PAC-Bayes Tree: weighted subtrees with guarantees

Tin Nguyen[*]
MIT EECS
tdn@mit.edu

Samory Kpotufe
Princeton University ORFE
samory@princeton.edu

Abstract

We present a weighted-majority classification approach over subtrees of a fixed tree, which provably achieves excess-risk of the *same* order as the best tree-pruning. Furthermore, the computational efficiency of pruning is maintained at both training and testing time despite having to aggregate over an exponential number of subtrees. We believe this is the first subtree aggregation approach with such guarantees.

The guarantees are obtained via a simple combination of insights from PAC-Bayes theory, which we believe should be of independent interest, as it generically implies consistency for weighted-voting classifiers w.r.t. Bayes – while, in contrast, usual PAC-bayes approaches only establish consistency of *Gibbs* classifiers.

1 Introduction

Classification trees endure as popular tools in data analysis, offering both efficient prediction and interpretability – yet they remain hard to analyze in general. So far there are two main approaches with generalization guarantees: in both approaches, a large tree (possibly overfitting the data) is first obtained; one approach is then to *prune* back this tree down to a subtree[2] that generalizes better; the alternative approach is to combine all possible subtrees of the tree by *weighted* majority vote. Interestingly, while both approaches are competitive with other practical heuristics, it remains unclear whether the alternative of *weighting subtrees* enjoys the same strong generalization guarantees as pruning; in particular, no weighting scheme to date has been shown to be *statistically consistent*, let alone attain the same tight generalization rates (in terms of excess risk) as pruning approaches.

In this work, we consider a new weighting scheme based on PAC-Bayesian insights [1], that (a) is consistent and attains the same generalization rates as the best pruning of a tree, (b) is efficiently computable at both training and testing time, and (c) competes against pruning approaches on real-world data. To the best of our knowledge, this is the first practical scheme with such guarantees.

The main technical hurdle has to do with a subtle tension between goals (a) and (b) above. Namely, let T_0 denote a large tree built on n datapoints, usually a binary tree with $O(n)$ nodes; the family of subtrees T of T_0 is typically of exponential size in n [2], so a naive voting scheme that requires visiting all subtrees is impractical; on the other hand it is known that if the weights decompose favorably over the leaves of T (e.g., multiplicative over leaves) then efficient classification is possible. Unfortunately, while various such multiplicative weights have been designed for voting with subtrees [3, 4, 5], they are not known to yield statistically consistent prediction. In fact, the best known result to date [5] presents a weighting scheme which can provably achieve an *excess* risk[3] (over the Bayes classifier) of the form $o_P(1) + C \cdot \min_T \mathcal{R}(h_T)$, where $\mathcal{R}(h_T)$ denotes the misclassification rate of a classifier h_T based on subtree T. In other words, the excess risk might never go to 0 as sample size increases, which in contrast is a basic property of the pruning alternative. Furthermore, the approach

[*]The majority of the research was done when the author was an undergraduate student at Princeton University ORFE.

[2]Considering only subtrees that partition the data space.

[3]The excess risk of a classifier h over the Bayes h_B (which minimizes $\mathcal{R}(h)$ over any h) is $\mathcal{R}(h) - \mathcal{R}(h_B)$.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

of [5], based on l_1-risk minimization, does not trivially extend to multiclass classification, which is most common in practice. Our approach is designed for multiclass by default.

Statistical contribution. PAC-Bayesian theory [1, 6, 7, 8] offers useful insights into designing weighting schemes with generalization guarantees (w.r.t. a *prior* distribution P over classifiers). However, a direct application of existing results fails to yield a consistent weighted-majority scheme. This is because PAC-Bayes results are primarily concerned with so-called *Gibbs classifiers*, which in our context corresponds to predicting with a random classifier h_T drawn according to a weight-distribution Q over subtrees of T_0. Instead, we are interested in Q-weighted majority classifiers h_Q. Unfortunately the corresponding error $\mathcal{R}(h_Q)$ can be twice the risk $\mathcal{R}(Q) = \mathbb{E}_{h_T \sim Q}\mathcal{R}(h_T)$ of the corresponding Gibbs classifier: this then results (at best – see overview in Section 2.2) in an excess risk of the form $(\mathcal{R}(h_Q) - \mathcal{R}(h_B)) \leq (\mathcal{R}(Q) - \mathcal{R}(h_B)) + \mathcal{R}(Q) = o_P(1) + \mathcal{R}(h_B)$, which, similar to [5], does not go to 0. So far, this problem is best addressed in PAC-Bayes results such as the MinCq bound in [6, 8] on $\mathcal{R}(h_Q)$, which is tighter in the presence of low correlation between base classifiers. In contrast, our PAC-Bayes result applies even without low correlation between base classifiers, and allows an excess risk $o_P(1) + (C/n) \cdot \min_T \log(1/P(T)) \to 0$ (Proposition 2). This first result is in fact of general interest since it extends beyond subtrees to any family of classifiers, and is obtained by carefully combining existing arguments from PAC-Bayes analysis.

However, our basic PAC-Bayes result alone does not ensure convergence at the same rate as that of the best pruning approaches. This requires designing a prior P that scales properly with the size of subtrees T of T_0. For instance, suppose P were uniform over all subtrees of T_0, then $\log(1/(P(T)) = \Omega(n)$, yielding a vacuous excess risk. We show through information-theoretic arguments that an appropriate prior P can be designed to yield rates of convergence of the same order as that of the best pruning of T_0. In particular, our resulting weighting scheme maintains ideal properties of pruning approaches such as *adaptivity* to the intrinsic dimension of data (see e.g. [9]).

Algorithmic contribution. We show that we can design a prior P which, while meeting the above statistical constraints, yields *posterior* weights that decompose favorably over the leaves of a subtree T. As a result of this decomposition, the weights of all subtrees can be recovered by simply maintaining corresponding weights at the nodes of the original tree T_0 for efficient classification in time $O(\log n)$ (this is illustrated in Figure 1). We then propose an efficient approach to obtain weights at the nodes of T_0, consisting of concurrent top-down and bottom-up dynamic programs that run in $O(n)$ time. These match the algorithmic complexity of the most efficient pruning approaches, and thus offer a practical alternative.

Our theoretical results are then verified in experiments over many real-world datasets. In particular we show that our weighted-voting scheme achieves similar or better error than pruning on practical problems, as suggested by our theoretical results.

Paper Organization. We start in Section 2 with theoretical setup and an overview of PAC-Bayes analysis. This is followed in Section 3 with an overview of our statistical results, and in Section 4 with algorithmic results. Our experimental analysis is then presented in Section 5.

2 Preliminaries

2.1 Classification setup

We consider a multiclass setup where the input $X \subset \mathcal{X}$, for a bounded subset $\mathcal{X}$ of $\mathbb{R}^D$, possibly of lower intrinsic dimension. For simplicity of presentation we assume $\mathcal{X} \subset [0,1]^D$ (as in normalized data). The output $Y \subset [L]$, where we use the notation $[L] = \{1, 2, \ldots, L\}$ for $L \in \mathbb{N}$.

We are to learn a *classifier* $h : \mathcal{X} \mapsto [L]$, given an i.i.d. training sample $\{X_i, Y_i\}_{i=1}^{2n}$ of size $2n$, from an unknown distribution over X, Y. Throughout, we let $\mathcal{S} \doteq \{X_i, Y_i\}_{i=1}^{n}$ and $\mathcal{S}_0 \doteq \{X_i, Y_i\}_{i=n+1}^{2n}$, which will serve later to simplify dependencies in our analysis.

Our performance measure is as follows.

Definition 1. *The* **risk** *of a classifier h is given as $\mathcal{R}(h) = \mathbb{E}[h(X) \neq Y]$. This is minimized by the* Bayes *classifier $h_B(x) \doteq \operatorname{argmax}_{l \in [L]} \mathbb{P}(Y = l | X = x)$. Therefore, for any classifier $\hat{h}$ learned over a sample $\{X_i, Y_i\}_i$, we are interested in the* **excess-risk** *$\mathcal{E}(\hat{h}) \doteq \mathcal{R}(\hat{h}) - \mathcal{R}(h_B)$.*

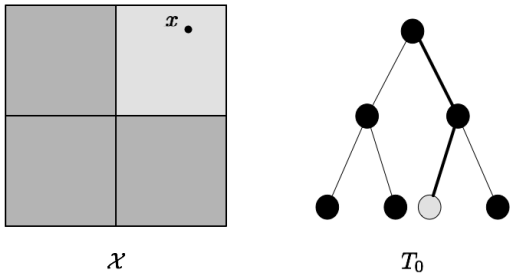


Figure 1: A partition tree T_0 over input space $\mathcal{X}$, and a query $x \in \mathcal{X}$ to classify. The leaves of T_0 are the 4 cells shown left, and the root is $\mathcal{X}$. A query x follows a single path (shown in bold) from the root down to a leaf. A key insight towards efficient weighted-voting is that this path visits all leaves (containing x) of any subtree of T_0. Therefore, weighted voting might be implemented by keeping a weight $w(A)$ at any node A along the path, where $w(A)$ aggregates the weights $Q(T)$ of every subtree T that has A as a leaf. This is feasible if we can restrict $Q(T)$ to be multiplicative over the leaves of T, without trading off accuracy.

Here we are interested in aggregations of *classification trees*, defined as follows.

Definition 2. *A* hierarchical partition *or* (space) partition-tree T *of $\mathcal{X}$ is a collection of nested partitions of $\mathcal{X}$; this is viewed as a tree where each node is a subset A of $\mathcal{X}$, each child A' of a node A is a subset of A, and whose collection of leaves, denoted $\pi(T)$, is a partition of $\mathcal{X}$. A* **classification tree** h_T *on $\mathcal{X}$ is a* labeled *partition-tree T of $\mathcal{X}$: each leaf $A \in \pi(T)$ is assigned a label $l = l(A) \in [L]$; the classification rule is simply $h_T(x) = l(A)$ for any $x \in A$.*

Given an initial tree T_0, we will consider only **subtrees** *T of T_0 that form a hierarchical partition of $\mathcal{X}$, and we henceforth use the term* subtrees *(of T_0) without additional qualification.*

Finally, aggregation (of subtrees of T_0) consists of *majority-voting* as defined below.

Definition 3. *Let $\mathcal{H}$ denote a discrete family of classifiers $h : \mathcal{X} \mapsto [L]$, and let Q denote a distribution over $\mathcal{H}$. The Q-***majority** *classifier $h_Q \doteq h_Q(\mathcal{H})$ is one satisfying for any $x \in \mathcal{X}$*

$$h_Q(x) = \operatorname*{argmax}_{l \in [L]} \sum_{h \in \mathcal{H}, h(x)=l} Q(h).$$

Our oracle rates of Theorem 1 requires no additional assumptions; however, the resulting corollary is stated under standard distributional conditions that characterize convergence rates for tree-prunings.

2.2 PAC-Bayes Overview

PAC-Bayes analysis develops tools to bound the error of a Gibbs classifier, i.e. one that randomly samples a classifier $h \sim Q$ over a family of classifiers $\mathcal{H}$. In this work we are interested in families $\{h_T\}$ defined over subtrees of an initial tree T_0. Here we present some basic PAC-Bayes result which we extend for our analysis. While these results are generally presented for classification risk $\mathcal{R}$ (defined above), we keep our presentation generic, as we show later that a different choice of risk leads to stronger results for $\mathcal{R}$ than what is possible through direct application of existing results.

Generic Setup. Consider a random vector Z, and an i.i.d sample $Z_{[n]} = \{Z_i\}_{i=1}^n$. Let $\mathcal{Z}$ be the support of Z, and $\mathcal{L} = \{\ell_h : h \in \mathcal{H}\}$ be a *loss class* indexed by $h \in \mathcal{H}$ – *discrete*, and where $\ell_h : \mathcal{Z} \to [0, 1]$. For $h \in \mathcal{H}$, the loss ℓ_h induces the following risk and empirical counterparts:

$$\mathcal{R}_{\mathcal{L}}(h) \doteq \mathbb{E}_Z \ell_h(Z), \quad \widehat{\mathcal{R}}_{\mathcal{L}}(h, Z_{[n]}) \doteq \frac{1}{n} \sum_{i=1}^n \ell_h(Z_i).$$

In particular, for the above classification risk $\mathcal{R}$, and $Z \triangleq (X, Y)$, we have $\ell_h(Z) = \mathbb{1}\{h(X) \neq Y\}$.

Given a distribution Q over $\mathcal{H}$, the risk (and empirical counterpart) of the Gibbs classifier is then

$$\mathcal{R}_{\mathcal{L}}(Q) \doteq \mathbb{E}_{h \sim Q} \mathcal{R}_{\mathcal{L}}(h), \quad \widehat{\mathcal{R}}_{\mathcal{L}}(Q, Z_{[n]}) \doteq \mathbb{E}_{h \sim Q} \widehat{\mathcal{R}}_{\mathcal{L}}(h, Z_{[n]}).$$

PAC-Bayesian results bound $\mathcal{R}_{\mathcal{L}}(Q)$ in terms of $\widehat{\mathcal{R}}_{\mathcal{L}}(Q, Z_{[n]})$, *uniformly* over any distribution Q, provided a fixed *prior* distribution P over $\mathcal{H}$. We will build on the following form of [10] which yields an upper-bound that is convex in Q (and therefore can be optimized for a good *posterior* Q^*).

Proposition 1 (PAC-Bayes on $\mathcal{R}_{\mathcal{L}}$ [10])**.** *Fix a prior P supported on $\mathcal{H}$, and let $n \geq 8$ and $\delta \in (0,1)$. With probability at least $1-\delta$ over $Z_{[n]}$, simultaneously for all $\lambda \in (0,2)$ and all posteriors Q over $\mathcal{H}$:*

$$\mathcal{R}_{\mathcal{L}}(Q) \leq \frac{\widehat{\mathcal{R}}_{\mathcal{L}}(Q, Z_{[n]})}{1-\lambda/2} + \frac{\mathcal{D}_{kl}\left(Q\|P\right) + \log\left(2\sqrt{n}/\delta\right)}{\lambda(1-\lambda/2)n},$$

where $\mathcal{D}_{kl}\left(Q\|P\right) \doteq \mathbb{E}_Q \log \frac{Q(h)}{P(h)}$ is the Kullback-Leibler divergence between Q and P.

Choice of posterior Q^*. Let Q^* minimize the above upper-bound, and let h^* minimize $\mathcal{R}_{\mathcal{L}}$ over $\mathcal{H}$. Then, by letting Q_{h^*} put all mass on h^*, we automatically get that, with probability at least $1-2\delta$:

$$\begin{aligned}\mathcal{R}_{\mathcal{L}}(Q^*) &\leq \mathcal{R}_{\mathcal{L}}(Q_{h^*}) \leq C \cdot \left(\widehat{\mathcal{R}}_{\mathcal{L}}(h^*, Z_{[n]}) + \frac{\log(1/P(h^*)) + \log(n/\delta)}{n}\right) \\ &\leq C \cdot \left(\mathcal{R}_{\mathcal{L}}(h^*) + \frac{\log(1/P(h^*)) + \log(n/\delta)}{n} + \sqrt{\frac{\log(1/\delta)}{n}}\right), \end{aligned} \tag{1}$$

where the last inequality results from bounding $|\mathcal{R}_{\mathcal{L}}(h^*) - \widehat{\mathcal{R}}_{\mathcal{L}}(h^*, Z_{[n]})|$ using Chernoff.

Unfortunately, such direct application is not enough for our purpose when $\mathcal{R}_{\mathcal{L}} = \mathcal{R}$. We want to bound the excess risk $\mathcal{E}(h_Q)$ for a Q-majority classifier h_Q over $h's \in \mathcal{H}$. It is known that $\mathcal{R}(h_Q) \leq 2\mathcal{R}(Q)$ which yields a bound of the form (1) on $\mathcal{R}(h_{Q^*})$; however this implies at best that $\mathcal{R}(h_{Q^*}) \to 2\mathcal{R}(h_B)$ even if $\mathcal{E}(h^*) \to 0$ (which is generally the case for optimal tree-pruning h_T^* [9]). This is a general problem in converting from Gibbs error to that of majority-voting, and is studied for instance in [6, 8] where it is shown that $\mathcal{R}(h_Q)$ can actually be smaller in some situations.

Improved choice of Q^*. Here, we want to design Q^* such that $\mathcal{R}(h_{Q^*}) \to \mathcal{R}(h_B)$ (i.e. $\mathcal{E}(h_{Q^*}) \to 0$) at the same rate as $\mathcal{E}(h_T^*) \to 0$ always. Our solution relies on a proper choice of loss ℓ_h that relates most directly to excess risk $\mathcal{E}$ that the 0-1 loss $\mathbb{1}\{h(x) \neq y\}$. A first candidate is to define $\ell_h(x,y)$ as $e_h(x,y) \doteq \mathbb{1}\{h(x) \neq y\} - \mathbb{1}\{h_B(x) \neq y\}$ since $\mathcal{E}(h) = \mathbb{E}\, e_h(X,Y)$; however $e_h(x,y) \notin [0,1]$ and can take negative values. This is resolved by considering an intermediate loss $e_h(x) = \mathbb{E}_{Y|x} e_h(x,Y) \in [0,1]$ to be related back to $e_h(x,y)$ by integration in a suitable order.

3 Statistical results

3.1 Basic PAC-Bayes result

We start with the following intermediate loss family over classifiers h, w.r.t. the Bayes classifier h_B.

Definition 4. *Let $e_h(x,y) \doteq \mathbb{1}\{h(x) \neq y\} - \mathbb{1}\{h_B(x) \neq y\}$, and $e_h(x) = \mathbb{E}_{Y|x} e_h(x,Y)$, and*

$$\widetilde{\mathcal{E}}(h, \mathcal{S}) \doteq \frac{1}{n}\sum_{i=1}^n e_h(X_i), \text{ and } \widehat{\mathcal{E}}(h, \mathcal{S}) \doteq \frac{1}{n}\sum_{i=1}^n e_h(X_i, Y_i).$$

Our first contribution is a basic PAC-Bayes result which the rest of our analysis builds on.

Proposition 2 (PAC-Bayes on excess risk)**.** *Let $\mathcal{H}$ denote a discrete family of classifiers, and fix a* prior *distribution P with support $\mathcal{H}$. Let $n \geq 8$ and $\delta \in (0,1)$. Suppose, there exists bounded functions $\widehat{\Delta}_n(h, \mathcal{S}), \Delta_n(h), h \in \mathcal{H}$ (depending on δ) such that*

$$\mathbb{P}\left(\forall h \in \mathcal{H},\, \widetilde{\mathcal{E}}(h, \mathcal{S}) \leq \widehat{\mathcal{E}}(h, \mathcal{S}) + \widehat{\Delta}_n(h, \mathcal{S})\right) \geq 1-\delta, \quad \inf_{h \in \mathcal{H}} \mathbb{P}\left(\widehat{\Delta}_n(h, \mathcal{S}) \leq \Delta_n(h)\right) \geq 1-\delta.$$

For any $\lambda \in (0,2)$, consider the following posterior over $\mathcal{H}$:

$$Q_\lambda^*(h) = \frac{1}{c}\, e^{-n\lambda\left(\widehat{\mathcal{R}}(h,\mathcal{S}) + \widehat{\Delta}_n(h,\mathcal{S})\right)} P(h), \quad \text{for } c = \mathbb{E}_{h\sim P}\, e^{-n\lambda\left(\widehat{\mathcal{R}}(h,\mathcal{S}) + \widehat{\Delta}_n(h,\mathcal{S})\right)}. \tag{2}$$

Then, with probability at least $1-4\delta$ over $\mathcal{S}$, simultaneously for all $\lambda \in (0,2)$:

$$\mathcal{E}(h_{Q_\lambda^*}) \leq \frac{L}{1-\lambda/2} \inf_{h \in \mathcal{H}} \left(\mathcal{E}(h) + \Delta_n(h) + \frac{\log(1/P(h))}{\lambda n} + \frac{\log\frac{2\sqrt{n}}{\delta} + \lambda\sqrt{2n\log\frac{1}{\delta}}}{\lambda n}\right).$$

Proposition 2 builds on Proposition 1 by first taking $\mathcal{R}_{\mathcal{L}}(h)$ to be $\mathcal{E}(h)$, $\widehat{\mathcal{R}}_{\mathcal{L}}(h)$ to be $\widetilde{\mathcal{E}}(h)$, and Z to be X. The bound in Proposition 2 is then obtained by optimizing over Q for fixed λ. Since this bound is on excess error (rather than error), optimizing over λ can only improve constants, while the choice of prior P is crucial in obtaining optimal rates as $|\mathcal{H}| \to \infty$. Such choice is treated next.

3.2 Oracle risk for trees (as $\mathcal{H} \doteq \mathcal{H}(T_0)$ grows in size with T_0)

We start with the following definitions on classifiers of interest and related quantities.

Definition 5. *Let T_0 be a binary partition-tree of $\mathcal{X}$ obtained from data $\mathcal{S}_0$, of depth D_0. Consider a family of classification trees $\mathcal{H}(T_0) \doteq \{h_T\}$ indexed by subtrees T of T_0, and where h_T defines a fixed labeling $l(A)$ of nodes $A \in \pi(T)$, e.g., $l(A) \doteq$ majority label in Y if $A \cap \mathcal{S}_0 \neq \emptyset$.*

Furthermore, for any node A of T_0, let $\hat{p}(A, \mathcal{S})$ denote the empirical mass of A under $\mathcal{S}$ and $p(A)$ be the population mass. Then for any subtree T of T_0, let $|T|$ be the number of nodes in T and define

$$\widehat{\Delta}_n(h_T, \mathcal{S}) \doteq \sum_{A \in \pi(T)} \sqrt{\hat{p}(A, \mathcal{S}) \frac{2 \log(|T_0| / \delta)}{n}}, \text{ and} \tag{3}$$

$$\Delta_n(h_T) \doteq \sum_{A \in \pi(T)} \sqrt{8 \max\left(p(A), \frac{(2 + \log D) \cdot D_0 + \log(1/\delta)}{n}\right) \frac{\log(|T_0| / \delta)}{n}}. \tag{4}$$

Remark 1. *In practice, we might start with a space partitioning tree T_0' (e.g., a dyadic tree, or KD-tree) which partitions $[0,1]^D$, rather than the support $\mathcal{X}$. We then view T_0 as the intersection of T_0' with $\mathcal{X}$.*

Our main theorem below follows from Proposition 2 on excess risk, by showing (a) that the above definition of $\widehat{\Delta}_n(h_T, \mathcal{S})$ and $\Delta_n(h_T)$ satisfies the conditions of Proposition 2, and (b) that there exists a proper prior P such that $\log(1/P(T)) \sim |\pi(T)|$, i.e., depends just on the subtree complexity rather than on that of T_0. The main technicality in showing (b) stems from the fact that P needs to be a proper distribution (i.e. $\sum_T P(T) = 1$) without requiring too large a normalization constant (remember that the number of subtrees can be exponential in the size of T_0). This is established through arguments from coding theory, and in particular Kraft-McMillan inequality.

Theorem 1 (Oracle risk for trees)**.** *Let the prior satisfy $P(h_T) \doteq (1/C_P) e^{-3D_0 \cdot |\pi(T)|}$ for a normalizing constant C_P, and consider the corresponding posterior Q_λ^* as defined in Equation 2, such that, with probability at least $1 - 4\delta$ over $\mathcal{S}$, for all $\lambda \in (0, 2)$, the excess risk $\mathcal{E}(h_{Q_\lambda^*})$ of the majority-classifier is at most*

$$\left(\frac{L}{1 - \lambda/2}\right) \cdot \min_{h_T \in \mathcal{H}(T_0)} \left(\mathcal{E}(h_T) + \Delta_n(h_T) + \frac{3D_0 \cdot |\pi(T)|}{\lambda n} + \frac{\log \frac{2\sqrt{n}}{\delta} + \lambda \sqrt{2n \log \frac{1}{\delta}}}{\lambda n} \right).$$

From Theorem 1 we can deduce that the majority classifier $h_{Q_\lambda^*}$ is consistent whenever the approach of pruning to the best subtree is consistent (typically, $\min_{h_T} \mathcal{E}(h_T) + (D_0 |\pi(T)|)/n = o_P(1)$). Furthermore, we can infer that $\mathcal{E}(h_{Q_\lambda^*})$ converges at the same rate as pruning approaches: the terms $\Delta_n(h_T)$ and $D_0 \cdot |\pi(T)|/n$ can be shown to be typically, of lower or similar order as $\mathcal{E}(h_T)$ for the best subtree classifier h_T. These remarks are formalized next and result in Corollary 1 below.

3.3 Rate of convergence

Much of known rates for tree-pruning are established for dyadic trees (see e.g. [9, 11]), due to their simplicity, under nonparametric assumptions on $\mathbb{E}[Y|X]$. Thus, we adopt such standard assumptions here to illustrate the rates achievable by $h_{Q_\lambda^*}$, following the more general statement of Theorem 1.

The first standard assumption below restricts how fast class probabilities change over space.

Assumption 1. *Consider the so-called* **regression function** $\eta(x) \in \mathbb{R}^L$ *with coordinate $\eta_l(x) \doteq \mathbb{E}_{Y|x} \mathbb{1}\{Y = l\}, l \in [L]$. We assume η is α-Hölder for $\alpha \in (0, 1]$, i.e.,*

$$\exists \lambda \text{ such that } \forall x, x' \in \mathcal{X}, \quad \|\eta(x) - \eta(x')\| \leq \lambda \|x - x'\|^\alpha.$$

Next, we illustrate some of the key conditions verified by dyadic trees which standard results build on. In particular, we want the *diameters* of nodes of T_0 to decrease relatively fast from the root down.

Assumption 2 (Conditions on T_0)**.** *The tree T_0 is obtained as the intersection of $\mathcal{X}$ with dyadic partition of $[0,1]^D$ (e.g. by cycling though coordinates) of depth $D_0 = O(D \log n)$ and partition size $|T_0| = O(n)$. In particular, we emphasize that the following conditions on subtrees then hold.*

For any subtree T of T_0, let $r(T)$ denote the maximum diameter of leaves of T (viewed as subsets of $\mathcal{X}$). There exist $C_1, C_2, d > 0$ such that:
For all $(C_1/n) < r \leq 1$, there exists a subtree T of T_0 such that $r(T) \leq r$ and $|\pi(T)| \leq C_2 r^{-d}$.

The above conditions on subtrees are known to *approximately* hold for other procedures such as KD-trees, and PCA-trees; in this sense, analyses of dyadic trees do yield some insights into the performance other approaches. The quantity d captures the *intrinsic dimension* (e.g., *doubling* or *box* dimension) of the data space $\mathcal{X}$ or is often of the same order [12, 13, 14].

Under the above two assumptions, it can be shown through standard arguments that the excess error of the best pruning, namely $\min_{h_T \in \mathcal{H}(T_0)} \mathcal{E}(h_T)$ is of order $n^{-\alpha/(2\alpha+d)}$, which is tight (see e.g. minimax lower-bounds of [15]). The following corollary to Theorem 1 states that such a rate, up to a logarithmic factor of n, is also attained by majority classification under Q^*_λ.

Corollary 1 (Adaptive rate of convergence)**.** *Assume that for any cell A of T_0, the labeling $l(A)$ corresponds to the majority label in A (under $\mathcal{S}_0$) if $A \cap \mathcal{S}_0 \neq \emptyset$, or $l(A) = 1$ otherwise. Then, under Assumptions 1 and 2, and the conditions of Theorem 1, there exists a constant C such that:*

$$\mathbb{E}_{\mathcal{S}_0, \mathcal{S}} \mathcal{E}(h_{Q^*_\lambda}) \leq C \left(\frac{\log n}{n} \right)^{\alpha/(2\alpha+d)}.$$

4 Algorithmic Results

Here we show that h_Q can be *efficiently* implemented by storing appropriate weights at nodes of T_0. Let $w_Q(A) \doteq \sum_{h_T : A \in \pi(T)} Q(h_T)$ aggregate weights over all subtrees T of T_0 having A as a leaf. Then $h_Q(x) = \operatorname{argmax}_{l \in [L]} \sum_{A \in \text{path}(x), l(A) = l} w_Q(A)$, where path$(x)$ denotes all nodes of T_0 containing x. Thus, $h_Q(x)$ is computable from weights proportional to $w_Q(A)$ at every node.

We show in what follows that we can efficiently obtain $w(A) = C \cdot w_{Q^*_\lambda}(A)$ by dynamic-programming by ensuring that $Q^*_\lambda(h_T)$ is multiplicative over $\pi(T)$. This is the case, given our choice of prior from Theorem 1: we have $Q^*_\lambda(h_T) = (1/C_{Q^*_\lambda}) \cdot \exp(\sum_{A \in \pi(T)} \phi(A))$ where

$$\phi(A) \doteq -\lambda \sum_{i : X_i \in A \cap \mathcal{S}} \mathbb{1}\{Y_i \neq l(A)\} - n\lambda \sqrt{\hat{p}(A, \mathcal{S}) \frac{2 \log(|T_0|/\delta)}{n}} - 3D_0.$$

We can then compute $w(A) \doteq C_{Q^*_\lambda} \cdot w_{Q^*_\lambda}(A)$ via dynamic-programming. The intuition is similar to that in [5], however, the particular form of our weights require a two-pass dynamic program (bottom-up and top-down) rather than the single pass in [5]. Namely, $w(A)$ divides into subweights that any node A' might contribute up or down the tree. Let

$$\alpha(A) \doteq \sum_{h_T : A \in \pi(T)} \exp\left(\sum_{A' \neq A, A' \in \pi(T)} \phi(A') \right), \tag{5}$$

so that $w(A) = e^{\phi(A)} \cdot \alpha(A)$. As we will show (proof of Theorem 2), $\alpha(A)$ decomposes into contributions from the parent A_p and sibling A_s of A, i.e., $\alpha(A) = \alpha(A_p)\beta(A_s)$ where $\beta(A_s)$ is given as (writing T_0^A for the subtree of T_0 rooted at A, and $T \preceq T'$ when T is a subtree of T'):

$$\beta(A_s) = \sum_{T \preceq T_0^{A_s}} \exp\left(\sum_{A' \in \pi(T)} \phi(A') \right). \tag{6}$$

The contributions $\beta(A)$ are first computed using the bottom-up Algorithm 1, and the contributions $\alpha(A)$ and final weights $w(A)$ are then computed using the top-down Algorithm 2. For ease of presentation, these routines run on a full-binary tree version $\bar{T}_0$ of T_0, obtained by adding a dummy child to each node A that has a single child in T_0. Each dummy node A' has $\phi(A') = 0$.

Algorithm 1 Bottom-up pass

for $A \in \pi(\bar{T}_0)$ **do**
  $\beta(A) \leftarrow e^{\phi(A)}$
end for
for $i \leftarrow D_0$ **to** 0 **do**
  $\mathcal{A}_i \leftarrow$ set of nodes of $\bar{T}_0$ at depth i
  for $A \in \mathcal{A}_i \setminus \pi(\bar{T}_0)$ **do**
    $\mathcal{N} \leftarrow$ the children nodes of A
    $\beta(A) \leftarrow e^{\phi(A)} + \prod_{A' \in \mathcal{N}} \beta(A')$
  end for
end for

Algorithm 2 Top-down pass

$\alpha(\text{root}) \leftarrow 1$
for $i \leftarrow 1$ **to** D_0 **do**
  $\mathcal{A}_i \leftarrow$ set of nodes of $\bar{T}_0$ at depth i
  for $A \in \mathcal{A}_i$ **do**
    $A_p, A_s \leftarrow$ parent of node A, sibling of node A
    $\alpha(A) \leftarrow \alpha(A_p)\beta(A_s)$
    $w(A) \leftarrow e^{\phi(A)}\alpha(A)$
  end for
end for

Theorem 2 (Computing $w(A)$)**.** *Running Algorithm 1, then 2, we obtain* $w(A) \doteq C_{Q^*_\lambda} \cdot w_{Q^*_\lambda}(A)$*, where* Q^*_λ *is as defined in Theorem 1. Furthermore, the combined runtime of Algorithms 1, then 2 is* $2|\bar{T}_0| \leq 4|T_0|$*, where* $|T|$ *is the number of nodes in* T*.*

5 Experiments

Table 1: UCI datasets

Name (abbreviation)	Features count	Labels count	Train size
Spambase (spam)	57	2	2601
EEG Eye State (eeg)	14	2	12980
Epileptic Seizure Recognition (epileptic)	178	2	9500
Crowdsourced Mapping (crowd)	28	6	8546
Wine Quality (wine)	12	11	4497
Optical Recognition of Handwritten Digits (digit)	64	10	3620
Letter Recognition (letter)	16	26	18000

Here we present experiments on real-world datasets, for two common partition-tree approaches, dyadic trees and KD-trees. The various datasets are described in Table 1.

The **main baseline** we compare against, is a popular efficient pruning heuristic where a subtree of T_0 is selected to minimize the penalized error $\mathcal{C}_1(h_T) = \widehat{\mathcal{R}}(h_T, \mathcal{S}) + \lambda \frac{|\pi(T,\mathcal{S})|}{n}$.

We also compare against other tree-based approaches that are theoretically driven and efficient. First is a pruning approach proposed in [16], which picks a subtree minimizing the penalized error $\mathcal{C}_2(h_T) = \widehat{\mathcal{R}}(h_T, \mathcal{S}) + \lambda \sum_A \sqrt{\max\left(\hat{p}(A,\mathcal{S}), \frac{\|A\|}{n}\right) \cdot \frac{\|A\|}{n}}$, where $\|A\|$ denotes the depth of node A in T_0. We note that, here we choose a form of $\mathcal{C}_2$ that avoids theoretical constants that were of a technical nature, but instead let λ account for such. We report this approach as **SN-pruning**. Second is the majority classifier of [5], which however is geared towards binary classification as it requires *regression*-type estimates in $[0, 1]$ at each node. This is denoted **HS-vote**.

All the above approaches have efficient dynamic programs that run in time $O(|T_0|)$, and all predict in time $O(\text{height}(T_0))$. The same holds for our PAC-Bayes approach as discussed above in Section 4.

Practical implementation of PAC-Bayes tree. Our implementation rests on the theoretical insights of Theorem 1, however we avoid some of the technical details that were needed for rigor,

such as sample splitting and overly conservative constants in concentration results. Instead we advise cross-validating for such constants in the prior and posterior definitions. Namely, we first set $P(h_T) \propto \exp(-|\pi(T,\mathcal{S})|)$, where $\pi(T,\mathcal{S})$ denotes the leaves of T containing data. We set $\Delta_n(h_T,\mathcal{S}) = \sum_{A\in\pi(T,\mathcal{S})}\sqrt{\frac{\hat{p}(A,\mathcal{S})}{n}}$. The posterior is then set as $Q^*(h_T) \propto \exp(-n(\lambda_1\widehat{\mathcal{R}}(h_T,\mathcal{S}) + \lambda_2\Delta_n(h_T,\mathcal{S})))P(h_T)$, where λ_1, λ_2 account for concentration terms to be tuned to the data.

Finally, we use the entire data to construct T_0 and compute weights, i.e., $\mathcal{S}_0 = \mathcal{S}$, as inter-dependencies are in fact less of an issue in practice. We note, that the above alternative theoretical approaches, SN-pruning and HS-vote, are also assumed (in theory) to work on a sample independent choice of T_0 (or equivalently built and labeled on a separate sample S_0), but are implemented here on the entire data to similarly take advantage of larger data sizes. The baseline pruning heuristic is by default always implemented on the full data.

Experimental setup and results. The data is preprocessed as follows: for dyadic trees, data is scaled to be in $[0,1]^D$, while for KD-trees data is normalized accross each coordinate by standard deviation.

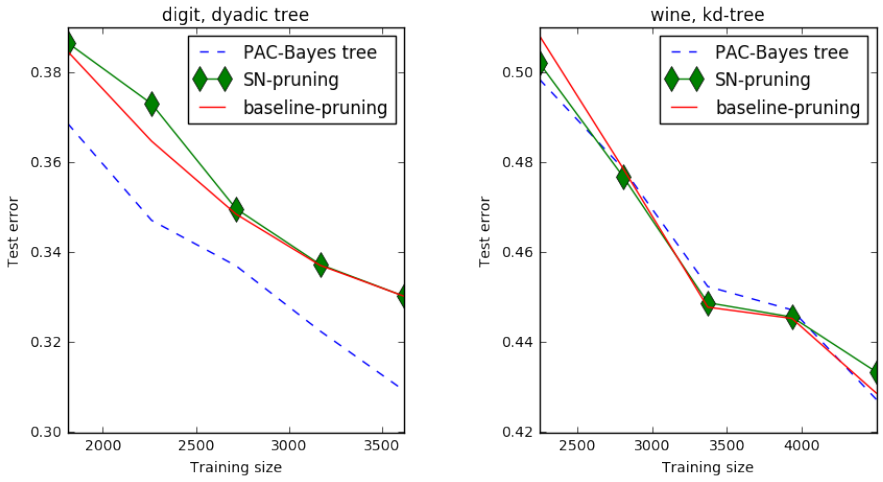
Testing data is fixed to be of size 2000, while each experiment is ran 5 times (with random choice of training data of size reported in Table 1) and average performance is reported. In each experiment, all parameters are chosen by 2-fold cross-validation for each of the procedures. The log-grid is 10 values, equally spaced in logarithm, from 2^{-8} to 2^6 while the linear-grid is 10 linearly-spaced values between half the best value of the log-search and twice the best value of the log-search.

Table 2 reports classification performance of the various theoretical methods relative to the baseline pruning heuristic. We see that proposed PAC-Bayes tree achieves competitive performance against all other alternatives. All the approaches have similar performance accross datasets, with some working slightly better on particular datasets. Figure 2 further illustrates typical performance on multiclass problems as training size varies.

Table 2: Ratio of classification error over that of the default pruning baseline: bold indicates best results across methods, while blue indicates improvement over baseline; N/A means the algorithm was not run on the task.

	$T_0 \equiv$ dyadic tree			$T_0 \equiv$ KD tree		
Dataset	SN-pruning	PAC-Bayes tree	HS-vote	SN-pruning	PAC-Bayes tree	HS-vote
spam	1.118	**0.975**	1.224	1.048	1.020	1.075
eeg	**0.979**	0.993	1.029	1.000	0.990	1.000
epileptic	0.993	0.992	0.951	0.977	0.987	**0.907**
crowd	**0.991**	1.020	N/A	1.001	1.017	N/A
wine	1.035	**0.991**	N/A	1.010	0.997	N/A
digit	1.000	**0.936**	N/A	0.994	0.997	N/A
letter	1.005	**0.993**	N/A	1.000	1.001	N/A

Figure 2: Classification error versus training size



References

[1] David A McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

[2] László A Székely and Hua Wang. On subtrees of trees. *Advances in Applied Mathematics*, 34(1):138–155, 2005.

[3] Trevor Hastie and Daryl Pregibon. *Shrinking trees*. AT & T Bell Laboratories, 1990.

[4] Wray Buntine and Tim Niblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8(1):75–85, 1992.

[5] David P Helmbold and Robert E Schapire. Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, 27(1):51–68, 1997.

[6] Alexandre Lacasse, François Laviolette, Mario Marchand, Pascal Germain, and Nicolas Usunier. PAC-Bayes bounds for the risk of the majority vote and the variance of the Gibbs classifier. In *Advances in Neural information processing systems*, pages 769–776, 2007.

[7] John Langford and John Shawe-Taylor. PAC-Bayes & margins. In *Advances in neural information processing systems*, pages 439–446, 2003.

[8] Pascal Germain, Alexandre Lacasse, Francois Laviolette, Mario Marchand, and Jean-Francis Roy. Risk Bounds for the Majority Vote: From a PAC-Bayesian Analysis to a Learning Algorithm. *Journal of Machine Learning Research*, 16:787–860, 2015.

[9] C. Scott and R.D. Nowak. Minimax-optimal classification with dyadic decision trees. *IEEE Transactions on Information Theory*, 52, 2006.

[10] Niklas Thiemann, Christian Igel, Olivier Wintenberger, and Yevgeny Seldin. A Strongly Quasiconvex PAC-Bayesian bound. In Steve Hanneke and Lev Reyzin, editors, *Proceedings of the 28th International Conference on Algorithmic Learning Theory*, volume 76 of *Proceedings of Machine Learning Research*, pages 466–492, Kyoto University, Kyoto, Japan, 15–17 Oct 2017. PMLR.

[11] L. Gyorfi, M. Kohler, A. Krzyzak, and H. Walk. *A Distribution Free Theory of Nonparametric Regression*. Springer, New York, NY, 2002.

[12] Nakul Verma, Samory Kpotufe, and Sanjoy Dasgupta. Which spatial partition trees are adaptive to intrinsic dimension? In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 565–574. AUAI Press, 2009.

[13] Samory Kpotufe and Sanjoy Dasgupta. A tree-based regressor that adapts to intrinsic dimension. *Journal of Computer and System Sciences*, 78(5):1496–1515, 2012.

[14] Santosh Vempala. Randomly-oriented kd trees adapt to intrinsic dimension. In *FSTTCS*, volume 18, pages 48–57. Citeseer, 2012.

[15] Jean-Yves Audibert and Alexandre B Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2):608–633, 2007.

[16] Clayton Scott. *Dyadic Decision Trees*. PhD thesis, Rice University, 2004.

[17] Olivier Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*. Institute of Mathematical Statistics, 2007.

[18] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[19] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[20] Colin McDiarmid. *On the method of bounded differences*. Cambridge University Press, Cambridge, 1989.

6 Appendix

6.1 Proposition 2

The proof Proposition 2 requires the following lemma, which states that the excess risk of the voting classifier is always at most L times the excess risk of the Gibbs classifier defined by the same distribution. It has the same spirit as the well-known result for the binary case where where the classification risk of the voting classifier is at most 2 times that of the stochastic classifier.

Lemma 1. *For any Q, it holds that:*

$$\mathcal{E}(h_Q) \leq L \cdot \mathcal{E}(Q).$$

Proof. For any classifier h, we have the following decomposition:

$$\mathcal{E}(h) = \mathbb{E}_X[\mathbb{P}\left(h_B(x) = Y | X = x\right) - \mathbb{P}\left(h(x) = Y | X = x\right)]$$

Interchanging order of integration, we write the excess risk of the Gibbs classifier as:

$$\mathcal{E}(Q) = \mathbb{E}_{h\sim Q}\mathcal{E}(h) = \mathbb{E}_X\left[\mathbb{E}_{h\sim Q}\big(\mathbb{P}\left(Y = h_B(x) | X = x\right) - \mathbb{P}\left(Y = h(x) | X = x\right)\big)\right]$$

Similarly, the excess risk of the majority classifier is:

$$\mathcal{E}(h_Q) = \mathbb{E}_X[\mathbb{P}\left(h_B(x) = Y | X = x\right) - \mathbb{P}\left(h_Q(x) = Y | X = x\right)]$$

Now, we observe the point-wise relationship between regression gaps, which is true for all x:

$$\begin{aligned}\mathbb{P}\left(h_B(x) = Y | X = x\right) - &\mathbb{P}\left(h_Q(x) = Y | X = x\right) \\ &\leq L \cdot \mathbb{E}_{h\sim Q}\big(\mathbb{P}\left(h_B(x) = Y | X = x\right) - \mathbb{P}\left(Y = h(x) | X = x\right)\big)\end{aligned}$$

Because of the majority rule nature of h_Q, if the output label $h_Q(x)$ is l, there must be at least $\frac{1}{L}$ (under Q) classifiers in $\mathcal{H}$ which predicts l. Hence, there must be at least $\frac{1}{L}$ (under Q) classifiers with the same gap in regression value as the voting classifier. In addition, because the other classifiers whose prediction is different from $h_Q(x)$ have non-negative regression gap, the statement is established. Integrating this point-wise inequality over X and using monotonicity of integration, we are done. □

In addition, the following lemma states that the distribution minimizing objectives that is the sum of a linear function in the distribution plus the Kullback-Leibler divergence w.r.t to a given priror has a particular exponential form.

Lemma 2 (Lemma 1.1.3 of [17])**.** *Suppose $\mathcal{H}$ is a hypothesis class. Let $G : \mathcal{H} \to \mathbb{R}$ be a bounded function. For a reference distribution P, define the Q^* distribution over $\mathcal{H}$:*

$$Q^*(h) = \frac{1}{c'} e^{-G(h)} P(h),$$

where c' is the normalization constant $c' = \mathbb{E}_{h\sim P} e^{-G(h)}$. Then, for all distributions Q over $\mathcal{H}$:

$$\mathbb{E}_{h\sim Q} G(h) + \mathcal{D}_{kl}\left(Q \| P\right) = -\log \mathbb{E}_{h\sim P} e^{-G(h)} + \mathcal{D}_{kl}\left(Q \| Q^*\right).$$

We are ready to delve into the proof of Proposition 2.

Proof of Proposition 2. First, we show how the PAC-Bayes theorem of Proposition 1 applies to excess risk. It was necessary to introduce the intermediate loss function $e_h(x)$, which is valued in $[0, 1]$ while $e_h(x, y)$ might not be. The reason why $e_h(X_i)$ is valued in $[0, 1]$ is that:

$$e_h(x) = \mathbb{P}\left(h_B(X_i) = Y | X = x\right) - \mathbb{P}\left(h(x) = Y | X = x\right)$$

where $0 \leq \mathbb{P}\left(h(x) = Y | X = x\right) \leq \mathbb{P}\left(h_B(x) = Y | X = x\right) \leq 1$. In addition, $\mathcal{E}(h) = \mathbb{E}e_h(X)$. Therefore we can apply Proposition 1 with $l_h(x, y) = e_h(x)$ to conclude that, with probability at least $1 - \delta$ over the sampling of $\mathcal{S}$, simultaneously for all $\lambda \in (0, 2)$ and all posteriors Q:

$$\mathcal{E}(Q) \leq \frac{\widetilde{\mathcal{E}}(Q, \mathcal{S})}{1 - \lambda/2} + \frac{\mathcal{D}_{\text{kl}}\left(Q \| P\right) + \log \frac{2\sqrt{n}}{\delta}}{\lambda(1 - \lambda/2)n}. \tag{7}$$

Now, by the definition of $\widehat{\Delta}_n(h, \mathcal{S})$, with probability at least $1-\delta$ over the sampling of $\mathcal{S}$, $\forall \lambda \in (0, 2)$, for all $h \in \mathcal{H}$ it holds that $\widetilde{\mathcal{E}}(h, \mathcal{S}) \leq \widehat{\mathcal{E}}(h, \mathcal{S}) + \widehat{\Delta}_n(h, \mathcal{S})$. Because the upper bound holds for all classifiers, when we take expectation on both sides using the same distribution, the inequality still holds. In other words, for a distribution Q, if we define $\Delta_n(Q, \mathcal{S}) \doteq \mathbb{E}_{h\sim Q}\Delta_n(h, \mathcal{S})$ then with probability at least $1-\delta$ over $\mathcal{S}$, simultaneously for all Q:

$$\widetilde{\mathcal{E}}(Q, \mathcal{S}) \leq \widehat{\mathcal{E}}(Q, \mathcal{S}) + \widehat{\Delta}_n(Q, \mathcal{S}) \tag{8}$$

A simple union bound guarantees that the probability of both events in Equation 7 and Equation 8 occurring is at least $1-2\delta$. In such situations, simultaneously for all $\lambda \in (0, 2)$ and distributions Q:

$$\mathcal{E}(Q) \leq \frac{\widehat{\mathcal{E}}(Q, \mathcal{S}) + \widehat{\Delta}_n(Q, \mathcal{S})}{1 - \lambda/2} + \frac{\mathcal{D}_{\text{kl}}\left(Q\|P\right) + \log\frac{2\sqrt{n}}{\delta}}{\lambda(1-\lambda/2)n} \tag{9}$$

Second, we prove that for any fixed $\lambda \in (0, 2)$, Q^*_λ as defined in Equation 2 minimizes the right hand side of Equation 9 out of all distributions over $\mathcal{H}$, which is equivalent to minimizing:

$$\phi(Q) \doteq \mathbb{E}_{h\sim Q}[n\lambda\widehat{\mathcal{E}}(h, \mathcal{S}) + n\lambda\widehat{\Delta}_n(h, \mathcal{S})] + \mathcal{D}_{\text{kl}}\left(Q\|P\right). \tag{10}$$

This objective has the right form for us to apply Lemma 2. We define $G(h) = n\lambda\widehat{\mathcal{E}}(h, \mathcal{S}) + n\lambda\widehat{\Delta}_n(h, \mathcal{S})$. Because $\widehat{\Delta}_n(h, \mathcal{S})$ is bounded, $G(h)$ satisfies the boundedness assumption and we can apply the Lemma. Since the Kullback-Leibler divergence is non-negative, it is true that

$$\min_Q \phi(Q) = \min_Q(\mathbb{E}_{h\sim Q}G(h) + KL(Q||P)) = -\log \mathbb{E}_{h\sim P}e^{-G(h)},$$

The minimum is attained at $Q = Q^*$. Upon closer inspection, it turns out that $Q^* = Q^*_\lambda$ as defined in Equation 2. Clearly, empirical excess risk is equal to the difference between empirical risks:

$$\widehat{\mathcal{E}}(h, \mathcal{S}) = \widehat{\mathcal{R}}(h, \mathcal{S}) - \widehat{\mathcal{R}}(h_B, \mathcal{S}).$$

Therefore:

$$\begin{aligned} Q^*(h) &= \frac{e^{-n\lambda(\widehat{\mathcal{E}}(h,\mathcal{S})+\widehat{\Delta}_n(h,\mathcal{S})}P(h)}{\mathbb{E}_{h'\sim P}e^{-n\lambda(\widehat{\mathcal{E}}(h,\mathcal{S})+\widehat{\Delta}_n(h,\mathcal{S})}} = \frac{e^{n\lambda\widehat{\mathcal{R}}(h_B,\mathcal{S})}\cdot e^{-n\lambda(\widehat{\mathcal{R}}(h,\mathcal{S})+\widehat{\Delta}_n(h,\mathcal{S})}P(h)}{e^{n\lambda\widehat{\mathcal{R}}(h_B,\mathcal{S})}\cdot\mathbb{E}_{h'\sim P}e^{-n\lambda(\widehat{\mathcal{R}}(h',\mathcal{S})+\widehat{\Delta}_n(h',\mathcal{S}))}} \\ &= \frac{e^{-n\lambda(\widehat{\mathcal{R}}(h,\mathcal{S})+\widehat{\Delta}_n(h,\mathcal{S})}P(h)}{\mathbb{E}_{h'\sim P}e^{-n\lambda(\widehat{\mathcal{R}}(h',\mathcal{S})+\widehat{\Delta}_n(h',\mathcal{S}))}} = Q^*_\lambda(h). \end{aligned}$$

Third, we put the results of the first two steps to formulate an oracle inequality for the Gibbs classifier. For simplicity assume that some $h^* \in \mathcal{H}$ achieves the infimum in the right hand side of Proposition 2. By Hoeffding's inequality [18], since $\widehat{\mathcal{E}}(h^*, \mathcal{S}) = \frac{1}{n}\sum_{i=1}^n e_h(X_i, Y_i)$ where $e_h(X_i, Y_i)$ are i.i.d valued in $\{-1, 1\}$, it holds with probability at least $1-\delta$ over $\mathcal{S}$ that:

$$\widehat{\mathcal{E}}(h^*, \mathcal{S}) - \mathcal{E}(h^*) \leq \sqrt{\frac{2\log\frac{1}{\delta}}{n}}. \tag{11}$$

In addition, also consider the following event, which has probability at least $1-\delta$, where:

$$\widehat{\Delta}_n(h^*, \mathcal{S}) \leq \Delta_n(h^*) \tag{12}$$

Supposing that the statements of Equation 9, Equation 11 and Equation 12 hold, which is an event of probability at least $1-4\delta$. Because Q^*_λ minimizes the right hand side of Equation 9, in particular it does better than Q_{h^*} i.e.

$$\begin{aligned} \mathcal{E}(Q^*_\lambda) &\leq \frac{n\lambda(\widehat{\mathcal{E}}(h^*, \mathcal{S}) + \widehat{\Delta}_n(h^*, \mathcal{S})) - \log P(h^*) + \log\frac{2\sqrt{n}}{\delta}}{\lambda(1-\frac{\lambda}{2})n} \\ &\leq \frac{n\lambda(\mathcal{E}(h^*) + \sqrt{2\frac{\log\frac{1}{\delta}}{n}} + \Delta_n(h^*)) - \log P(h^*) + \log\frac{2\sqrt{n}}{\delta}}{\lambda(1-\frac{\lambda}{2})n} \\ &\leq \frac{\mathcal{E}(h^*) + \Delta_n(h^*)}{1-\frac{\lambda}{2}} + \frac{\log(1/P(h^*)) + \log\frac{2\sqrt{n}}{\delta} + \lambda\sqrt{2n\log\frac{1}{\delta}}}{\lambda(1-\frac{\lambda}{2})n}. \end{aligned}$$

Finally, combining this upper bound on the excess risk of the Gibbs classifier with Lemma 1 to conclude that with probability at least $1 - 4\delta$ over $\mathcal{S}$:

$$\mathcal{E}(h_{Q_\lambda^*}) \leq \frac{L}{1 - \lambda/2} \inf_{h \in \mathcal{H}} \left(\mathcal{E}(h) + \Delta_n(h) + \frac{\log(1/P(h))}{\lambda n} + \frac{\log \frac{2\sqrt{n}}{\delta} + \lambda \sqrt{2n \log \frac{1}{\delta}}}{\lambda n} \right).$$

□

6.2 Theorem 1

We need a few lemmas to prove Theorem 1. The first is Kraft's inequality, a standard tool in coding theory.

Lemma 3 (Kraft's inequality [19])**.** *For any prefix-free code over an alphabet of size D, the codeword lengths $l_1, l_2, \ldots, l_m$ must satisfy:*

$$\sum_i D^{-l_i} \leq 1$$

The next lemma shows that the normalization constant defining the prior in Theorem 1 is at most 1.

Lemma 4. *Recall that the prior in Theorem 1 has the normalization constant $C_P = \sum_{h_T \in \mathcal{H}(T_0)} e^{-3D_0 \cdot |\pi(T)|}$. It is true that:*

$$C_P \leq 1$$

Proof. The idea is to design a prefix-free codebook for $\mathcal{H}(T_0)$ over the binary alphabet $\{0, 1\}$ and use Kraft's inequality, a standard tool in coding theory (A prefix-free code is a codebook such that no codeword is a prefix of another). Because of the one-to-one correspondence between the classifier h_T and the subtree T, it suffices to encode T.

First, we define a prefix-free code for the set of nodes in T_0 by modifying the strategy in Section 2.3.2 of [16]. The code for a node A consists of three components, concatenated in order of appearance:

- Depth encoding: if the depth of A is k, the encoding is string of k ones.
- A delimiting 0 to signify that the depth encoding has ended
- Path encoding: the sequence of left and right links that is the path from the root to A, where a left link is encode as 0 and a right link is encoded as 1.

This is a prefix-free code for the set of nodes of T_0: the sequence of ones at the start eliminates the possibilities of two nodes at different depths having codewords that are prefix of each other, and for nodes of the same depth, there will be a discrepancy in the paths from root to the nodes that makes it impossible for prefixing. Given this codebook of nodes, the encoding E of subtree T, it suffices to concatenate the codewords for $A \in \pi(T)$: deeper nodes are put in front, and among those at the same depth, go from left to right.

Second, we prove that E is a prefix-free code for the family of subtrees T over the alphabet $\{0, 1\}$ i.e. for subtrees $T_1 \neq T_2$, neither $E(T_1)$ is a prefix of $E(T_2)$ nor vice versa. On one hand, consider the case that $E(T_1) = E(T_2)$ i.e. two different subtrees having the same encoding. However, since no two different subtrees can have the same leaf set, this is not possible.

On the other hand, consider the case when one codeword is a proper prefix of the other: without loss of generality, assume $E(T_1)$ is a proper prefix of $E(T_2)$. Because E is the concatenation of prefix-free codes, from $E(T_1)$ we reconstruct uniquely the leaf set $\pi(T_1)$ and from $E(T_2)$ we reconstruct uniquely the leaf set $\pi(T_2)$. Since $E(T_1)$ is a proper prefix of $E(T_2)$, it must be true that one leaf set is contained in the other $\pi(T_1) \subset \pi(T_2)$ and there exists A' such that $A' \in \pi(T_2)$ but $A' \notin \pi(T_1)$. But because both $\pi(T_1)$ and $\pi(T_2)$ partition $\mathcal{X}$, it means that $A' \cap \mathcal{X} = \emptyset$. This is a contradiction since A' is supposed to intersect $\mathcal{X}$

Overall, there can be no code that is a prefix of another code. We now analyze the length of the encoding E. It is easy to see that each node A has a codelength at most 3 times its depth. Therefore, to encode a subtree T, whose maximum depth is bounded by D_0 and has $|\pi(T)|$ leaves we can

use codewords whose lengths are upper bounded by $3\log_2 e \cdot D_0 \cdot |\pi(T)|$. We now employ Kraft's inequality. In our case, the prefix-free codebook for subtrees T is over the alphabet $\{0,1\}$ and $3D_0 \log_2 e \cdot |\pi(T)|$ are upper bounds on the codelengths. Hence:

$$\sum_{h_T \in \mathcal{H}(T_0)} 2^{-(3D_0 \log_2 e \cdot |\pi(T)|)} \leq 1 \Longrightarrow \sum_{h_T \in \mathcal{H}(T_0)} e^{-3D_0 \cdot |\pi(T)|} \leq 1.$$

□

The next lemma proves that $\widehat{\Delta}_n(h_T, \mathcal{S})$ and $\Delta_n(h_T)$ defined in Equation 3 and Equation 4 satisfies the conditions of Proposition 2 for the hypothesis class $\mathcal{H}(T_0)$.

Lemma 5. *With $\widehat{\Delta}_n(h_T, \mathcal{S})$ defined as in Equation 3, with probability at least $1-\delta$ over $\mathcal{S}$, for all $h_T \in \mathcal{H}(T_0)$:*

$$\widetilde{\mathcal{E}}(h_T, \mathcal{S}) \leq \widehat{\mathcal{E}}(h_T, \mathcal{S}) + \widehat{\Delta}_n(h_T, \mathcal{S})$$

Proof. Let $n(A, \mathcal{S}) = n\hat{p}(A, \mathcal{S})$ be the number of data points in $\mathcal{S}$ which falls into A. Define

$$c(A, \mathcal{S}) \doteq \begin{cases} \frac{1}{n(A,\mathcal{S})} \sum_{X_i \in A} e_h(X_i, Y_i) & \text{if } n(A, \mathcal{S}) > 0 \\ 0 & \text{otherwise} \end{cases}$$

For fixed $X^n = \{X_i\}_{i=1}^n$, denote by $\bar{c}(A, X^n)$ the condition expectation over $Y^n = \{Y_i\}_{i=1}^n$ is:

$$\begin{aligned} \bar{c}(A, X^n) &\doteq \mathbb{E}_{Y^n|X^n} c(A, \mathcal{S}) \\ &= \begin{cases} \frac{1}{n(A,\mathcal{S})} \sum_{X_i \in A} e_h(X_i) & \text{if } n(A, \mathcal{S}) > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

First, for fixed X^n, we derive a uniform concentration result for all nodes A in T_0, with the randomness from Y^n. For nodes that contain data, $c(A, \mathcal{S})$ is the average of $n(A, \mathcal{S})$ independent random variables since $Y_1, Y_2, \ldots, Y_n$ are independent conditioned on X^n. Furthermore, $c(A, \mathcal{S})$ satisfies a bounded variation condition: a change in some Y_i for results in at most a change of $\frac{2}{n(A,\mathcal{S})}$ in $c(A, \mathcal{S})$. Hence, for any node A of T_0, for any $\epsilon_A > 0$, by McDiarmid's inequality [20]:

$$\Pr_{Y^n|X^n}(\bar{c}(A, X^n) - c(A, \mathcal{S}) > \epsilon_A) \leq \exp\left(- \frac{2\epsilon_A^2}{n(A, \mathcal{S})(\frac{2}{n(A,\mathcal{S})})^2} \right) = e^{-n(A,\mathcal{S})\epsilon_A^2/2}$$

We set $\epsilon_A = \sqrt{2\log(|T_0|/\delta)/n(A, \mathcal{S})}$. In addition, we multiply both sides of the event $\bar{c}(A, X^n) - c(A, \mathcal{S}) > \epsilon_A$ by $\hat{p}(A, \mathcal{S})$ to have that:

$$\Pr_{Y^n|X^n}\left(\hat{p}(A, \mathcal{S})(\bar{c}(A, X^n) - c(A, \mathcal{S})) \leq \sqrt{\hat{p}(A, \mathcal{S}) \frac{2\log(|T_0|/\delta)}{n}} \right) \leq \frac{\delta}{|T_0|}$$

Now, we perform a union bound, with at most $|T_0|$ elements to conclude that for fixed X^n, with probability at least $1-\delta$ over Y^n, simultaneously for all A such that $\hat{p}(A, \mathcal{S}) > 0$:

$$\hat{p}(A, \mathcal{S})(\bar{c}(A, X^n) - c(A, \mathcal{S})) \leq \sqrt{\hat{p}(A, \mathcal{S}) \frac{2\log(|T_0|/\delta)}{n}} \tag{13}$$

As for nodes A such that $\hat{p}(A, \mathcal{S}) = 0$, by definition, for all Y^n it holds that that $\bar{c}(A, X^n) - c(A, \mathcal{S}) = 0$. Hence, the statement in Equation 13 is also true for empty cells. Moving on to the tail bound for the subtree classifiers' excess risk. The empirical excess risk, the intermediate losss and the penalty of each h_T is decomposable over the leaves of T:

$$\widetilde{\mathcal{E}}(h_T, \mathcal{S}) - \widehat{\mathcal{E}}(h_T, \mathcal{S}) = \sum_{A \in \pi(T)} \hat{p}(A, \mathcal{S})[\bar{c}(A, X^n) - c(A, \mathcal{S})]$$

$$\widehat{\Delta}_n(h_T, \mathcal{S}) = \sum_{A \in \pi(T)} \sqrt{\hat{p}(A, \mathcal{S}) \frac{2\log(|T_0|/\delta)}{n}}$$

Because of this decomposition, the following inclusion of events is true:

$$\{\exists h_T : \widetilde{\mathcal{E}}(h_T, \mathcal{S}) - \widehat{\mathcal{E}}(h_T, \mathcal{S}) > \widehat{\Delta}_n(h_T, \mathcal{S})\}$$
$$\subseteq \left\{ \exists A : \hat{p}(A, \mathcal{S})[\bar{c}(A, X^n) - c(A, \mathcal{S})] > \sqrt{\hat{p}(A, \mathcal{S}) \frac{2\log(|T_0|/\delta)}{n}} \right\}$$

According to Equation 13, the probability over Y^n of the later event is at most δ. Therefore:

$$\Pr_{Y^n|X^n}(\exists h_T : \widetilde{\mathcal{E}}(h_T, \mathcal{S}) - \widehat{\mathcal{E}}(h_T, \mathcal{S}) > \widehat{\Delta}_n(h_T, \mathcal{S})) \leq \delta$$

Taking the expectation of both sides w.r.t to X^n, and taking the complement event, we conclude that:

$$\Pr_{\mathcal{S}}(\forall h_T : \widetilde{\mathcal{E}}(h_T, \mathcal{S}) - \widehat{\mathcal{E}}(h_T, \mathcal{S}) \leq \widehat{\Delta}_n(h_T, \mathcal{S})) \geq 1 - \delta$$

□

Lemma 6. *With $\Delta_n(h_T)$ defined in Equation 4 for any $h_T \in \mathcal{H}(T_0)$, with probability at least $1 - \delta$ over $\mathcal{S}$:*

$$\widehat{\Delta}_n(h_T, \mathcal{S}) \leq \Delta_n(h_T)$$

Proof. It is implied by Lemma 1 [16]. Each node in T_0 can be associated with a codeword $\|A\|$ that is proportional to its depth in T_0, with the constant being upper bounded by $2 + \log D$. Then, with probability at least $1 - \delta$, for all A:

$$\hat{p}(A, \mathcal{S}) \leq 4 \max\left(p(A), \frac{\|A\| + \log(1/\delta)}{n}\right)$$

Since the maximal depth of a node in T_0 is D_0, we replace $\|A\|$ by $(2 + \log D) \cdot D_0$:

$$\hat{p}(A, \mathcal{S}) \leq 4 \max\left(p(A), \frac{(2 + \log D) \cdot D_0 + \log(1/\delta)}{n}\right)$$

If the inequality above holds for all A, for any h_T, by taking the summation of $A \in \pi(T)$ on both sides to prove the statement of the lemma. □

Proof of Theorem 1. Theorem 1 is a direct application of Proposition 2 for the family $\mathcal{H}(T_0)$.

It is clear from Lemma 5 and Lemma 6 that $\widehat{\Delta}_n(h_T, \mathcal{S})$ and $\Delta_n(h_T)$ satisfy the conditions of Proposition 2 (the boundedness condition is automatically satisfied because $\mathcal{H}(T_0)$ is finite). Hence, for the choice of prior $P(h_T) = \frac{1}{C_p} e^{-3D_0 \cdot |\pi(T)|}$, with probability at least $1 - 4\delta$ over $\mathcal{S}$, simultaneously for all $\lambda \in (0, 2)$:

$$\mathcal{E}(h_{Q^*_\lambda})$$
$$\leq \frac{L}{1 - \lambda/2} \min_{h_T \in \mathcal{H}(T_0)} \left(\mathcal{E}(h_T) + \Delta_n(h_T) + \frac{\log C_p + 3D_0 \cdot |\pi(T)|}{\lambda n} + \frac{\log \frac{2\sqrt{n}}{\delta} + \lambda\sqrt{2n \log \frac{1}{\delta}}}{\lambda n} \right).$$

By Lemma 4, $\log C_P \leq 0$. This shows the statement of the Theorem. □

6.3 Corollary 1

As usual when it comes to tree-based classification, we first go through tree-based regression. Each label Y is converted to a vector one-hot encoding $b(Y)$ where the l coordinate $b_l(Y) = 1$ if $Y = l$ and 0 otherwise. Then we have a family of regression trees $\{\eta_T\}$ indexed by subtrees T of T_0 where η_T defines a fixed regression value $s(A)$ of nodes $A \in \pi(T)$, e.g. $s(A) \doteq$ average value of $b(Y)$ if $A \cap \mathcal{S}_0 \neq \emptyset$.

The following proposition, implied by Equation A.1 in [13], shows the bias-variance decomposition of L_2 excess risk for regressors based on dyadic trees.

Proposition 3 (Bias-variance decomposition [13])**.** *There are absolute constants C_3, C_4 such that the following hold. For any T that induces a dyadic partition of $\mathcal{X}$, it is true that:*

$$\mathbb{E}_{\mathcal{S}_0,X}\|\eta_T(X) - \eta(X)\|^2 \leq C_3\lambda^2 r(T)^{2\alpha} + C_4\frac{|\pi(T)|}{n}$$

The next lemma shows how the L_2 excess risk of a regressor is an upper bound on the excess risk of the associated plug-in classifier. The result for binary classification is well-established: here we prove in the multiclass case for completeness.

Lemma 7. *Suppose $\hat{\eta}(x)$ is a regressor (trained from data) and $\hat{h}(x)$ is the associated plug-in classifier $\hat{h}(x) = \mathrm{argmax}_{l\in[L]}\,\hat{\eta}_l(x)$. Then:*

$$\mathcal{E}(\hat{h}) \leq 2\sqrt{L}\cdot\sqrt{\mathbb{E}_X\|\hat{\eta}(X) - \eta(X)\|_2^2}$$

Proof. We first prove that:

$$\mathcal{E}(\hat{h}) \leq 2\mathbb{E}_X\|\hat{\eta}(X) - \eta(X)\|_1$$

The idea is to prove the point-wise inequality:

$$\mathbb{P}\left(h_B(x) = Y|X = x\right) - \mathbb{P}\left(\hat{h}(x) = Y|X = x\right) \leq 2|\hat{\eta}(x) - \eta(x)|$$

(Integration over x of the left hand side gives the excess risk of $\hat{h}$ while integrating over the right hand side gives the L_1 regression risk.) The inequality can be rewritten as:

$$\eta_{h_B(x)}(x) - \eta_{\hat{h}(x)}(x) \leq 2|\hat{\eta}(x) - \eta(x)|$$

Let $|\hat{\eta}(x) - \eta(x)| = u$. Then, for all $l \in [L]$, $|\hat{\eta}_l(x) - \eta_l(x)| \leq u$. In particular:

$$\left|\hat{\eta}_{h_B(x)}(x) - \eta_{h_B(x)}(x)\right| \leq u$$
$$\left|\hat{\eta}_{\hat{h}(x)}(x) - \eta_{\hat{h}(x)}(x)\right| \leq u$$

From the first equation, we have $\eta_{h_B(x)}(x) \leq \hat{\eta}_{h_B(x)}(x) + u$, so that $\eta_{h_B(x)}(x) - \eta_{\hat{h}(x)}(x) \leq \hat{\eta}_{h_B(x)}(x) - \eta_{\hat{h}(x)}(x) + u$. Because $\hat{h}(x)$ is the plug-in of $\hat{\eta}(x)$, we have $\hat{\eta}_{h_B(x)}(x) \leq \hat{\eta}_{\hat{h}(x)}(x)$. Combined this fact with the second equation, which says $\hat{\eta}_{\hat{h}(x)}(x) - \eta_{\hat{h}(x)}(x) \leq u$, overall we have shown that $\eta_{h_B(x)}(x) - \eta_{\hat{h}(x)}(x) \leq 2u$.

We then combine with the well-known inequality between L_p norms:

$$\mathbb{E}_X\|\hat{\eta}(X) - \eta(X)\|_1 \leq \sqrt{L\cdot\mathbb{E}_X\|\hat{\eta}(X) - \eta(X)\|_2^2}$$

Truly:

$$\|\hat{\eta}(X) - \eta(X)\|_1^2 = (\sum_l |\hat{\eta}_l(X) - \eta_l(X)|)^2 \leq L\sum_l(\hat{\eta}_l(X) - \eta_l(X))^2$$

so we have, by Jensen's inequality:

$$(\mathbb{E}_X\|\hat{\eta}(X) - \eta(X)\|)^2 \leq \mathbb{E}_X\|\hat{\eta}(X) - \eta(X)\|_1^2 \leq L\cdot\mathbb{E}\|\hat{\eta}(X) - \eta(X)\|_2^2$$

□

Proof of Corollary 1. We first convert Theorem 1, a statement in high probability over $\mathcal{S}$, to a statement in expectation over $\mathcal{S}$. We select $\delta = \frac{1}{n}$. In the rare event (probability at most $\frac{4}{n}$) that the upper bound in Theorem 1 does not hold, we still have the trivial upper bound on excess risk $\mathcal{E}(h_{Q_\lambda^*}) \leq 1$. Therefore, for fixed $\mathcal{S}_0$ but taking expectation over $\mathcal{S}$, we have:

$$\mathbb{E}_{\mathcal{S}}\mathcal{E}(h_{Q_\lambda^*}) \leq C_0 \min_{h_T\in\mathcal{H}(T_0)}\left(\mathcal{E}(h_T) + \Delta_n(h_T) + \frac{3D_0|\pi(T)|}{\lambda n} + \frac{\log(2n\sqrt{n}) + \lambda\sqrt{2n\log n} + 4\lambda}{\lambda n}\right). \tag{14}$$

where $C_0 = \frac{L}{1-\lambda/2}$. We now show that the expectation of the right hand side has the right dependencies on n, α, d. The strategy is to demonstrate the existence of a right resolution r that results in a classification tree with the right excess risk. By Assumption 2, for any $(C_1/n) < r \leq 1$, there exists a subtree T^r of T_0 such that $r(T^r) \leq r$, $|\pi(T^r)| \leq C_2 r^{-d}$. Combined with Proposition 3, the excess risk of the regressor has the form:

$$\mathbb{E}_{\mathcal{S}_0,X}\|\eta_{T^r}(X) - \eta(X)\|^2 \leq C_3\lambda^2 r^{2\alpha} + C_4\frac{r^{-d}}{n}.$$

Consider the choice $r_* \doteq (\frac{C_4}{C_3\lambda^2})^{1/(2\alpha+d)}(\frac{\log n}{n})^{1/(2\alpha+d)}$. Such r_* is permissible since $\frac{1}{n}$ is smaller that $(\frac{\log n}{n})^{1/(2\alpha+d)}$ for all large n. Therefore, for some constant C_5, the L_2 excess risk of $\eta_{T^{r^*}}$ satisfies:

$$\mathbb{E}_{\mathcal{S}_0,X}\|\eta_{T^{r_*}}(X) - \eta(X)\|^2 \leq C_5\left(\frac{\log n}{n}\right)^{2\alpha/(2\alpha+d)}.$$

Because of how we define η_T, h_T is the plug-in classifier associated with η_T. Using Lemma 7, we have for some constant C_8:

$$\mathbb{E}_{\mathcal{S}_0}\mathcal{E}(h_{T^{r_*}}) \leq C_8\left(\frac{\log n}{n}\right)^{\alpha/(2\alpha+d)}.$$

We move on to bound $\Delta_n(h_{T^{r_*}})$. Observe that $\sqrt{\max(a,b)} \leq \sqrt{a} + \sqrt{b}$ for any $a, b \geq 0$. Hence:

$$\begin{aligned}
\Delta_n(h_{T^{r_*}}) &= \sqrt{\frac{\log(n\,|T_0|)}{n}}\sum_{A\in\pi(T^{r_*})}\sqrt{4\max\left(p(A), \frac{D_0+\log n}{n}\right)} \\
&\leq 2\sqrt{\frac{\log(n\,|T_0|)}{n}}\sum_{A\in\pi(T^{r_*})}\left(\sqrt{p(A)} + \sqrt{\frac{D_0+\log n}{n}}\right) \\
&\leq 2\sqrt{\frac{\log(n\,|T_0|)}{n}}\sum_{A\in\pi(T^{r_*})}\sqrt{p(A)} + 2\sqrt{\log(n\,|T_0|)(D_0+\log n)}\frac{|\pi(T^{r_*})|}{n}.
\end{aligned}$$

We bound the first summation by Jensen's inequality for concave $\sqrt{x}$, supposing that the summation is over A such that $p(A) > 0$:

$$\sum_{A\in\pi(T^{r_*})}\sqrt{p(A)} = \sum_{A\in\pi(T^{r_*})}p(A)\frac{1}{\sqrt{p(A)}} \leq \sqrt{\sum_{A\in\pi(T^{r_*})}\frac{p(A)}{p(A)}} = \sqrt{|\pi(T^{r_*})|}.$$

Combining the fact that the maximal depth $D_0 = O(D\log n)$ and that T_0 has $O(n)$ leaves, it means that $|T_0| = O(Dn\log n)$. Therefore, there exists a constant C_6 such that:

$$\Delta_n(h_{T^{r_*}}) \leq C_6(\sqrt{\frac{\log n\cdot|\pi(T^{r_*})|}{n}} + \frac{\log n\cdot|\pi(T^{r_*})|}{n}).$$

Recall that $|\pi(T^{r_*})| \leq r_*^{-d}$. This implies that for setting $r = r^*$ there exists some constant C_7 such that

$$\sqrt{\frac{\log n\cdot|\pi(T^{r_*})|}{n}} \leq C_7\left(\frac{\log n}{n}\right)^{\alpha/(2\alpha+d)},$$

which leads to the overall conclusion that there exists some constant C_8 satisfying:

$$\mathbb{E}_{\mathcal{S}_0}\big(\mathcal{E}(h_{T^{r_*}}) + \Delta_n(h_{T^{r_*}}) + \frac{3D_0\cdot|\pi(T^{r_*})|}{\lambda n}\big) \leq C_8\left(\frac{\log n}{n}\right)^{\alpha/(2\alpha+d)}. \tag{15}$$

The order of growth on the right hand side of the above inequality dominates the $\frac{\log(2n\sqrt{n})+\lambda\sqrt{2n\log n}+4\lambda}{\lambda n}$ component of the right hand side of Equation 14. We now show the rate of convergence over $\mathcal{S}_0$ and $\mathcal{S}$. Clearly, for any $\mathcal{S}_0$:

$$\min_{h_T\in\mathcal{H}(T_0)}\left(\mathcal{E}(h_T) + \Delta_n(h_T) + \frac{3D_0\cdot|\pi(T)|}{\lambda n}\right) \leq \mathcal{E}(h_{T^{r_*}}) + \Delta_n(h_{T^{r_*}}) + \frac{3D_0\cdot|\pi(T^{r_*})|}{\lambda n}. \tag{16}$$

Therefore, by taking expectation of both sides of Equation 14, combining Equations 15 and 16, we conclude that there exists a constant C such that:

$$\mathbb{E}_{\mathcal{S}_0,\mathcal{S}}\mathcal{E}(h_{Q^*_\lambda}) \leq C\left(\frac{\log n}{n}\right)^{\alpha/(2\alpha+d)}.$$

□

6.4 Theorem 2

Proof. That the combined runtime of Algorithm 1 and Algorithm 2 is $2\left|\bar{T}_0\right|$ is clear: and each algorithm will visit each node of $\bar{T}_0$ exactly once, doing a constant amount of computation. In addition, $\left|\bar{T}_0\right| \leq 2\left|T_0\right|$ since we only add one dummy node to at most $|T_0|$ nodes.

Regarding the correctness of the procedure: if we show that indeed, the $\beta(A)$ computed through Algorithm 1 is equal to the right hand side of Equation 6 and the $\alpha(A)$ computed through Algorithm 2 is equal to the right hand side of Equation 5, then we will have proven that after the two algorithms, $w(A) = Cw_{Q^*_\lambda}(A)$ for some positive constant C, since if $\alpha(A)$ is the expression in Equation 5, then

$$\begin{aligned} w(A) &= e^{\phi(A)} \sum_{h_T : A\in\pi(T)} \exp\left(\sum_{A'\neq A, A'\in\pi(T)} \phi(A')\right) = \sum_{h_T : A\in\pi(T)} \exp\left(\sum_{A'\in\pi(T)} \phi(A')\right) \\ &= C_{Q^*_\lambda} \cdot w_{Q^*_\lambda}(A) \end{aligned}$$

The strategy is to prove the correctness on $\bar{T}_0$: because the dummy nodes A' that we add to T_0 to form $\bar{T}_0$ have zero contribution $\phi(A') = 0$, we also have correctness on T_0. For A in $\bar{T}_0$, we denote by

$$\beta^*(A) \doteq \sum_{T\preceq \bar{T}_0^A} \exp\left(\sum_{A'\in\pi(T)} \phi(A')\right),$$

and

$$\alpha^*(A) \doteq \sum_{T : A\in\pi(T)} \exp\left(\sum_{A'\neq A, A'\in\pi(T)} \phi(A')\right).$$

and set out to prove that $\beta(A) = \beta^*(A)$ and $\alpha(A) = \alpha^*(A)$. For any node A, denote A_L to be the left child of A and A_R to be the right child, in the augmented tree $\bar{T}_0$.

Regarding $\beta(A) = \beta^*(A)$, it suffices to show that $\beta^*(A)$ satisfies the base case and the recurrence relation defining $\beta(A)$ in Algorithm 1, namely:

$$\beta^*(A) = \begin{cases} e^{\phi(A)} & \text{if } A \in \pi(\bar{T}_0) \\ e^{\phi(A)} + \beta^*(A_L)\beta^*(A_R) & \text{otherwise} \end{cases}$$

When A is a leaf, $\beta^*(A) = e^{\phi(A)}$ is immediate since the only the subtree rooted at A is A itself, viewed as a subtree. When A is an internal node, any $T \preceq T_0^A$ is either just the node A or can be decomposed into a left subtree T_L and a right subtree T_R that are rooted at A_L and A_R, respectively. The former case contributes the term $e^{\phi(A)}$ to the sum. In the later case, the product over $A' \in \pi(T)$ is the same as the product over $A_1 \in \pi(T_L)$ times the product over $A_2 \in \pi(T_R)$. In other words:

$$\begin{aligned} \beta^*(A) &= e^{\phi(A)} + \sum_{T_L\preceq T_0^{A_L}} \sum_{T_R\preceq T_0^{A_R}} \prod_{A_1\in\pi(T_L)} \prod_{A_2\in\pi(T_R)} e^{\phi(A_1)}e^{\phi(A_2)} \\ &= e^{\phi(A)} + \Big(\sum_{T_L\preceq T_0^{A_L}} \prod_{A_1\in\pi(T_L)} e^{\phi(A_1)}\Big)\Big(\sum_{T_R\preceq T_0^{A_R}} \prod_{A_2\in\pi(T_R)} e^{\phi(A_2)}\Big) \\ &= e^{\phi(A)} + \beta^*(A_L)\beta^*(A_R) \end{aligned}$$

Hence, we have $\beta(A) = \beta^*(A)$.

Regarding $\alpha(A) = \alpha^*(A)$, again we aim to show that $\alpha^*(A)$ satisfies the base case and recurrence relation defining $\alpha(A)$ in Algorithm 2, namely:

$$\begin{aligned}\alpha^*(\text{root}) &= 1\\ \alpha^*(A_L) &= \alpha^*(A)\beta^*(A_R)\\ \alpha^*(A_R) &= \alpha^*(A)\beta^*(A_L)\end{aligned}$$

The first equation is clear: when A is the root node, the summation defining $\alpha^*(A)$ is over the subtree with only the root, and there are no leaves in this tree except the root, so the summation inside the exponential is 0. We only need to prove the second equation: the third follows in the same manner. The partition T of any subtree classifier h_T such that $A_L \in \pi(T)$ can be decomposed into three parts: the part T_1 which is a pruned subtree of T_0 such that $A \in \pi(T_1)$, the extension of T_1 by a pruned subtree rooted at A_R (which we denote T_2) and the inclusion of itself A_L as a leaf. There are no constraints between T_1 and T_2 except that the former must have A as a leaf and T_2 is rooted at A_R. Therefore:

$$\begin{aligned}\alpha^*(A_L) &= \sum_{T_1 : A \in \pi(T_1)} \sum_{T_2 \preceq T_0^{A_R}} \exp\left(\sum_{A' \neq A, A' \in \pi(T_1)} \phi(A')\right) \exp\left(\sum_{A' \in \pi(T_2)} \phi(A')\right)\\ &= \left(\sum_{T_1 : A \in \pi(T_1)} \exp\left(\sum_{A' \neq A, A' \in \pi(T_1)} \phi(A')\right)\right)\left(\sum_{T_2 \preceq T_0^{A_R}} \exp\left(\sum_{A' \in \pi(T_2)} \phi(A')\right)\right)\\ &= \alpha^*(A)\beta^*(A_R)\end{aligned}$$

Hence we have $\alpha(A) = \alpha^*(A)$. □