Uniform Approximation of Functions with Random Bases

Ali Rahimi[†] and Benjamin Recht[♯]

† Intel Research Berkeley, Berkeley, CA
ali.rahimi@intel.com

♯ Center for the Mathematics of Information, Caltech, Pasadena, CA
brecht@ist.caltech.edu

***Abstract*— Random networks of nonlinear functions have a long history of empirical success in function fitting but few theoretical guarantees. In this paper, using techniques from probability on Banach Spaces, we analyze a specific architecture of random nonlinearities, provide L_∞ and L_2 error bounds for approximating functions in Reproducing Kernel Hilbert Spaces, and discuss scenarios when these expansions are dense in the continuous functions. We discuss connections between these random nonlinear networks and popular machine learning algorithms and show experimentally that these networks provide competitive performance at far lower computational cost on large-scale pattern recognition tasks.**

I. Introduction

Conventional wisdom in the earliest days of artificial intelligence held that randomly connected "associator units" that computed random binary functions of their inputs were sufficient for a great many pattern recognition tasks [1], but these randomized networks have largely been superseded by deterministic architectures trained by back propagation (such as multi-layer neural networks), convex optimization (such as Support Vector Machines) or by greedy algorithms (such as Adaboost). Recently, largely motivated by the fact that randomization is computationally cheaper than optimization, architectures based on random nonlinearities have been regaining popularity in the machine learning community for large scale data analysis [2], [3], [4], [5]. There is ample evidence that these randomized function fitting algorithms are fast and accurate, but very few theoretical guarantees are available.

In this paper, we analyze the ability of certain randomized function classes to approximate other well-studied classes of functions. Using tools from probability theory on Banach spaces, we show that with high probability, a fixed target function in a Reproducing Kernel Hilbert Space can be approximated well in the L_∞ and the L_2 sense as a linear combination of a few randomly chosen basis functions. For the class of functions we consider, the approximation rate turns out to be the same as that obtain by choosing the basis optimally. This result implies that learning architectures that fit data sets with a linear combination of randomly selected basis functions can approximate a variety of canonical learning algorithms that select the basis functions by costly optimization procedures. We also provide empirical evidence that randomizing instead of optimizing over the choice of the bases dramatically decreases the computational effort needed for practical learning tasks while still producing accurate input-output mappings.

II. Related Work

Seminal work by Barron and Jones provided a large class of functions that could be approximated to high accuracy with finite sums of K tunable basis functions:

$$\hat{f}(x) = \sum_{k=1}^{K} c_k \, \phi(w_k' x + b_k) \,. \tag{1}$$

Here, ϕ is a nonlinear function, and w_k and b_k are the parameters and c_k are the weights of $\hat{f}$. When ϕ is the cosine function, Jones showed that functions with absolutely integrable Fourier transforms could be approximated with L_2 error below $O(1/\sqrt{K})$ using only K terms [6]. When ϕ is sigmoidal, Barron showed that any function whose derivative was an absolutely integrable in the Fourier domain could also be approximated with L_2 error below $O(1/\sqrt{K})$ using only K terms [7]. Barron also investigated approximation by sigmoids in the L_∞ norm over a compact set [8], and this result was tightened and generalized to other nonlinearities by quantifying their VC dimension [9] or their Rademacher complexity [10].

The proofs of these approximation results are typically either existential—relying, for example, on random sampling of the weights c_k from an unknown distribution—or are constructive and achieved via a greedy algorithm. The constructive proofs are particularly interesting because they imply specific algorithms for fitting functions to data sets by simply setting the target function to be the empirical distribution of the data. In fact, the greedy constructive proofs of these results resemble popular greedy data fitting algorithms such as Adaboost [11] and Matching Pursuit [12]. These constructive proofs build up $\hat{f}$ stagewise by greedily adding at each stage a term $c_k \phi(w_k' x + b_k)$ to $\hat{f}$ to reduce the discrepancy between $\hat{f}$ and the target function. In a data fitting setting, when the dataset is large, computing w_k and b_k at each stage involves a computationally expensive optimization procedure. The approximation bounds presented here are constructive yet probabilistic: the parameters w_k and b_k are selected via a simple sampling procedure and the weights c_k are then the minimizers of a convex cost function.

A number of popular data fitting algorithms such as the SVM and Kriging search over an infinite-dimensional

Reproducing Kernel Hilbert Space (RKHS) of functions [13], [14]. These Hilbert spaces are making them particularly attractive in data fitting algorithms because they are dense in the space of square-integrable functions [15], but still admit a finite representation that grows only linearly with the number of data points to fit [16]. We show that weighted sums of random functions are dense in an RKHS defined by the choice of basis functions and sampling distribution. When the number of data points is large, weighted sums of random functions provide a more parsimonious functional representation that is much faster to compute than fitting the RKHS representation.

This work builds on two of our previous papers [4], [17] by providing L_∞ approximation error bounds, and additional results about the denseness of random function spaces in their corresponding RKHSs.

III. Approximation with Random Features

Let $\{\phi(\cdot;\theta) \;:\; w \in \Theta\}$ be a family of functions on a compact set $X \subset \mathbb{R}^d$ parameterized over the set Θ. For example, we will consider $\phi(x;\theta) = \cos(w'x+b)$ with $\theta = (w,b)$ and $\Theta = \mathbb{R}^d \times [-\pi,\pi]$. We are interested in approximating mixtures of the form $f(x) = \int_\Theta \alpha(\theta)\phi(x;\theta)\,d\theta$ by a finite sum of the form (1). For these mixtures, define a norm $\|f\|_p := \sup_\theta \left|\frac{\alpha(\theta)}{p(\theta)}\right|$ where p is a fixed probability distribution on Θ. We also define the set of mixture of ϕ with finite $\|\cdot\|_p$ norm:

$$\mathscr{F}(X,\Theta,\phi,p)=\left\{f(x)=\int_\Theta \alpha(\theta)\phi(x;\theta)d\theta \,\middle|\, \|f\|_p<\infty\right\}.$$

To simplify the presentation, unless there is confusion, we will denote $\mathscr{F}(X,\Theta,\phi,p)$ by $\mathscr{F}$. Furthermore, by linearity, we may assume throughout that $|\phi(x;\theta)| \le 1$ for all x and θ.

The following two theorems show that a given $f \in \mathscr{F}$ can be approximated to resolution $O(\|f\|_p/\sqrt{K})$ by a function of the form

$$\hat{f}(x) = \sum_{k=1}^{K} c_k \phi(x;\theta_k) \tag{2}$$

where $\theta_1,\dots,\theta_K$ are sampled iid from $p(\theta)$.

The first theorem concerns the L_2 approximation error considered in [7], [6], [15] and applies to arbitrary basis functions ϕ and distributions p.

Theorem 3.1: Let μ be any probability measure on X, and define the norm $\|f\|_\mu^2 = \int_X f^2(x)\,\mu(dx)$. Suppose ϕ satisfies $\sup_{x,\theta}|\phi(x;\theta)| \le 1$. Fix $f \in \mathscr{F}$. Then for any $\delta > 0$, with probability at least $1-\delta$ over $\theta_1,\dots,\theta_K$ drawn iid from p, there exist $c_1,\dots,c_K$ so that the function

$$\hat{f}(x) = \sum_{k=1}^{K} c_k \phi(x;\theta_k) \tag{3}$$

satisfies

$$\|\hat{f}-f\|_\mu < \frac{\|f\|_p}{\sqrt{K}}\left(1+\sqrt{2\log\tfrac{1}{\delta}}\right). \tag{4}$$

In contrast to the approximation bounds reviewed in Section II, which use the probabilistic method to show the *existence* of θ_k's that yield a good $\hat{f}$, the random sampling employed here actually *produces* parameters θ_k which yield a good $\hat{f}$ with very high probability. When μ is the empirical measure over some data set, this theorem gives a bound on the distortion incurred by fitting the data with random basis instead of a function in $\mathscr{F}$. The proof appears in [17].

One can obtain much stronger approximation guarantees if the nonlinear function ϕ is well-behaved. The uniform pointwise error in the approximation of f by $\hat{f}$ can be bounded if ϕ is of the form $\phi(x;\theta) = \phi(\theta'x)$ with $\phi : \mathbb{R} \to \mathbb{R}$ Lipshitz.

Theorem 3.2: Let $\phi(x;\theta) = \phi(\theta'x)$, with $\phi : \mathbb{R} \to \mathbb{R}$ L-Lipschitz, $\phi(0) = 0$, and $|\phi| < 1$. Suppose furthermore that p has a finite second moment. Fix $f \in \mathscr{F}$. Then for any $\delta > 0$, with probability at least $1-\delta$ over $\theta_1,\dots,\theta_K$ drawn iid from p there exist $c_1,\dots,c_K$ so that the function

$$\hat{f}(x) = \sum_{k=1}^{K} c_k \phi(\theta_k' x) \tag{5}$$

satisfies

$$\|\hat{f}-f\|_\infty < \frac{\|f\|_p}{\sqrt{K}}\left(\sqrt{\log\tfrac{1}{\delta}}+4LB\sqrt{\mathbb{E}\theta'\theta}\right), \tag{6}$$

where $B = \sup_{x\in X}\|x\|_2$.

As above, $\hat{f}$ converges to the target f as $O(1/\sqrt{K})$. The rate depends on δ only logarithmically. It depends more strongly on the norm of the target function and the variance of the sampling distribution p. The proof, which appears in the appendix, is similar to that of Theorem 3.1, and additionally borrows the notion of the Rademacher complexity of ϕ from estimation error bounds in statistical learning theory [18]. Indeed, a similar statement to Theorem 3.2 can be obtained for any function class $\phi(x;\theta)$ with low Rademacher Complexity.

IV. Relationship to Reproducing Kernel Hilbert Spaces

While the function class $\mathscr{F}$ is well approximated by a random set of bases, it appears at first glance that this a rather small set of functions. However this class of functions is dense in a Reproducing Kernel Hilbert Spaces (RKHS) defined by ϕ and p, implying that it is quite rich. RKHSs are commonly used in machine learning to represent complicated functions in a non-parametric way because they are dense in the set of continuous functions. But algorithms for fitting functions in an RKHS to data have superlinear complexity in the number of data points. When the number of data points is large, randomized nonlinear expansions provide a compact and computationally efficient alternative to the RKHS representations.

For a given function $\phi(x;\theta) : X \times \Theta \to \mathbb{R}$ and probability distribution $p(\theta)$ on Θ, we can define the corresponding kernel k on $X \times X$ as

$$k(x,y) = \int_\Theta p(w)\phi(x;\theta)\phi(y;\theta)d\theta. \tag{7}$$

This is clearly positive definite as for any $x_1,\dots,x_m \in X$, the matrix K with entries $K_{ij} = k(x_i,x_j)$ is an integral of rank-one outer product matrices $Z_\theta = [\phi(x_i;\theta)\phi(x_j;\theta)]$. The RKHS defined by the kernel k, denoted $\mathscr{H}$, is the completion of the set of all finite linear combinations of the form

$$f(x) = \sum_t a_t k(x, x_t), \qquad x_t \in X, \tag{8}$$

with the inner product that satisfies $\langle k(\cdot, x_t), k(\cdot, x_s)\rangle = k(x_t, x_s)$.

The following proposition introduces an alternative representation of this Reproducing Kernel Hilbert Space using standard definitions and results [19]:

Proposition 4.1: Let X, Θ, ϕ, p, and $\mathscr{H}$ be as above and let the space $\hat{\mathscr{H}}$ be the completion of the set of all functions of the form $f(x) = \int_\Theta \alpha(\theta)\phi(x;\theta)\, d\theta$ such that

$$\int_\theta \frac{\alpha(\theta)^2}{p(\theta)} d\theta < \infty, \tag{9}$$

with the inner product

$$\langle f, g\rangle = \int_\theta \frac{\alpha(\theta)\beta(\theta)}{p(\theta)} d\theta, \tag{10}$$

where $g(x) = \int_\Theta \beta(\theta)\phi(x;\theta)\, d\theta$. Then $\hat{\mathscr{H}} = \mathscr{H}$.

This result follows immediately from Theorem 2 in §III.3 of [20] which is attributed to Aronszajn: to every positive definite kernel, $k(x,y)$ there corresponds one and only one Hilbert space with $k(x,y)$ as a reproducing kernel. For completeness, we provide a proof in the Appendix. Using this proposition, we see that $\mathscr{F}$ is a subset of the RKHS $\mathscr{H}$: For any $f \in \mathscr{F}$, by Hölder's inequality,

$$\int_\theta \frac{\alpha(\theta)^2}{p(\theta)} d\theta = \int_\theta \frac{\alpha(\theta)^2}{p(\theta)^2} p(\theta) d\theta \leq \|f\|_p^2, \tag{11}$$

which, by the definition of $\mathscr{F}$, is finite. In fact, $\mathscr{F}$ is a *dense* subset of $\mathscr{H}$:

Theorem 4.2: Let $\mathscr{F}$ and $\mathscr{H}$ be defined as above for a given function $\phi(x;\theta)$ and probability distribution $p(\theta)$. Then $\mathscr{F}$ is dense in $\mathscr{H}$.

This implies that whenever $\mathscr{H}$ is dense in the space of continuous functions, $\mathscr{F}$ is also dense in the space of continuous functions. To prove the theorem, observe that functions of the form of (8) are dense in $\mathscr{H}$. But these functions can also be written in the form $\int_\Theta \alpha(\theta)\phi(x;\theta)\, d\theta$ via the identification $\alpha(\theta) := p(\theta)\sum_t a_t \phi(x_t;\theta)$. Since $|\alpha(\theta)| \leq p(\theta)\sum_t |a_t|$, and since a_t are finite, $|\alpha(\theta)|/p(\theta)$ is also finite, implying that f is in $\mathscr{F}$

The denseness of $\mathscr{F}$ in $\mathscr{H}$ implies that in a data fitting setting, one has the luxury of choosing whether to fit a function in $\mathscr{H}$ or in $\mathscr{F}$. The Representer Theorem [16] guarantees that for a large number of function fitting problems, the optimal $f \in \mathscr{H}$ takes the finite form $f(x) = \sum_{i=1}^N a_i k(x, x_i)$ where $x_1, \ldots, x_N$ are the examples provided for the fit. If N (the number of data points) is smaller than K (the number of random basis functions needed for a desired quality-of-fit) one can directly compute the kernel k corresponding to p and ϕ via (7) and fit the N parameters a_i. On the other hand, when N is very large, optimizing over the a_i is expensive, and fitting the K parameters c_i of a random expansion of the form (2) is much faster. Moreover, even if the Representer Theorem does not apply, such as when the optimization problem involves derivatives of the function to be fit, sums of random bases provide excellent approximations to the true optimum.

V. Examples

This section provides several examples of random bases and their corresponding sampling distributions. In each example, we describe the RKHS that is being approximated by $\mathscr{F}$ and examine the relation to existing supervised learning techniques. The discussion is summarized in Figure 1.

a) Random Fourier Bases: Sinusoidal nonlinearities of the form $\phi(x,\theta) = \cos(\omega' x + b)$ with $\theta = (w, b)$ and $\Theta = \mathbb{R}^d \times [-\pi, \pi]$ are 1-Lipschitz and satisfy the assumptions of Theorem 3.2. These features project their input onto a randomly chosen line, and then pass the resulting scalar through a sinusoid.

When b is drawn from a distribution on $[-\pi, \pi]$ that is symmetric about 0, the corresponding kernel according to (7) is

$$\begin{aligned} k(x,y) &= \int\int_0^{2\pi} p(b)p(\omega)\cos(\omega' x + b)\cos(\omega' y + b)\, db d\omega \\ &= \frac{1}{2}\int p(\omega)\cos(\omega'(x-y))\, d\omega. \end{aligned} \tag{12}$$

This kernel is shift invariant, meaning $k(x,y) = k(x-y)$.

In fact, any shift invariant kernel can be represented using random cosine features. Given a shift invariant kernel, the corresponding ϕ and p can be recovered by simply letting $\phi(x,\theta) = \cos(\omega' x + b)$, and setting $p(w)$ to the inverse Fourier transform of k, and $p(b)$ to the uniform distribution on $[-\pi, \pi]$. We explored this result, which is based on Bochner's theorem [21], in [4].

This observation is of practical importance because certain RKHSs are known to work well in many data fitting applications. Approximating these RKHSs with a small number of random features enables us to fit huge datasets cheaply, by avoiding the machinery of kernel machines. For example, to approximate the RKHS induced by the Gaussian kernel, $k(x,y) = \exp(-\gamma\|x-y\|_2^2)$, it suffices to sample w from the inverse Fourier transform of k, which is just a Gaussian with mean 0 and covariance $2\gamma I$.

b) Random Stumps: Decision stumps are sigmoidal basis functions commonly used with the Adaboost algorithm. They have the form $\phi(x;\theta) = \mathrm{sgn}(x_i - t)$, with $\theta = (t, i)$, where the threshold t is a real number in some interval and $i \in [1 \ldots d]$ is some integer that indexes a component of $x \in \mathbb{R}^d$. To alleviate the computational cost of the greedy algorithm employed in Adaboost to fit θ, one can select i and t randomly.

To identify the RKHS that corresponds to this sampling distribution, consider first the one-dimensional example where X is contained in the interval $[a,b]$. Suppose t is selected uniformly at random from $[a,b]$. Then, for $a \leq x < y \leq b$,

$$\frac{1}{b-a}\int_a^b \mathrm{sgn}(x-t)\,\mathrm{sgn}(y-t)\, dt = 1 - 2\frac{y-x}{b-a}, \tag{13}$$

	$\phi(x;\theta)$	θ	$p(\theta)$	$k(x,y)$
Fourier	$\cos(\omega' x + b)$	$(\omega, b) \in \mathbb{R}^{d+1}$	$b \sim \text{unif}[-\pi,\pi]$ $p(\omega) \sim \mathcal{N}(0, 2\gamma I)$	$\exp(-\gamma\|x-y\|^2)$
Stump	$\text{sgn}(x_k - t)$	$t \in \mathbb{R}$, $k \in \{1,\dots,d\}$	k uniform $t \ \sim \text{unif}[-a,a]$	$1 - \frac{1}{a}\|x-y\|_1$
Bins	Indicator vector of x in random grid	$\delta \in [0,\infty)^d$, $u \in [0,\delta]$	$\prod_{k=1}^d \gamma^2 \delta_k \exp(-\gamma\delta_k)$	$\exp(-\gamma\|x-y\|_1)$

Fig. 1. Examples of Random Features, their sampling distributions, and their corresponding kernels.

so the induced kernel is $k(x,y) = 1 - 2\frac{|x-y|}{b-a}$. In higher dimensions, when t is sampled in the interval $[-a,a]$ and the components are selected uniformly at random, assuming that X is contained in a hypercube $[-a,a]^d$, the kernel becomes $k(x,y) = 1 - \frac{1}{a}\|x-y\|_1$.

c) Random Bins: Binning basis functions partition the input space using an axis-aligned grid, and assign a binary indicator to each partition. Such partitionings are known to be universal approximators and are sometimes referred to as axis-aligned "arrangements." [22] Computing optimal values for the parameters of these grids is difficult. Here we summarize why randomly selecting the pitch and shift of the grids results in arrangements that can approximate complex target functions with little computational effort. For more details, see [4].

Unlike the previous examples, this set of bases is vector-valued, with $\phi(x;\theta)$ mapping X to a bit string in $\{0,1\}^N$. We first describe ϕ for $X \subset [-a,a]$ and then extend to the multidimensional setting. For a given grid pitch δ, consider the grid shift $u \in [0,\delta]$ and a binning function $\phi(x,u)$ that returns an bit vector that indicates x's bin by mapping $x \in \mathbb{R}$ into a binary bit string $\{0,1\}^{\lceil 2a/\delta \rceil}$ with the nth bit set if x falls in the interval $[u + n\delta, u + (n+1)\delta]$ and 0 otherwise. Thus $\phi'(x,u)\phi(y,u)$ is 1 if x and y fall in the same bin, and zero otherwise. When the grid shift u is drawn uniformly at random from the interval $[0,\delta]$, the corresponding kernel can be shown to be

$$k_{\text{hat}}(x,y;\delta) = \frac{1}{\delta}\int \phi(x;u)'\phi(y;u)du \tag{14}$$

$$= \max\left(0, 1 - \frac{|x-y|}{\delta}\right). \tag{15}$$

Other one-dimensional kernels can be obtained as mixtures of this kernel. Draw the pitch δ from some distribution $p(\delta)$ and sample u uniformly from $[0,\delta]$. In this case, the parameter set is $\theta = (\delta, u)$, with $\delta \in [0,\infty)$, and $u \in [0,\delta]$, and the resulting kernel is given by $k(x,y) = \int_0^\infty k_{\text{hat}}(x,y;\delta)p(\delta)\ d\delta$. For example, when $p(\delta)$ is the Gamma distribution $\delta\exp(-\delta)$, the resulting kernel is the Laplacian kernel, $k_{\text{Laplacian}}(x,y) = \exp(-|x-y|)$. The RKHS associated with this kernel is also dense in the set of continuous functions.

To represent multi-dimensional kernels of the form $\prod_{m=1}^d k(|x^m - y^m|)$, the binning process is applied over each dimension of $X \subset \mathbb{R}^d$ independently. The probability that the mth components of $x, y \in X$ are binned together in dimension m by the above process is $k(|x^m - y^m|)$, where x^m and y^m are the mth components of x and y respectively. Since the binning process is independent across dimensions, the probability that x and y are binned together in every dimension is $\prod_{m=1}^d k(|x^m - y^m|)$. In the case of the Laplacian kernel, the resulting d-dimensional kernel is $\exp(-\|x-y\|_1)$.

In this multivariate case, $\phi(x;\theta)$ encodes the integer vector $[\hat{x}^1, \cdots, \hat{x}^d]$ corresponding to each bin of the d-dimensional grid as a binary indicator vector. In data fitting applications, the total number of occupied bins is at most the number of data points, so unused bins can be eliminated from the representation. With simple data structures, function expansions of this sort can be computed and stored efficiently [4].

VI. Numerical Experiments

To show that the space $\mathscr{F}$ is rich enough for typical data fitting problems, Table I summarizes the results of some experiments. The table also provides wall clock times to showcase the speed of randomized fitting. These experiments compare the results of fitting data with functions in $\mathscr{F}$ via least squares and of fitting data with the corresponding RKHS $\mathscr{H}$ using state-of-the-art algorithms for kernel machines. The experiments were conducted on five standard large-scale datasets from the UCI machine learning repository [23]. The results in the literature pertaining to the SVM solvers SVM$^{\text{light}}$ and libSVM were replicated using binaries provided by the respective authors. For the random feature experiments, regressors and classifiers were trained by solving the ridge regression problem $\min_w \|\Phi' w - y\|_2^2$, where y denotes the vector of desired outputs and Φ denotes the matrix of random bases evaluated on the training data. To evaluate the resulting function on a data point x, it suffices to compute $w'\phi(x;\theta)$. Despite its simplicity, regression with random bases is faster than, and provides competitive accuracy with, alternative methods. Random Fourier bases perform better on the tasks that largely rely on interpolation. On the other hand, Random Bins perform better on those for which the standard SVM requires many support vectors, because they explicitly preserve locality in the input space. This difference is most dramatic in the `Forest` dataset. Figure 2 shows that good performance can be obtained even from a modest number of bases.

Figure 3 compares regression with Random Stumps against Adaboost on the UCI `adult` dataset. Since is much faster than Adaboost, we could afford to run Random Stumps for larger K than Adaboost. These additional runs are included in the plots. Adaboost expends considerable effort

Dataset	Fourier+LS	Binning+LS	Exact SVM
CPU regression 6500 instances 21 dims	3.6% 20 secs $D = 300$	5.3% 3 mins $P = 350$	11% 31 secs ASVM
Census regression 18,000 instances 119 dims	5% 36 secs $D = 500$	7.5% 19 mins $P = 30$	9% 13 mins SVMTorch
Adult classification 32,000 instances 123 dims	14.9% 9 secs $D = 500$	15.3% 1.5 mins $P = 30$	15.1% 7 mins SVM$^{\text{light}}$
Forest Cover classification 522,000 instances 54 dims	11.6% 71 mins $D = 5000$	2.2% 25 mins $P = 50$	2.2% 44 hrs libSVM

TABLE I

Comparison of testing error and training time between ridge regression with random bases and various state-of-the-art exact kernel methods reported in the literature. For classification tasks, the percent of testing points incorrectly predicted is reported, and for regression tasks, the RMS error normalized by the norm of the ground truth.

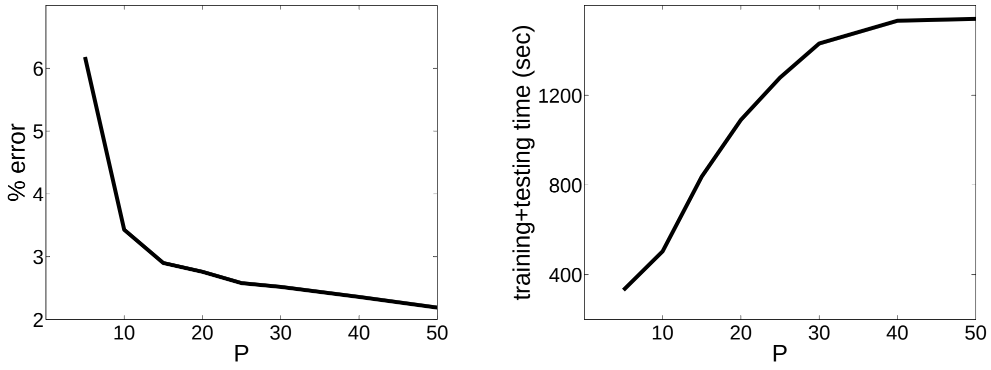

Fig. 2. On the Forest dataset, using random binning, error decays quickly as the number of random bases grows (left). Training time grows slowly as the number of random bases grows (right).

in choosing its decision stumps and obtains low test error using only a few of them.

Random Stumps achieve similar accuracy as Adaboost in orders of magnitude less time. It requires many more stumps than Adaboost because it chooses them randomly. But because it is fast, the function can be learned much more quickly with Random Stumps.

Appendix

a) Proof of Theorem 3.2: In what follows, for convenience, we use the notation $\|f\|_\infty$ and $\|f(x)\|_\infty$ interchangeably to denote $\sup_{x\in X}|f(x)|$. Construct $\hat{f}$ as defined in Equation (5) with $c_k \equiv \frac{\alpha(\theta_k)}{Kp(\theta_k)}$ and the random variable

$$v(\theta_1,\dots,\theta_K) = \|\hat{f} - f\|_\infty. \tag{16}$$

We bound the deviation of v from its expectation using McDiarmid's inequality.

First, observe that v is stable under perturbation of any one of its arguments. Indeed, for any $\theta_1,\dots,\theta_K$ and $\tilde{\theta}_k$, by the triangle inequality and the boundedness of ϕ, we have

$$|v(\theta_1,\dots,\theta_K) - v(\theta_1,\dots,\tilde{\theta}_k,\dots,\theta_K)| \tag{17}$$

$$\le \frac{1}{K}\left\|\frac{\alpha(\theta_k)}{p(\theta_k)}\phi(\theta_k'x) - \frac{\alpha(\tilde{\theta}_k)}{p(\tilde{\theta}_k)}\phi(\tilde{\theta}_k'x)\right\|_\infty \le \frac{2\|f\|_p}{K}. \tag{18}$$

Call this quantity Δ.

Next, bound the expectation of v. The choice of $c_1,\dots,c_K$ ensures that $\mathbb{E}_\theta \hat{f} = f$. By a standard argument [18],

$$\mathbb{E}v = \mathbb{E}\sup_{x\in X}|\hat{f}(x) - \mathbb{E}\hat{f}(x)| \tag{19}$$

$$\le \tfrac{2}{K}\mathbb{E}_{\theta,\varepsilon}\sup_{x\in X}\left|\sum_{k=1}^{K}\varepsilon_k\, c_k\phi(\theta_k'x)\right|, \tag{20}$$

where $\varepsilon_1,\dots,\varepsilon_K$ is a sequence of Rademacher random variables.

Since the function $c_k\phi(\cdot)$ is $L\|f\|_p$-Lipschitz in its scalar argument and $c_k\phi(0) = 0$, by Theorem 4.12 of [24] (replicated in Theorem 12(4) of [18]), Cauchy-Schwartz, and

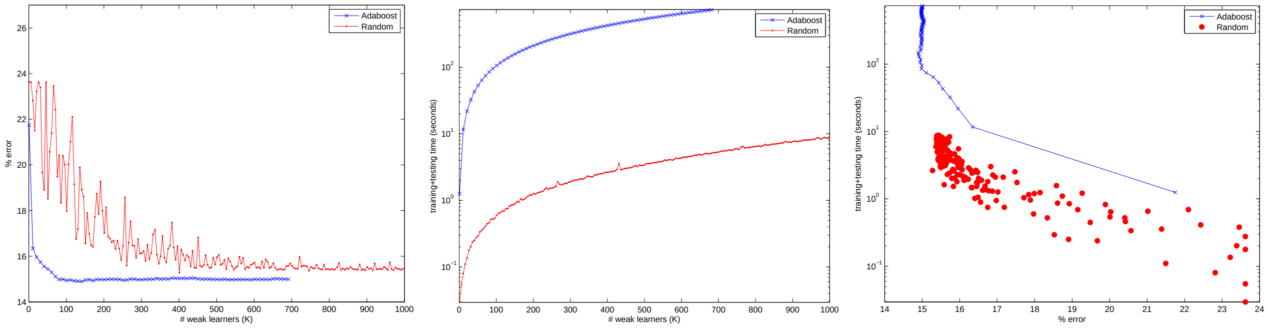

Fig. 3. Comparisons between Random Stumps and Adaboosted decision stumps on the `adult` dataset. The leftmost figure plots test error of each classifier as a function of K. The accuracy of Random Stumps catches up to that of Adaboost as K grows. The second column plots the total training and testing time as a function of K. The third column combines the previous two columns. It plots testing+training time required to achieve a desired error rate. For a given error rate, Random Stumps are an order of magnitude faster than Adaboost.

Jensen's inequality,

$$\begin{aligned}\mathbb{E}\nu &\leq \tfrac{2}{K}\mathbb{E}\sup_{x\in X}\left|\sum_{k=1}^{K}\varepsilon_k\, c_k\phi(\theta_k' x)\right| &(21)\\ &\leq \tfrac{4L\|f\|_p}{K}\mathbb{E}\sup_{x\in X}\left|\sum_{k=1}^{K}\varepsilon_k\ \theta_k' x\right| &(22)\\ &\leq \tfrac{4L\|f\|_pB}{K}\mathbb{E}\left\|\sum_{k=1}^{K}\varepsilon_k\ \theta_k\right\|_2 &(23)\\ &\leq \frac{4L\|f\|_pB}{\sqrt{K}}\sqrt{\mathbb{E}\,\|\theta_1\|_2^2}. &(24)\end{aligned}$$

Call this quantity μ.

Using McDiarmid's concentration inequality, we now have

$$\Pr\left[\nu \geq \mu+\varepsilon\right] \leq \Pr\left[\nu \geq \mathbb{E}\nu+\varepsilon\right] \leq \exp\left(-\frac{2\varepsilon^2}{K\Delta^2}\right). \qquad (25)$$

Setting the right hand side to δ and solving for ε yields the theorem.

b) Proof of Proposition 4.1: By Theorem 2 in §III.3 of [20], the proposition follows if we can show the following three facts

1) $k(x,\cdot)\in\hat{\mathscr{H}}$ for all $x\in X$.
2) For all $f\in\hat{H}$, $x\in X$, $f(x)=\langle f,k(x,\cdot)\rangle$
3) The span of $k(x,\cdot)$ is dense in $\hat{\mathscr{H}}$

Since $k(x,y)=\int_\Theta (p(\theta)\phi(x;\theta))\phi(y;\theta)\,d\theta$,

$$\|k(x,\cdot)\|_k^2=\int_\Theta p(\theta)\phi(x;\theta)^2\,d\theta\leq 1$$

so $k(x,\cdot)\in\hat{\mathscr{H}}$. To prove 2, let $f(x)=\int_\Theta\alpha(\theta)\phi(x;\theta)\,d\theta$ and observe

$$\begin{aligned}\langle f,k(x,\cdot)\rangle &= \int_\Theta \frac{\alpha(\theta)\,(p(\theta)\phi(x;\theta))}{p(\theta)}\,d\theta\\ &= \int_\Theta \alpha(\theta)\phi(x;\theta)\,d\theta = f(x).\end{aligned}$$

The proof of 3 is now immediate: if $k(x,\cdot)$ is not dense in $\hat{\mathscr{H}}$, there exists a $g\in\hat{\mathscr{H}}$ which is orthogonal to $k(x,\cdot)$ for all $x\in X$. But, by 2, this means $g=0$, completing the proof.

References

[1] H. D. Block, "The perceptron: a model for brain functioning," *Review of modern physics*, vol. 34, pp. 123–135, Jan. 1962.

[2] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Computation*, vol. 9, no. 7, pp. 1545–1588, 1997.

[3] F. Moosmann, B. Triggs, and F. Jurie, "Randomized clustering forests for building fast and discriminative visual vocabularies," in *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[4] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[5] W. Maass and H. Markram, "On the computational power of circuits of spiking neurons," *Journal of Computer and System Sciences*, vol. 69, pp. 593–616, Dec. 2004.

[6] L. K. Jones, "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training," *The Annals of Statistics*, vol. 20, no. 1, pp. 608–613, Mar. 1992.

[7] A. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Transactions on Information Theory*, vol. 39, pp. 930–945, May 1993.

[8] A. R. Barron, "Neural net approximation," in *Proceedings of the 7th Yale Workshop on Adaptive and Learning Systems*, K. S. Narendra, Ed., 1992, pp. 69–72.

[9] F. Girosi, "Approximation error bounds that use VC-bounds," in *International Conference on Neural Networks*, 1995, pp. 295–302.

[10] G. Gnecco and M. Sanguineti, "Approximation error bounds via Rademacher's complexity," *Applied Mathematical Sciences*, vol. 2, no. 4, pp. 153–176, 2008.

[11] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear Estimation and Classification*, D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, and B. Yu, Eds. Springer, 2003.

[12] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[13] V. Vapnik, *Statistical learning theory*. Wiley, 1998.

[14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.

[15] F. Girosi and G. Anzellotti, "Convergence rates of approximation by translates," Massachusetts Institute of Technology AI Lab, Tech. Rep. AIM-1288, 1992.

[16] B. Schölkopf, R. Herbrich, A. Smola, and R. Williamson, "A generalized representer theorem," NeuroCOLT, Tech. Rep. 81, 2000.

[17] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[18] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research (JMLR)*, vol. 3, pp. 463–482, 2002.

[19] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[20] F. Cucker and S. Smale, "On the mathematical foundations of learning," *Bulletin of the American Mathematical Society*, vol. 39, no. 1, pp. 1–49, 2001.

[21] W. Rudin, *Fourier Analysis on Groups*, reprint edition ed., ser. Wiley Classics Library. New York: Wiley-Interscience, 1994.

[22] L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition*. New York: Springer, 1996.

[23] C. L. B. D. J. Newman, S. Hettich and C. J. Merz, "UCI repository of machine learning databases," 1998. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[24] M. Ledoux and M. Talagrand, *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, 1991.