Treelets — A Tool for Dimensionality Reduction and Multi-Scale Analysis of Unstructured Data

Ann B. Lee
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15206
annlee@stat.cmu.edu

Boaz Nadler
Department of Computer Science
and Applied Mathematics
Weizmann Institute of Science
P.O.Box 26, Rehovot 76100, Israel
boaz.nadler@weizmann.ac.il

Abstract

In many modern data mining applications, such as analysis of gene expression or word-document data sets, the data is *high-dimensional* with hundreds or even thousands of variables, *unstructured* with no specific order of the original variables, and *noisy*. Despite the high dimensionality, the data is typically redundant with underlying structures that can be represented by only a few features. In such settings and specifically when the number of variables is much larger than the sample size, standard global methods may not perform well for common learning tasks such as classification, regression and clustering. In this paper, we present *treelets* — a new tool for multi-resolution analysis that extends wavelets on smooth signals to general unstructured data sets. By construction, treelets provide an orthogonal basis that reflects the internal structure of the data. In addition, treelets can be useful for feature selection and dimensionality reduction prior to learning. We give a theoretical analysis of our algorithm for a linear mixture model, and present a variety of situations where treelets outperform classical principal component analysis, as well as variable selection schemes such as supervised (sparse) PCA.

1 Introduction

A well-known problem in statistics is that estimation and prediction tasks become increasingly difficult with the dimensionality of the observations. This "curse of dimensionality" [1] highlights the necessity of more efficient data representations that reflect the inherent, often simpler, structures of naturally occurring data. Such low-dimensional compressed representations are required to both (i) reflect the geometry of the data, and (ii) be suitable for later tasks such as regression, classification, and clustering. Two standard tools for dimensionality reduction and feature selection are Principal Component Analysis (PCA) and wavelets. Each one of these techniques has its own strengths and weaknesses. As described below, both methods are inadequate for the analysis of noisy unstructured high-dimensional data of intrinsic low dimensionality, which is the interest of this work.

Principal component analysis (PCA) is a popular feature selection method due to both its simplicity and theoretical property as providing a sequence of "best" linear approximations in a least square sense [2]. PCA, however, has two main limitations. First, PCA computes a *global* representation, where each basis vector is a linear combination of *all* the original variables. Thus, interpretation of its results is often a difficult task and may not help in unraveling internal localized structures in a data set. For example, in DNA microarray data, it can be quite difficult to detect small sets of highly correlated genes from a global PCA analysis. The second limitation of PCA is that for noisy input data, it constructs an optimal representation of the noisy data but not necessarily of the (unknown) underlying noiseless data. When the number of variables p is much larger than the number of observations n, the true underlying principal factors may be masked by the noise, yielding an inconsistent estimator in the joint limit $p(n) \to \infty$ and $n \to \infty$ [3]. Even for a finite sample size n, this property of PCA and other global methods including partial least squares and ridge regression can lead to large prediction errors in regression and classification [4, 5].

In contrast to PCA, wavelet methods describe the data in terms of *localized* basis functions. The representations are multi-scale, and for smooth data, also sparse [6]. Wavelets are often used in many non-parametric statistics tasks, including regression and density estimation [7]. Their main limitation is the

implicit assumption of smoothness of the (noiseless) data as a function of its variables. Wavelets are thus not suited for the analysis of unstructured data.

In this paper, we are interested in the analysis of high-dimensional, unstructured and noisy data, as it typically appears in many modern applications (gene expression microarrays, word-document arrays, consumer data sets). We present a novel multi-scale representation of unstructured data, where variables are randomly ordered and do not necessarily satisfy any specific smoothness criteria. We call the construction *treelets*, as the method is inspired by both wavelets and hierarchical clustering trees. The treelet algorithm starts from a pairwise similarity measure between features and constructs, step by step, a data-driven multiscale orthogonal basis whose basis functions are supported on nested clusters in a hierarchical tree. As in PCA, we explore the covariance structure of the data but — unlike PCA — the analysis is *local* and *multiscale*.

There are also other methods related to treelets. In recent years, hierarchical clustering algorithms have been widely used for identifying diseases and groups of co-expressed genes [8]. The novelty and contribution of our approach, compared to such clustering methods, is the simultaneous construction of a data-driven multiscale basis and a cluster tree. The introduction of a basis enables application of the well-developed machinery of orthonormal expansions, wavelets and wavelet packets for e.g. reconstruction, compression, and denoising of *general*, non-structured, data arrays. The treelet algorithm bears some similarities to the local Karhunen-Loeve Basis for smooth structured data by Saito [9], where the basis functions are data-driven but the tree structure is fixed. Our work is also related to a recent paper by Murtagh [10], which also suggests constructing basis functions on data-driven cluster trees but uses fixed Haar wavelets. The treelet algorithm offers the advantages of both approaches as it incorporates *adaptive basis functions* as well as *a data-driven tree structure*.

In Sec. 2, we describe the treelet algorithm. In Sec. 3, we provide analysis of its performance on a linear mixture error-in-variable model and give a few illustrative examples of its use in representation, regression and classification problems. In particular, in Sec. 3.2, (unsupervised) treelets are compared to supervised dimensionality reduction schemes by variable selection, and are shown to outperform these under some settings, whereas in Sec. 3.3 we present application of treelets on a classification problem with a real dataset of internet advertisements.

2 The Treelet Transform

In many modern data sets (e.g. DNA microarrays, word-document arrays, financial data, consumer databases, etc.), the data is noisy and high-dimensional but also highly redundant with many variables (the genes, words, etc) related to each other. Clustering algorithms are typically used for the organization and internal grouping of the coordinates of such data sets, with *hierarchical clustering* being one of the common choices. These methods offer an easily interpretable description of the data structure in terms of a dendrogram, and only require the user to specify a measure of similarity between groups of observations, or in this case, groups of variables. So called agglomerative methods start at the bottom of the tree and at each level merge the two groups with highest inter-group similarity into one larger cluster.

The novelty of the proposed treelet algorithm is in constructing not only clusters or groupings of variables, but also a *multi-resolution representation* of the data: At each level of the tree, we group together the most similar variables and replace them by a coarse-grained "sum variable" and a residual "difference variable", both computed from a local principal component analysis (or Jacobi rotation) in two dimensions. We repeat this process recursively on the sum variables, until we reach either the root node at level $L = p - 1$ (where p is the total number of original variables) or a maximal level $J \leq p - 1$ selected by cross-validation or other stopping criteria. As in standard multi-resolution analysis, the treelet algorithm results in a set of "scaling functions" defined on nested subspaces $V_0 \supset V_1 \supset \ldots \supset V_J$, and a set of orthogonal "detail functions" defined on residual spaces $\{W_j\}_{j=1}^{J}$ where $W_j \oplus V_j = V_{j-1}$.

The decision as to which variables to merge in the tree is determined by a similarity score M_{ij} computed for all pairs of sum variables v_i and v_j. One choice for M_{ij} is the correlation coefficient

$$M_{ij} = C_{ij}/\sqrt{C_{ii}C_{jj}} \tag{1}$$

where $C_{ij} = \mathbb{E}\left[(v_i - \mathbb{E}v_i)(v_j - \mathbb{E}v_j)^T\right]$ is the familiar covariance. For this measure $|M_{ij}| \leq 1$ with equality if and only if $x_j = ax_i + b$ for some constants $a, b \in \mathbb{R}$. Other information-theoretic or graph-theoretic similarity measures are also possible and can potentially lead to better results.

The Treelet Algorithm: Jacobi Rotations on Pairs of Similar Variables

• At level $L = 0$ (the bottom of the tree), each observation or "signal" is represented by the original variables x_k $(k = 1, \ldots, p)$. For convenience, introduce a

p-dimensional coordinate vector

$$\mathbf{x}^{(0)} \doteq [s_{0,1}, \ldots, s_{0,p}]$$

where $s_{0,k} = x_k$, and associate these coordinates to the Dirac basis $B_0 \doteq [\mathbf{v}_{0,1}, \mathbf{v}_{0,2}, \ldots, \mathbf{v}_{0,p}]$ where B_0 is the $p \times p$ identity matrix. Compute the sample covariance and similarity matrices $C^{(0)}$ and $M^{(0)}$. Initialize the set of "sum variables", $S = \{1, 2, \ldots, p\}$.

- Repeat for $L = 1, \ldots, J$

1. **Find the two most similar sum variables according to the similarity matrix $M^{(L-1)}$.** Let

$$(\alpha, \beta) = \arg\max_{i,j \in S} M_{ij}^{(L-1)} \ . \tag{2}$$

   where $i < j$, and maximization is only over pairs of sum variables that belong to the set S. As in standard wavelet analysis, "difference variables" (defined in step 3) are not processed.

2. **Perform a local PCA on this pair.** Find a Jacobi rotation matrix [11]

$$J(\alpha, \beta, \theta_L) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \tag{3}$$

   where $c = \cos(\theta_L)$ and $s = \sin(\theta_L)$, that decorrelates x_α and x_β; i.e. find a rotation angle θ_L such that $C_{\alpha\beta}^{(L)} = C_{\beta\alpha}^{(L)} = 0$ and $C_{\alpha\alpha}^{(L)} \geq C_{\beta\beta}^{(L)}$, where $C^{(L)} = JC^{(L-1)}J^T$. This transformation corresponds to a change of basis $B_L = JB_{L-1}$, and new coordinates $\mathbf{x}^{(L)} = J\mathbf{x}^{(L-1)}$. Update the similarity matrix $M^{(L)}$ accordingly.

3. **Multi-resolution analysis.** Define the sum and difference variables at level L as $s_L = x_\alpha^{(L)}$ and $d_L = x_\beta^{(L)}$. Similarly, define the scaling and detail functions $\mathbf{v}_L$ and $\mathbf{w}_L$ as columns α and β of the basis matrix B_L. Remove the difference variable from the set of sum variables, $S = S \backslash \{\beta\}$. At level L, we have the *orthogonal treelet decomposition*

$$\mathbf{x} = \sum_{i=1}^{p-L} s_{L,i}\mathbf{v}_{L,i} + \sum_{i=1}^{L} d_i \mathbf{w}_i. \tag{4}$$

   where the new set of scaling vectors $\{\mathbf{v}_{L,i}\}_{i=1}^{p-L}$ is the union of the vector $\mathbf{v}_L$ and the scaling vectors $\{\mathbf{v}_{L-1,j}\}_{j \neq \alpha,\beta}$ from the previous level, and the new coarse-grained sum variables $\{s_{L,i}\}_{i=1}^{p-L}$ are the projections of the original data onto these vectors.

The output of the algorithm can be summarized in terms of a cluster tree with a height $J \leq p-1$ and an ordered set of rotations and pairs of indices, $\{(\theta_j, \alpha_j, \beta_j)\}_{j=1}^{J}$. The treelet decomposition of a signal $\mathbf{x}$ has the general form in Eq. 4 with $L = J$. As in standard multi-resolution analysis, the first sum is the coarse-grained representation of the signal, while the second sum captures the residuals at different scales. In particular, for a maximum height tree with $J = p-1$, we have $\mathbf{x} = s_J\mathbf{v}_J + \sum_{j=1}^{J} d_j\mathbf{w}_j$, with a single coarse-grained variable (the root of the tree) and $p-1$ difference variables. Fig. 1 (left) shows an example of a treelet construction for a signal of length $p = 5$, with the signal representations $\mathbf{x}^{(L)}$ at the different levels of the tree shown on the right.

In a naive implementation with an exhaustive search for the optimal pair (α, β) in Eq. 2, the overall complexity of the treelet algorithm is $O(Jp^2)$ operations. However, by storing the similarity matrices $C^{(0)}$ and $M^{(0)}$ and keeping track of their local changes, the complexity is reduced to $O(p^2)$.

3 Theory and Examples

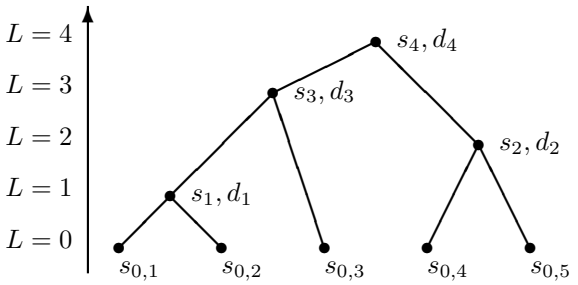
The motivation for the treelets is two-fold: One goal is to find a "natural" system of coordinates that reflects the underlying internal structures of the data. A second goal is to improve the performance of conventional regression and classification techniques in the "large p, small n" regime by compressing the data prior to learning. In this section, we study a few illustrative supervised and unsupervised examples with treelets and a linear error-in-variables mixture model that address both of these issues.

In the unsupervised setting, we consider a data set $\{\mathbf{x}_i\}_{i=1}^{n} \subset \mathbb{R}^p$ that follows a linear mixture model with K components and additive Gaussian noise,

$$\mathbf{x} = \sum_{j=1}^{K} u_j \mathbf{v}_j + \sigma\mathbf{z} \ . \tag{5}$$

The components or "factors" u_j are random variables, the "loading vectors" $\mathbf{v}_j$ are fixed but typically unknown linearly independent vectors, σ is the noise level, and $\mathbf{z} \sim \mathcal{N}_p(0, I)$ is the noise vector. Unsupervised learning tasks include inference on the number of components K and the underlying vectors $\mathbf{v}_j$ in, for example, data representation, compression and smoothing.

In the supervised case, we consider a data set $\{\mathbf{x}_i, y_i\}_{i=1}^{n}$, where the response value y is a linear com-



$$\mathbf{x}^{(4)} = [\, s_4 \quad d_1 \quad d_3 \quad d_4 \quad d_2 \,]$$
$$\mathbf{x}^{(3)} = [\, s_3 \quad d_1 \quad d_3 \quad s_2 \quad d_2 \,]$$
$$\mathbf{x}^{(2)} = [\, s_1 \quad d_1 \quad s_{0,3} \quad s_2 \quad d_2 \,]$$
$$\mathbf{x}^{(1)} = [\, s_1 \quad d_1 \quad s_{0,3} \quad s_{0,4} \quad s_{0,5}]$$
$$\mathbf{x}^{(0)} = [\, s_{0,1} \quad s_{0,2} \quad s_{0,3} \quad s_{0,4} \quad s_{0,5}]$$

Figure 1: (**Left**) A toy example of a hierarchical tree for data of dimension $p = 5$. At $L = 0$, the signal is represented by the original p variables. At each successive level $L = 1, 2, \ldots, p-1$ the two most similar sum variables are combined and replaced by the sum and difference variables s_L, d_L corresponding to the first and second local principal components. (**Right**) Signal representation $\mathbf{x}^{(L)}$ at different levels. The s- and d-coordinates represent projections along scaling and detail functions in a multi-scale treelet decomposition.

bination of the variables u_j above according to

$$y = \sum_{j=1}^{K} \alpha_j u_j + \epsilon \;, \tag{6}$$

where ϵ represents random noise. A supervised learning task is prediction of y for new data $\mathbf{x}$ given a training set $\{\mathbf{x}_i, y_i\}_{i=1}^n$ in regression or classification.

Linear mixture models are common in many fields, including spectroscopy and gene expression analysis. In spectroscopy Eq. 5 is known as Beer's law, where $\mathbf{x}$ is the logarithmic absorbance spectrum of a chemical substance measured at p wavelengths, u_j are the concentrations of constituents with pure absorbance spectra $\mathbf{v}_j$, and the response y is typically one of the components, $y = u_i$. In gene data, $\mathbf{x}$ is the measured expression level of p genes, u_j are intrinsic activities of various pathways, and each vector $\mathbf{v}_j$ represents the set of genes in a pathway. The quantity y is typically some measure of severity of a disease such as time until recurrence of cancer. A linear relation between y and the values of u_j as in Eq. 6 is commonly assumed.

3.1 Linear Mixture Model with Block Structures

We first consider the unsupervised problem of uncovering the internal structure of a given data set. Specifically, we consider a set $\{\mathbf{x}_i\}_{i=1}^n$ from the model (5) with $K = 3$ components and with loading vectors

$$\begin{array}{rcl} \mathbf{v}_1 & = & \frac{1}{\sqrt{p_1}} [\overbrace{1\,1\ldots1}^{B_1} \; \overbrace{0\,0\ldots0}^{B_2} \; \overbrace{0\,0\ldots0}^{B_3}]^T \\ \mathbf{v}_2 & = & \frac{1}{\sqrt{p_2}} [0\,0\ldots0 \;\; 1\,1\ldots1 \;\; 0\,0\ldots0]^T \\ \mathbf{v}_3 & = & \frac{1}{\sqrt{p_3}} [0\,0\ldots0 \;\; 0\,0\ldots0 \;\; 1\,1\ldots1]^T \;. \end{array} \tag{7}$$

where each set of variables B_j is disjoint with p_j elements ($j = 1, 2, 3$). For illustrative purposes, the variables are ordered; shuffling the variables does not affect the results of the treelet algorithm. Our aim is to recover the unknown vectors $\mathbf{v}_i$ and the relationships between the variables $\{x_1, \ldots, x_p\}$. We present two examples. In the first example, PCA is able to find the hidden vectors, while it fails in the second one. Treelets, in contrast, are able to unravel these structures in both cases.

Example 1: Uncorrelated Blocks. Suppose that the random variables $u_j \sim N(0, \sigma_j^2)$ are *independent* for $j = 1, 2, 3$. The population covariance matrix of $\mathbf{x}$ is then given by $C = \Sigma + \sigma^2 I_p$ where

$$\Sigma = \begin{pmatrix} \Sigma_{11} & 0 & 0 \\ 0 & \Sigma_{22} & 0 \\ 0 & 0 & \Sigma_{33} \end{pmatrix} \tag{8}$$

is a 3×3 block matrix with $\Sigma_{kk} = \sigma_k^2 1_{p_k \times p_k}$. Assume that $\sigma_j \gg \sigma$ for all j. As $n \to \infty$, PCA recovers the hidden vectors $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$, as these three vectors are the principal eigenvectors of the system. A treelet transform with a height determined by cross-validation (see below), given that $K = 3$, returns the same results.

Example 2: Correlated Blocks. We now consider the case of correlations between the random variables u_j. Specifically, assume they are *dependent* according to

$$u_1 \sim N(0, \sigma_1^2), \;\; u_2 \sim N(0, \sigma_2^2), \; u_3 = c_1 u_1 + c_2 u_2 \;. \tag{9}$$

The covariance matrix is now given by $C = \Sigma + \sigma^2 I_p$ where

$$\Sigma = \begin{pmatrix} \Sigma_{11} & 0 & \Sigma_{13} \\ 0 & \Sigma_{22} & \Sigma_{23} \\ \Sigma_{13}{}^T & \Sigma_{23}{}^T & \Sigma_{33} \end{pmatrix} \tag{10}$$

with $\Sigma_{kk} = \sigma_k^2 1_{p_k \times p_k}$ (note that $\sigma_3^2 = c_1^2\sigma_1^2 + c_2^2\sigma_2^2$), $\Sigma_{13} = c_1\sigma_1^2 I_{p_1 \times p_3}$ and $\Sigma_{23} = c_2\sigma_2^2 I_{p_2 \times p_3}$. Due to

the correlations between u_j, the loading vectors of the block model no longer coincide with the principal eigenvectors, and it is quite difficult to extract them with PCA.

We illustrate this problem by the example considered in [12]. Specifically, let $\sigma_1^2 = 290$, $\sigma_2^2 = 300$, $c_1 = -0.3$, $c_2 = 0.925$, $p_1 = p_2 = 4, p_3 = 2$, and $\sigma = 1$. The corresponding variance σ_3^2 of u_3 is 282.8. The first three PCA vectors are shown in Fig. 2 (left). As expected, it is difficult to detect the underlying vectors $\mathbf{v}_i$ from these results. Other methods, such as PCA with thresholding also fail to achieve this goal [12], *even with an infinite number of observations*, i.e. in the limit $n \to \infty$. This example illustrates the limitations of a global approach, since ideally, we should detect that the variables (x_5, x_6, x_7, x_8) are all related and then extract the latent vector $\mathbf{v}_2$ from these variables. In [12], Zou et al show by simulation that a combined L_1 and L_2-penalized least squares method, which they call "sparse PCA" or "elastic nets", correctly identifies the sets of important variables if given "oracle information" on the number of variables p_1, p_2, p_3 in the different blocks. The treelet transform is similar in spirit to elastic nets as both methods tend to group highly correlated variables together. Treelets however are able to find the vectors $\mathbf{v}_i$ knowing only K, the number of components in the linear mixture model, and also do not require tuning of any additional sparseness parameters.

Let us start with a theoretical analysis in the limit $n \to \infty$, assuming pairwise correlation as the similarity measure. At the bottom of the tree, i.e. for $L = 0$, the correlation coefficients of pairs of variables in same block B_k $(k = 1, 2, 3)$ are given by

$$\rho_{kk} = \frac{1}{1 + \sigma^2/\sigma_k^2} \approx 1 - \sigma^2/\sigma_k^2 \ , \qquad (11)$$

while variables in different blocks are related according to

$$\begin{aligned} \rho_{13} &= \frac{sgn(c_1)}{\sqrt{1+(c_2^2\sigma_2^2)/(c_1^2\sigma_1^2)}} + \mathcal{O}\left(\frac{\sigma^2}{\sigma_3^2}\right) \approx -0.30 \\ \rho_{23} &= \frac{sgn(c_2)}{\sqrt{1+(c_1^2\sigma_1^2)/(c_2^2\sigma_2^2)}} + \mathcal{O}\left(\frac{\sigma^2}{\sigma_3^2}\right) \approx 0.95. \end{aligned} \qquad (12)$$

The treelet algorithm is bottom-up, and thus combines within-block variables before it merges (weaker correlated) variables between different blocks. While the order in which within-block variables are paired depends on the exact realization of the noise, the coarse scaling functions are very robust to this noise thanks to the adaptive nature of the treelets. Moreover, variables in the same block that are *statistically exchangeable* will (in the limit $n \to \infty$, $\sigma \to 0$) have the *same* weights in all scaling functions, at all levels in the tree.

For example, suppose that the noise realization is such that at level $L = 1$, we group together variables x_5 and x_6 in block B_2. A local PCA on this pair gives the rotation angle $\theta_1 \approx \pi/4$ and

$$s_1 \approx \frac{x_5 + x_6}{\sqrt{2}}, \quad d_1 \approx \frac{-x_5 + x_6}{\sqrt{2}} \ . \qquad (13)$$

The updated correlation coefficients are $\rho(s_1, x_7) \approx \rho(s_1, x_8) \approx \rho(x_7, x_8) \approx 1$; hence any of these three pairs may be chosen next. Suppose that at $L = 2$, s_1 and x_8 are grouped together. A theoretical calculation gives the rotation angle $\theta_2 \approx \arctan\left(1/\sqrt{2}\right)$ and principal components

$$s_2 \approx \frac{x_5 + x_6 + x_8}{\sqrt{3}}, \quad d_2 \approx -\frac{x_5 + x_6 - 2x_8}{\sqrt{6}} \ . \qquad (14)$$

Finally, "merging" s_2 and the remaining variable x_7 in the set B_2 leads to $\theta_3 \approx \pi/6$ and

$$s_3 \approx \frac{x_5 + x_6 + x_7 + x_8}{2}, \quad d_3 \approx -\frac{x_5 + x_6 - 3x_7 + x_8}{2\sqrt{3}} \ . \qquad (15)$$

The corresponding scaling and detail functions $\{\mathbf{v}_3, \mathbf{w}_3, \mathbf{w}_2, \mathbf{w}_1\}$ in the basis are localized and supported on nested clusters in the block B_2. In particular, the maximum variance function $\mathbf{v}_3 \approx \left[0, \ldots, 0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, \ldots, 0\right]^T$ only involves variables in B_2 with the statistically equivalent variables $\{x_5, x_6, x_7, x_8\}$ all having equal weights. A similar analysis applies to the remaining two blocks B_1 and B_3. With a tree of height $J = 7$, the treelet algorithm returns the hidden loading vectors in Eq. 7 as the three maximum variance basis vectors. Fig. 2 (center and right) shows results from a treelet simulation with a finite but large sample size. To determine the height J of the tree, we use cross-validation and choose the "best basis" with the larges variance using K vectors, where $K = 3$ is given.

Finally, to illustrate the importance of an adaptive data-driven construction, we compare treelets to [10], which suggests a *fixed* Haar wavelet transform. For example, suppose that a particular realization of the noise leads to the grouping order $\{\{x_5, x_6\}, x_8\}, x_7\}$ described above. A *fixed* rotation angle of $\pi/4$ gives the following sum coefficient and scaling function at level $L = 3$, $s_3 = \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(x_5 + x_6) + x_8\right) + x_7\right)$ and $\mathbf{v}_3{}^{\text{Haar}} = [0, \ldots, 0, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, 0, \ldots, 0]^T$, respectively. Thus, although the random variables x_1, x_2, x_3 and x_4 are statistically exchangeable and of equal importance, they have different weights in the scaling function. Furthermore, a different noise realization can lead to very different sum variables and scaling functions. Note also that *only* if p_1, p_2 and p_3 are powers of 2, and if all the random variables are grouped dyadically as in $\{\{x_5, x_6\}, \{x_7, x_8\}\}$ etc, are we able to recover the loading vectors in Eq. 7 by this method.

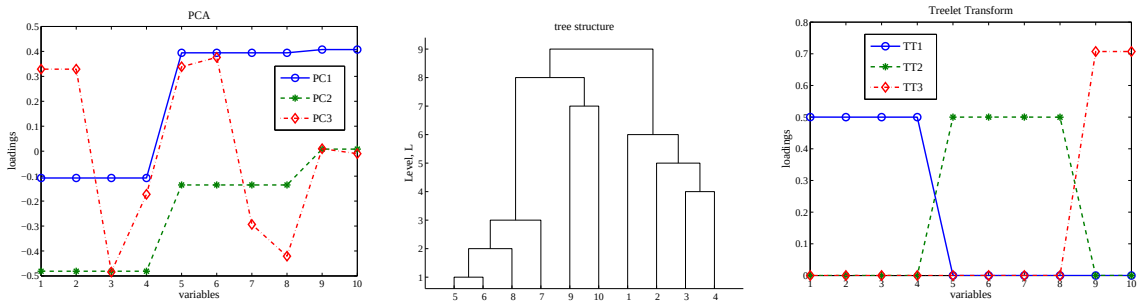


Figure 2: In Example 2, PCA fails to find the important variables in the model, while a treelet transform is able to uncover the underlying data structures. **Top:** The loadings of the first three eigenvectors in PCA. **Bottom left:** The tree structure in a simulation with the treelet transform. **Bottom right:** The loadings of the three dominant treelets.

3.2 The Treelet Transform as a Feature Selection Scheme Prior to Regression

Now consider a typical regression or classification problem with a training set $\{\mathbf{x}_i, y_i\}_{i=1}^n$, from Eqs. (5) and (6). Since the data $\mathbf{x}$ is noisy, this is an *error-in-variables* type problem. Given the finite training set the goal is to construct a linear function $f : \mathbb{R}^p \to \mathbb{R}$ to predict $\hat{y} = f(\mathbf{x})$ for a new observation $\mathbf{x}$. In typical applications, the number of variables is much larger than the number of observations ($p \gg n$). Two common approaches to overcome this problem include principal component regression (PCR) and partial least squares (PLS). Both methods first perform a global dimensionality reduction from p to k variables, and then apply linear regression on these k features. The main limitation of these global methods is that *the computed projections are noisy themselves*, see [3, 4]. In fact, the averaged prediction error of these methods has the form [4]

$$\mathbb{E}\{(\hat{y} - y)^2\} \simeq \frac{\sigma^2}{\|\mathbf{v}_y\|^2}\left[1 + \frac{c_1}{n} + \frac{c_2\ \sigma^2}{\mu\|\mathbf{v}_y\|^2}\frac{p^2}{n^2}(1 + o(1))\right] \tag{16}$$

where $\|\mathbf{v}_y\|$ is the norm of the orthogonal response vector of y (see Eq. 19 for an example), μ is a measure of the variance and covariance of the components u_i, and c_1, c_2 are both $O(1)$ constants, independent of σ, p, n. This formula shows that when $p \gg n$ the last term in (16) can dominate and lead to large prediction errors, thus emphasizing the need for robust feature selection and dimensionality reduction of the underlying noise-free data *prior* to application of learning algorithms such as PCR and PLS.

Variable selection schemes, and specifically those that choose a small subset of variables based on their individual correlation with the response y are also common approaches to dimensionality reduction in this setting. To analyze their performance we consider a more general transformation $T : \mathbb{R}^p \to \mathbb{R}^k$ defined by k orthonormal projections $\mathbf{w}_i$,

$$T\mathbf{x} = (\mathbf{x} \cdot \mathbf{w}_1, \mathbf{x} \cdot \mathbf{w}_2, \ldots, \mathbf{x} \cdot \mathbf{w}_k) \tag{17}$$

This family of transformations includes variable selection methods, where each projection $\mathbf{w}_j$ selects a single variable, as well as wavelet-type methods and our treelet transform. Since an orthonormal projection of a Gaussian noise vector in $\mathbb{R}^p$ is a Gaussian vector in $\mathbb{R}^k$, the prediction error in the new variables admits the form

$$\mathbb{E}\{(\hat{y}-y)^2\} \simeq \frac{\sigma^2}{\|T\mathbf{v}_y\|^2}\left[1 + \frac{c_1}{n} + \frac{c_2\ \sigma^2}{\mu\|T\mathbf{v}_y\|^2}\frac{k^2}{n^2}(1 + o(1))\right] \tag{18}$$

Eq. (18) indicates that a dimensionality reduction scheme should ideally preserve the signal vector of y ($\|T\mathbf{v}_y\| \simeq \|\mathbf{v}_y\|$) while at the same time representing the signals by as few features as possible ($k \ll p$). The main problem of PCA is that it optimally fits the noisy data, yielding for the noise-free response $\|T\mathbf{v}_y\|/\|\mathbf{v}_y\| \simeq (1 - C\sigma^2 p^2/n^2)$. The main limitation of variable selection schemes is that in complex settings with *overlapping* vectors $\mathbf{v}_j$, such schemes may at best yield $\|T\mathbf{v}_y\|/\|\mathbf{v}_y\| = r < 1$. However, due to high dimensionality, variable selection schemes may still achieve better prediction errors than methods that use all the original variables. If the data $\mathbf{x}$ is apriori known to be smooth continuous signals, then this feature selection can be done by wavelet compression, which is known to be asymptotically optimal. In the case of unstructured data, we propose to use treelets.

We present a simple example and compare the performance of treelets to the variable selection scheme of [13] for PLS. Specifically, we consider a training set of $n = 100$ observations from (5) in $p = 2000$ dimensions with $\sigma = 0.5$, $K = 3$ components and $y = u_1$, where $u_1 = \pm 1$ with equal probability, $u_2 = I(U_2 < 0.4)$, $u_3 = I(U_3 < 0.3)$ where $I(x)$ is the indicator of x, and U_j are all independent uniform random variables

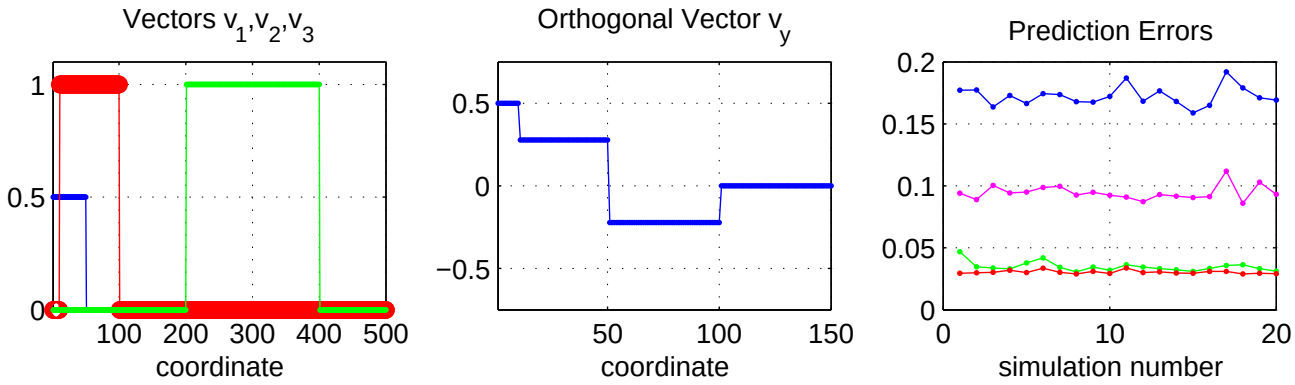


Figure 3: **Left:** The vectors $\mathbf{v}_1$ (blue), $\mathbf{v}_2$ (red), and $\mathbf{v}_3$ (green). **Center:** The vector $\mathbf{v}_y$ (only first 150 coordinates are shown, the rest are zero). **Right:** Averaged prediction errors of 20 simulation results for the methods from top to bottom: PLS on all variables (blue), supervised PLS with variable selection (purple), PLS on treelet features (green), PLS on projections onto the true vectors $\mathbf{v}_i$ (red).

in [0,1]. The vectors $\mathbf{v}_j$ are shown in figure 3 (left). In this example, the two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ overlap. Therefore, the response vector unique to y, known in chemometrics as the *net analyte signal*, is given by (see Fig. 3, center)

$$\mathbf{v}_y = \mathbf{v}_1 - \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_2\|^2}\mathbf{v}_2 \tag{19}$$

To compute $\mathbf{v}_y$, all the 100 first coordinates are needed. However, a feature selection scheme that chooses variables based on their correlation to the response will pick the first 10 coordinates and then most of the next 40. Variables numbered 51 to 100, although critical for prediction of the response $y = u_1$, are uncorrelated with it (as u_1 and u_2 are uncorrelated) and are thus *not* chosen. In contrast, even in the presence of moderate noise, the treelet algorithm correctly joins together the subsets of variables 1-10, 11-50, 51-100 and 201-400. The rest of the variables, which contain only noise are combined only at much higher levels in the treelet algorithm, as they are asymptotically uncorrelated. Therefore, using only coarse-grained sum variables in the treelet transform yields near optimal prediction errors. In Fig. 3 (right) we plot the mean squared error of prediction (MSEP) for 20 different simulations tested on an independent test set of 500 observations. The different methods are PLS on all variables (MSEP=0.17), supervised PLS with variable selection as in [13] (MSEP=0.09), PLS on the 50 treelet features with highest variance, with the level of the treelet determined by leave-one-out cross validation (MSEP=0.035), and finally PLS on the projection of the noisy data onto the true vectors $\mathbf{v}_i$ (MSEP = 0.030). In all cases, the optimal number of PLS projections (latent variables) is also determined by leave-one-out cross validation. Due to the high dimensionality of the data, choosing a subset of the original variables performs better than full-variable methods. However, choosing a subset of treelet features performs even better yielding almost optimal errors ($\sigma^2/\|\mathbf{v}_y\|^2 \approx 0.03$).

3.3 A Classification Example with an Internet-Ad Dataset

We conclude with an application of treelets on the internet advertisement dataset [14], from the UCI ML repository. After removal of the first three continuous variables, this dataset contains 1555 binary variables and 3278 observations, labeled as belonging to one of two classes. The goal is to predict whether a new sample (an image in an internet page) is an internet advertisement or not, given values of its 1555 variables (various features of the image).

With standard classification algorithms, one can easily obtain a generalization error of about 5%. For example, regularized linear discriminant analysis (LDA), with the additional assumption of a diagonal covariance matrix, achieves an average misclassification error rate of about 5.5% for a training set of 3100 observations and a test set of 178 observations (the average is taken over 10 randomly selected training and test sets). Nearest neighbor classification with $k = 1$ achieves a slightly better performance with an error rate of roughly 4%.

This data set, however, has several distinctive properties that are clearly revealed if one applies the treelet algorithm as a pre-processing step *prior* to learning: First of all, several of the original variables are *exactly* linearly related. As the data is binary (-1 or 1), these variables are either identical or with opposite values. In fact, one can reduce the dimensionality of the data from 1555 to 760 without loss of information. (Such

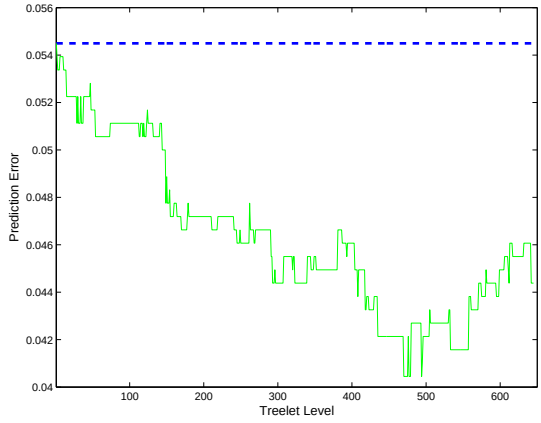


Figure 4: Averaged test classification error of LDA at different levels of the treelet algorithm (green) compared to LDA on the full uncompressed data (blue line).

a lossless compression reduces the error rate of LDA slightly, to roughly 4.8%, while the error rate of k-NN obviously remains the same). Furthermore, of these remaining 760 variables, many are highly related. There are more than 200 distinct pairs of variables with a correlation coefficient larger than 0.95. Not surprisingly, treelets not only reduce the dimensionality but also increase the predictive performance on this dataset. In figure 4 a plot of the LDA error on the 200 highest variance treelet features is shown as a function of the level of the tree. As seen from the graph, at a treelet level of $L = 450$ the error of LDA is decreased to roughly 4.2%. Similar results hold for k-NN, where the error is decreased from 4% for the full dimensional data to around 3.3% for a treelet-compressed version. The above results with treelets are competitive with recently published results on this data set using other feature selection methods in the literature [15].

3.4 Summary and Discussion

To conclude, in this paper we presented *treelets* – a novel construction of a multi-resolution representation of unstructured data. Treelets have many potential applications for dimensionality reduction, feature extraction, denoising etc, and enable use of wavelet-type methods including wavelet-packets and joint best basis, to unstructured data. In particular, we presented a few simulated examples of situations where treelets outperform other common dimensionality reduction methods (e.g. linear mixture models with overlapping loading vectors or correlated components). We have also shown the potential applicability of treelets on real data sets for a specific example with an internet-ad dataset.

Acknowledgments: The authors would like to thank R.R. Coifman and S. Lafon for interesting discussions. The research of ABL was funded in part by NSF award CCF-0625879, and the research by BN was supported by the Hana and Julius Rosen fund and by the William Z. and Eda Bess Novick Young Scientist fund.

References

[1] D.L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. In *American Math. Society Conference on "Math Challenges of the 21st Century"*, 2000.

[2] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[3] I.M. Johnstone and A.Y. Lu. Sparse principal component analysis. 2004. Submitted.

[4] B. Nadler and R.R. Coifman. The prediction error in cls and pls: the importance of feature selection prior to multivariate calibration. *Journal of Chemometrics*, 19:107–118, 2005.

[5] J. Buckheit and D.L. Donoho. Improved linear discrimination using time frequency dictionaries. In *Proc. SPIE*, volume 2569, pages 540–551, 1995.

[6] D.L. Donoho and I.M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90:1200–1224, 1995.

[7] R. T. Ogden. *Essential Wavelets for Statistical Applications and Data Analysis*. Birkhäser, 1997.

[8] L.J. van't Veer *et. al.* Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(31):530–536, 2002.

[9] R.R. Coifman and N. Saito. The local karhunen-loeve basis. In *Proc. IEEE International Symposium on Time-Frequency and Time-Scale Analysis*, pages 129–132. IEEE Signal Processing Society, 1996.

[10] F. Murtagh. The haar wavelet transform of a dendrogram - i. 2005. Submitted.

[11] G.H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[12] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

[13] E. Bair, T. Hastie, D. Paul, and R. Tibshirani. Prediction by supervised principal components. *Journal of the American Statistical Association*, 101(473):119–137, 2006.

[14] N. Kushmerick. Learning to remove internet advertisements. In *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 175–181, 1999.

[15] Z. Zhao and H. Liu. Searching for interacting features. In *Proceedings of the 20th International Joint Conference on AI (IJCAI-07)*, 2007.