon(T)$ as well. We will need notation to describe these local changes. Thus, if $\ell \in leaves(T)$ and $h \in F$, we use $T(\ell, h)$ to denote the tree that is the same as T, except that we now make a new internal node at ℓ and label this node by the function h. The newly created child leaves ℓ_0 and ℓ_1 (corresponding to the outcomes $h(x) = 0$ and $h(x) = 1$ at the new internal node) are labeled by their majority labels with respect to f and P.

For node function class F and splitting criterion G, we give pseudocode for the algorithm $\mathbf{TopDown}_{F,G}(t)$ in Figure 1. This algorithm takes as input an integer $t \geq 0$, and outputs a decision tree T with t internal nodes. Note that line 5 is the only place in the code where we make use of the assumption that we can compute probabilities with respect to P exactly, and as mentioned above, these probabilities are sufficiently simple that they can be replaced by a polynomial amount of sampling from an oracle for labeled examples via standard arguments (see Theorem 4.1). Two other aspects of the algorithm deserve mention here. First, although we have written the search for (ℓ_{best}, h_{best}) explicitly as an exhaustive search in the **for** loop line 4, clearly for certain F a more direct minimization might be possible, and in fact necessary in the case that F is uncountably infinite (for example, if F were the class of all thresholded linear combinations of two input variables). Second, all of our results still hold (with appropriate quantification) if we can only find a pair (ℓ, h) that *approximates* Δ_{best}. For example, if we have a heuristic method for always finding (ℓ, h) such that $\Delta = G(T) - G(T(\ell,h)) \geq \Delta_{best}/2$, then our main results hold without modification.

Our primary interest will be in the error of the tree T output by $\mathbf{TopDown}_{F,G}(t)$ as a function of t. To emphasize the dependence on t, for fixed F and G we use ϵ_t to denote $\epsilon(T)$ and G_t to denote $G(T)$. We think of G_t as a "potential function" that (hopefully) decreases with each new split.

Algorithms similar to $\mathbf{TopDown}_{F,G}(t)$ are in widespread use in both applications and the experimental machine learning community [1, 15, 3]. There are of course many important issues of implementation that we have omitted that must be addressed in practice. Foremost among these is the fact that in applications, one does

[2] Note that if we choose $G(q) = 2\min(q, 1-q)$ we have a permissible splitting criterion (we have multiplied by 2 to force $G(1/2) = 1$), and thus $G(T) = 2\epsilon(T)$. It will be clear shortly that this choice of G corresponds to always making the split that most reduces the error, and is *not* a good choice.

$\mathbf{TopDown}_{F,G}(t)$:

1 Initialize T to be the single-leaf tree, with binary label equal to the majority label.
2 **while** T has fewer than t internal nodes:
3 $\quad\Delta_{best} \leftarrow 0$.
4 $\quad$**for** each pair $(\ell, h) \in leaves(T) \times F$:
5 $\quad\quad\Delta \leftarrow G(T) - G(T(\ell, h))$.
6 $\quad\quad$**if** $\Delta \geq \Delta_{best}$ **then** :
7 $\quad\quad\quad\Delta_{best} \leftarrow \Delta; \ell_{best} \leftarrow \ell; h_{best} \leftarrow h$.
8 $\quad T \leftarrow T(\ell_{best}, h_{best})$.
9 Output T.

Figure 1: Algorithm $\mathbf{TopDown}_{F,G}(t)$.

not usually have an oracle for the exact probabilities, or even an oracle for random examples, but only a random sample S of fixed (often small) size. To address this, it is common practice to use the empirical distribution on S to first grow the tree in the top-down fashion of Figure 1 until ϵ_t is actually zero (which may require t to be on the order of $|S|$), and then use the splitting criterion G again to *prune* the tree. This is done in order to avoid the phenomenon known as *overfitting*, in which the error on the sample and the error on the distribution diverge [15, 3]. Despite such issues, our idealization $\mathbf{TopDown}_{F,G}(t)$ captures the basic algorithmic ideas behind many widely used decision tree algorithms. For example, it is fair to think of the popular C4.5 software package as a variant of $\mathbf{TopDown}_{F,G}(t)$ in which $G(q)$ is the binary entropy function, and F is the class of single variables (projection functions) [15]. Similarly, the CART program uses the splitting criterion $G(q) = 4q(1-q)$, known as the *Gini* criterion [3]. We feel that the analysis given here provides some insight into why such top-down decision tree algorithms succeed, and what their limitations are.

Let us now discuss the choice of the node function class F in more detail. Intuitively, the more powerful the class of node functions, the more rapidly we might expect G_t and ϵ_t to decay with t. This is simply because more powerful node functions may allow us to represent the same function more succinctly as a decision tree. Of course, the more powerful F is, the more computationally expensive $\mathbf{TopDown}_{F,G}(t)$ becomes: even if the choice of F is such that we can replace the naive **for** loop of line 4 by a direct computation of (ℓ_{best}, h_{best}), we still expect this minimization to require more time as F becomes more powerful. Thus there is a trade-off between the expressive power of F, and the running time of $\mathbf{TopDown}_{F,G}(t)$.

In the software packages C4.5 and CART, the default is to err on the side of simplicity: typically just the projections of the input variables (or slightly more powerful classes) are used as the node functions. The implicit attitude taken is that in practical applications, expensive searches over powerful classes F are simply not feasible, so we ensure the computational efficiency of computing (ℓ_{best}, h_{best}) (or at least a good approximation), and hope that a simple F is sufficiently powerful to significantly reduce G_t with each new internal node added. In our analysis, the class F is a parameter. Our approach may be paraphrased as follows: *assuming* that we have made a "favorable" choice of F, what can we say about the rate of decay of G_t and ϵ_t as a function of t? Of course, for this approach to be interesting, we need a reasonable definition of what it means to have made a "favorable" choice of F, and it is clear that this definition must say something about the relationship between F and the target function f. In the next section, we adopt the *Weak Hypothesis Assumption* (motivated by and closely related to the model of Weak Learning [12, 16, 10]) to quantify this relationship.

We defer detailed discussion of the choice of the permissible splitting criterion G, since one of the main results of our analysis is a rather precise reason why some choices may be vastly preferable to others. Here it suffices to simply emphasize that different choices of G can in fact result in different trees being built: thus, both ℓ_{best} and h_{best} may depend strongly on G. The best choice of G in practice is far being from a settled issue, as evidenced by the fact that the two most popular decision tree learning packages (C4.5 and CART) use different choices for G. There have also been a number of experimental papers examining various choices [14, 6]. Perhaps the insights in this paper most relevant to the practice of machine learning are those regarding the behavior of $\mathbf{TopDown}_{F,G}$ as a function of G; these insights are investigated experimentally in a recent paper [7].

3 The Weak Hypothesis Assumption

We now quantify what we mean by a "favorable" choice of the node splitting class F. The definition we adopt is essentially the one used by a number of previous papers on the topic of weak learning [16, 9, 10].

Definition 1 *Let f be any boolean function over an input space X. Let F be any class of boolean functions over X. Let $\gamma \in (0, 1/2]$. We say that f* **γ-satisfies the Weak Hypothesis Assumption with respect to** *F if for any distribution P over X, there exists a function $h \in F$ such that $\Pr_P[h(x) \neq f(x)] \leq 1/2 - \gamma$. If $\mathcal{P}$ is a class of distributions, we say that f γ-satisfies the Weak Hypothesis Assumption with respect to F* **over** *$\mathcal{P}$ if the statement holds for any $P \in \mathcal{P}$. We call the parameter γ the* **advantage**.

Note that if F actually contains the function f, then f trivially $1/2$-satisfies the Weak Hypothesis Assumption with respect to F. If F does not contain f, then the Weak Hypothesis Assumption amounts to an assumption on f. More precisely, it is known that Weak Hypothesis Assumption is equivalent to the assumption that on any distribution, f can be approximated by thresholded linear combinations of functions in F [9]. Thus, the Weak Hypothesis Assumption is cosmetically similar to the PAC model, in that we make *a priori* assumptions on the target function.

However, compared to the PAC model and its variants, in the Weak Hypothesis Assumption we take a more *incremental* view of learning: rather than assuming that the target function lies in some large, fixed class, and trying to design an efficient algorithm for searching this class, we instead hope that "simple" functions (in our case, the decision tree node functions) are already slightly correlated with the target function, and analyze an algorithm's ability to amplify these slight correlations to achieve arbitrary accuracy (invariably by exploiting the fact that the slight advantage holds for a wide class of distributions). Based on our results here and our remarks in the introduction regarding the known results for decision tree learning in the PAC model, it seems that the weak learning approach is perhaps more enlightening for the decision tree heuristics we analyze.

Under the Weak Hypothesis Assumption, several previous papers have proposed *boosting* algorithms that combine many different functions from F, each with a small predictive advantage over random guessing on a different *filtered distribution*, to obtain a single hybrid function whose generalization error on the target distribution P is less than any desired value ϵ [16, 9, 10]. The central question examined for such algorithms is: As a function of the advantage γ and the desired error ϵ, how many functions must be combined, and how many random examples drawn? Several boosting algorithms enjoy very strong upper bounds on these quantities [16, 9, 10]: the number of functions that must be combined is polynomial in $1/\gamma$ and $\log(1/\epsilon)$, and the number of examples required is polynomial in $1/\gamma$ and $1/\epsilon$.

The goal of this paper is to give a similar analysis for top-down decision tree algorithms: as a function of γ and ϵ, how many nodes

must the constructed tree have (that is, how large must t be) before the error ϵ_t is driven below ϵ? However, we can immediately dismiss the idea that *any* decision tree learning algorithm (top-down or otherwise) can achieve performance comparable to that achieved by algorithms specifically designed for the weak learning model. To see this, suppose the input space X is $\{0,1\}^n$, and that f is the majority function. Then if F is the class of single-variable projection functions, it is known that f $1/n$-satisfies the Weak Hypothesis Assumption with respect to F. However, it is not hard to show that the smallest decision tree with generalization error less than $1/4$ with respect to f on the uniform input distribution has size exponential in n. >From this example we can immediately infer that t must be exponential in $1/\gamma$ in order for $\mathbf{TopDown}_{F,G}(t)$ to achieve any nontrivial performance. Note, however, that this limitation is entirely *representational* — that is, it arises solely from the fact that we have chosen to represent our hypothesis as a decision tree over F, not from any properties of the algorithm we use to construct the tree. We will show that for appropriately chosen G, algorithm $\mathbf{TopDown}_{F,G}(t)$ in fact achieves tree size bounds that are exponential in $1/\gamma$ but polynomial in $1/\epsilon$ [3]. In contrast, the recent boosting methods specifically designed for the weak learning model [9, 10] represent their hypothesis as a thresholded linear combination of functions from F, a choice that is obviously well-suited to the Weak Hypothesis Assumption in light of the remarks above.

It is worth noting here that despite the fact that existing boosting algorithms enjoy significantly better bounds than those given here for top-down heuristics, the algorithms are actually fairly comparable in practice [8, 7]. In a recent experimental paper investigating some of the theoretical issues raised here, we argue that this disparity between theory and experiment can largely be explained by regarding the advantage γ as an *algorithm-dependent* quantity that varies according to the relative "difficulty" of the filtered distributions generated by the algorithm in question [7].

It is reasonable to ask whether some assumption other than the Weak Hypothesis Assumption might permit even more favorable analyses of top-down decision tree algorithms. In this regard, we briefly note that, like other boosting analyses, for our analysis a significant weakening of the Weak Hypothesis Assumption in fact suffices. Namely, our results hold if f satisfies the Weak Hypothesis Assumption over $\mathcal{P}$, where $\mathcal{P}$ contains the target distribution P and all those filterings of P that are constructed by the algorithm (this notion will be made more precise in the analysis). Furthermore, it can be shown that this weaker assumption is in fact *implied* by the rapid decay of ϵ_t. In other words, this weakened assumption holds if and only if $\mathbf{TopDown}_{F,G}(t)$ performs well. Thus the Weak Hypothesis Assumption seems to be the correct framework in which to analyze algorithm $\mathbf{TopDown}_{F,G}(t)$.

4 Statement of the Main Result

Our main result gives upper bounds on the error ϵ_t of $\mathbf{TopDown}_{F,G}(t)$ for three different choices of the splitting criterion G, under the assumption that the target function γ-satisfies the Weak Hypothesis Assumption for F. Equivalently, we give upper bounds on the value of t required to satisfy $\epsilon_t \leq \epsilon$ for any given ϵ.

Two of our choices for G are motivated by the popular implementations of $\mathbf{TopDown}_{F,G}(t)$ previously discussed. The Gini criterion used by the CART program is $G(q) = 4q(1-q)$, and the entropy criterion used by the C4.5 software package is $G(q) = H(q) = -q\log(q) - (1-q)\log(1-q)$. It is easily verified that both are permissible. The third criterion we examine is $G(q) = 2\sqrt{q(1-q)}$. This new choice is motivated by our analysis: it is the splitting criterion for which we obtain the best bounds. A plot of these three functions is given in Figure 2.

Theorem 4.1 *Let F be any class of boolean functions, let $\gamma \in (0, 1/2]$, and let f be any target boolean function that γ-satisfies the Weak Hypothesis Assumption with respect to F. Let P be any target distribution, and let T be the tree output by* $\mathbf{TopDown}_{F,G}(t)$. *Then for any ϵ, the error $\epsilon(T)$ is less than ϵ provided that*

$$t \geq \left(\frac{1}{\epsilon}\right)^{c/(\gamma^2\epsilon^2\log(1/\epsilon))} \quad \text{if} \quad G(q) = 2q(1-q), \tag{3}$$

for some constant c; or provided that

$$t \geq \left(\frac{1}{\epsilon}\right)^{c\log(1/\epsilon)/\gamma^2} \quad \text{if} \quad G(q) = H(q), \tag{4}$$

for some constant c; or provided that

$$t \geq \left(\frac{1}{\epsilon}\right)^{c/\gamma^2} \quad \text{if} \quad G(q) = 4\sqrt{q(1-q)}, \tag{5}$$

for some constant c. Furthermore, suppose that instead of the ability to compute probabilities exactly with respect to P, we are only given an oracle for random examples $\langle x, f(x)\rangle$. Consider taking a random sample S of m examples, and running algorithm $\mathbf{TopDown}_{F,G}(t)$ *using the empirical distribution on S. There exists a polynomial, $p(\log(1/\delta), 1/\epsilon, t)$, such that if m is larger than $p(\log(1/\delta), 1/\epsilon, t)$ and t satisfies the above conditions, then with probability greater than $1-\delta$, the error $\epsilon(T)$ of T with respect to f and P will be smaller than ϵ* [4].

In terms of the dependence on the advantage parameter γ, all three bounds are exponential in $1/\gamma$. As we have discussed, the majority example shows that this dependence is an unavoidable consequence of using decision trees. In terms of dependence on ϵ, however, there are vast differences between the bounds for different choices of G. For the Gini criterion, the bound is exponential in $1/\epsilon$, and for the entropy criterion of C4.5 it is superpolynomial but subexponential. Only for the new criterion $G(q) = 2\sqrt{q(1-q)}$ do we obtain a bound polynomial in $1/\epsilon$. Although at this writing we do not have matching lower bounds for the three criteria, the analysis will show that there are good technical reasons for believing that there are real differences between the criteria.

The remainder of the paper is devoted to the proof of Theorem 4.1. We emphasize and discuss the parts of the analysis that shed light on important issues such as the reason the Weak Hypothesis Assumption is helpful for $\mathbf{TopDown}_{F,G}(t)$, and the differences between the various choices for G.

5 Proof of the Main Result

Our approach is to bound G_t as a function of t and γ, and therefore bound ϵ_t as well. The main idea is to show that G_t decreases by at least a certain amount with each new split. This magnitude of this decrease depends critically on the concavity properties of G.

Let us fix a leaf ℓ of T, and use the shorthand $w = w(\ell)$ and $q = q(\ell)$ (recall that $w(\ell)$ is the probability of reaching ℓ, and $q(\ell)$ is the probability that the target function is 1 given that we have reached ℓ). Now suppose we split the leaf ℓ by replacing it by a new internal node labeled by the test function $h \in F$ (that is, T

[3] Note that once we consider the sample size needed to achieve error ϵ, any learning algorithm requires $\Omega(1/\epsilon)$ examples, and thus $\Omega(1/\epsilon)$ computation time. Thus in terms of computation time, our bounds show that the best top-down algorithm is within a polynomial factor of optimal for fixed γ. However, our results leave open the possibility of improved bounds for the *tree size* dependence on ϵ.

[4] This statment follows from the given bounds on t by standard Occam's Razor arguments, and we will omit the proof details.

becomes $T(\ell, h)$). Then we have two new child leaves of the now-internal node ℓ, one corresponding to the test outcome $h(x) = 0$ and the other corresponding to the test outcome $h(x) = 1$. Let us refer to these leaves as ℓ_0 and ℓ_1 respectively, and let τ denote the probability that x reaches ℓ_1 given that x reaches the internal node ℓ (thus, $w(\ell_0) = (1-\tau)w$ and $w(\ell_1) = \tau w$). Let $p = q(\ell_0)$ and $r = q(\ell_1)$; then we have $q = (1-\tau)p + \tau r$. This simply says that the total probability that $f(x) = 1$ must be preserved after splitting at ℓ. See Figure 4 for a diagram summarizing the split parameters.

We are now in position to make some simple but central insights regarding how the split at ℓ reduces the quantity G_t. Note that since $q = (1-\tau)p + \tau r$ and $\tau \in [0, 1]$, one of p and r is less than or equal to q and the other is greater than or equal to q. Without loss of generality, let $p \leq q \leq r$. Now before the split, the contribution of the leaf ℓ to G_t was $w \cdot G(q)$. After the split, the contribution of the now-internal node ℓ to G_{t+1} is $w \cdot ((1-\tau)G(p) + \tau G(r))$. Thus the decrease to G_t is caused by the split is exactly $w \cdot (G(q) - (1-\tau)G(p) - \tau G(r))$. It will be convenient to analyze the *local* decrease to G_t, thus assuming $w = 1$, and reintroduce the w factor later. Hence, we are interested in *lower bounding* the quantity

$$G(q) - (1-\tau)G(p) - \tau G(r). \tag{6}$$

This quantity is simply the difference between the value of the concave function G at point q, which is a linear combination of p and r, and the value of the same linear combination of $G(p)$ and $G(r)$; see Figure 5. Define $\delta = r - p$. It is easy to see that if δ is small, or if τ is near 0 or 1, then $G(q) - (1-\tau)G(p) - \tau G(r)$ will be small. Thus to lower bound the decrease to G_t, we will first need to argue that under the Weak Hypothesis Assumption these conditions cannot occur. Towards this goal, let P_ℓ denote the distribution of inputs reaching ℓ, and let the "balanced" distribution P'_ℓ be defined by $P'_\ell(x) = P_\ell(x)/(2q)$ if $f(x) = 1$ and $P'_\ell(x) = P_\ell(x)/(2(1-q))$ if $f(x) = 0$. Thus P'_ℓ is simply P_ℓ modified to give equal weight to the positive and negative examples of f (see Figure 6). Lemma 5.1 below shows that if the function h used to split at ℓ witnesses the Weak Hypothesis Assumption for the balanced distribution at ℓ, then δ cannot be too small, and τ cannot be too near 0 or 1.

5.1 Constraints on the Parameters τ and δ from the Weak Hypothesis Assumption.

Lemma 5.1 *Let q, τ and δ be as defined above for the split at leaf ℓ. Let P_ℓ be the distribution induced by P on those inputs reaching ℓ, and let P'_ℓ be the balanced distribution at ℓ. If the function h placed at ℓ obeys $\Pr_{P'_\ell}[h(x) \neq f(x)] \leq 1/2 - \gamma$ (thus, h witnesses the Weak Hypothesis Assumption for P'_ℓ), then $\tau(1-\tau)\delta \geq \gamma q(1-q)$.*

Proof: Recall that $\tau = \Pr_{P_\ell}[h(x) = 1]$ and $r = \Pr_{P_\ell}[f(x) = 1 | h(x) = 1]$. It follows that $\Pr_{P_\ell}[f(x) = 1, h(x) = 1] = \tau r$ and therefore $\Pr_{P_\ell}[f(x) = 0, h(x) = 1] = \tau - \tau r = \tau(1-r)$. In going from P_ℓ to the balanced distribution P'_ℓ, inputs x such that $f(x) = 0$ are scaled by the factor $1/(2(1-q))$ and thus $\Pr_{P'_\ell}[f(x) = 0, h(x) = 1] = (1/2(1-q))\tau(1-r)$. Similarly, since $q = \Pr_{P_\ell}[f(x) = 1]$, we have $\Pr_{P_\ell}[f(x) = 1, h(x) = 0] = q - \tau r$, and this is scaled by the factor $1/2q$ to obtain $\Pr_{P'_\ell}[f(x) = 1, h(x) = 0] = (1/2q)(q - r\tau)$. See Figure 6. Thus, we may write an exact expression for $\Pr_{P'_\ell}[h(x) \neq f(x)]$:

$$\Pr_{P'_\ell}[h(x) \neq f(x)] = \frac{1}{2(1-q)}\tau(1-r) + \frac{1}{2q}(q - \tau r) \tag{7}$$

$$= \frac{1}{2} + \frac{\tau}{2}\left(\frac{1-r}{1-q} - \frac{r}{q}\right). \tag{8}$$

By the assumption on h, $\Pr_{P'_\ell}[h(x) \neq f(x)] \leq 1/2 - \gamma$. Thus,

$$\frac{\tau}{2}\left(\frac{r}{q} - \frac{1-r}{1-q}\right) \geq \gamma. \tag{9}$$

Substituting $r = q + (1-\tau)\delta$ yields

$$\frac{\tau(1-\tau)\delta}{2}\left(\frac{1}{q} + \frac{1}{1-q}\right) \geq \gamma. \tag{10}$$

Since $1/q + 1/(1-q) = 1/q(1-q) \leq 2/q(1-q)$, the lemma follows. □(Lemma 5.1)

It is important to note that Lemma 5.1 does *not* hold in general if we let the function h placed at ℓ be the Weak Hypothesis Assumption witness for the *unbalanced* distribution P_ℓ. The reason is that if q and $1-q$ are unbalanced (say, 0.7 and 0.3) then the Weak Hypothesis Assumption witness for P_ℓ may be the constant function $h \equiv 1$, in which case we will have $\tau = 1$, and there is no decrease to G_t. Lemma 5.1 is also the only place in the proof in which the Weak Hypothesis Assumption will be used. Thus, the "weaker" assumption mentioned above that suffices for our main result to hold is simply that the Weak Hypothesis Assumption holds over all balanced distributions P'_ℓ, where ℓ is any node in any decision tree over F. Note that this class of distributions is always a subclass of those distributions that can be obtained by taking a subset of the support of P, setting the probability of this subset to zero, and renormalizing. Thus, the resulting class of filtered distributions is in some sense simpler than those generated by other boosting algorithms [16, 9, 10].

Our goal now is to obtain for each G a lower bound on the local drop $G(q) - (1-\tau)G(p) - \tau G(r)$ to G_t under the condition $\tau(1-\tau)\delta \geq \gamma q(1-q)$ given by Lemma 5.1. We emphasize at the outset that it is of the utmost importance to obtain the best (largest) lower bounds possible, since eventually these lower bounds directly influence the exponent of the resulting bounds on t. Obtaining these lower bounds will be the most involved step of our analysis. Before taking it, however, let us look ahead slightly and observe that we expect the lower bound we obtain to depend critically on properties of the splitting criterion $G(q)$. In particular, it seems that a good choice of $G(q)$ should have large curvature for all values of q, for such a function will maximize the difference between $G(q)$ and the linear combination $(1-\tau)G(p) + \tau G(r)$ (see Figure 5). In this light, we see that the choice $G(q) = 2\min(q, 1-q)$ may be an especially *poor* choice, because even under the condition $\tau(1-\tau)\delta \geq \gamma q(1-q)$ (or other similar conditions), $G(q) = (1-\tau)G(p) + \tau G(r)$ may hold. In fact, the only situation in which the decrease to G_t will be non-zero for this choice of $G(q)$ will be when $p < 1/2$ and $r > 1/2$, a situation which is certainly not implied by the Weak Hypothesis Assumption. The reason that $2\min(q, 1-q)$ is a bad choice is that it exhibits concavity only around the point $q = 1/2$, rather than for all values of q. In Figures 2 and 3, we plot the three choices for G that we will examine, so that their relative concavity properties can be compared. These concavity properties play a crucial role in the final bounds we obtain for each of the choices for G. In this regard, the important point to keep in mind during the coming minimization is: for each G, how does $G_t - G_{t+1}$ — the amount by which we reduce our "potential" with each split — compare to G_t itself, which is the potential remaining?

5.2 A Constrained Multivariate Minimization Problem.

In order to analyze the effects of the condition $\tau(1-\tau)\delta \geq \gamma q(1-q)$ on the quantity $G(q) - (1-\tau)G(p) - \tau G(r) = G_t - G_{t+1}$, it will be helpful to rewrite this quantity in terms of q, τ and δ. Thus we substitute $p = q - \tau\delta$ and $r = q + (1-\tau)\delta$ and define

$$\Delta_G(q, \tau, \delta) = G(q) - (1-\tau)G(q - \tau\delta) - \tau G(q + (1-\tau)\delta). \tag{11}$$

To obtain the desired lower bound, we must solve a constrained multivariate minimization problem: namely, we must minimize $\Delta_G(q,\tau,\delta)$ subject to the constraint $\tau(1-\tau)\delta \geq \gamma q(1-q)$. As mentioned previously, we would like to express the resulting constrained minimum of $\Delta_G(q,\tau,\delta)$ as a function of γ and q only. Towards this goal, we first fix τ and δ and lower bound $\Delta_G(q,\tau,\delta)$ by an algebraic expression of its parameters.

Lemma 5.2 *Let $G(q)$ be one of $4q(1-q)$, $H(q)$ and $2\sqrt{q(1-q)}$. Then for any fixed values for $\tau, \delta \in [0,1]$,*

$$\Delta_G(q,\tau,\delta) \geq -\frac{\tau(1-\tau)\delta^2}{2}G''(q) - \frac{\tau(1-\tau)(1-2\tau)\delta^3}{6}G'''(q). \tag{12}$$

Proof: With τ and δ fixed, we perform a Taylor expansion of $G(q)$ at q, and replace the occurrences of $G(q)$, $G(q-\tau\delta)$ and $G(q+(1-\tau)\delta)$ in $\Delta_G(q,\tau,\delta)$ by their Taylor expansions around q. It is easily verified that the terms involving zero-order and first-order derivatives of G cancel in the resulting expression for $\Delta_G(q,\tau,\delta)$. The contribution to $\Delta_G(q,\tau,\delta)$ involving second-order derivatives is

$$-\frac{1-\tau}{2}G''(q)(-\tau\delta)^2 - \frac{\tau}{2}G''(q)((1-\tau)\delta)^2 = -\frac{G''(q)}{2}\tau(1-\tau)\delta^2. \tag{13}$$

The contribution involving third-order derivatives is

$$\begin{aligned} &-\frac{1-\tau}{6}G'''(q)(-\tau\delta)^3 - \frac{\tau}{6}G'''(q)((1-\tau)\delta)^3 \\ &= -\frac{G'''(q)}{6}\tau(1-\tau)(1-2\tau)\delta^3. \end{aligned} \tag{14}$$

The lower bound claimed in the lemma is simply the sum of the second-order and third-order contributions. The fact that it is actually a lower bound on $\Delta_G(q,\tau,\delta)$ follows from the fact that for the three G under consideration, the derivatives are either all negative (and thus all make positive contributions to the bound, allowing us to take any prefix of the Taylor expansion to obtain a lower bound), or have alternating signs with negative even-order derivatives (and thus even-order derivatives make a positive contribution to the bound and odd-order derivatives make a negative contribution). In the latter case, the terms of the Taylor expansion can be grouped in adjacent pairs, with each pair giving positive contribution. Thus by summing through an odd-order derivative we obtain a lower bound. □(Lemma 5.2)

Lemma 5.2 provides us with an algebraic expression lower bounding $\Delta_G(q,\tau,\delta)$. We now wish to show that at the constrained minimum of $\Delta_G(q,\tau,\delta)$, this expression can be further simplified to obtain a lower bound involving only q and γ. Thus, we wish to use the constraint of Lemma 5.1 to eliminate the parameters δ and τ.

We begin by substituting for δ under the given constraint $\tau(1-\tau)\delta \geq \gamma q(1-q)$. It is clear from Equation 11 and Figure 5 that for *any* fixed values $q, \tau \in [0,1]$, $\Delta_G(q,\tau,\delta)$ is minimized by choosing δ as small as possible. Thus let us set $\delta = \gamma q(1-q)/(\tau(1-\tau))$ and define

$$\Delta_G(q,\tau) = \Delta_G\left(q,\tau,\frac{\gamma q(1-q)}{\tau(1-\tau)}\right) \tag{15}$$

$$= G(q) - (1-\tau)G\left(q - \frac{\gamma q(1-q)}{1-\tau}\right) - \tau G\left(q + \frac{\gamma q(1-q)}{\tau}\right). \tag{16}$$

Our next lemma shows that for the three choices of $G(q)$ under consideration, for all values of q the minimizing value of τ can be bounded away from 0 and 1. This will allow us to replace occurrences of τ with constant values.

Lemma 5.3 *Let $G(q)$ be one of $4q(1-q)$, $H(q)$ and $2\sqrt{q(1-q)}$. Let $\gamma \in [0,1]$ be any fixed value, and let $\Delta_G(q,\tau)$ be as defined in Equation 16. Then for any fixed $q \in [0,1]$, $\Delta_G(q,\tau)$ is minimized by a value of τ falling in the interval $[0.4, 0.6]$.*

Proof: We differentiate $\Delta_G(q,\tau)$ with respect to τ:

$$\begin{aligned} \frac{\partial\Delta_G(q,\tau)}{\partial\tau} &= G(q-\tau\delta) + \tau\delta\cdot G'(q-\tau\delta) \\ &\quad -[G(q+(1-\tau)\delta) \\ &\quad -(1-\tau)\delta\cdot G'(q+(1-\tau)\delta)] \end{aligned} \tag{17}$$

where we recall $\delta = \gamma q(1-q)/(\tau(1-\tau))$. This last expression for $\partial\Delta_G(q,\tau)/\partial\tau$ has a natural and helpful geometric interpretation: it is the difference between two tangent lines to the function G, both evaluated at the point q. (See Figure 7.) The term $G(q-\tau\delta) + \tau\delta\cdot G'(q-\tau\delta)$ computes the line tangent to G at the point $q-\tau\delta$, and evaluates this line at the point q, while the term $G(q+(1-\tau)\delta) - (1-\tau)\delta\cdot G'(q+(1-\tau)\delta)$ computes the line tangent to G at the point $q+(1-\tau)\delta$, and again evaluates this line at q. Thus, for the G that we are considering, for any fixed values of q and γ the minimizing values of τ satisfy $G(q-\tau\delta) + \tau\delta\cdot G'(q-\tau\delta) = G(q+(1-\tau)\delta) - (1-\tau)\delta\cdot G'(q+(1-\tau)\delta)$ (and therefore $\partial\Delta_G(q,\tau)/\partial\tau = 0$). We would like to show that for some particular choices for G, any minimizing value of τ is always (for all $q, \in [0,1]$, $\gamma \in (0,1/2]$, and $\delta = \gamma q(1-q)/(\tau(1-\tau))$ bounded away from 0 and 1. For the choice $G(q) = 2\sqrt{q(1-q)}$ and $\gamma = 0.1$, it can be seen in Figures 8 and 9 that for all $q, \in [0,1]$, if $\tau \leq 0.4$ then $\partial\Delta_G(q,\tau)/\partial\tau \leq 0$, and if $\tau \geq 0.6$ then $\partial\Delta_G(q,\tau)/\partial\tau \geq 0$. (Formal proof for this choice of G is given in Lemma 5.8 of the Appendix, and for the other choices for G in the full version of the paper.) Thus for our three choices for G, all the minimizing values of τ lie in the interval [0.4,0.6] for all values of q and γ, as desired. □(Lemma 5.3)

5.3 Effects of the Choice of G on the Drop to G_t.

Let us review where we are: in Lemma 5.2 we have given a lower bound on $\Delta_G(q,\tau,\delta)$ that involves all three parameters and derivatives of $G(\cdot)$, and in Lemma 5.3 we have shown that under the constraint $\delta = \gamma q(1-q)/(\tau(1-\tau))$, to lower bound $\Delta_G(q,\tau)$ for the $G(\cdot)$ we examine, we may assume that $\tau \in [0.4, 0.6]$. We now apply these general results to obtain lower bounds for specific choices of $G(\cdot)$.

Lemma 5.4 *Under the constraint $\tau(1-\tau)\delta \geq \gamma q(1-q)$ given by Lemma 5.1, if $G(q) = 4q(1-q)$ then for any $q \in [0,1]$*

$$\Delta_G(q,\tau,\delta) \geq 16\gamma^2(q(1-q))^2. \tag{18}$$

Proof: Here we have derivatives $G''(q) = -8$ and $G'''(q) = 0$. Application of Lemma 5.2 gives

$$\Delta_G(q,\tau,\delta) \geq 4\tau(1-\tau)\delta^2 \tag{19}$$

and the substitution $\delta = \gamma q(1-q)/(\tau(1-\tau))$ yields

$$\Delta_G(q,\tau) \geq 16\gamma^2(q(1-q))^2. \tag{20}$$

Here we have used the fact that $\tau(1-\tau) \leq 1/4$. □(Lemma 5.4)

Lemma 5.5 *Under the constraint $\tau(1-\tau)\delta \geq \gamma q(1-q)$ given by Lemma 5.1, if $G(q) = H(q)$ then for any $q \in [0,1]$*

$$\Delta_G(q,\tau,\delta) \geq \gamma^2 q(1-q). \tag{21}$$

Proof: Using the fact that for $G(q) = H(q)$ we have derivatives $G''(q) = -(1/(1-q)+1/q)$ and $G'''(q) = -(1/(1-q)^2 - 1/q^2)$, application of Lemma 5.2 yields

$$\begin{aligned}\Delta_G(q,\tau,\delta) &\geq \frac{\tau(1-\tau)\delta^2}{2}\left(\frac{1}{1-q}+\frac{1}{q}\right) \\ &\quad + \frac{\tau(1-\tau)(1-2\tau)\delta^3}{6}\left(\frac{1}{(1-q)^2}-\frac{1}{q^2}\right) \quad (22)\\ &= \frac{\tau(1-\tau)\delta^2}{2q(1-q)} \\ &\quad - \frac{\tau(1-\tau)(1-2\tau)\delta^3(1-2q)}{6(q(1-q))^2}. \quad (23)\end{aligned}$$

Substituting $\delta = \gamma q(1-q)/(\tau(1-\tau))$ yields

$$\begin{aligned}\Delta_G(q,\tau) &\geq \frac{\gamma^2 q(1-q)}{2\tau(1-\tau)} - \frac{\gamma^3 q(1-q)(1-2q)(1-2\tau)}{6(\tau(1-\tau))^2} \quad (24)\\ &\geq 2\gamma^2 q(1-q) - \frac{\gamma^2 q(1-q)(1-2\cdot 0.4)}{12(0.4)^2(0.4)^2} \quad (25)\\ &\geq \gamma^2 q(1-q) \quad (26)\end{aligned}$$

as desired. Here to lower bound the first term we have used the fact that $\tau(1-\tau) \leq 1/4$ always, and for the second term we have used $1-2q \leq 1$, $\gamma \leq 1/2$, and Lemma 5.3, which states that the minimizing value of τ lies in the interval $[0.4, 0.6]$ for all q. □(Lemma 5.5)

Lemma 5.6 *Under the constraint $\tau(1-\tau)\delta \geq \gamma q(1-q)$ given by Lemma 5.1, if $G(q) = 2\sqrt{q(1-q)}$ then for any $q \in [0,1]$*

$$\Delta_G(q,\tau,\delta) \geq \frac{\gamma^2(1-2q)^2(q(1-q))^{1/2}}{2} + 2\gamma^2(q(1-q))^{3/2}. \quad (27)$$

Proof: Here we have derivatives

$$G''(q) = \frac{-(1-2q)^2}{2(q(1-q))^{3/2}} - \frac{2}{(q(1-q))^{1/2}} \quad (28)$$

and

$$G'''(q) = \frac{3(1-2q)^3}{4(q(1-q))^{5/2}} + \frac{3(1-2q)}{(q(1-q))^{3/2}}. \quad (29)$$

Plugging these derivatives into Lemma 5.2 yields

$$\begin{aligned}\Delta_G(q,\tau,\delta) &\geq \frac{\tau(1-\tau)\delta^2}{2}\left[\frac{(1-2q)^2}{2(q(1-q))^{3/2}} + \frac{2}{(q(1-q))^{1/2}}\right] \\ &\quad - \frac{\tau(1-\tau)(1-2\tau)\delta^3}{6} \times \\ &\quad \left[\frac{3(1-2q)^3}{4(q(1-q))^{5/2}} + \frac{3(1-2q)}{(q(1-q))^{3/2}}\right]. \quad (30)\end{aligned}$$

Substituting $\delta = \gamma q(1-q)/(\tau(1-\tau))$ into this final equation and applying Lemma 5.3 can be shown to give

$$\Delta_G(q,\tau) \geq \frac{\gamma^2(1-2q)^2(q(1-q))^{1/2}}{2} + 2\gamma^2(q(1-q))^{3/2} \quad (31)$$

where we have omitted the algebraic details for brevity; however, Lemma 5.3 is used to replace occurrences of τ by constant values while preserving a lower bound. □(Lemma 5.6)

5.4 Finishing Up: Solution of Recurrences.

Let us take stock of the lower bounds on $G_t - G_{t+1}$ that we have proven, and assume that q is small for simplicity. Recall that we have analyzed the *local* drop to G_t, so to compute the global drop we must reintroduce $w = w(\ell)$. For $G(q) = 4q(1-q)$, Lemma 5.4 shows that $G_t - G_{t+1}$ is on the order of $w \cdot \gamma^2 q^2$. For $G(q) = H(q)$, Lemma 5.5 shows that $G_t - G_{t+1}$ is on the order of $w \cdot \gamma^2 q$. Ignoring the dependence on w and γ, neither of these drops is on the order of $G(q)$ itself. For $G(q) = 2\sqrt{q(1-q)}$, Lemma 5.6 shows that (provided q is bounded away from $1/2$) $G_t - G_{t+1}$ is on the order of $w \cdot \gamma\sqrt{q}$. This is the only case in which the drop is on the order of $G(q)$. We will now see the strong effect that these drops have on our final bounds, beginning with the choice $G(q) = 2\sqrt{q(1-q)}$.

Theorem 5.7 *Let $G(q) = 2\sqrt{q(1-q)}$, and let ϵ_t and G_t be as defined in Equations 1 and 2. Then under the Weak Hypothesis Assumption, for any $\epsilon \in [0,1]$, to obtain $\epsilon_t \leq \epsilon$ it suffices to make*

$$t \geq \left(\frac{1}{\epsilon}\right)^{32/\gamma^2} \quad (32)$$

splits.

Proof: By the definition of G_t, there must exist a leaf ℓ such that $2w(\ell)(q(\ell)(1-q(\ell)))^{1/2} \geq G_t/t$. By Lemma 5.6, if we split at ℓ using the witness h to the Weak Hypothesis Assumption on the balanced distribution at ℓ, then the resulting drop to G_t will be at least

$$\frac{\gamma^2(1-2q(\ell))^2 G_t}{4t} + \frac{\gamma^2 q(\ell)(1-q(\ell))G_t}{t}. \quad (33)$$

Now if $q(\ell) \in [1/4, 3/4]$ then the second term above is at least $3\gamma^2 G_t/16$, and if $q(\ell) \notin [1/4, 3/4]$ then the first term above is at least $\gamma^2 G_t/16$. Thus we obtain the recurrence inequality

$$G_{t+1} \leq G_t - \frac{\gamma^2 G_t}{16t} = \left(1 - \frac{\gamma^2}{16t}\right) G_t. \quad (34)$$

We would like to find a t such that $G_t \leq \epsilon$, since $e_t \leq G_t$ this would give us the right bound on t. Solving the above recurrence we have

$$G_t \leq e^{-\gamma^2 \log(t)/32} \quad (35)$$

as desired. □(Theorem 5.7)

We defer proofs of the bounds given in Theorem 4.1 for the choices $G(q) = 4q(1-q)$ and $G(q) = H(q)$ to the Appendix (Theorem 5.9 and Theorem 5.10), but the basic idea is the same: we apply Lemmas 5.4 and 5.5 to obtain recurrence inequalities for G_t; the stated bounds on t reflect the solutions to these recurrences. For $G(q) = H(q)$, we obtain

$$G_{t+1} \leq G_t - \frac{\gamma^2 G_t}{4t\log(2/G_t)}. \quad (36)$$

For $G(q) = 4q(1-q)$, we obtain

$$G_{t+1} \leq G_t - \frac{\gamma^2 G_t^3}{128t}. \quad (37)$$

These recurrence inequalities of Equations 34, 36 and 37 are in order of decreasing rate of decay of G_t, thus explaining the superiority of the choice $G(q) = 2\sqrt{q(1-q)}$ by our analysis.

Acknowledgements

Thanks to Tom Dietterich, Yoav Freund, Andrew Ng and Rob Schapire for discussions on the material presented here.

References

[1] In *Machine Learning: Proceedings of the International Conference*. Morgan Kaufmann, 1982–1995.

[2] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the 26th ACM Symposium on the Theory of Computing*. ACM Press, New York, NY, 1994.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.

[4] N. Bshouty and Y. Mansour. Simple learning algorithms for decision trees and multivariate polynomials. In *Proceedings of the 36th IEEE Symposium on the Foundations of Computer Science*, pages 304–311. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[5] N. H. Bshouty. Exact learning via the monotone theory. In *Proceedings of the 34th IEEE Symposium on the Foundations of Computer Science*, pages 302–311. IEEE Computer Society Press, Los Alamitos, CA, 1993.

[6] W. Buntine and T. Niblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8:75–86, 1992.

[7] T. Dietterich, M. Kearns, and Y. Mansour. Applying the weak learning framework to understand and improve C4.5. Unpublished manuscript, 1996.

[8] Y. Freund and R. Schapire. Some experiments with a new boosting algorithm. Unpublished manuscript, 1996.

[9] Yoav Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, September 1995.

[10] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Second European Conference on Computational Learning Theory*, pages 23–37. Springer-Verlag, 1995.

[11] J. Jackson. An efficient membership query algorithm for learning DNF with respect to the uniform distribution. In *Proceedings of the 35th IEEE Symposium on the Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 1994.

[12] M. Kearns and L. G. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

[13] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. In *Proc. of the 23rd Symposium on Theory of Computing*, pages 455–464. ACM Press, New York, NY, 1991.

[14] J. Mingers. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3:319–342, 1989.

[15] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[16] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[17] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Appendix

Lemma 5.8 below is needed for the proof of Lemma 5.3 for the choice $G(q) = 2\sqrt{q(1-q)}$, which shows that the minimizing value of τ for the function $\Delta_G(q,\tau)$ lies in the range $[0.4, 0.6]$. Proofs for the choices $G(q) = 4q(1-q)$ and $G(q) = H(q)$ will be given in the full paper.

Lemma 5.8 *Let $G(q) = 2\sqrt{q(1-q)}$, and let $\delta = \gamma q(1-q)/(\tau(1-\tau))$. If $\gamma < 0.2$ and q is sufficiently small, then for $\tau \leq 0.4$,*

$$\begin{aligned} & G(q-\tau\delta) + \tau\delta \cdot G'(q-\tau\delta) \\ & \quad \leq \quad G(q+(1-\tau)\delta) - (1-\tau)\delta \cdot G'(q+(1-\tau)\delta) \end{aligned} \tag{38}$$

and for $\tau \geq 0.6$,

$$\begin{aligned} & G(q-\tau\delta) + \tau\delta \cdot G'(q-\tau\delta) \\ & \quad \geq \quad G(q+(1-\tau)\delta) - (1-\tau)\delta \cdot G'(q+(1-\tau)\delta). \end{aligned} \tag{39}$$

Proof: For this choice of $G(q)$, we have $G'(q) = (1-2q)\sqrt{q(1-q)}$. To simplify notation a bit, let us define

$$F(q,\Delta) = G(q-\Delta) + \Delta G'(q-\Delta). \tag{40}$$

We will be interested in the choices $\Delta = \tau\delta$ and $\Delta = -(1-\tau)\delta$, and we will find it convenient to divide $F(q,\Delta)$ by $G(q)$. In particular, algebraic manipulations (details omitted) can be shown to yield

$$\frac{F(q,\tau\delta)}{G(q)} = \frac{1 + \frac{\gamma q}{1-\tau} - \frac{\gamma}{2(1-\tau)}}{\sqrt{\left(1 - \frac{\gamma(1-q)}{1-\tau}\right)\left(1 + \frac{\gamma q}{1-\tau}\right)}}. \tag{41}$$

Thus, we find that as $q \to 0$,

$$\frac{F(q,\tau\delta)}{G(q)} \to \frac{1 - \frac{\gamma}{2(1-\tau)}}{\sqrt{1 - \frac{\gamma}{1-\tau}}}. \tag{42}$$

Similarly, it can be shown that as $q \to 0$,

$$\frac{F(q,-(1-\tau)\delta)}{G(q)} \to \frac{1 + \frac{\gamma}{2\tau}}{\sqrt{1 + \frac{\gamma}{\tau}}}. \tag{43}$$

Now to complete the proof of the statement of the lemma for $\tau \leq 0.4$ we need to show:

$$\frac{1 - (5/6)\gamma}{\sqrt{1 - (5/3)\gamma}} < \frac{1 + (5/4)\gamma}{\sqrt{1 + (5/2)\gamma}}. \tag{44}$$

This inequality holds for $\gamma < 0.2$, as does the reverse inequality obtained for the value $\tau \geq 0.6$. □(Lemma 5.8)

Analogous to Theorem 5.7, Theorems 5.9 and 5.10 set up and solve the recurrence equations for the choices $G(q) = 4q(1-q)$ and $G(q) = H(q)$.

Theorem 5.9 *Let $G(q) = 4q(1-q)$, and let ϵ_t and G_t be as defined in Equations 1 and 2. Then under the Weak Hypothesis Assumption, for any $\epsilon \in [0,1]$, to obtain $\epsilon_t \leq \epsilon$ it suffices to make*

$$t \geq 2^{c/\gamma^2\epsilon^2} \tag{45}$$

splits, for some constant c.

Proof: At round t, there must be a leaf ℓ such that (1) $w(\ell) \geq \epsilon_t/(2t)$ and (2) $min\{q(\ell), 1-q(\ell)\} \geq \epsilon_t/2$, because the total weight of the leaves with weight less than $\epsilon_t/(2t)$ is at most $\epsilon_t/2$, the remaining leaves have, on average, an error of $\epsilon_t/2$, therefore at least one of them has error at least $\epsilon_t/2$. By Lemma 5.4, splitting at ℓ using the witness for the Weak Hypothesis Assumption on the balanced distribution at ℓ will result in a reduction to G_t of at least

$$\begin{aligned} w(\ell) \cdot 16\gamma^2(q(\ell)(1-q(\ell))^2 &\geq 4\gamma^2 w(\ell)\min(q(\ell), 1-q(\ell))^2 \\ &\geq \gamma^2\epsilon_t^3/(2t) \\ &\geq \gamma^2 G_t^3/(128t). \end{aligned} \tag{46}$$

In the last inequality we have used the fact that for $G(q) = 4q(1-q)$, $\epsilon_t \geq G_t/4$. This enables us to derive a recurrence inequality

$$G_{t+1} \leq G_t - \frac{\gamma^2 G_t^3}{128t}. \tag{47}$$

We are interested in the value of t for which $G_t \leq \epsilon$, since we are guaranteed that $\epsilon_t \leq G_t$. It is not difficult to verify that $t \geq 2^{c/(\gamma^2\epsilon^2)}$, for some constant c, suffices. □(Lemma 5.9)

Theorem 5.10 *Let $G(q) = H(q)$, and let ϵ_t and G_t be as defined in Equations 1 and 2. Then under the Weak Hypothesis Assumption, for any $\epsilon \in [0,1]$, to obtain $\epsilon_t \leq \epsilon$ it suffices to make*

$$t \geq \left(\frac{1}{\epsilon}\right)^{c\log(1/\epsilon)/\gamma^2} \tag{48}$$

splits, for some constant c.

Proof: By Equation 1, after t splits there must be a leaf ℓ such that $w(\ell)\min(q(\ell), 1-q(\ell)) \geq \epsilon_t/t$. If we now split at ℓ using the witness for the Weak Hypothesis Assumption on the balanced distribution at ℓ, then by Lemma 5.5, we know that the resulting reduction to G_t will be at least $w(\ell)\cdot\gamma^2 q(\ell)(1-q(\ell)) \geq w(\ell)\cdot\gamma^2\min(q(\ell), 1-q(\ell)) \geq \gamma^2\epsilon_t/(2t)$. In other words, we may write

$$G_{t+1} \leq G_t - \frac{\gamma^2\epsilon_t}{2t}. \tag{49}$$

Now for the choice $G(q) = H(q)$, we are interested in the value of t for which, $G_t \leq H(\epsilon_t)$ since we know that $\epsilon_t \leq G_t$ by Jensen's Inequality. Thus we are interested in t such that $H^{-1}(G_t) \leq \epsilon$, where $H^{-1}(y)$ is defined to be the unique $x \in [0, 1/2]$ such that $H(x) = y$. It can be shown that $H^{-1}(y) \geq y/(2\log(2/y))$ for all $y \in [0,1]$. Thus we have

$$G_{t+1} \leq G_t - \frac{\gamma^2 H^{-1}(G_t)}{2t} \tag{50}$$

$$\leq G_t - \frac{\gamma^2 G_t}{4t\log(2/G_t)}. \tag{51}$$

It can be verified that $G_t \leq e^{-\gamma\sqrt{\log(t)/c}}$ is a solution to this recurrence inequality, as desired. Thus to obtain $\epsilon_t \leq G_t \leq \epsilon$, it suffices to have $e^{-\gamma\sqrt{\log(t)}} \leq \epsilon$, which requires $t \geq (1/\epsilon)^{c\log(1/\epsilon)/\gamma^2}$. □(Lemma 5.10)

Figure 2: Plots of the three splitting criterion G that we examine, from top to bottom: $G(q) = 2\sqrt{q(1-q)}$, $G(q) = H(q)$ and $G(q) = 4q(1-q)$. The bottom (piecewise linear) plot is $2\min(q, 1-q)$, the splitting criterion corresponding to direct minimization of the error.

Figure 3: Same as Figure 2, but for small q. Notice that the choice $G(q) = 2\sqrt{q(1-q)}$ (top plot) enjoys strong concavity in this regime of q, while $G(q) = H(q)$ and $G(q) = 4q(1-q)$ (second and third from top) are nearly linear in this regime.

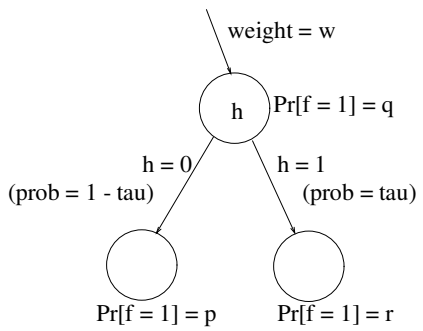

Figure 4: A typical split labeled by the function h, showing the split parameters used in the analysis.

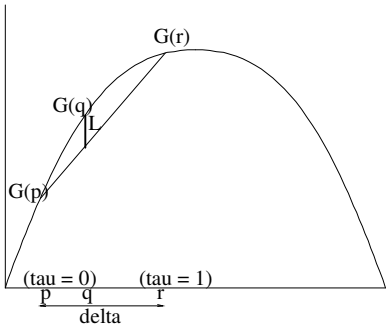

Figure 5: Effects of the concavity of G on the local drop to G_t. Here $q = (1-\tau)p + \tau r, 0 \leq \tau \leq 1$. The local drop to G_t is equal to the length of the vertical line segment L. Intuitively, for fixed p, q and r, the greater the concavity of G, the greater the drop. To show that this drop is significant, we must bound the three points away from each other — that is, lower bound $\delta = r - p$ and bound τ away from 0 and 1.

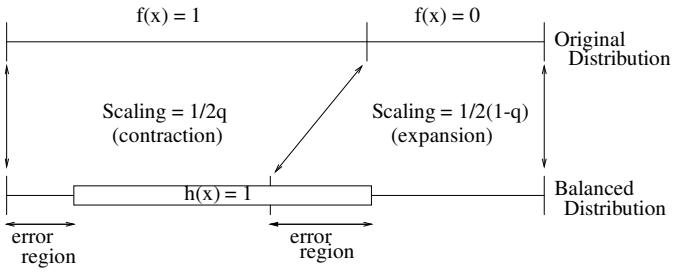

Figure 6: Illustration of the proof of Lemma 5.1. In going from the original distribution P_ℓ to the balanced distribution P'_ℓ inputs x such that $f(x) = 1$ have their probabilities "contracted" and x such that $f(x) = 0$ are "expanded" (since in this figure $q = \Pr_{P_\ell}[f(x) = 1] > 1/2$). We compute the error of the Weak Hypothesis Assumption witness h for the balanced distribution P'_ℓ by computing the weight of the error regions under P_ℓ and applying the contraction and expansion operations.

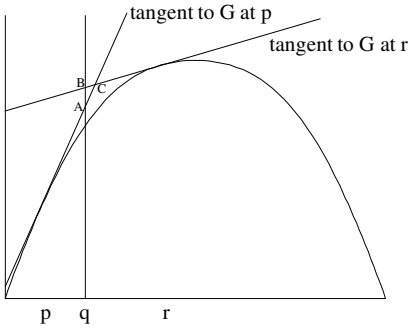

Figure 7: Illustration of the proof of Lemma 5.3. Here $q = (1-\tau)p + \tau r$, and the minimizing value of τ for the function $\Delta_G(q, \tau)$ occurs when the intersection of tangent line at $G(p)$ with the vertical line at q (point A) coincides with the intersection of the tangent line at $G(r)$ with the vertical line at q (point B). In the figure, these points do not coincide: point A still lies below point B, and thus q must be moved away from p, to lie directly below the desired tangent intersection point C. Thus, we must increase τ to reach the minimizing value.

Figure 8: For the proof of Lemma 5.3. Plot of $G(q - \tau\delta) + \tau\delta \cdot G'(q - \tau\delta) - G(q + (1-\tau)\delta) - (1-\tau)\delta \cdot G'(q + (1-\tau)\delta)$ for $G(q) = 2\sqrt{q(1-q)}$, with $\gamma = 0.1$ and $\tau = 0.4$. For this value of τ, we see that the function is negative for all values of q, indicating that the minimizing value of τ for the function $\Delta_G(q, \tau)$ is larger than 0.4. Formal proof is given in Lemma 5.8 of the Appendix.

Figure 9: For the proof of Lemma 5.3. Plot of $G(q - \tau\delta) + \tau\delta \cdot G'(q - \tau\delta) - G(q + (1-\tau)\delta) - (1-\tau)\delta \cdot G'(q + (1-\tau)\delta)$ for $G(q) = 2\sqrt{q(1-q)}$, with $\gamma = 0.1$ and $\tau = 0.6$. For this value of τ, we see that the function is positive for all values of q, indicating that the minimizing value of τ for the function $\Delta_G(q, \tau)$ is smaller than 0.6. Formal proof is given in Lemma 5.8 of the Appendix.