Wavelet decompositions of Random Forests

Shai Dekel and Iryna Nemirovsky

Abstract—The work brings together an intersection of Harmonic Analysis and Machine Learning. The wavelet decomposition of random forests allows establishing ordering of the random forest components: from 'significant' features to 'less significant' to 'insignificant' noise. This allows for example, to replace the limits typically imposed on tree depths (to avoid over-fitting) by the more robust technique of wavelet thresholding. Our approach could also be considered as a promising research direction to the problem of simplifying or compressing neural-networks. In this paper, we present a few sample experimental results, with the hope that the machine learning community will find interesting applications for this very simple, yet powerful tool.

Index Terms—Machine Learning, Geometric Wavelets, Adaptive Approximation, Non-linear Approximation, Wavelet thresholding.

1 Introduction

THIS work brings together an intersection of Harmonic Analysis and Machine Learning. Wavelets [11], [18], and geometric wavelets [13], [12] are a powerful yet simple tool for finding sparse representations of 'complex' functions. Decision or Random forests [4], [5], [9] are effective machine learning methods that can be considered as a way to overcome the 'greedy' nature of a single decision tree. When combined, the wavelet decomposition of the random forest allows establishing ordering of the random forest data: from 'significant' features to 'less significant' features to 'insignificant' noise. Therefore, the method provides a better understanding of any constructed random forest. This allows to avoid over-fitting even with a small number of trees, to remove noise or provide compression [2]. Also, thresholding the wavelet components is a robust alternative to trying to pre-determine the maximal level of decision trees so as to avoid over-fitting. Lastly, our approach could also be considered as an alternative method of pruning of ensembles [6], [17], [24], [15] where the most important decision nodes of a huge and complex ensemble of models can be quickly and efficiently extracted. In Section 2, we review the two essential ingredients of our method: Geometric Wavelets and Random Forests. In Section 3 we present our main construction and list some of its properties. Finally in Section 4, we present a few, but convincing, sample experimental results, with the hope that the machine learning community will find interesting applications for this very simple, yet powerful tool.

- *S. Dekel is with GE Global Research and the School of Mathematical Sciences, Tel-Aviv University, Israel.*
  E-mail: shai.dekel@ge.com
- *I. Nemirovsky is with the School of Mathematical Sciences, Tel-Aviv University, Israel*

2 Preliminaries

In this section we provide an overview of the two components we are brining together: Geometric Wavelets and Random Forests.

2.1 Geometric Wavelets

The Binary Space Partition (BSP) technique is widely used in image processing, computer graphics [21], [22]. In statistics and machine learning [3], [9] it is called a Decision Tree or the Classification and Regression Tree (CART). In the functional setting we are given a function $f \in L_2(\Omega_0)$ and in the practical setting (e.g. 'classification' problems) the input is point values $f(x_i)$, $x_i \in \Omega_0$, $i \in I$, where $\Omega_0 \subset \mathbb{R}^n$ is a bounded convex domain. The goal is to find an efficient representation of this data, overcoming the complexity, geometry and possibly non-smooth nature of the function values. To this end, we subdivide the initial domain Ω_0 into two subdomains, e.g. by intersecting it with an hyper-plane. The subdivision is performed such that a given cost function is minimized (see below). This subdivision process then continues recursively on the subdomains until some exit criterion is met, which in turn, determines the leaves of the tree. We now describe one instance of the cost function. At each stage of the BSP process the algorithm finds, for a given convex polyhedral domain Ω, two convex subdomains Ω', Ω'' (see Figure 1) and two multivariate polynomials $Q_{\Omega'}, Q_{\Omega''}$ of fixed (typically low) total degree $r-1$ (order r), that minimize the following quantity

$$\|f - Q_{\Omega_1}\|^2_{L_2(\Omega')} + \|f - Q_{\Omega_2}\|^2_{L_2(\Omega'')}, \quad \Omega' \cup \Omega'' = \Omega. \quad (1)$$

If the input consists of non-uniform point values, then the discrete version of (1) is minimized

$$\sum_{x_i \in \Omega'} |f(x_i) - Q_{\Omega'}|^2 + \sum_{x_i \in \Omega''} |f(x_i) - Q_{\Omega''}|^2, \quad \Omega' \cup \Omega'' = \Omega. \quad (2)$$

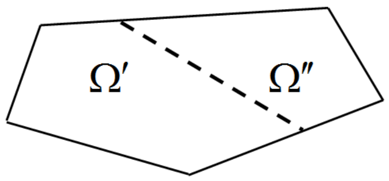


Fig. 1. Illustration of a subdivision by a hyper-plane of a domain Ω.

Observe that the primary degrees of freedom are associated with the subdividing hyperplane. For the order $r = 1$, the approximating polynomials are nothing but the mean of the function values over each of the subdomians, while for higher orders they can be computed using least-squares minimization.

In many applications of decision trees, the high-dimensionality of the data does not allow to search through all possible subdivisions. For example, one may restrict the allowed subdivisions to the class of hyperplanes aligned with the main axes. In contrast, there are cases where one would like to consider more advanced form of subdivisions, where they take certain hyper-surface form, such as conic-sections. Our paradigm of wavelet decompositions can support in principal all of these forms.

For some applications, there is a need to understand which nodes of the tree hold more information than other. Furthermore, in the presence of noise, one popular approach is to limit the levels of the tree, so as not to over-fit and contaminate the decisions by noise. Following the classic paradigm of nonlinear approximation using wavelets [11], [18], and based in part on the theory presented in [16], the authors in [13] proposed the following construction of wavelets over the tree representation. Let Ω' be a child of Ω in a tree $\mathcal{T}$, i.e. $\Omega' \subset \Omega$ and Ω' was created by a partition of Ω, as in Figure 1. We use the polynomial approximations $Q_{\Omega'}, Q_{\Omega}$ associated with these domains, computed by the local minimization (1) and define

$$\psi_{\Omega'} := \psi_{\Omega'}(f) := \mathbf{1}_{\Omega'}\left(Q_{\Omega'} - Q_{\Omega}\right), \tag{3}$$

as the geometric wavelet associated with the subdomain Ω' and the function f (or the given discrete data $\{f(x_i)\}$). As explained in [13], each $\psi_{\Omega'}$ is a local difference component that belongs to the detail space between two levels in the BSP tree, a low resolution level associated with Ω and a high resolution level associated with Ω'. Also, the wavelets (3) also have the 'zero moments' property, i.e., if the data consists of point values of a certain polynomial over Ω, then we get $f(x) = Q_{\Omega'}(x) = Q_{\Omega}(x)$, $\forall x \in \Omega$ and therefore $\psi_{\Omega'} = 0$.

Under certain mild conditions on the tree $\mathcal{T}$ (e.g the diameters of the subdomains should diminish as the levels increase) and the function f we have from by the nature of the wavelets, the 'telescopic' sum of differences

$$f = \sum_{\Omega \in \mathcal{T}} \psi_{\Omega}, \quad \text{where} \quad \psi_{\Omega_0} := Q_{\Omega_0}.$$

A key insight of [13], [16] is that the significance of the geometric wavelet components $\{\psi_{\Omega}\}_{\Omega \in \mathcal{T}}$, can be evaluated based on their norm., i.e. they should be sorted by,

$$\left\|\psi_{\Omega_{k_1}}\right\|_2 \geq \left\|\psi_{\Omega_{k_2}}\right\|_2 \geq \left\|\psi_{\Omega_{k_3}}\right\|_2 \cdots .$$

Observe that in the discrete case, the norm is computed by

$$\left\|\psi_{\Omega'}\right\|_2^2 = \sum_{x_i \in \Omega'} \left|Q_{\Omega}(x_i) - Q_{\Omega'}(x_i)\right|^2, \tag{4}$$

where Ω' is the child of Ω. Therefore, for a given integer $M \in \mathbb{N}$, we can approximate the function by the M-term geometric wavelet sum

$$\sum_{m=1}^{M} \psi_{\Omega_{k_m}}. \tag{5}$$

The sum (5) is, in some sense, a generalization of the classical M-term wavelet approximation, where the wavelets are constructed over dyadic cubes. In Figure 2, we see an M-term approximation using 2048 geometric wavelet terms of the test-image "Peppers" with $r = 2$.

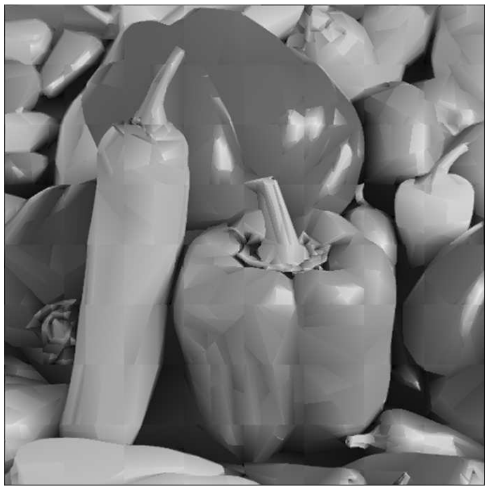


Fig. 2. 'Sparse' Geometric Wavelet approximation of the 512×512 "Peppers" image using 2048 terms. PSNR=31.32.

From an approximation theoretical perspective, the above form of adaptive wavelet approximation performs better than the common non-adaptive mechanism of selecting maximal tree levels on real-life data that is typically non smooth. In applications, one can replace the selection of the parameter M in (5), by a threshold parameter $\epsilon > 0$, chosen suitably for the problem. One then creates an M-term sum from all wavelet terms with norm greater than ϵ. This form of approximation is what allows, for example, in digital cameras to consistently meet a specified visual quality level for the compression algorithm, which is achieved by setting the quantization parameters of the transform coefficients.

As mentioned above, the geometric wavelet approach is not restricted to trees generated by hyperplane subdivision. For example, it can support the conic section subdivisions that are sometimes applied in the context of decision forests [9]. Moreover, in [12] the authors describe a geometric wavelet construction over bivariate trees where each node is split by an active contour segmentation. In this setting, the functional (1), is replaced by a Mumford-Shah type functional

$$\left\|f - Q_{in(\gamma)}\right\|^2_{L_2(in(\gamma))} + \left\|f - Q_{out(\gamma)}\right\|^2_{L_2(out(\gamma))} + \mu \cdot length\left(\gamma\right), \quad (6)$$

where γ is a closed curve and $in\left(\gamma\right)$ and $out\left(\gamma\right)$ are its inside and outside domains, respectively. In Fig. 3 we see an example from [12], where more segmented body parts in a Computed Tomography image are added as wavelet components of an M-term approximation are added.

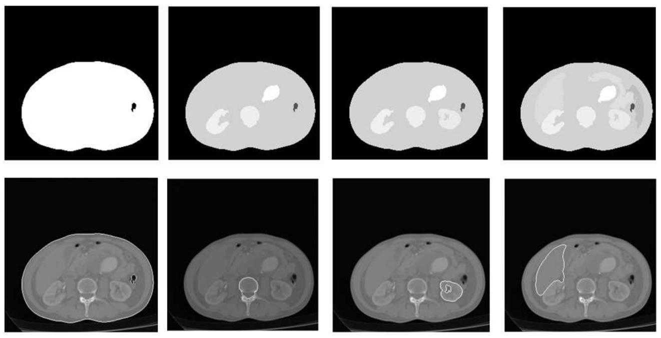


Fig. 3. [12] Using Geometric Wavelet terms to ‘add in’ more segmented organs in a Computed Tomography image.

Typically, in classification problems, the input training set consists not of function values, but of labeled data using L classes. In this scenario, each input training point x_i is assigned a class $\mathcal{C}(x_i)$. To convert the problem to the ‘functional’ setting described above one assigns to each class $\mathcal{C}$ the value of a node on the symmetric simplex of dimension $L-1$. Thus, we may assume that the input data is in the form

$$(x_i, \mathcal{C}(x_i)) \in \left(\mathbb{R}^n, \mathbb{R}^{L-1}\right).$$

In this case, if we choose the approximation order to be $r = 1$, then the calculated mean over any subdomain Ω is in fact a point $E_\Omega \in \mathbb{R}^{L-1}$, inside the simplex. Therefore, the ‘multi-valued’ wavelet $\psi_{\Omega'} : \mathbb{R}^n \to \mathbb{R}^{L-1}$ is (compare with (3))

$$\psi_{\Omega'} = \mathbf{1}_{\Omega'}\left(E_{\Omega'} - E_\Omega\right),$$

and its ‘norm’ in the discrete case is given by

$$\|\psi_{\Omega'}\|_2^2 = \|E_{\Omega'} - E_\Omega\|^2_{l_2} \#\{x_i \in \Omega'\},$$

where for $v \in \mathbb{R}^{L-1}$, $\|v\|_{l_2} := \sqrt{\sum_{i=1}^{L-1} v_i^2}$,

Obviously, any value inside the multidimensional simplex, can be mapped back to a class, along with an estimated certainty level, by calculating the closest vertex of the simplex to it. In particular, these mappings can be applied to any wavelet approximation of functions receiving values in the simplex.

2.2 Random Forests

Even before we inject any notion of randomness into our algorithm, we recall, that in many applications, the amount of input data is large or the problem is embedded in very high dimensions (i.e n is large). One then may partition the data by tiling the input domain Ω_0 and then computing decision trees over each separate piece. Consequently, the partition of the problem domain creates a forest of ‘non-overlapping’ decision trees. Our wavelet decomposition paradigm can easily support this optimization. One creates the M-term approximation (5), by extracting the most significant terms from the full list of all wavelet components of all the trees (supported over the separate tiles).

Our preferred form of injecting randomness is based on ‘bagging’ [4], [5]. From an approximation theoretical perspective [13], this form of random forest allows to create an over-complete representation [7] that overcomes the ‘greedy’ nature of a single tree.

For $j = 1, \ldots, N$, one creates a tree $\mathcal{T}_j$, based on a subset of the data, X^j. This can be achieved by simply selecting randomly a fixed portion of the input data points, say two-thirds. One then provides a weight (score) w_j to the tree $\mathcal{T}_j$, based on the estimated performance of the tree. In the supervised learning, one typically uses the remaining data points $x_i \notin X^j$. We note that the approximation associated with the jth tree, denoted by $\tilde{f}_j\left(x_i\right)$, is computed by finding the leaf $\Omega \in \mathcal{T}_j$ in which x_i is contained and then evaluating $\tilde{f}_j\left(x_i\right) := Q_\Omega\left(x_i\right)$, where Q_Ω is the corresponding polynomial associated with Ω computed by the minimization (1). Observe that this method can also be used to provide the regression $\tilde{f}_j\left(x\right)$, for any point $x \in \Omega_0$. Thus, in supervised learning, the ‘score’ assigned to the j-th tree can be based on the mean squared error

$$\tilde{w}_j := \frac{1}{\#\left\{x_i \notin X^j\right\}} \sum_{x_i \notin X^j} \left|f\left(x_i\right) - \tilde{f}_j\left(x_i\right)\right|^2.$$

Using normalized weights,

$$w_j := \frac{1 - \tilde{w}_j}{\sum\limits_{i=1}^{N}\left(1 - \tilde{w}_i\right)}, \quad (7)$$

one then assigns a value to any point $x \in \Omega_0$ by

$$\tilde{f}\left(x\right)=\sum_{j=1}^{N} w_j \tilde{f}_j\left(x\right).$$

The weights described above are 'global' weights, but one could also assign local weights to specific regions or subsets of the data.

Another form of injecting randomness into the forest is using the random subspace method (see e.g. [14], [9]). However, we believe that with our introduction of wavelet decompositions, this method should only be used in cases where n (e.g. the number of parameters) is very large, and searching at each tree node for an optimal split of the training data could be computationally overwhelming. To overcome that, one resorts to selecting at each node Ω, a random subset of $\tilde{n} < n$ variables and search for the optimal subdivision of Ω using the corresponding variable subspace.

3 Wavelet Decomposition of Random Forests

3.1 Collecting the tree decompositions

Based on the tools described in Section 2, we create a wavelet decomposition of each of the trees in the random forest

$$\tilde{f}_j=\sum_{\Omega\in\mathcal{T}_j}\psi_\Omega, \quad j=1,\ldots,N.$$

Using the weights (7) provides a wavelet representation of the entire random forest

$$\tilde{f}\left(x\right)=\sum_{j=1}^{N}\sum_{\Omega\in\mathcal{T}_j} w_j\psi_\Omega\left(x\right) \tag{8}$$

The theory tells us that one should order the wavelet components of the random forest by

$$w_{j\left(\Omega_{k_1}\right)}\left\|\psi_{\Omega_{k_1}}\right\|_2 \geq w_{j\left(\Omega_{k_2}\right)}\left\|\psi_{\Omega_{k_2}}\right\|_2\cdots, \tag{9}$$

with the notation $\Omega\in\mathcal{T}_j \Rightarrow j\left(\Omega\right)=j$. Thus, the M-term approximation of a random forest is

$$\tilde{f}_M\left(x\right)=\sum_{m=1}^{M} w_{j\left(\Omega_{k_m}\right)}\psi_{\Omega_{k_m}}\left(x\right). \tag{10}$$

3.2 Measuring the strength of a tree using wavelet sparsity

In many application of machine learning, one tries to assign a quality indicator to each tree of the forest. Here we propose that one such useful property of a tree is the sparsity of its wavelet decomposition. Roughly speaking, for a tree whose number of levels were not limited a-priory, this indicator will be small if only a few decisions nodes in the tree are important and capture the essence of the classification. For a given tree $\mathcal{T}$ and parameter $0<\tau\leq 2$, we denote the τ-strength of the tree by

$$\mathcal{N}_\tau\left(f,\mathcal{T}\right):=\left(\sum_{\Omega\in\mathcal{T}}\left\|\psi_\Omega\right\|_2^\tau\right)^{1/\tau}. \tag{11}$$

To explain the role of τ, we recall the following example from [13]. Let $\widetilde{\Omega}\subset[0,1]^n$ be a convex polytope, and denote $f(x):=\mathbf{1}_{\widetilde{\Omega}}(x)$. Assume $\mathcal{T}$ is a partition such that for each $\Omega\in\mathcal{T}$, either $\widetilde{\Omega}\subseteq\Omega$, $\Omega\subseteq\widetilde{\Omega}$ or $int(\Omega\cap\widetilde{\Omega})=\emptyset$, where $int(E)$ denotes the interior of $E\subset\mathbb{R}^n$. Then, for $r=1$, it is easy to see that

$$Q_\Omega=\begin{cases}\frac{|\widetilde{\Omega}|}{|\Omega|}, & \widetilde{\Omega}\subseteq\Omega,\\ 1, & \Omega\subseteq\widetilde{\Omega},\\ 0, & int(\Omega\cap\widetilde{\Omega})=\emptyset.\end{cases}$$

As shown in [13], for the choice $\tau=2$ and any tree that satisfies the above conditions

$$\mathcal{N}_2^2(f,\mathcal{T})=|\widetilde{\Omega}|=\|f\|_2^2\,. \tag{12}$$

it follows that $\mathcal{N}_2\left(f,\mathcal{T}\right)$ is not a good sparsity strength measure.

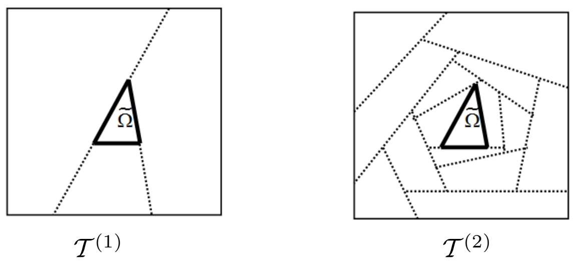


Fig. 4. Two BSP partitions with $\mathcal{N}_2\left(f,\mathcal{T}^{(1)}\right)=\mathcal{N}_2\left(f,\mathcal{T}^{(2)}\right)=\|f\|_2$

Referring to Figure 4, we see that the partition $\mathcal{T}^{(1)}$ is 'optimal' since its BSP lines coincide with the hyper-planes that describe $\partial\widetilde{\Omega}$, while $\mathcal{T}^{(2)}$ contains subdivisions that do not reduce the entropy and hence unuseful. Nevertheless, the equality $\mathcal{N}_2\left(f,\mathcal{T}^{(1)}\right)=\mathcal{N}_2\left(f,\mathcal{T}^{(2)}\right)=\|f\|_2$ holds. However, things change dramatically when we choose a sufficiently small τ. For small τ, the ℓ_τ norm serves almost as counting measure and since the wavelet decomposition associated with $\mathcal{T}^{(1)}$ contains significantly less non-zero elements, we obtain that $\mathcal{N}_\tau\left(f,\mathcal{T}^{(1)}\right)$ is much smaller than $\mathcal{N}_\tau\left(f,\mathcal{T}^{(2)}\right)$.

4 Experimental Results

To demonstrate the potential applicability of wavelet decompositions of random forests we show experimental results for two basic problems: image denoising and supervised learning of highly non-linear and highly noisy training data. We then also discuss the computational complexity of the algorithm.

4.1 Image denoising

As is standard practice in the image processing community, we took well-known test images and added Guassian noise to them. We then constructed a random forest over the noisy data using bagging, in the following sense: At each iteration i, $1 \leq i \leq \tilde{N}$, we selected randomly a fixed percentage (e.g. 80%) of the noisy pixels. At each iteration, we then tiled the image in different ways, so as to speedup the computations of the trees. For example, we tiled test images of size 256×256 into 4 horizontal or vertical strips of typical sizes 64×256. We then constructed trees over each tile separately using only the random input data selected out of the bag for this iteration. During the tree constructions, we stored for each node the geometry of the associated subdomain and the associated approximating low order polynomial. Observe that from this information we can easily extract the wavelet component (3) associated with each node.

Once all $\tilde{N}$ iterations were completed, we obtained trees, $\mathcal{T}_j$, $1 \leq j \leq N$, where $N = K\tilde{N}$ and K, the number of tiles at each iteration. For the denoising algorithm the weights w_j, used in the representation (8) were set to $w_j = 1/\tilde{N}$. We then retained the most significant wavelet components of forest representation of the noisy image using thresholding of wavelet norms (4). Assuming σ^2 is the additive noise variance, we retain only the wavelet components whose norm satisfies

$$\|\psi_{\Omega'}\|_2^2 \geq c\sigma^2 \# \{x_i \in \Omega'\}. \tag{13}$$

The constant in (13) can be crudely estimated as $c = 4$, but is further tuned using experiments to $c = 0.6$. Obtaining better theoretical performance estimates of denoising using wavelet thresholding, in the context of random forests, is part of our ongoing research.

In Figure 5 we see the test image "Lena" with added noise and the result of our denoising algorithm. So as to compare with state-of-the-art algorithm of [26], as well as the cited results therein, the noise was added as per the the MatLab code provided in [25]. From Table 1, one can see that our algorithm (right most column labeled as **WF**) is quite competitive with the algorithms of [19], [20], [1], [10] and [26].

TABLE 1
Comparison with state-of-the-art denoising algorithms

All PSNR values in dB. Left column: PSNR of noisy images.

Test image	[19]	[20]	[1]	[10]	[26]	**WF**
"Lena" 16.57	25.7	26.0	26.2	27.3	26.0	**26.4**
"Peppers" 22.22	29.8	30.1	30.3	30.6	30.1	**30.7**

4.2 Data classification

In these experiments we demonstrate the advantage of applying the wavelet decomposition to learning of non-linear and noisy classification training sets. In Figure 6(a) we see a data set of four classes, each class containing 500 points generated from adding Gaussian noise (in the direction of the curve normal) to four spiral curves. We then further contaminated the training set with noise, by randomly mislabeling 40% of each spiral's data points as belonging to one of the three other spirals. Here, as explained in Section 2.1, we applied the mapping of the four labels to the simplex with four vertices in $\mathbb{R}^3$: $v1 = (1/2, 0, -1/2\sqrt{2}), v2 = (-1/2, 0, -1/2\sqrt{2}), v3 = (0, 1/2, 1/2\sqrt{2})$ and $v4 = (0, -1/2, 1/2\sqrt{2})$. Thus, our noisy training data set containing misclassifications is of the form $(x_i, v_j(i)) \in (\mathbb{R}^2, \mathbb{R}^3)$.

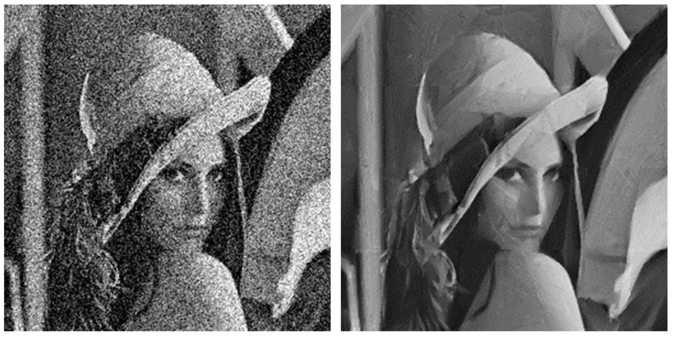
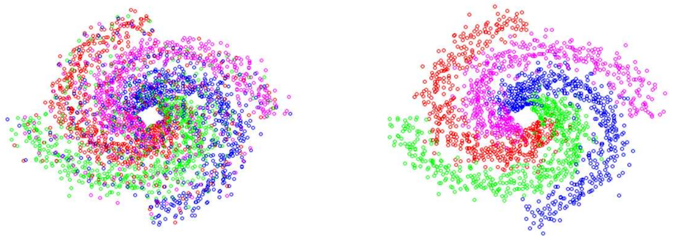

(a) Noisy PSNR=16.57 dB. (b) De-noised PSNR=26.4 dB.

Fig. 5. Image denoising of "Lena". The algorithm selected 11805 significant wavelets using (13), from a forest of $N = 64$ trees, with $\tilde{N} = 16$, $K = 4$, containing 165822 wavelet components. The local polynomial approximation is of order $r = 2$ (linear).

Over this highly noisy data, we constructed a random forest consisting of 5 trees, each of 19 levels. During the construction, the subdivision process was terminated at nodes containing ≤ 8 points, so as to avoid overfitting. We then computed a sparse wavelet approximation of the random forest using only $M = 144$ wavelet components (out of 1641). Observe that M is selected with the goal of minimizing the classification error on the training set by the following algorithm: we initialize $M = M_0$ for some small $M_0 \geq 1$ and compute the classification error for $\tilde{f}_{M_0}$. We then add the next significant wavelet components one by one, using (9) updating the wavelet representation locally on their support only as well as the global error. If the error reaches some minima at M and begins to increase we terminate the search and declare $\tilde{f}_M$ as the optimal sparse representation for our classification problem.

Indeed, when the training data is contaminated with noise, one of the challenges of the standard random forest algorithm is selecting the maximal number of levels to use from the trees. Too few and the data can be under-fitted, too many and the result could be over-fitting. The wavelet decomposition overcomes this problem by picking only the significant components.

We generated testing data from the spiral curves to test our trained classification (see 6(b)). On this testing data, the sparse wavelet approximation obtained a classification accuracy of 88.3%, while the standard random forest classification using all 19 levels obtained only an accuracy of 81.4%.



(a) Noisy training dataset. (b) Testing dataset.

Fig. 6. 4 spiral data used for a classification experiment

4.3 Computational complexity

Here, we do not discuss the computational complexity involved in the construction of decision forests, since there is a significant body of existing literature as well as many open source tools that implement the construction. However, for our wavelet decomposition tool to work, it is required that each decision node will store the 'geometry' of its support and well as the local estimate it gave to the data within its support. In a recommended embodiment, one selects a 'quality' threshold and then iterates over the decision nodes computing the associated wavelet components and retaining them if their norms exceed the threshold. Therefore, the computational complexity is linear in the number of decision nodes. If one wants to construct the M-term approximation (10) for some given M, then the complexity is $\#nodes \times \log(\#nodes)$.

5 Conclusions

In this paper we presented an algorithm for sparse representation of decision forests based on theoretical concepts from approximation theory and harmonic analysis. Although we only demonstrated a limited range of experimental results, we hope that the machine learning community will find a significant number of applications for this approach.

References

[1] M. Aharon, M. Elad, and A.M. Bruckstein, The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation, IEEE Trans. On Signal Processing, Vol. 54, No. 11, pp. 4311-4322, 2006.

[2] D. Alani, A. Averbuch and S. Dekel, Image coding using geometric wavelets , IEEE Trans. Image Processing 16 (2007), 69-77.

[3] E. Alpaydin, *Introduction to machine learning*, MIT Press, 2004.

[4] L. Breiman. "Random forests", Machine Learning, Vol. 45, pp 5-32, 2001.

[5] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*, Chapman and Hall/CRC, 1984.

[6] H. Chen, P. Tino and X. Yao, Predictive Ensemble Pruning by Expectation Propagation, IEEE Knowledge and data engineering, Vo. 21, No. 7, pp. 999-1013, 2009.

[7] O. Christensen, *An introduction to Frames and Riesz Bases*, Birkäuser, 2002.

[8] R. Coifman and D. Donoho, "Translation invariant denoising, Wavelets and Statistics", Lecture Notes in Statistics, Vol. 103, 1995, pp 125-15.

[9] A. Criminisi, J. Shotton and E. Konukoglu, "Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning", Microsoft Research technical report TR-2011-114, 2011.

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, Image denoising by sparse 3D transform-domain collaborative filtering, IEEE Trans. Image Processing, Vol. 16, No. 8, pp. 2080-2095, 2007.

[11] I. Daubechies, *Ten lectures on wavelets*, 1992.

[12] S. Dekel and I. Gershtansky, "Active Geometric Wavelets", Proc. Approximation Theory XIII, pp.95-109, 2012.

[13] S. Dekel and D. Leviatan, "Adaptive multivariate approximation using binary space partitions and geometric wavelets", SIAM J. Num. Anal. Vol. 43, pp. 707-732, 2005.

[14] T. K. Ho, The random subspace method of constructing decision forests, IEEE Pattern analysis and machine intelligence, Vol. 20, No. 8, pp. 832-844, 1998.

[15] A. Joly, F. Schnitzler, P. Geurts and L. Wehenkel, L_1-based compression of random forest models, proceedings of ESANN, pp. 375-380, 2012.

[16] B. Karaivanov and P. Petrushev, "Nonlinear piecewise polynomial approximation beyond Besov spaces", Appl. Comput. Harmon. Anal., Vol. 15, No. 3, pp. 177-223, 2003.

[17] V. Kulkarni and P. Sinha, Pruning of Random Forest classifiers: A survey and future directions, proceedings of ICDSE, pp. 64-68, 2012

[18] S. Mallat, *A Wavelet tour of signal processing, 3rd edition (the sparse way)*, 2009.

[19] A. Pizurica and W. Philips, Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising, IEEE Trans. on Image Processing, Vol. 15, No. 3, pp. 654-665, 2006.

[20] J. Portilla, V. Strela, M. J. Wainwright and E. P. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, IEEE Trans. on Image Processing, Vol. 12, No. 11, pp. 1338 1351, 2003.

[21] H. Radha, M. Vetterli and R. Leonardi, "Image compression using binary space partitioning trees", IEEE Trans. Image Proc. Vol. 5 (1996), pp. 1610-1624.

[22] P. Salembier and L. Garrido, Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval, IEEE Trans. Image Proc. 9 (2000), 561-576.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, Image quality assessment: From error visibility to structural similarity, IEEE Transactions on Image Processing, Vol. 13, No. 4, Apr. 2004.

[24] F. Yang, W. Lu, L. Luo, T. Li, Margin optimization based pruning for random forest, Neurocomputing, Vol. 94, pp. 54-63, 2012.

[25] L. Zhanga, LPG-PCA denoising experimental results, avaliable at http://www4.comp.polyu.edu.hk/ cslzhang/LPG-PCA-denoising.htm

[26] L. Zhanga, W. Dong, D. Zhanga and G. Shi, Two-stage image denoising by principal component analysis with local pixel grouping, Pattern Recognition, Vol. 43, No. 4, pp. 1531-1549, 2010.

Shai Dekel Shai received his Ph.D. degree in mathematics from the Tel-Aviv University, Tel-Aviv, Israel, in 2000. Presently, he is a Researcher at GE Global Research and a visiting associate professor at the school of mathematical sciences, Tel-Aviv University. His research interests include approximation theory, harmonic analysis, and their applications in data science.

Iryna Nemirovsky Iryna is presently a M.Sc. student in the school of mathematical sciences, Tel-Aviv University.