Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Random Composite Forests

Giulia DeSalvo
Courant Institute of Mathematical Sciences
251 Mercer St.
New York, NY 10012
desalvo@cims.nyu.edu

Mehryar Mohri
Courant Institute and Google Research
251 Mercer St.
New York, NY 10012
mohri@cs.nyu.edu

Abstract

We introduce a broad family of decision trees, *Composite Trees*, whose leaf classifiers are selected out of a hypothesis set composed of p subfamilies with different complexities. We prove new data-dependent learning guarantees for this family in the multi-class setting. These learning bounds provide a quantitative guidance for the choice of the hypotheses at each leaf. Remarkably, they depend on the Rademacher complexities of the sub-families of the predictors and the fraction of sample points correctly classified at each leaf. We further introduce *random composite trees* and derive learning guarantees for random composite trees which also apply to Random Forests. Using our theoretical analysis, we devise a new algorithm, RANDOMCOMPOSITEFOREST (RCF), that is based on forming an ensemble of random composite trees. We report the results of experiments demonstrating that RCF yields significant performance improvements over both Random Forests and a variant of RCF in several tasks.

Introduction

Random Forests (RFs) are ensemble learning models introduced by Breiman (2001) combining the bagging approach (Breiman 1996) and the random subspace method (Ho 1995; Amit and Geman 1997). They are used in a variety of applications for classification and regression tasks ranging from computer vision and medical imaging to finance (Criminisi, Shotton, and Konukoglu 2012), often outperforming existing methods (Schroff, Criminisi, and Zisserman 2008; Shotton et al. 2011; Montillo et al. 2011; Xiong et al. 2012).

In contrast with their empirical success, the theoretical analysis of RFs is still subject to several questions, as pointed out by Denil, Matheson, and de Freitas (2014). There has been considerable work seeking to prove consistency results for RFs under some simplifying assumptions (Biau, Devroye, and Lugosi 2008; Biau 2012) including more recent work presenting a finer analysis (Denil, Matheson, and de Freitas 2014), but few theoretical publications have focused on deriving finite-sample generalization bounds for these models. It is worth mentioning that a by-product of our analysis is a new data-dependent finite-sample learning guarantee for RFs using the random subspace method.

To achieve a higher accuracy in some difficult tasks, more complex decision trees are needed, with leaf classifiers (or node questions) selected out of richer hypothesis sets than those commonly used by RFs. We will consider a broad family of decision trees, *Composite Trees*, whose leaf classifiers are selected out of a hypothesis set composed of p subfamilies with different complexities. Of course, trees with leaf classifiers consistently picked from a rich hypothesis set would be prone to overfitting. However, when the leaf classifiers are selected more rarely from more complex subfamilies and more frequently from the less complex ones, then learning can be successful, while providing the flexibility of using richer hypotheses when needed. We will present new data-dependent learning guarantees in the multi-class setting that provide a quantitative guidance for the choice of the hypotheses at each leaf of a composite tree. Our bounds are given in terms of the Rademacher complexities of the p subfamilies and, remarkably, depend on the fraction of correctly classified points at each leaf.

Previous empirical papers have examined the performance of decision trees whose leaves have a specific classifier, but little theoretical analysis of this problem has been developed. The classifiers used at the leaves have been generated from various algorithms: neural nets (Zhou and Chen 2002), linear regression and nearest neighbor algorithm (Seewald, Petrak, and Widmer 2001), and Gaussian processes (Kohavi 1996; Frohlich et al. 2013; Seewald, Petrak, and Widmer 2001). Moreover, several papers combine decision trees with the SVM algorithm, but none provide generalization bounds or any other strong theoretical justification (Chang et al. 2010; Bennet and Blue 1998; Takahashi and Abe 2002; Dong and Chen 2008; Madjarov and Gjorgjevikj 2012; Arreola, Fehr, and Burkhardt 2006; 2007; Rodriguez-Lujan, Cruz, and Huerta 2012; Kumar and Gopal 2010; Wang and Saligrama 2012). One paper that is close to our theoretical analysis is by Golea et al. (1997). They provide generalization bounds for decision trees in terms of the VC dimension, but they assume boolean leaf functions under the setting of binary classification.

We also define randomized versions of composite trees. As already mentioned, randomization appears in the definition of RFs in two principle ways: through bagging and randomized node optimization (RNO) (Ho 1995; Amit and Geman 1997; Ho 1998; Dietterich et al. 2000). Bagging gen-

erates several samples of the same size from the training set by sampling uniformly and with replacement in order to then independently train a decision tree for each sample. RNO selects a random subset of the features at each node of the decision tree and optimizes the information gain over this subset of features to choose the best splitting criteria. Random forests may use either RNO, bagging, or both when generating the decision trees in a forest. We define *Random Composite Trees* as composite trees where the same technique as RNO is used to randomly sample features for each node question of the tree.

We further extend our analysis of composite trees to RFs and also derive learning guarantees for random composite trees. The main contribution of our paper is the theoretical analysis of generalized decision trees and random forests, but we further supplement the theory with an algorithm and preliminary experimental results. Using our theoretical analysis, we derive and implement an algorithm, named RANDOMCOMPOSITEFOREST (RCF), that is based on forming an ensemble of random composite trees. We report the results of experiments demonstrating that RCF yields significant performance improvements over RFs in several tasks. Moreover, we provide results for a variant of RCF showing that the theoretical bound used to derived the RCF is a key factor in the algorithm.

The rest of this paper is organized as follows. In the Preliminaries, we introduce some initial concepts and notation. In the Composite Trees Section, we give a formal definition of composite trees. Next, in the Learning Guarantee Section, we derive data-dependent learning bounds for composite trees and random composite trees. In the Algorithm Section, we describe our RCF algorithm, including its pseudocode. Finally, we report the results of a series of experiments with RCF in the Experiments Section.

Preliminaries

We consider the familiar set-up of supervised learning in a multi-class setting. Let $\mathcal{X}$ denote the input space and $\mathcal{Y} = \{1, \ldots, c\}$ a set of c classes indexed by integers. We assume that training and test points are drawn i.i.d. according to some distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ and denote by $S = ((x_1, y_1), \ldots, (x_m, y_m))$ a training sample of size m drawn according to $\mathcal{D}^m$.

The label associated by a hypothesis $f \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ to $x \in \mathcal{X}$ is given by $\operatorname{argmax}_{y \in \mathcal{Y}} f(x, y)$. The margin $\rho_f(x, y)$ of the function f for a labeled example $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is defined by $\rho_f(x, y) = f(x, y) - \max_{y' \neq y} f(x, y')$. Thus, f misclassifies (x, y) iff $\rho_f(x, y) \leq 0$. Let $\rho > 0$. The generalization error $R(f)$ of f, its empirical margin error $\widehat{R}_{S,\rho}(f)$, and its empirical error $\widehat{R}_S(f)$ for a sample S are defined as follows:

$$R(f) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[1_{\rho_f(x,y)\leq 0}], \widehat{R}_{S,\rho}(f) = \mathop{\mathbb{E}}_{(x,y)\sim S}[1_{\rho_f(x,y)\leq \rho}],$$

$$\text{and } \widehat{R}_S(f) = \mathop{\mathbb{E}}_{(x,y)\sim S}[1_{\rho_f(x,y)\leq 0}],$$

where the notation $(x, y) \sim S$ indicates that (x, y) is drawn according to the empirical distribution defined by S. For any family of hypotheses G mapping $\mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$, we denote by $\widetilde{G}$ the family of functions that G defines based on its first argument:

$$\widetilde{G} = \{x \mapsto g(x, y) \colon y \in \mathcal{Y}, g \in G\}.$$

Composite Trees

Let $\mathcal{H}$ denote a family of p distinct hypothesis sets of functions mapping $\mathcal{X} \times \mathcal{Y}$ to $[0, 1]$. For any $k \in \{1, \ldots, l\}$ with $l \geq 1$, let $\mathcal{S}_k$ denote a family of functions mapping $\mathcal{X}$ to $\{0, 1\}$. A *Composite Tree* with $l \geq 1$ leaves is a tree of classifiers which, in the most generic view, can be defined by a triplet $(\mathbf{H}, \mathsf{s}, \mathbf{h})$ where

- $\mathbf{H} = (H_1, \ldots, H_l)$ is an element of $\mathcal{H}^l$; the hypothesis used at leaf $k \in \{1, \ldots, l\}$ is selected from $H_k \in \mathcal{H}$;
- $\mathsf{s} \colon \mathcal{X} \times [1, l] \to \{0, 1\}$ is a *leaf selector*, that is $\mathsf{s}(x, k) = 1$ if x is assigned to leaf k, $\mathsf{s}(x, k) = 0$ otherwise. The function $\mathsf{s}(\cdot, k)$ is an element of $\mathcal{S}_k$ for each leaf k;
- $\mathbf{h} = (h_k)_{k\in[1,l]}$ is a *leaf classifier* such that $h_k \colon \mathcal{X} \times \mathcal{Y} \to [0, 1]$ and $\sum_{y\in\mathcal{Y}} h_k(x, y) = 1$. The function h_k is an element of H_k for each leaf k.

We define $\mathcal{H}_k = \{x \mapsto \mathsf{s}(x, k)h_k(x, y) \colon \mathsf{s}(\cdot, k) \in \mathcal{S}_k, h_k \in H_k\}$ to be the family made of the products of a k-leaf selector and a k-leaf classifier.

In standard decision trees, the leaf selector s can be decomposed into *node questions* (or their complements) and we will later assume this decomposition: for any $x \in \mathcal{X}$ and $k \in [1, l]$, $\mathsf{s}(x, k) = \prod_{j=1}^{d_k} q_j(x)$, where d_k is the depth of leaf k and where each function $q_j \colon \mathcal{X} \to \{0, 1\}$ is an element of a family Q_j.[1] However, the first part of our analysis does not require that assumption.

Each triplet $(\mathbf{H}, \mathsf{s}, \mathbf{h})$ defines a Composite Tree Function $f \colon \mathcal{X} \times \mathcal{Y} \to [0, 1]$ as follows:

$$f(x, y) = \sum_{k=1}^{l} \mathsf{s}(x, k)h_k(x, y).$$

We denote by $\mathcal{T}_l$ the family of all composite tree functions with l leaves thereby defined.

Learning Guarantees

In this section, we first present new learning guarantees for composite trees. Next, we define *random composite trees* and further derive learning bounds for this family.

Generalization bounds for composite trees

Let m_k be the number of points at leaf k and let m_k^+ be the number of sample points at leaf k that are classified correctly, $m_k^+ = |\{i \colon \rho_{h_k}(x_i, y_i) > 0, \mathsf{s}(x_i, k) = 1\}|$. Similarly, let m_k^- be the number of sample points at leaf k that are misclassified, $m_k^- = |\{i \colon \rho_{h_k}(x_i, y_i) \leq 0, \mathsf{s}(x_i, k) = 1\}|$.

[1] Each q_j is either a node question q or its complement $\bar{q}$ defined by $\bar{q}(x) = 1$ iff $q(x) = 0$. The family Q_j is assumed symmetric: it contains $\bar{q}$ when it contains q.

The main result of this section is Theorem 1, which gives data-dependent learning bounds for multi-class composite trees in terms of the quantity m_k^+ and the Rademacher complexities of the families $\widetilde{\mathcal{H}}_k$ made of products of a leaf selector and a leaf classifier (see Preliminaries). The following is a simpler form of our learning bound; with high probability, for all $f \in \mathcal{T}_l$, the following inequality holds:

$$R(f) \leq \widehat{R}_S(f) + \sum_{k=1}^{l} \min\left(8c\,\mathfrak{R}_m(\widetilde{\mathcal{H}}_k), \frac{m_k^+}{m}\right) + \widetilde{O}\left(\frac{1}{\rho}\sqrt{\frac{\log pl}{m}}\right) + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}. \quad (1)$$

Remarkably, the bound suggests that a composite tree using a very complex hypothesis set $\mathcal{H}_k$ can nevertheless benefit from favorable learning guarantees when the fraction of correctly classified points at the leaves labeled with functions from $\mathcal{H}_k$ is relatively small. Note further that, in this comparison, the complexity term $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k)$ is scaled by the number of distinct classes c, which, in some applications could be relatively large. Even in the simple case of $p = 1$ where there is only one hypothesis class H_1 for leaf classifiers, the result is striking: in the worst case, the complexity term could be in $O(l\mathfrak{R}_m(\widetilde{\mathcal{H}}_1))$; however, this result shows that it could be substantially less for some composite trees since it may be that for many leaves $m_k^+/m \ll c\,\mathfrak{R}_m(\widetilde{\mathcal{H}}_1)$. The fraction of points m_k^+/m at each leaf k depends on the choice of the composite tree. Thus, the bound can guide the design of different algorithms by selecting composite trees benefitting from better guarantees. Finally, note that the bound admits only a logarithmic dependency on the number of distinct hypothesis sets p.

In the following, we assume that the leaves are numbered in order of increasing depth and at times use the shorthand $r_k = \mathfrak{R}_m(\widetilde{\mathcal{H}}_k)$. Let $\mathcal{K}$ denote the set of leaves whose fraction of correctly classified points at leaf k are greater than $8cr_k$: $\mathcal{K} = \{k \in [1, l] : \frac{m_k^+}{m} > 8cr_k\}$. The following learning bound holds in the case of leaf predictors taking values in $\{0, 1\}$. We have also extended our analysis to the case of leaf predictors taking values in $[0, 1]$ (full version of our paper).

Theorem 1. *Fix $\rho > 0$. Assume that for all $k \in [1, l]$, the functions in H_k take values in $\{0, 1\}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S of size $m \geq 1$, the following holds for all $l \geq 1$ and all $f \in \mathcal{T}_l$ defined by $(\mathbf{H}, \mathsf{s}, \mathbf{h})$:*

$$R(f) \leq \widehat{R}_S(f) + \sum_{k=1}^{l} \min\left(8c\,\mathfrak{R}_m(\widetilde{\mathcal{H}}_k), \frac{m_k^+}{m}\right) + \min_{\substack{\mathcal{L}\subseteq\mathcal{K} \\ |\mathcal{L}|\geq|\mathcal{K}|-\frac{1}{\rho}}} \sum_{k\in\mathcal{L}} \left(\frac{m_k^+}{m} - 8c\,\mathfrak{R}_m(\widetilde{\mathcal{H}}_k)\right) + C(m, p, \rho) + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}$$

where $C(m, p, \rho) = \frac{2}{c\rho}\sqrt{\frac{\log pl}{m}} + \sqrt{\left\lceil \frac{4}{\rho^2}\log\left(\frac{c^2\rho^2 m}{4\log pl}\right)\right\rceil \frac{\log pl}{m}}$

$= \widetilde{O}\left(\frac{1}{\rho}\sqrt{\frac{\log pl}{m}}\right)$.

Proof. Let Δ denote the simplex in $\mathbb{R}^l$ and $\mathrm{int}(\Delta)$ its interior. For any $\boldsymbol{\alpha} \in \mathrm{int}(\Delta)$, let $g_{\boldsymbol{\alpha}}$ be the function defined for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ by

$$g_{\boldsymbol{\alpha}}(x, y) = \sum_{k=1}^{l} \alpha_k \mathsf{s}(x, k) h_k(x, y).$$

Observe that $g_{\boldsymbol{\alpha}}$ generates the same classifications as f since for any $x \in \mathcal{X}$, $s(x, k) = 1$ for only one leaf k and since $\alpha_k \geq 0$. Thus, we can equivalently analyze $R(g_{\boldsymbol{\alpha}})$ instead of $R(f)$, for any $\boldsymbol{\alpha} \in \mathrm{int}(\Delta)$.

Since $g_{\boldsymbol{\alpha}}$s are convex combinations of the functions $x \mapsto \mathsf{s}(x, k)h_k(x, y)$, we can apply to the set of functions $g_{\boldsymbol{\alpha}}$ the learning guarantees for convex ensembles with multiple hypothesis sets given by Kuznetsov, Mohri, and Syed (2014), which show that, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for any $\boldsymbol{\alpha} \in \mathrm{int}(\Delta)$ and any $f \in \mathcal{T}_l$ defined by $(\mathbf{H}, \mathsf{s}, \mathbf{h})$:

$$R(f) \leq \widehat{R}_{S,\rho}(g_{\boldsymbol{\alpha}}) + \frac{8c}{\rho}\sum_{k=1}^{l} \alpha_k \mathfrak{R}_m(\widetilde{\mathcal{H}}_k) + C(m, p, \rho) + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}$$

where $\mathcal{H}_k = \{(x, y) \mapsto \mathsf{s}(x, k)h_k(x, y) \colon h_k \in H_k,\ \mathsf{s}(\cdot, k) \in \mathcal{S}_k\}$. Since the inequality holds for any $\boldsymbol{\alpha} \in \mathrm{int}(\Delta)$, it implies that with probability at least $1 - \delta$, the following holds any $f \in \mathcal{T}_l$ defined by $(\mathbf{H}, \mathsf{s}, \mathbf{h})$:

$$R(f) \leq \inf_{\boldsymbol{\alpha}\in\mathrm{int}(\Delta)} \left[\widehat{R}_{S,\rho}(g_{\boldsymbol{\alpha}}) + \frac{8c}{\rho}\sum_{k=1}^{l} \alpha_k \mathfrak{R}_m(\widetilde{\mathcal{H}}_k)\right] + C(m, p, \rho) + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}. \quad (2)$$

This bound depends on the choice of $\boldsymbol{\alpha} \in \mathrm{int}(\Delta)$. The crux of our proof consists now of removing $\boldsymbol{\alpha}$ to derive an explicit bound. The first terms of the right-hand side can be re-written as

$$B(\boldsymbol{\alpha}) = \frac{1}{m}\sum_{k=1}^{l} \sum_{\mathsf{s}(x_i,k)=1} 1_{\alpha_k \rho_{h_k}(x_i,y_i)<\rho} + \frac{8c}{\rho}\sum_{k=1}^{l} \alpha_k \mathfrak{R}_m(\widetilde{\mathcal{H}}_k) \quad (3)$$

by definition of the empirical margin error $\widehat{R}_{S,\rho}(g_{\boldsymbol{\alpha}}) = \frac{1}{m}\sum_{k=1}^{l}\sum_{\mathsf{s}(x_i,k)=1} 1_{\alpha_k\rho_{h_k}(x_i,y_i)<\rho}$ and that of $\rho_{h_k}(x_i, y_i)$ given in the Preliminaries Section. Observe that function B can be decoupled as a sum, $B(\boldsymbol{\alpha}) = \sum_{k=1}^{l} B_k(\alpha_k)$, where

$$B_k(\alpha_k) = \frac{1}{m}\sum_{\mathsf{s}(x_i,k)=1} 1_{\alpha_k\rho_{h_k}(x_i,y_i)<\rho} + \frac{8c}{\rho}\alpha_k r_k.$$

For any $k \in [1, l]$, $B_k(\alpha_k)$ can be rewritten as follows in terms of m_k^- and m_k^+: $B_k(\alpha_k) = \frac{m_k^-}{m} + \frac{m_k^+}{m}1_{\alpha_k<\rho} + \frac{8c}{\rho}\alpha_k r_k$. This implies $\inf_{\alpha_k>0} B_k(\alpha_k) = \frac{m_k^-}{m} + \min\left(\frac{m_k^+}{m}, 8cr_k\right)$. However, we need to ensure the global

condition $\sum_{k=1}^{l} \alpha_k \leq 1$. First, let $l' = \min(|\mathcal{K}|, \frac{1}{\rho})$. For any $\mathcal{J} \subseteq \mathcal{K}$ with $|\mathcal{J}| \leq l'$, we choose $\alpha_k = \rho$ for $k \in \mathcal{J}$, $\alpha_k \to 0$ otherwise, which guarantees $\sum_{k=1}^{l} \alpha_k = \rho l' \leq 1$ and implies that $\inf_{\boldsymbol{\alpha} \in \mathrm{int}(\Delta)} B(\boldsymbol{\alpha})$ is equal to

$$\min_{\mathcal{J}} \Big(8c \sum_{k \in \mathcal{J}} r_k + \sum_{k \in \mathcal{K} - \mathcal{J}} \frac{m_k^+}{m} \Big) + \sum_{k=1}^{l} \frac{m_k^-}{m} + \sum_{k \notin \mathcal{K}} \frac{m_k^+}{m}.$$

Moreover, to simplify the bound, observe that

$$\begin{aligned}
&\min_{\mathcal{J}} \Big(8c \sum_{k \in \mathcal{J}} r_k + \sum_{k \in \mathcal{K} - \mathcal{J}} \frac{m_k^+}{m} \Big) \\
&= \min_{\mathcal{J}} \Big(8c \sum_{k \in \mathcal{J}} r_k + \sum_{k \in \mathcal{K} - \mathcal{J}} \frac{m_k^+}{m} + \sum_{k \in \mathcal{K} - \mathcal{J}} 8cr_k - \sum_{k \in \mathcal{K} - \mathcal{J}} 8cr_k \Big) \\
&= 8c \sum_{k \in \mathcal{K}} r_k + \min_{\mathcal{J}} \Big(\sum_{k \in \mathcal{K} - \mathcal{J}} \frac{m_k^+}{m} - 8cr_k \Big).
\end{aligned}$$

Also, by definition, we can write: $\sum_{k \in \mathcal{K}} 8cr_k + \sum_{k \notin \mathcal{K}} \frac{m_k^+}{m} = \sum_{k=1}^{l} \min\Big(8cr_k, \frac{m_k^+}{m}\Big)$. Now, define $\mathcal{L} = \mathcal{K} - \mathcal{J}$. Since $|\mathcal{J}| \leq l'$, $|\mathcal{L}| = |\mathcal{K}| - |\mathcal{J}| \geq |\mathcal{K}| - l' = |\mathcal{K}| - \min(|\mathcal{K}|, \frac{1}{\rho}) = \max(0, |\mathcal{K}| - \frac{1}{\rho})$, hence $|\mathcal{L}| \geq |\mathcal{K}| - \frac{1}{\rho}$. Finally, this allows us to write the bound in a simpler form:

$$\begin{aligned}
\inf_{\boldsymbol{\alpha} \in \mathrm{int}(\Delta)} A(\boldsymbol{\alpha}) &= \sum_{k=1}^{l} \min\Big(8cr_k, \frac{m_k^+}{m}\Big) \\
&\quad + \min_{\substack{\mathcal{L} \subseteq \mathcal{K} \\ |\mathcal{L}| \geq |\mathcal{K}| - \frac{1}{\rho}}} \Big(\sum_{k \in \mathcal{L}} \frac{m_k^+}{m} - 8cr_k \Big) + \sum_{k=1}^{l} \frac{m_k^-}{m}.
\end{aligned}$$

Since $\widehat{R}_S(f) = \sum_{k=1}^{l} \frac{m_k^-}{m}$, this coincides with the bound in the statement of the theorem. □

Let d_k be the VC-dimension of $\widetilde{\mathcal{H}}_k$. We can replace $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k)$ in the bound by an upper bound in terms of the VC-dimension (Mohri, Rostamizadeh, and Talwalkar 2012): $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k) \leq \sqrt{\frac{2d_k \log(m+1)}{m}}$. When m_k^+, the number of correctly classified points at each leaf k, is sufficiently small, the contribution of leaf k to the complexity term of the bound only depends on m_k^+. This is likely since it suffices that the following inequality holds: $m_k^+ \leq 8c\sqrt{2d_k m \log(m+1)}$. At a fundamental level, the crucial measure appears to be a balance of the Rademacher complexities and the fractions m_k^+/m. This bound directly applies to standard multi-class decision trees, which could lead to many different application in this setting. We will later apply it to the random composite trees.

The bound of Theorem 1 can be generalized to hold uniformly for all $\rho > 0$ at the price of an additional term in $O\Big(\frac{\log \log_2 \frac{1}{\rho}}{m}\Big)$ using standard techniques for margin bounds (see for example (Mohri, Rostamizadeh, and Talwalkar 2012)). Note that for $|\mathcal{K}| \leq \frac{1}{\rho}$, choosing $\mathcal{L} = \emptyset$ gives the following simpler bound:

$$\begin{aligned}
R(f) \leq \widehat{R}_S(f) + \sum_{k=1}^{l} \min\Big(8c\mathfrak{R}_m(\widetilde{\mathcal{H}}_k), \frac{m_k^+}{m}\Big) \\
+ \widetilde{O}\left(\frac{1}{\rho}\sqrt{\frac{\log pl}{m}}\right) + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}.
\end{aligned}$$

Thus, we can choose $\rho = \frac{1}{|\mathcal{K}|}$ at an additional price only in $O\Big(\frac{\log \log_2 |\mathcal{K}|}{m}\Big) \leq O\Big(\frac{\log \log_2 l}{m}\Big)$. This gives the simpler form (1) of the bound of Theorem 1, with $C(m, p, \rho) = C(m, p, \frac{1}{|\mathcal{K}|}) \leq C(m, p, \frac{1}{l})$.

The learning bounds just discussed are given in terms of the complexity terms $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k)$. To derive more explicit guarantees, we need to express them instead in terms of the Rademacher complexities $\mathfrak{R}_m(\widetilde{H}_k)$. The following result will provide us the tool to do so.

Lemma 1. *Let H_1 and H_2 be two families of functions mapping $\mathcal{X}$ to $\{0, 1\}$ and let $H = \{h_1 h_2 \colon h_1 \in H_1, h_2 \in H_2\}$. Then, the empirical Rademacher complexity of H for any sample S of size m can be bounded as follows:*

$$\mathfrak{R}_m(H) \leq \mathfrak{R}_m(H_1) + \mathfrak{R}_m(H_2).$$

Proof. Observe that for any $h_1 \in H_1$ and $h_2 \in H_2$, we can write $h_1 h_2 = (h_1 + h_2 - 1)1_{h_1 + h_2 - 1 \geq 0} = (h_1 + h_2 - 1)_+$. Since $x \mapsto (x-1)_+$ is 1-Lipschitz over $[0, 2]$, by Talagrand's contraction in (Ledoux and Talagrand 1991), the following holds: $\mathfrak{R}_m(H) \leq \mathfrak{R}_m(H_1 + H_2) \leq \mathfrak{R}_m(H_1) + \mathfrak{R}_m(H_2)$. □

We now assume, as previously discussed, that leaf selectors are defined via node questions $q_j \colon \mathcal{X} \to \{0, 1\}$, with $q_j \in Q_j$. Then, by Lemma 1, we can write $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k) \leq \Big(\sum_{j=1}^{d_k} \mathfrak{R}_m(\widetilde{Q}_j) + \mathfrak{R}_m(\widetilde{H}_k) \Big)$. If we use the same hypothesis at each node, $Q_j = Q$ for all j for some Q, then the bound simply becomes: $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k) \leq \big(d_k \mathfrak{R}_m(\widetilde{Q}) + \mathfrak{R}_m(\widetilde{H}_k)\big)$.

Generalization bounds for random composite trees

Consider composite trees with leaf selectors that are composed of node questions. Here, we assume that the node questions are defined as threshold functions based on a family of features F. As in RNO, a random composite tree is a composite tree where at each node n, the node question is based a random subset $F_n \subseteq F$ of size r. More precisely, the node question of a random composite tree is

$$q_n(x) = 1_{\Phi(x) \cdot \theta_n \leq 0} \; s.t. \; \theta_n = \operatorname*{argmax}_{\theta \in F_n} I_n \tag{4}$$

where Φ is the feature mapping and I_n is the information gain of node n. For simplicity, let $\eta = \frac{r}{|F|}$.

Let $\Pi_G(m)$ denote the growth function of a family of functions G. Then, the following learning bound for random composite trees can be derived using our analysis of composite trees, which inherently also provides a generalization guarantee for RNO.

Proposition 2. *Fix $\rho > 0$. Assume that for all $k \in [1, l]$, the functions in H_k take values in $\{0, 1\}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S of size $m \geq 1$, the following holds for all $l \geq 1$ and all random composite tree function f:*

$$R(f) \leq \widehat{R}_S(f) + \sum_{k=1}^{l} \min\left(8cD_{\mathcal{H}_k}, \frac{m_k^+}{m}\right)$$

$$+ \min_{\substack{\mathcal{L}\subseteq\mathcal{K} \\ |\mathcal{L}|\geq|\mathcal{K}|-\frac{1}{\rho}}} \sum_{k\in\mathcal{L}} \left(\frac{m_k^+}{m} - 8cD_{\mathcal{H}_k}\right) + C(m, p, \rho) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}},$$

where $D_{\mathcal{H}_k} = \sqrt{\frac{2\log\Pi_{\widetilde{H}_k}(m)}{m}} + \sqrt{\frac{2d_k(r\log\frac{e}{\eta}+\log 2mr)}{m}}$, and d_k is depth of leaf k for a given tree.

Proof. By Lemma 1, the following inequality holds: $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k) \leq \sum_{j=1}^{d_k} \mathfrak{R}_m(\widetilde{Q}_j) + \mathfrak{R}_m(\widetilde{H}_k)$. By Massart's lemma, the Rademacher complexities can be bounded as follows in terms of the growth function: $\sum_{j=1}^{d_k} \mathfrak{R}_m(\widetilde{Q}_j) + \mathfrak{R}_m(\widetilde{H}_k) \leq \sqrt{(2\log\Pi_{\widetilde{H}_k}(m))/m} + \sqrt{(2\sum_{j=1}^{d_k}\log\Pi_{\widetilde{Q}_j}(m))/m}$. Now, using the fact that there are $\binom{|F|}{r}$ ways of choosing r features out of $|F|$ and the upper bound $\binom{|F|}{r} \leq (\frac{|F|e}{r})^r$ for $1 \leq r \leq |F|$, we can write $\log\Pi_{\widetilde{Q}_j}(m) \leq \log\binom{|F|}{r}(2mr) \leq r\log\frac{e}{\eta} + \log(2mr)$, since there are $2m$ distinct threshold functions for each dimension with m points and there are r dimensions. Using this upper bound on $\mathfrak{R}_m(\widetilde{\mathcal{H}}_k)$ in Theorem 1 concludes the proof. □

As indicated before, the bound of this proposition can be generalized to hold uniformly for all $\rho > 0$ at the price of an additional term $O\left(\frac{\log\log_2\frac{1}{\rho}}{m}\right)$. For $|\mathcal{K}| \leq \frac{1}{\rho}$, choosing $\mathcal{L} = \emptyset$ gives the following simpler bound:

$$R(f) \leq \widehat{R}_S(f) + \sum_{k=1}^{l} \min\left(8cD_{\mathcal{H}_k}, \frac{m_k^+}{m}\right) + \widetilde{O}\left(\frac{1}{\rho}\sqrt{\frac{\log pl}{m}}\right). \tag{5}$$

Algorithm

In this section, we derive an algorithm, RANDOMCOMPOSITEFOREST, which is based on averaging an ensemble of random composite trees and which directly benefits from the bound (5). Thus, we define a *Random Composite Forest* function f as the uniform average of B composite tree functions f: $\mathsf{f}(x, y) = \frac{1}{B}\sum_{b=1}^{B} f_b(x, y)$. The label returned by f for each input point $x \in \mathcal{X}$ is given by $\operatorname{argmax}_{y\in\mathcal{Y}} \mathsf{f}(x, y)$.

Algorithm 1 gives the pseudocode of our algorithm. RCF generates several random composite trees independently and then returns the uniform average of the scoring functions defined by these trees. For any random composite tree, the previous section described how the node questions are chosen. Yet, the type of leaf classifiers still needs to be determined: let the leaf hypothesis sets H_k at leaf k be decision surfaces defined by a polynomial kernel. We choose the hypothesis

Algorithm 1 RANDOMCOMPOSITEFOREST(B, r, γ)

```
for b = 1 to B do
  for n = 1 to d(M) do
    q_n ← RNO(r)
  end for
  for (δ_k)_{1≤k≤l} ⊆ G do
    for k = 1 to l do
      h_k ← SVM(δ_k)
    end for
    f_b(·) = Σ_{k=1}^{l} Π_{j=1}^{d_k} q_j(·) h_k(·, y).
    F_b ← F_b ∪ {f_b}
  end for
    f_b^* ← argmin_{f∈F_b} R̂_S(f) + Σ_{k=1}^{l} min(8cγA, m_k^+/m)
end for
return f = (1/B) Σ_{b=1}^{B} f_b^*
```

h_k that minimizes a surrogate loss (hinge loss) of the empirical error by using the multi-class SVM algorithm via the one-versus-one technique on the sample points that reached leaf k. In the pseudocode, we denote by SVM(δ_k) the multi-class SVM algorithm at leaf k with polynomial kernels of degree δ_k. Only relatively short trees are formed since the depth M of the tree scales with the bound and since enough sample points need to reach each leaf to learn via SVM.

The first step of the algorithm is to learn the node questions q_n of a random composite tree via RNO for a given size, r, of the randomly chosen subset of features and it thereby forms a tree with l leaves. In the pseudocode, we denote this step by $q_n(\cdot) \leftarrow \text{RNO}(r)$ where $d(M)$ is the number of nodes for a tree with maximum depth M. Then, the algorithm generates p different sequences of degree values $(\delta_k)_{1\leq k\leq l} \subseteq \mathcal{G}$. For each sequence, the algorithm learns the leaf classifiers as described above and defines a new random composite tree function f_b. We denote by $H_{f_b,k}$ the hypothesis set at leaf k that served to define f_b and denote by $\mathcal{F}_b$ the set containing all f_bs. The algorithm then chooses the best $f_b \in \mathcal{F}_b$ that minimizes the generalization bound (5). More precisely, we first upper bound the growth function of the leaf classifier hypothesis set $H_{f,k}$ in terms of the VC-dimension of the hypothesis set: $\sqrt{(2\log\Pi_{\widetilde{H}_{f,k}}(m))/m} \leq \sqrt{(2\mathsf{d}_{f,k}\log(\frac{em}{\mathsf{d}_{f,k}}))/m}$, where $\mathsf{d}_{f,k}$ is the VC-dimension of $H_{f,k}$. Then, we rescale the complexity term by a parameter γ, which we will determine by cross-validation. For a given γ, the algorithm chooses f_b^*, the composite tree $f \in \mathcal{F}_b$ with the smallest value of the generalization bound:

$$R(f) \leq \widehat{R}_S(f) + \sum_{k=1}^{l} \min\left(8c\gamma A, \frac{m_k^+}{m}\right), \tag{6}$$

where $A = \sqrt{\frac{2d_k(r\log\frac{e}{\eta}+\log 2mr)}{m}} + \sqrt{\frac{2\mathsf{d}_{f,k}\log(\frac{em}{\mathsf{d}_{f,k}})}{m}}$. The algorithm then repeats the process above for each value of $b = 1, ..., B$ and returns the random composite forest function f that is the uniform average of f_b^*s: $\mathsf{f} = \frac{1}{B}\sum_{b=1}^{B} f_b^*$.

The step of using bound 6 is at the heart of our algorithm

Table 1: The table reports the average test error (%) and standard deviation for RCF, RFs, and RF-SVM algorithm. For each dataset, the table also indicates the sample size, the number of features and number of classes.

Dataset	Examples	Features	RFs Error	RCF Error	RF+SVM Error	Classes
vowel	528	10	5.28 ± 0.46	3.77 ± 0.45	4.5 ± 1.38	11
vehicle	846	18	28.4 ± 0.53	27.9 ± 1.10	31.2 ± 2.19	4
dna	2000	180	3.55 ± 0.10	3.30 ± 0.10	6.3 ± 1.79	3
pendigits	7494	16	1.25 ± 0.14	0.29 ± 0.03	0.31 ± 0.08	10
german	1000	24	26.6 ± 0.37	24.2 ± 0.60	23.0 ± 0.63	2
iris	150	4	13.3 ± 1.10	10.0 ± 1.50	12.0 ± 1.6	3
abalone	4177	8	75.3 ± 1.15	71.7 ± 0.66	75.1 ± 0.24	29
a2a	2265	123	18.1 ± 0.49	17.5 ± 0.18	19.47 ± 0.21	2
australian	690	14	19.4 ± 0.85	17.4 ± 0.98	17.4 ± 0.55	2
usps	2007	256	9.80 ± 0.30	7.48 ± 0.16	7.7 ± 0.15	10
sonar	208	60	21.9 ± 4.85	16.2 ± 3.80	28.09 ± 2.3	2

since it enables us to select the random composite tree that admits the best generalization guarantee. By generating different random composite trees and minimizing this bound, the algorithm is directly exploiting the key theoretical result of balancing the complexity of the families of predictors and the fraction of correctly classified points.

Experiments

This section reports the results of the experiments with the RCF algorithm. We tested RCF on eleven datasets from UCI's data repository: german, vehicle, vowel, dna, pendigits, iris, abalone, and a2a. Table 1 gives the sample size, the number of features, and the number of classes of each datasets. For each dataset, we randomly divided the data into training, validation, and test sets in order to run the RCF algorithm. We repeated the experiment five times where each time we used a different random partition of the set. We compare our results to the Random Forests algorithm that uses bagging and RNO. The trees of RFs were grown without pruning and the leaf classifiers were averaged over the labels of the training points that reached the leaf. We also tested a variant of RCF, named RF-SVM, which simply places classifiers generated by the SVM algorithm at the leaves without using the bound 6. We implemented RCF, RFs, and RF-SVM by using scikit-learn, (Pedregosa et al.).

For the SVM algorithm at the leaves of each composite tree, we allowed the set of polynomial degrees $\mathcal{G} = \{1, \ldots, 9\}$. The number of different sequences of degree values was $p = 10$. For each polynomial degree $\delta \in \mathcal{G}$, the regularization parameter $C_\delta \in \{10^i \colon i = -3, \ldots, 2\}$ of SVMs was selected via cross-validation and at each leaf k, it was simply scaled by $\sqrt{\frac{m_k}{m}} C_\delta$. The r parameter which determined the size of the subset of features was in the following range: $r \in \{1, \sqrt{|F|}/2, \sqrt{|F|}, 2\sqrt{|F|}, \ldots, |F|\}$. The γ parameter that rescales the bound was $\gamma \in \{10^i \colon i = -3, \ldots, 0\}$ and the maximum depth of each tree varied within $M \in \{2, \ldots, 8\}$. The number of trees averaged in the random forest function f was $B \in \{100, \ldots, 900\}$. For each of these parameter settings, we ran the experiments as described above and then picked the parameters with the smallest validation error and reported the test error of RCF in Table 1. The range of all the parameter values for the RF-SVM algorithm are the same as the RCF and the RFs were tested using the same range of values for r and B as in the RCF. The test errors of the parameters with the smallest validation error for RFs and RF-SVM are also given in Table 1.

The results of Table 1 show that RCF yields a significant improvement in accuracy compared with that of RFs. The results are statistically significant at the 0.5% level for all datasets except for a2a, sonar, and vehicle when using a one-sided paired t-test. The datasets a2a and sonar are significant at the 2.5% level and 5% level respectively while the dataset vehicle is not statistically significant. The fact that RCF substantially outperforms the RF-SVM algorithm on almost all the data sets directly shows both the usefulness and the effectiveness of the bound 6 in practice.

For the RCF algorithm, we have chosen to use SVMs with polynomial kernels, but we could instead employ different hypothesis sets for the leaf classifiers and replace SVMs with another algorithm. Moreover, there are several components that could be easily optimized over to further improve performance such as optimizing over the regularization parameter C_δ at each leaf and testing a wider range of values for all the parameters. Lastly, we would like to emphasize that RCF is one of several algorithms that could be derived from the generalization bounds since we exploited only the simpler form of Theorem 1 and since this simpler form could guide on its own the design of various algorithms.

Conclusion

We introduced a broad learning model, Composite Trees, structured as decision trees with leaf selectors and node questions that can be chosen from complex hypothesis sets. For multi-class classification, we derived novel theoretical guarantees which suggest that generalization depends on a balance between the Rademacher complexities of the subfamilies of classifiers and the fraction of sample points correctly classified at each leaf. We further derived learning guarantees for Random Composite Trees which we used to devise and implement a new algorithm, RCF. The algorithm benefits from the theoretical guarantees derived since it performs better than RF-SVM and it significantly outperforms RFs in experiments with several datasets.

Acknowledgments This work was partly funded by the NSF awards IIS-1117591, CCF-1535987 and DGE 1342536.

References

Amit, Y., and Geman, D. 1997. Shape quantization and recognition with randomized trees. In *Neural Computation*.

Arreola, K.; Fehr, J.; and Burkhardt, H. 2006. Fast support vector machine classification using linear SVMs. In *ICPR*.

Arreola, K.; Fehr, J.; and Burkhardt, H. 2007. Fast support vector machine classification of very large datasets. In *GfKl Conference*.

Bennet, K., and Blue, J. 1998. A support vector machine approach to decision trees. In *IJCNN*.

Biau, G.; Devroye, L.; and Lugosi, G. 2008. Consistency of random forests and other averaging classifiers. *JMLR*.

Biau, G. 2012. Analysis of a random forests model. *JMLR*.

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24(2):123–140.

Breiman, L. 2001. Random forests. In *Mach. Learn.*

Chang, F.; Guo, C.-Y.; Lin, X.-R.; and Lu, C.-J. 2010. Tree decomposition for large-scale SVM problems. *JMLR*.

Criminisi, A.; Shotton, J.; and Konukoglu, E. 2012. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Found. Trends. Comput. Graph. Vis.*

Denil, M.; Matheson, D.; and de Freitas, N. 2014. Narrowing the gap: Random forests in theory and in practice. In *ICML*.

Dietterich, T.; Shotton, J.; Winn, J.; Iglesias, J. E.; Metaxas, D.; and Criminisi, A. 2000. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. In *Machine Learning*.

Dong, G., and Chen, J. 2008. Study on support vector machine based decision tree and application. In *ICNC-FSKD*.

Frohlich, B.; Rodner, E.; Kemmler, M.; and Denzler, J. 2013. Large-scale gaussian process multi-class classification for semantic segmentation and facade recognition. In *Machine Vision and Applications*.

Golea, M.; Bartlett, P. L.; Lee, W.; and Mason, L. 1997. Generalization in decision trees and DNF: Does size matter? In *NIPS*.

Ho, T. 1995. Random decision forest. In *International Conference on Document Analysis and Recognition*.

Ho, T. 1998. The random subspace method for constructing decision forests. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Kohavi, R. 1996. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *KDD*.

Kumar, A., and Gopal, M. 2010. A hybrid SVM based decision tree. *JPR*.

Kuznetsov, V.; Mohri, M.; and Syed, U. 2014. Multi-class deep boosting. In *NIPS*.

Ledoux, M., and Talagrand, M. 1991. *Probability in Banach Spaces: Isoperimetry and Processes*. New York: Springer.

Madjarov, G., and Gjorgjevikj, D. 2012. Hybrid decision tree architecture utilizing local SVMs for multi-label classification. In *HAIS*.

Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2012. *Foundations of Machine Learning*. The MIT Press.

Montillo, A.; Shotton, J.; Winn, J.; Iglesias, J. E.; Metaxas, D.; and Criminisi, A. 2011. Entangled decision forests and their application for semantic segmentation of ct images. In *Medical Imaging*.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *JMLR*.

Rodriguez-Lujan, I.; Cruz, C. S.; and Huerta, R. 2012. Hierarchical linear support vector machine. *JPR*.

Schroff, F.; Criminisi, A.; and Zisserman, A. 2008. Object class segmentation using random forests. In *British Machine Vision Conference*.

Seewald, A.; Petrak, J.; and Widmer, G. 2001. Hybrid decision tree learner with alternative leaf classifiers: an empirical study. In *FLAIRS*.

Shotton, J.; Fitzgibbon, A.; Cook, M.; Sharp, T.; Finocchio, M.; Moore, R.; Kipman, A.; and Blake, A. 2011. Real-time human pose recognition in parts from single depth images. In *IEEE Computer Vision and Pattern Recognition*.

Takahashi, F., and Abe, S. 2002. Decision tree based multiclass support vector machines. In *ICONIP*.

Wang, J., and Saligrama, V. 2012. Local supervised learning through space partitioning. In *NIPS*.

Xiong, C.; Johnson, D.; Xu, R.; and Corso, J. 2012. Random forests for metric learning with implicit pairwise position dependence. In *ACM SIGKDD*.

Zhou, Z.-H., and Chen, Z.-Q. 2002. Hybrid decision trees. In *Knowledge-Based Systems*.