Wavelet-Based Statistical Signal Processing Using Hidden Markov Models

Matthew S. Crouse,[◇] *Robert D. Nowak,*[□] *and Richard G. Baraniuk*[◇] [*]

[◇] Department of Electrical and Computer Engineering
Rice University
6100 South Main Street
Houston, TX 77005–1892
Email: mcrouse@ece.rice.edu, richb@rice.edu
Web: http://www.dsp.rice.edu
Fax: (713) 524–5237

[□] Department of Electrical Engineering
Michigan State University
260 Engineering Building
East Lansing, MI 48824–1226
Email: nowak@egr.msu.edu
Web: http://www.egr.msu.edu/spc/
Fax: (517) 353–1980

Submitted to *IEEE Transactions on Signal Processing*
(Special issue on Wavelets and Filterbanks, April 1998)
Revised: August 1997
EDICS Numbers: 2.4.4 Wavelets and Filterbanks
and 3.5 Statistical Modeling

Abstract

Wavelet-based statistical signal processing techniques such as denoising and detection typically model the wavelet coefficients as independent or jointly Gaussian. These models are unrealistic for many real-world signals. In this paper, we develop a new framework for statistical signal processing based on wavelet-domain hidden Markov models (HMMs) that concisely models the statistical dependencies and nonGaussian statistics encountered in real-world signals. Wavelet-domain HMMs are designed with the intrinsic properties of the wavelet transform in mind and provide powerful, yet tractable, probabilistic signal models. Efficient Expectation Maximization algorithms are developed for fitting the HMMs to observational signal data. The new framework is suitable for a wide range of applications, including signal estimation, detection, classification, prediction, and even synthesis. To demonstrate the utility of wavelet-domain HMMs, we develop novel algorithms for signal denoising, classification, and detection.

[*]This work was supported by the National Science Foundation, grant nos. MIP 94–57438 and MIP–9701692, and by the Office of Naval Research, grant no. N00014–95–1–0849.

1 Introduction

The wavelet transform has emerged as an exciting new tool for statistical signal and image processing. The wavelet domain provides a natural setting for many applications involving real-world signals, including estimation [1–3], detection [4], classification [4], compression [5], prediction and filtering [6], and synthesis [7]. The remarkable properties of the wavelet transform have led to powerful signal processing methods based on simple scalar transformations of individual wavelet coefficients. These methods implicitly treat each wavelet coefficient as though it were independent of all others. Methods that exploit dependencies between wavelet coefficients should perform even better. The goal of this paper is to develop new wavelet-domain probability models that match the statistical dependencies and nonGaussian statistics often encountered in practice. These new models lead to sophisticated processing techniques that coordinate the nonlinear processing amongst coefficients to outperform current wavelet-based algorithms. The models are designed with the intrinsic properties of the wavelet transform in mind.

1.1 The discrete wavelet transform

The wavelet transform is an atomic decomposition that represents a one-dimensional signal $z(t)$ in terms of shifted and dilated versions of a prototype bandpass wavelet function $\psi(t)$, and shifted versions of a lowpass scaling function $\phi(t)$ [8,9]. For special choices of the wavelet and scaling functions, the atoms

$$\begin{aligned}
\psi_{J,K}(t) &\equiv 2^{-J/2}\,\psi\Big(2^{-J}t - K\Big), && (1)\\
\phi_{J_0,K}(t) &\equiv 2^{-J_0/2}\,\phi\Big(2^{-J_0}t - K\Big), && J,K \in \mathbb{Z}
\end{aligned}$$

form an orthonormal basis, and we have the signal representation [8,9]

$$z(t) = \sum_K u_K\,\phi_{J_0,K}(t) + \sum_{J=-\infty}^{J_0}\sum_K w_{J,K}\,\psi_{J,K}(t), \qquad (2)$$

with $w_{J,K} \equiv \int z(t)\,\psi^*_{J,K}(t)\,dt$, and $u_K \equiv \int z(t)\,\phi^*_{J_0,K}(t)\,dt$.

In this representation, J indexes the *scale* or resolution of analysis — smaller J corresponds to higher resolution analysis. J_0 indicates the coarsest scale or lowest resolution of analysis. K indexes the spatial location of analysis. For a wavelet $\psi(t)$ centered at time zero and frequency f_0, the *wavelet coefficient* $w_{J,K}$ measures the signal content around time $2^J K$ and frequency $2^{-J} f_0$. The scaling coefficient u_K measures the local mean around time $2^{J_0} K$. In practice, we work with a finite-resolution representation of $z(t)$, replacing the semi-infinite sum in (2) with a sum over a finite number of scales $J_1 \le J \le J_0$, $J_1, J_0 \in \mathbb{Z}$. Furthermore, multidimensional wavelet bases for higher-dimensional data are easily formed using tensor products of wavelet and scaling functions [8,9]. To keep the notation manageable in the sequel, we will emphasize the one-dimensional case and apply an abstract single index system for wavelet atoms and coefficients: $\psi_{J,K} \to \psi_i$, $w_{J,K} \to w_i$. Figure 1 provides a graphical depiction of the time-frequency analysis affected by the wavelet transform.

In wavelet-based signal and image processing, we process the signal $z(t)$ by operating on its wavelet coefficients $\{w_i\}$ and scaling coefficients

$\{u_K\}$. The wavelet transform has several attractive properties that make it natural for signal and image processing. We call these the *primary properties* of the wavelet transform:

Locality: Each wavelet atom ψ_i is localized simultaneously in time and frequency.

Multiresolution: Wavelet atoms are compressed and dilated to analyze at a nested set of scales.

Compression: The wavelet transforms of real-world signals tend to be sparse.

Together the Locality and Multiresolution properties enable the wavelet transform to efficiently match a wide range of signal characteristics, from high-frequency transients and edges to slowly-varying harmonics. The wavelet transform's ability to match a wide variety of signals leads to the Compression property. Complicated signals can often be represented using only a handful of wavelet and scaling functions. As a result of these properties, statistical signal modeling and processing methods based in the wavelet-domain are, in many cases, much more effective than classical time-domain or frequency-domain approaches.

1.2 Wavelet-domain statistical models

In this paper, we adopt a statistical approach to wavelet-based signal processing, in which we regard the signal and its wavelet coefficients as random realizations from a family or distribution of signals. To this end, our objective is to develop probability models for the wavelet transform of signals that are rich and flexible enough to capture the structure of a wide variety of data, yet concise, tractable, and efficient for practical application in real-world problems.

Until recently, wavelet coefficients have been modeled either as jointly Gaussian [4,6,10,11], or as nonGaussian but independent [2, 3, 12, 13]. Jointly Gaussian models can efficiently capture linear correlations between wavelet coefficients. However, Gaussian models are in conflict with the Compression property, which implies the wavelet transforms of most signals are sparse, resulting in a large number of small coefficients and a small number of large coefficients. A typical wavelet coefficient density or histogram is thus much more "peaky" at zero and heavy-tailed than the Gaussian.

NonGaussian models have also been formulated, but usually the coefficients are assumed to be statistically independent of each other. Justification for independent nonGaussian models is based on the primary properties plus the interpretation of the wavelet transform as a "decorrelator" that attempts to make each wavelet coefficient statistically independent of all others. However, the wavelet transform cannot completely decorrelate real-world signals — a *residual dependency structure* always remains between the wavelet coefficients. In words, we have the following *secondary properties* of the wavelet transform:

Clustering: If a particular wavelet coefficient is large/small, then adjacent coefficients are very likely to also be large/small [14].

Persistence: Large/small values of wavelet coefficients tend to propagate across scales [15, 16].

As we see in Figure 2, these are striking features of the wavelet transform. They have been exploited with great success by the compression community [5, 14]. Our goal is to do the same for signal processing.

1.3 Modeling Framework

Completely modeling the joint probability density function of all of the wavelet coefficients, $f(\mathbf{w})$ with $\mathbf{w} = \{w_i\}$, would characterize the dependencies between the coefficients. However, the complete joint probability density is usually intractable to use and impossible to estimate. At the other extreme, modeling the wavelet coefficients as statistically independent, with $f(\mathbf{w}) = \prod_i f(w_i)$, is simple but disregards the inter-coefficient dependencies. To strike a balance between these two extremes, we must represent the key dependencies, and only the key dependencies. The primary and secondary properties of the wavelet transform suggest natural candidates: Persistence suggests that wavelet coefficients can have strong dependencies across scale (vertically in Figure 1), while Clustering and Locality suggest that coefficients can have strong dependencies within scale (horizontally in Figure 1).

In this paper, we introduce a new modeling framework that neatly summarizes the probabilistic structure of the coefficients of the wavelet transform [17]. Our models owe their richness and flexibility to the following features:

Mixture Densities: To match the nonGaussian nature of the wavelet coefficients, we model the marginal probability $f(w_i)$ of each coefficient as a mixture density with a hidden state variable (see Figure 3(a)).

Probabilistic Graphs: To characterize the key dependencies between the wavelet coefficients, we introduce Markovian dependencies between the hidden state variables. These dependencies are described by a probabilistic graph or tree (see Figure 3(b)).

Models of this type, commonly referred to as *Hidden Markov Models* (HMMs), have proved tremendously useful in a variety of applications, including speech recognition [18, 19] and artificial intelligence [20].

We will investigate three simple probabilistic graphs, with state-to-state connectivities shown in Figure 3(b). The *Independent Mixture* (IM) model leaves the state variables unconnected and hence ignores any inter-coefficient dependencies. The *Hidden Markov Chain* model connects the state variables horizontally within each scale. The *Hidden Markov Tree* (HMT) model connects the state variables vertically across scale. We will refer to these models collectively as *wavelet-domain HMMs.*

We will show that the wavelet-domain HMM framework effectively captures key wavelet coefficient characteristics, leading to computationally efficient and statistically robust wavelet-based modeling of real-world signals. This framework provides a natural setting for exploiting the structure inherent in real-world signals for estimation, detection, classification, prediction and filtering, and synthesis. In Section 5.1, we will apply this machinery to signal estimation and derive a new wavelet denoising scheme that performs substantially better than current approaches (see Figure 10 and Table 1). In Section 5.2, we will apply our models to two difficult problems in detection and classification.

Although for illustrative purposes we will focus on one-dimensional signals, the results in this paper apply to wavelet-domain HMMs for images and multi-dimensional data as well. For example, HMT models for one-dimensional signals have a natural binary tree structure, with each wavelet state connected to the two "child" wavelet states below it (see Figure 3(b)). HMT models for images have a natural quadtree structure, with each wavelet state connected to the four "child" wavelet states below it (see Figure 4). HMT models for m-dimensional data have a natural 2^m tree structure.

1.4 Related work

Our approach in this paper differs considerably from previous approaches to modeling wavelet transforms. In the signal estimation arena, research has concentrated primarily on modeling the nonGaussianity of the wavelet coefficients rather than their inter-dependencies [2, 3, 12, 13]. However, in the compression arena, techniques incorporating both coefficient nonGaussianity and inter-coefficient dependence lie at the heart of the state-of-the-art compression systems. In particular, the zerotree coder of Shapiro [5] has revolutionized wavelet image compression, significantly improving compression performance by exploiting dependencies between wavelet coefficients. Recently, new compression algorithms have been developed that combine the idea of exploiting dependencies with probabilistic models for the wavelet coefficients [21,22]. Although similar in spirit to the probability models presented in this paper, none of these new compression algorithms use an HMM framework.

Wavelet-domain HMMs also differ considerably from the multiscale stochastic models developed in [6, 10]. In these multiscale stochastic models, the wavelet coefficients themselves (rather than the hidden state variables) are modeled using a Markov structure. In addition, Basseville et al. emphasize linear Gaussian models [6]. Wavelet-domain HMMs are nonlinear, nonGaussian, and do not constrain the wavelet coefficients to be strictly Markov.

The multiscale models of [23], used for segmentation, have a Markov tree of state variables similar to that of the HMT model. However, these models are applied directly to the signal (which is not tree-structured), rather than the wavelet transform of the signal (which is tree-structured). This distinction can be visualized as removing all the dark nodes from Figure 3(b) except those at the bottom scale, which now represent signal samples instead of wavelet coefficients. Differing both in form and purpose from wavelet-domain HMMs, these models clearly do not provide feasible wavelet-domain models.

Wavelet-domain HMMs are also distinct from the traditional HMMs used in time series analysis and speech recognition. Traditional HMMs model local correlations of a signal across time. Wavelet-domain HMMs model dependencies in the two-dimensional time-frequency plane — an entirely different topology requiring new modeling techniques. In addition, wavelet-domain HMMs are designed to characterize properties of wavelet coefficients such as Clustering, Persistence, and Compression. These properties are not necessarily present in time-series data, and hence can lead to substantially different modeling assumptions than those typically used for time-series data.

After dispensing with definitions and notation in Section 2, we turn to wavelet transform modeling using HMMs in Section 3. We discuss the training of these models in Section 4. (Details of a new EM training algorithm for the HMT model are provided in the Appendix.) In Section 5, we apply this machinery to several problems in signal estimation and detection and classification. We close in Section 6 with a discussion and conclusions.

2 Preliminaries

Graphs and trees will play a central role in this paper. An undirected *graph* consists of a set of *nodes* $\{v_1, v_2, \ldots, v_N\}$ and a set of *connections* linking the nodes. A *path* is a set of connections between two nodes. A rooted *tree* is an undirected acyclic graph. In a tree there is a unique path linking any two nodes. All nodes that lie on the path from v_i to the *root* are called *ancestors* of v_i; all nodes that lie on paths from v_i away from the root are called *descendants* of v_i. The *parent* of v_i is its immediate ancestor and is denoted by $v_{\rho(i)}$. A node is a *child* of v_i if v_i is its parent. We denote the children of node v_i by $\{v_j\}_{j \in c(i)}$. A node may have several children, but only one parent; nodes with no children are called *leaves* of the tree. In a *binary* tree, each node that is not itself a leaf has two children.

When viewed in the time-frequency plane as in Figure 1, a wavelet transforms has a natural organization as a *forest* of binary trees [24].[1] The tree(s) are rooted at the wavelet coefficients in the coarsest scale (lowest frequency) band; a single scaling coefficient sits above each root. Depending on the length of the signal and the number of scale bands computed in the transform, the forest of trees will contain from one to several distinct trees (see Figures 3(b) and Figure 5). For instance, if we analyze a length-N discrete signal over L wavelet scales, we obtain $N\,2^{-L}$ wavelet trees. In our abstract indexing scheme, we will denote the i-th wavelet coefficient from the k-th tree as w_i^k.

Finally, some simple notation: When dealing with random quantities, we will use capital letters to denote the random variable and lower case to refer to a realization of this variable. We will use $p_S(s)$ to denote the probability mass function (pmf) of the discrete random variable S and $f_W(w)$ to denote the probability density function (pdf) of the continuous random variable W. We will use the shorthand iid for independent and identically distributed and will denote vectors with boldface letters.

3 Wavelet Domain Probability Models

Recall that our objective is to develop probability models for the wavelet transform that capture complex dependencies and nonGaussian statistics, yet remain tractable so that they can be applied to real-world problems. To this end, we develop our model in two steps. We begin with a simple model in which the wavelet coefficients are assumed to be independent of each other. This model is based on the primary properties of the wavelet transform and motivated by the fact that the wavelet transform "nearly" decorrelates a wide variety of signal.

Next, we extend the independent coefficient model in order to account for residual dependencies that remain between the wavelet coefficients. This extension is accomplished with simple Markovian structures on the wavelet tree. We consider Markov models across both time and scale to account for

[1] Do not confuse our use of trees with so-called tree-structured filter banks [9].

the secondary properties of the wavelet transform: Clustering and Persistence. Our structures reflect Markov dependencies between the states of the wavelet coefficients, rather than the values of the wavelet coefficients themselves (as in [6]). The tandem of marginal Gaussian mixtures and first-order Markovian dependencies leads to practical HMMs for the wavelet coefficients.

3.1 Probabilistic Models for an Individual Wavelet Coefficient

The Compression property of the wavelet transform states that the transform of a typical signal consists of a small number of large coefficients and a large number of small coefficients. More precisely, most wavelet coefficients have small values and hence contain very little signal information. A few wavelet coefficients have large values that represent significant signal information. This property leads to the following simple model for an individual wavelet coefficient. We model each coefficient as being in one of two states: "high," corresponding to a wavelet component containing significant contributions of signal energy, or "low," representing coefficients with little signal energy. If we associate with each state a probability density — say a high-variance, zero-mean density for the "high" state and a low-variance, zero-mean density for the "low" state — the result is a two-state mixture model for each wavelet coefficient.

As we see from Figure 6, the two-state, zero-mean mixture model is completely parameterized by the pmf of the state variable S, $p_S(1), 1 - p_S(1)$, and the variances of the Gaussian pdfs corresponding to each state. In most applications of mixture models, the value of the coefficient W is observed, but the value of the state variable S is not; we say that the value of S is *hidden*.

Several factors support the model's validity. Empirical results from estimation have shown the this mixture model to be simple, yet effective [2,3]. Our experience corroborates these results; in Figure 7 we demonstrate the fit that this model provides for an actual signal. Furthermore, theoretical connections have been made between wavelet coefficient mixture models and the fundamental parameters of Besov spaces — function spaces that have proved extremely useful for characterizing real-world signals [12].

For any given set of wavelet data, the two-state, zero-mean Gaussian mixture model may not provide a fit to $f_W(w)$ with the desired fidelity. To improve accuracy, we can use Gaussian mixture models with $M > 2$ states or non-zero means in the Gaussian mixing densities. By increasing the number of states and allowing non-zero means, we can make the fit arbitrarily close for densities with a finite number of discontinuities [25]. We can even mix nonGaussian densities, such as conditional densities belonging to the exponential family of distributions [26]. However, the two-state, zero-mean Gaussian mixture model is simple, robust, and easy-to-use — attractive features for many applications. For purposes of instruction, we will focus on the simple two-state model in this paper, but develop machinery capable of handling more general mixture models.

In general, an M-state Gaussian mixture model for a random variable W consists of

1. a discrete random state variable S taking the values $s \in 1, 2, \ldots, M$ according to the pmf

$p_S(s)$.

2. the Gaussian conditional pdfs $f_{W|S}(w|S = s)$, $s \in 1, 2, \ldots, M$.

To generate a realization of W using the mixture model, we first draw a state value s according to $p_S(s)$ and then draw an observation w according to $f_{W|S}(w|S = s)$. The pdf of W is given by

$$f_W(w) = \sum_{m=1}^{M} p_S(m)\, f_{W|S}(w|S = m). \qquad (3)$$

Although each wavelet coefficient W is conditionally Gaussian given its state variable S, the wavelet coefficient has an overall nonGaussian density due to the randomness of S.

Unlike wavelet coefficients, scaling coefficients typically are not zero mean. Therefore, a two-state, zero-mean Gaussian mixture model may be inappropriate. One approach is to model the scaling coefficients as Gaussian with non-zero mean. Since scaling coefficients are essentially weighted averages of a large number signal samples, this approximation is reasonable in light of the Central Limit Theorem. A more flexible approach is to apply a Gaussian mixture model as before, but with non-zero-mean mixing densities.

3.2 Probabilistic Models for a Wavelet Transform

3.2.1 Motivation

Since a Gaussian mixture model can accurately characterize the pdf of a single wavelet coefficient, it seems logical to use Gaussian mixture models to characterize the joint pdf of the entire wavelet transform. The simplest approach would be to model the wavelet coefficients as independent Gaussian mixtures. We call this approach the *Independent Mixture (IM) model* (see Figure 3(a)). Because the wavelet transform nearly decorrelates a wide variety of signals, this model for the wavelet tree is intuitively plausible. Moreover, as demonstrated by the denoising results in [2, 3], the IM model is a substantial improvement over deterministic signal models that do not explicitly take the distribution of signal's wavelet coefficient values into account.

Nevertheless, the Clustering and Persistence properties lead to local dependencies between wavelet coefficients. Characterization of these dependencies has resulted in significant performance gains in compression [5, 14]. Ideally, we would like a model that both matches each individual coefficient's pdf and captures dependencies between coefficients.

We motivate our approach by extending the Gaussian mixture model for one wavelet coefficient to jointly model two wavelet coefficients that represent components of the signal close in time and/or scale. We say that two such coefficients are *neighbors*. By Clustering and Persistence, if one coefficient is in a high-variance (low-variance) state, then its neighbor is very likely to also be in a high-variance (low-variance) state. Thus, the two neighboring wavelet coefficients can be modeled as Gaussian mixtures with *inter-dependent state variables*. This two-coefficient example suggests a natural generalization to the multiple coefficients in a wavelet transform: model each coefficient as a Gaussian mixture, but allow probabilistic dependencies between the state variables of each mixture.

What remains is to specify an appropriate model for these dependencies between the state variables. A complete joint pdf taking into account all possible dependencies is clearly intractable, since the number of different state variable combinations grows exponentially in the number of wavelet coefficients. Fortunately, the Locality and Multiresolution properties of the wavelet transform suggest that dependencies die off quickly as we move away from the local neighborhood about a coefficient of interest. Hence, very accurate and practical models can be obtained with probabilistic links between the states of only neighboring wavelet coefficients. We will now apply probabilistic graph theory [20,27,28] to develop these models.

3.2.2 Graph models for wavelet transforms

Probabilistic graphs are useful tools for modeling the local dependencies between a set of random variables [20, 27, 28]. Roughly speaking, a probabilistic graph associates each random variable with a node in a graph; dependencies between pairs of variables are represented by connecting the corresponding nodes. The Locality and Multiresolution properties of the wavelet transform suggest three simple ways to "connect the dots" representing the wavelet coefficients and states in Figure 1: (1) a graph with no dependencies between wavelet state variables, (2) a graph linking wavelet state variables across time using chains, and (3) a graph linking wavelet state variables across scale using trees. In Figure 3, we illustrate these three simple graphs.

We are by no means limited to just these three graphs. We can develop graphs that capture even more inter-dependencies by placing additional connections between the states. Unfortunately, the computational complexity increases substantially for graphs more complicated than trees. Although we can still formulate algorithms for training and applying more general graphs [20, 28], to keep our presentation and analysis simple we will concentrate on the three special cases described in Figure 3:

Independent Mixture (IM) Model:

A mixture model with no connections, as in Figure 3(a), corresponds to the IM presented in [2, 3] and discussed above. It treats wavelet state variables (and hence wavelet coefficients) as independent random variables.

Hidden Markov Chain Model:

Connecting the state variables S_i horizontally in Figure 3(b) specifies a Markov-1 chain dependency between the state variables *within each scale* [18]. This model treats wavelet state variables as dependent within each scale, but independent from scale to scale.

Hidden Markov Tree (HMT) Model:

By connecting state variables vertically *across scale* in Figure 3(b), we obtain a graph with tree-structured dependencies between state variables. We call this new model a "tree model" to emphasize the underlying dependencies between parent and child state variables.

We will focus on the IM and HMT models in the sequel.[2]

[2]The IM and HMT models and their respective training algorithms generalize directly to multi-dimensional data. However, the chain model must be adapted to higher dimensions. For example, in 2-d the chain becomes a hidden Markov field, and more complicated training algorithms are required for this topology. See [28] for a discussion of possible algorithms and [29] for an alternative modeling approach potentially useful for reducing complexity.

3.2.3 HMT model

The HMT model matches both the Clustering and Persistence properties of the wavelet transform. Its structure is reminiscent of the zerotree wavelet compression system [5], which exploits tree-structured dependencies for substantial compression gains. Furthermore, this graph has a natural parent-child dependency interpretation. State variable dependencies are modeled via state transition probabilities from each parent state variable S_i to its childrens' states, the two state variables connected to it from below (if they exist). For example, in Figure 5(a), state variables S_4 and S_5 are both children of S_2, and hence causally dependent on S_2. Dependency is not simply limited to parent-child interactions, however. State variables S_4 and S_5 may be highly dependent due to their joint interaction with S_2.

Moreover, if we do not restrict ourselves to zero-mean, two-state Gaussian mixture model, but rather use Gaussian mixture models with more than two states and non-zero means, this simple tree-structure is capable of approximating the joint parent-child wavelet coefficient pdf to arbitrary precision.[3]

[3]This approximation capability improves to arbitrary precision as we increase the number of mixture components, M, used to model the marginal parent and child distributions. A sketch of the proof follows. Recall the components of the parent-child model. The parent is modeled using M Gaussian univariate mixing densities and an M-vector of probabilities that weight the densities. Conditioned on the parent state variable, the child wavelet coefficient is modeled by its own M Gaussian univariate densities and an $M \times M$ matrix of probabilities for transitions from the parent's state to the child's state. The joint model for parent and child is thus a mixture of M^2 bivariate Gaussians, which are the Cartesian products of the univariate mixing densities. The mixture weights for these M^2 bivariate Gaussians are products of the parent state probabilities with the $M \times M$ matrix of transition probabilities. Hence we have M^2 degrees of freedom in choosing the mixture weights and we can realize any weighted mixture of the M^2 bivariate Gaussians we desire. Appealing to the approximation capabilities of Gaussian mixtures [25] (analogous to radial basis function networks [30]), as M increases the bivariate Gaussian mixture is capable of approximating any bivariate parent-child pdf with a finite number of discontinuities to arbitrary precision.

Using an M-state Gaussian mixture model for each wavelet coefficient W_i, the parameters for the HMT model are:

1. $p_{S_1}(m)$, the pmf for the root node S_1.
2. $\epsilon_{i,\rho(i)}^{mr} = p_{S_i|S_{\rho(i)}}\left(m \middle| S_{\rho(i)} = r\right)$, the conditional probability that S_i is in state m given $S_{\rho(i)}$ is in state r.
3. $\mu_{i,m}$ and $\sigma_{i,m}^2$, the mean and variance, respectively, of the wavelet coefficient W_i given S_i is in state m.

These parameters can be grouped into a model parameter vector $\boldsymbol{\theta}$. Recall that we will primarily focus on the case $M = 2$ with the means $\mu_{i,m} = 0$.

In the HMT model, we have the following conditional independence relationships among the wavelet coefficients $\{W_i\}$. First, we observe that

$$\begin{aligned} f_{W_i}\left(w_i \middle| \{W_j\}_{j \neq i}, \{S_j = s_j\}_{j \neq i}, S_i = s_i\right) \\ = \; f_{W_i}\left(w_i | S_i = s_i\right). \end{aligned} \qquad (4)$$

In words, W_i is conditionally independent of all other random variables given its state S_i. Hence, the conditional independence properties for the states also lead to conditional independence properties for the wavelet coefficients. We next investigate the independence properties for the wavelet coefficients and wavelet states in tandem.

The tree-structured connections lead to several conditional independence relationships for the states and wavelet coefficients. Given the parent state $S_{\rho(i)}$, the pair of nodes $\{S_i, W_i\}$ are independent of the entire tree except for S_i's descendants. Conversely, given the child state $S_{j,\ j \in c(i)}$, the pair $\{S_i, W_i\}$ are independent of S_j's descendants. Combining these properties shows us that $\{S_i, W_i\}$ are conditionally independent of the entire tree given only the parent state $S_{\rho(i)}$ and the children states $\{S_j\}_{j \in c(i)}$.

In Figure 5(a) we can see concrete examples of these independence properties. Given the parent S_1, the pair of nodes $\{S_2, W_2\}$ are conditionally independent of the subtree rooted at S_3. Effectively, conditioning on the parent state separates $\{S_2, W_2\}$ from the right side of the tree. Conversely, given the child S_4, the pair $\{S_2, W_2\}$ are conditionally independent of the subtrees rooted at S_8 and S_9; given the other child S_5, $\{S_2, W_2\}$ are conditionally independent of the subtrees rooted at S_{10} and S_{11}. Applying these together, we see that that given the parent S_1 and children $\{S_4, S_5\}$, the pair $\{S_2, W_2\}$ are conditionally independent of the rest of the tree.

It is important to note that the Markov structure is on the states of the wavelet coefficients, not on the coefficients themselves. This is an important distinction between our model and other multiscale Markov signal representations such as those considered in [6, 10]. Because the states are never known exactly, our HMM framework does not place a Markov structure on the wavelet coefficients directly. Let $J(i)$ denote that the scale of W_i (and S_i), and assume that the scales are ordered from finest, $J = 1$, to coarsest, $J = L$. In our model

$$f_{W_i}\left(w_i \middle| \{W_l\}_{J(l)>J(i)}\right) \neq f_{W_i}\left(w_i \middle| W_{\rho(i)}\right). \qquad (5)$$

However, even though the wavelet coefficients are generally not Markov, signal processing using wavelet-domain HMMs remains efficient due to the Markov nature of the wavelet state variables.

The question remains of how to capture interdependencies in the scaling coefficients. One simple possibility is to use a jointly Gaussian model for the scaling coefficients; this captures correlations between scaling coefficients, but treats them as independent of the wavelet coefficients. If instead we use Gaussian mixture models, then the simplest approach is to treat the scaling coefficients as independent mixtures as shown in Figure 5(b). To capture scaling coefficient correlations, we can link the states of the scaling coefficients across time in a hidden Markov chain model. Alternatively, to capture dependencies between wavelet and scaling coefficients, we can link the states of the scaling coefficients across scale with those of the coarsest scale wavelet coefficients. Other possibilities exist, but the above models are natural and straightforward to implement. The question of which model is best, or even whether any model is necessary, depends on the application at hand. In wavelet denoising, for example, the noisy scaling coefficients typically contain much more signal energy than noise energy; hence, they can be viewed as reliable estimates of the signal scaling coefficients and left unprocessed [1, 3].

3.2.4 Three standard problems of HMMs

There are three canonical problems associated with the wavelet-domain HMMs we have described [18]:

Training: Given one or more sets of observed wavelet coefficients $\{w_i\}$, determine the wavelet-domain HMM parameters $\boldsymbol{\theta}$ that best characterize the wavelet coefficients.

Likelihood Determination: Given a fixed wavelet-domain HMM with parameters $\boldsymbol{\theta}$, determine the likelihood of an observed set of wavelet coefficients $\{w_i\}$.

State Estimation: Given a fixed wavelet-domain HMM with parameters $\boldsymbol{\theta}$, determine the most likely sequence of hidden states $\{s_i\}$ for an observed set of wavelet coefficients $\{w_i\}$. This is useful for problems such as segmentation (see [23]), where the hidden states represent a physically meaningful quantity.

We next focus on training and likelihood determination, since they are crucial for the applications that we develop in Section 5.

4 Training and Likelihood via the EM Algorithm

In training, we seek the parameters of a wavelet-based HMM that best fit a given set of data. The training data $\mathbf{w} = \{w_i\}$ consists of the wavelet coefficients of an observed signal(s); the model parameters $\boldsymbol{\theta}$ consist of the mixture state probabilities and the mean and variance of each Gaussian component. For parameter estimation, we apply the maximum likelihood (ML) principle. ML estimates are asymptotically efficient, unbiased, and consistent as the number of training observations increases.

Direct ML estimation of model parameters $\boldsymbol{\theta}$ from the observed data $\mathbf{w}$ is intractable, since in estimating $\boldsymbol{\theta}$ we are characterizing the states $\mathbf{S} = \{S_i\}$ of the wavelet coefficients $\mathbf{w}$, which are unobserved (hidden).[4] Yet, given the values of the states, ML estimation of $\boldsymbol{\theta}$ is simple (merely ML estimation of Gaussian means and variances). Therefore, we employ an iterative *Expectation Maximization* (EM) approach [31], which jointly estimates both the model parameters $\boldsymbol{\theta}$ and probabilities for the hidden states $\mathbf{S}$ given the observed wavelet coefficients $\mathbf{w}$. In the context of HMMs, the EM algorithm is also known as the *Baum-Welch* algorithm.

4.1 EM algorithms for Training

Our discussion of EM algorithms focuses on the specific problem of parameter estimation in wavelet-based HMMs; for a more general treatment, see [31]. We begin with some terminology. The *incomplete* data is our training data $\mathbf{w}$, and the *complete* data $(\mathbf{w}, \mathbf{s})$ is our training data augmented with the hidden states $\mathbf{s}$. Our goal is to maximize the incomplete log-likelihood function $\ln f(\mathbf{w}|\boldsymbol{\theta})$. The EM algorithm decouples this difficult maximization into iteration between two simpler steps: the E step and the M step.

At the lth iteration, the E step calculates the expected value $E_{\mathbf{S}}\left[\ln f\left(\mathbf{w}, \mathbf{S}|\boldsymbol{\theta}\right) \middle| \mathbf{w}, \boldsymbol{\theta}^l\right]$. The M step then maximizes this expression as a function of $\boldsymbol{\theta}$ to obtain $\boldsymbol{\theta}^{l+1}$ for the next iteration. Under mild conditions, this iteration converges to a local maximum of the log-likelihood function $\ln f(\mathbf{w}|\boldsymbol{\theta})$ [31]. Efficient EM algorithms for HMMs exist under the assumption that the underlying probabilistic graph is *chordal* [20, 28]. A graph is chordal if all cycles of length greater than 3 have a chord.[5] Since the HMMs considered in this paper do not contain cycles, they are trivially chordal and hence admit efficient EM training.

[4] Since the states are not observed, we will generally denote them using capital letters to emphasize their uncertain or random nature.

[5] A *cycle* of a graph is a path starting and ending at the same node — a closed-loop. A *chord* is a connection between two non-consecutive nodes in a cycle.

For HMMs, the complexity of one EM iteration is linear in the number of observations [18,28]. With intelligent initialization, the algorithm can converge in as few as ten iterations for a simple two-state HMT model. However, as the graph model underlying an HMM becomes more complicated, each iteration of the EM algorithm becomes more computationally intensive (still linear complexity, but with a large constant factor), and the algorithm may take longer to converge. Hence, it is important to keep the HMM as simple as possible. See [29] for a context-based approach to reduce complexity in HMMs, yet still model key wavelet-domain dependencies.

The specific EM steps for the IM and hidden Markov chain models have been developed thoroughly in [18, 26], so we do not include them in this paper. For more general tree models, Ronen et al. provide specific EM steps for discrete variables in [32]. Since the observed wavelet data in the HMT model is continuous-valued, we provide a new EM algorithm for this model in the Appendix.

4.2 Likelihood Determination

The E step of the EM algorithm is useful in its own right, since it calculates $\ln f(\mathbf{w}|\boldsymbol{\theta})$, the log-likelihood of the observed data given the model. This likelihood calculation basically measures how well the model $\boldsymbol{\theta}$ describes the data $\mathbf{w}$. Hence, it is useful for detection and classification applications, as we will see in Section 5.2. The calculation can also be used to predict or estimate the values of wavelet coefficients given the model.

4.3 Robust Training via Tying

HMMs are very rich models; thus we must ensure that we have enough training data to prevent "overfitting." By averaging over only one or very few signal observations, we cannot expect to robustly estimate the marginal densities of the wavelet coefficients, let alone a joint density for the entire wavelet transform. This brings us to a key problem: If limited training data are available, how can we make our modeling more robust? We do so by modeling random variables that have similar properties using a common density or a common set of density parameters. For example, if we expect two random variables to have roughly the same variability, we can describe them with a common variance. In this way, we obtain more reliable parameter estimates by increasing the amount of training data associated with each parameter. This practice is known as "tying" in the HMM literature [18], and we use it to more robustly estimate the means, variances, and transition probabilities of our wavelet-domain HMMs.

In Figure 8, we distinguish between two different types of tying in the HMT model, tying between wavelet trees and tying within wavelet trees. Recall from Section 2 (see also Figure 3(b) and Figure 5) that in general, the wavelet decomposition of even a single signal observation can result in multiple wavelet trees. By tying *across* trees — which assumes that the coefficients of these trees have the same density — we can train as if we had multiple signal observations. We can also tie *within* trees — by tying all coefficients within the same scale of a tree, for example. In the Appendix, we discuss both types of tying for training HMT models.

5 Applications

Our development of wavelet-domain HMMs has been motivated by the intrinsic properties of the wavelet transform, and we have discussed how several aspects of the model are supported by empirical and theoretical evidence. However, the true test of our modeling framework lies in its application to real signal processing "benchmark" problems. To this end, we consider applications in signal estimation and detection/classification.

We compare the estimation performance of our new models for signal estimation in additive noise to state-of-the-art wavelet denoising methods. We show that our new framework offers significant improvements in several well-studied benchmark problems. Wavelet-domain HMMs are also well-suited to signal detection and classification.

In this Section, we approach these problems by assuming that no prior signal models are available and that only "training" data are available for the design of the detector/classifier. We compare the wavelet-domain HMM-based detectors to classical detectors. Our results demonstrate the HMM's high performance and extremely efficient use of training data in two difficult signal detection problems.

5.1 Signal Estimation

Wavelets have proved remarkably successful for estimating signals in additive white Gaussian noise (WGN) [1, 3]. The Compression property indicates that the wavelet transform typically compacts signals into just a few coefficients of large magnitude. Because the wavelet transform is orthogonal, it leaves white noise evenly distributed across many coefficients of small magnitude. Therefore, by setting small wavelet coefficients to zero, one effectively removes noise without degrading the signal (see Figure 9(a)).

Existing denoising methods usually ignore possible dependencies between signal wavelet coefficients, and hence these methods do not exploit key Clustering and Persistence properties. In this Section, we illustrate the power of the HMT model by developing a novel signal denoising method based on this framework. The new denoising method coordinates the noise removal among the wavelet coefficients and automatically adjusts to subtle structure within the signal [17] (see Figure 9(b)).

Consider the problem of estimating a length-N signal $\mathbf{x}$ in zero-mean white Gaussian noise with power σ_n^2. Taking the L-scale wavelet transform of the noisy signal, we obtain $K = N\,2^{-L}$ trees of noisy wavelet coefficients $\{w_i^k\}$ (see Section 2). Since the orthogonal wavelet transform of zero-mean white Gaussian noise is zero-mean white Gaussian noise of the same power, the estimation problem can be expressed in the wavelet domain as

$$w_i^k \;=\; y_i^k + n_i^k, \qquad (6)$$

where w_i^k, y_i^k, and n_i^k denote the wavelet coefficients of the observed data, the signal, and the noise, respectively.

Our approach is succinctly described as follows. We first fit a HMT model to the y_i^k's from the noisy data and then use this model as a prior signal distribution to compute the conditional mean estimates of the y_i^k's given w_i^k. In effect, this approach is an "empirical" Bayesian estimation procedure under squared-error loss. It is empirical, since we estimate the parameters of our Bayesian prior from the data itself (see Figure 9(b)). To fit an HMT to the noisy wavelet coefficients, we apply the EM algorithm from the Appendix. We begin by estimating the parameters $\left\{p_{S_1}(m),\ \epsilon_{i,\rho(i)}^{mr},\ \sigma_{i,m}^2\right\}$ for the *signal* wavelet coefficients using the *noisy* signal observation.[6]

The key observation is that if the signal has a wavelet-domain HMM pdf, then the noisy signal does as well. This observation stems from two facts. First, the sum of independent Gaussian random variables is also Gaussian, with variance the sum of the two variances. Second, given the values of their hidden state variables, the signal wavelet coefficients are Gaussian. Therefore, adding the independent zero-mean white Gaussian noise n_i^k increases each mixture model variance $\sigma_{i,m}^2$ by σ_n^2, but leaves the other parameters unchanged. Hence, we can obtain the signal wavelet model from the noisy signal by fitting an HMM to the noisy signal wavelet coefficients and then subtracting the added variance due to noise. If we denote the mixture variance of the noisy wavelet coefficient at location i in the mth state as $\gamma_{i,m}^2$, then

$$\sigma_{i,m}^2 = \left(\gamma_{i,m}^2 - \sigma_n^2\right)_+, \tag{7}$$

with $(x)_+ = x$ for $x \geq 0$ and $(x)_+ = 0$ for $x < 0$. The noise power σ_n^2 can be estimated using the median estimate of [1] performed on the finest scale wavelet coefficients (where the signal energy is expected to be negligible).

Of course, we typically have only a single noisy signal observation at hand. Therefore, in order to insure reliable parameter estimation for the signal we must "share" statistical information between related wavelet coefficients. We accomplish this by assuming that all wavelet coefficients and state variables within a common scale are identically distributed, including identical parent-child state transition probabilities. (This model corresponds to tying both within and across trees from Section 4.3). The resulting HMT model is completely parameterized by two mixture variances for the wavelet coefficients at each scale, two probabilities for the root state variable at the coarsest scale, and 2×2 state transition probability matrices for the state variables at all other scales.

Once we have trained the HMT, estimation of the true signal wavelet coefficients (denoising) is straightforward. Note that if the states S_i^k of the signal wavelet coefficients y_i^k are known, then the estimation problem becomes a series of simple one-dimensional problems of estimating zero-mean Gaussian random variables in zero-mean additive Gaussian noise. The conditional mean estimate of y_i^k, given w_i^k and the state s_i^k, is

$$E\left[Y_i^k \middle| W_i^k = w_i^k, S_i^k = m\right] \;=\; \frac{\sigma_{i,m}^2}{\sigma_n^2 + \sigma_{i,m}^2}\, w_i^k. \tag{8}$$

Now recall that by-products of the EM algorithm are the hidden state probabilities $p\left(S_i^k \middle| \mathbf{w}^k, \boldsymbol{\theta}\right)$ given the model and the observed wavelet coefficients. (See the Appendix for how these probabilities are calculated.) Using these state probabilities, we obtain conditional mean estimates for y_i^k via the chain rule for conditional expectation

$$E\left[y_i^k \middle| \mathbf{w}^k, \boldsymbol{\theta}\right] \;=\; \sum_m p\left(S_i^k = m \middle| \mathbf{w}^k, \boldsymbol{\theta}\right) \times \frac{\sigma_{i,m}^2}{\sigma_n^2 + \sigma_{i,m}^2}\, w_i^k. \tag{9}$$

The final signal estimate (denoised signal) is computed as the inverse wavelet transform of these estimates of the signal wavelet coefficients. Note that

[6] As in [2, 3], we assume that the wavelet coefficients are zero-mean. The scaling coefficients, though not zero-mean, are relatively noise-free and hence are not processed.

only the wavelet coefficients are processed. The original scaling coefficients are used in the inverse transform.

We now compare our "empirical" Bayesian denoising procedure using the IM and HMT with current state-of-the-art wavelet denoising algorithms.[7] Table 1 compares the estimation performance of the IM and the HMT models with two state-of-the-art scalar algorithms. Donoho and Johnstone's SureShrink algorithm [1] performs scalar thresholding in the wavelet domain. The Bayesian mixture algorithm of Chipman et al. [3] operates in a similar fashion to the denoising method using the IM model, except that their mixture model is a true Bayesian prior and is not inferred from the data. Mean-squared-error (MSE) results are tabulated for denoising Donoho and Johnstone's length-1024 test signals Bumps, Blocks, Doppler, and Heavisine [1] in additive white Gaussian noise of power $\sigma_n^2 = 1$. Inspection of Table 1 shows that significant MSE gains can be achieved by exploiting wavelet-domain dependencies via the HMT model. The only exception is the Heavisine signal, which has less wavelet-domain structure for the HMT model to exploit. In this case, the IM and HMT model perform roughly equivalently.

Figure 10 illustrates the subjective improvement[8] of the HMT model for denoising a signal realization of the Doppler signal in white Gaussian noise of power $\sigma_n^2 = 2.25$. We see that the HMT denoising method offers two significant advantages over the other methods: (1) HMT denoising is often smoother than both SureShrink and IM, and (2) HMT denoising preserves the high-frequency components at the beginning of the signal better than the other methods. This demonstrates how exploiting the statistical dependencies between wavelet coefficients enables HMT denoising to better separate signal from noise — even in regions where signal and noise are visually indistinguishable.

5.2 Signal Detection and Classification

5.2.1 Motivation

Our marriage of wavelet transforms and HMMs yields a flexible framework for generalized likelihood-based signal detection and classification that both matches the properties of the wavelet transform and exploits the structure inherent in real-world signals. Given iid signal observations from two or more classes of signals, we can train HMMs for each class c, resulting in parameter vectors $\boldsymbol{\theta}_m$. We use the trained HMMs to detect or classify a new signal observation $\mathbf{w}$ by determining which describes the new observation best. This task boils down to computing the likelihood of the new signal observation for each HMM $f(\mathbf{w}|\boldsymbol{\theta}_m)$ and then selecting the class m whose HMM provides the greatest likelihood. This approach is analogous

[7] For each estimation algorithm, Bumps was transformed using the Daubechies-4 wavelet, Blocks using the Haar wavelet, and Doppler and Heavisine using the Daubechies-8 most-nearly-symmetric wavelet. The IM and HMT algorithms used a seven-scale wavelet decomposition. The error results of Table 1 for SureShrink and the Bayesian algorithm of Chipman et al. were quoted from [3]. More details of these two algorithms are provided in [1, 3]. Error results for IM and HMT were obtained by averaging over 1000 trials. For Figure 10, SureShrink was implemented using the "hybrid" shrinkage estimator in the WaveLab software. The Bayesian mixture algorithm [3] was not implemented for Figure 10, but is similar to IM both in its Bayesian formulation and MSE performance.

[8] Results vary depending on the noise realization, and there can be no guarantee of smoothness with any MSE-based optimality criterion. However, the figure is a typical example of the subjective improvements that can result from exploiting dependencies between wavelet coefficients.

to the use of HMMs for speech recognition [19], where each signal class is a specific word or utterance. A slightly different approach developed for time-domain HMMs has been shown to be asymptotically optimal in the Neyman-Pearson sense for two-class problems [33].

Several other wavelet-based detection and classification schemes have been proposed [4,24,34,35]. Our purpose is not to provide a comprehensive review of wavelet-based detection algorithms, but rather to demonstrate the potential of the new wavelet-domain HMM framework for signal detection and classification. Note, however, that this approach is quite different from the other wavelet-based detection schemes mentioned above.

The properties of the wavelet transform make our framework particularly appropriate for the classification and detection of real-world signals. To demonstrate the power and potential of wavelet-domain HMMs for signal classification, we tackle two difficult problems — classification of nonlinear processes and change detection. These problems arise in many applications, including sonar and radar, machinery and process monitoring, and biomedical signal analysis. We do not suggest that this framework is the optimal one for either specific problem, rather we chose these two examples to demonstrate the flexibility and adaptability of the approach. In situations where the data is known to obey a simple probability model, then optimal detection and classification methods should be used. However, in complicated real-world applications where the only prior information is a set of training data, our approach offers a useful framework for detection and classification. In combination, wavelet HMMs and training data provide an efficient and powerful framework for generalized likelihood ratio testing. Both examples considered here are binary hypothesis problems, but the framework is applicable to multiple hypothesis testing as well.

In these examples, we applied a Haar transform with a single wavelet tree and a single scaling coefficient. We modeled the wavelet coefficients using two-component ($M = 2$) IM and HMT models with non-zero mixture means. These models were trained using multiple signal observations (without tying). We did not model the scaling coefficient, since it provides the global mean of the signal, which in both examples was the same under the two hypotheses. In other scenarios the scaling coefficient(s) may provide vital discriminating information.

For the purposes of illustration, we only considered a very special form of the wavelet-domain HMM framework — one based on the Haar transform and two-component mixture models. Although the Haar transform is appropriate for edge detection, different wavelet transforms may work better for other applications. Generally, we expect the transform that best compresses the signals of interest to provide the best performance. In addition, we could use more flexible models with $M > 2$ mixture components, but in using such models we would risk "overfitting" the training data.

5.2.2 Classification and Detection of Nonlinearity

For the purposes of demonstration, we have designed a numerical experiment that captures many of the nuances that make nonlinearity classification/detection so difficult. We consider two classes of random processes described mathematically by:

$$\text{I:} \quad x_1(t) = ax_1(t-1) + n_1(t)$$

$$\text{II:} \quad x_2(t) = y_2(t) + 0.2y_2^3(t),$$
$$\text{with } y_2(t) = by_2(t-1) + n_2(t)$$

Both n_1 and n_2 are white Gaussian noise processes, and the autoregressive (AR) parameters a and b are iid and uniform over the interval $(0.4, 0.8)$. The signals are discrete-time and organized into signal vectors of length 128 with $(t = 1, 2, \ldots, 128)$. Class I signals are linear AR(1) processes. Class II signals are produced by passing linear AR(1) processes through a memoryless cubic nonlinearity. Examples of signals from each class are shown in Figure 11 (generated with the same AR parameter and white noise excitation for comparison).

The first task at hand is to train wavelet-domain HMMs for the two classes based on labeled observations from each class. We generated N_T iid AR signals from each class for training purposes. (Note: the AR parameter was varied independently for each realization.) For comparison, we constructed a minimum-probability-of-error quadratic detector under the assumption that the two classes have Gaussian distributions with different means and covariances [36], with the means and covariances estimated from the training data. The quadratic detector is not optimal, since the second class is nonGaussian. In cases where the number of training observations N_T was smaller than the dimension of the observations, we formed the quadratic detector in the subspace spanned by the training data.

After training the classifiers, we tested their performance with 1000 additional iid observations from each class. To obtain reliable estimates of the error rates, we repeated the training and testing procedure 10 times in each case. The error rates for the IM model, HMT model, and quadratic detector, as a function of the number of training vectors N_T from each class, are shown in Figure 12.

Given a limited amount of training data, the quadratic detector had a difficult time distinguishing the classes and thus offers very poor performance. In contrast, the wavelet-domain HMMs make much more efficient use of the training data. With only 128 training vectors from each class, the performances of the HMMs have stabilized to their minimum error rates. Additional training data does not increase their performance. The performance of the quadratic detector does improve as N_T increases, but requires nearly 10 times the amount of training data that the HMMs require for the same error rate. We see that asymptotically (in number of training data N_T) the quadratic detector has the best error rate, followed closely by the HMT model. The IM model has the worst asymptotic error performance. This demonstrates the performance gains associated with the HMT model. Also, this suggests that more complex wavelet-domain HMMs (that is, more probabilistic connections between states) may provide asymptotic performances that meet or even exceed that of the quadratic detector. Of course, more complex HMMs will also require more training data to achieve such performance. These and related issues are currently under investigation.

5.2.3 Detection of an Abrupt Change

In this example, we consider the following two-class problem. Class I consists of random discrete-time processes with an arbitrary mean value and additive white Gaussian noise. Class II consists of random discrete-time processes with an abrupt arbitrary change in the mean at some arbitrary point in the signal. Again, our signal observations are organized into length-128 observation vectors. Formally, our signal classes are defined by:

$$\text{I:} \quad x_1(t) = a_1 + n_1(t)$$

$$\text{II:} \quad y_2(t) = a_2 I_{t\in\{1,\ldots,\tau\}} + b_2 I_{t\in\{\tau+1,\ldots,128\}} + n_2(t)$$

Both n_1 and n_2 are white Gaussian noise processes. a_1, a_2, and b_2 are iid and uniform on $[-1, 1]$. $I_{t\in\{1,\ldots,\tau\}} = 1$ if $t \in \{1, \ldots, \tau\}$ and is zero otherwise. $I_{t\in\{\tau+1,\ldots,128\}}$ is defined in an analogous fashion. The change-point τ is uniformly distributed over the integers $\{16, \ldots, 112\}$. Examples of signals from each class are shown in Figure 13.

An excellent treatment of classical methods for the detection of abrupt changes is given in [37]. In addition, other wavelet-based approaches to the change point problem have been discussed in the literature [34, 35]. The purpose of this example is not to make an exhaustive comparison between our method and other existing techniques in the literature; rather, the intent is simply to demonstrate the versatility of the wavelet-based HMM approach to signal detection.

We again designed the wavelet-domain HMM classifiers (Haar-based IM model and Haar-based HMT model) with training data from each class, and then tested their performance with 1000 additional iid observations from each class. The error rates for the IM model and HMT model as a function of the number of training vectors M from each class, are shown in Figure 14. For comparison, we also implemented an "off-line" generalized likelihood ratio test (GLRT) detector [37]. Although the signal parameters (position and size of change) and noise variance are estimated from the data itself, the GLRT detector makes full-use of piecewise constant signal model and Gaussian noise model. Since the GLRT exploits all prior information, we may view the performance of the GLRT detector as a lower bound on detection performance in this case. Note that with just a few hundred training samples, the performances of the wavelet-domain HMM detectors approaches that of the GLRT detector.

Clearly the HMMs are not the optimal detector in this problem. With precise knowledge of the problem at hand, more efficient detectors such as the GLRT are easily designed. However, this experiment again demonstrates the utility of the wavelet-domain HMMs for modeling data with little or no prior information.

6 Conclusions

The primary properties of the wavelet transform — Locality, Multiresolution, and Compression — have led to powerful new approaches to statistical signal processing. However, existing methods usually model the wavelet coefficients as statistically independent or jointly Gaussian. The Compression property dictates the need for nonGaussian models for individual wavelet coefficients. Moreover, the secondary properties of the wavelet transform — Clustering and Persistence — indicate that statistical dependencies between coefficients must be characterized in order to derive optimal signal processing algorithms.

In this paper, we have developed a new framework for statistical signal processing based on wavelet-domain HMMs. The framework enables us to concisely model the non-Gaussian statistics of individual wavelet coefficients and capture statistical dependencies between coefficients. We have developed an efficient Expectation Maximization algorithm for fitting the HMMs to observational signal data, and we have demonstrated the utility, flexibility, and performance of our framework in several estimation and detection problems.

We believe that the HMM framework presented here could serve as a powerful new tool for wavelet-based statistical signal and image processing, with applications in signal estimation, detection, classification, compression, and synthesis. Although the examples we have provided here are one-dimensional, multi-dimensional wavelet domain HMTs are derived from our results, since the models and training algorithms apply to quad and higher-dimensional trees. Furthermore, these HMMs apply not only for modeling wavelet-domain data, but also for modeling data from other multiresolution transforms or signal representations. Finally, the knowledge base that has already accumulated in statistics, speech recognition, artificial intelligence, and related fields may lead to wavelet-domain HMMs that are even more accurate and sophisticated, yet still tractable, robust, and efficient for signal processing.

A Appendix — EM Algorithm for Hidden Markov Trees

Although the EM algorithm is classical with a well-known basic structure, the exact EM steps are problem dependent. In fact, the EM steps for estimating the parameters of tree-structured probability models have been derived only recently [27, 32], with work primarily focusing on trees of discrete-valued random variables. Following [32], we will develop an EM algorithm for HMTs generalized to handle continuous-valued wavelet coefficients and specialized to the tree structure provided by the wavelet transform. The major change from [32] is that for the leaves of the tree we replace maximum-likelihood (ML) estimation of probability mass function (pmf) values with ML estimation of Gaussian mixture means and variances. In addition, we will demonstrate the effect of tying on the algorithm.

In applying the EM algorithm to HMTs, our task is to fit an M-state HMT model, parameterized by $\boldsymbol{\theta} = \left\{p_{S_i}(m),\ \epsilon^{mn}_{i,\rho(i)},\ \mu_{i,m},\ \sigma^2_{i,m} \;\middle|\; i = 1,\ldots,P;\ n,m = 1,\ldots,M\right\}$,[9] to $K > 1$ trees of observed wavelet coefficients, with P the number of wavelet coefficients in each tree. We omit modeling the single scaling coefficient associated with each tree; as mentioned in Section 3.2, extensions to handle the scaling coefficients are straightforward.

We obtain the K trees either by wavelet-transforming K signal observations, each into a single tree, or by wavelet-transforming one signal observation into K different wavelet trees, as shown in Figure 5(b). In the latter case, we actually tie

[9] Except for i's corresponding to root states, the pmfs $p_{S_i}(m)$ are completely specified by the root state pmfs and the transition probabilities $\epsilon^{mn}_{i,\rho(i)}$.

across trees, modeling different trees using the same set of parameters (see Section 4.3 for details). The EM steps are identical for either case.

Recall from Section 4.1 that the EM algorithm is iterative, and that for HMTs it converges to a local-maximum of the incomplete log-likelihood $f(\mathbf{w}|\boldsymbol{\theta})$. The iterative structure is as follows:

EM algorithm for HMT

Initialize:

Select an initial model estimate $\boldsymbol{\theta}^0$.

Set iteration counter $l = 0$.

1. E step:
Calculate $p\left(\mathbf{S}\middle|\mathbf{w}, \boldsymbol{\theta}^l\right)$, the joint pmf for the hidden state variables (used in the maximization of $E_{\mathbf{S}}\left[\ln f\left(\mathbf{w}, \mathbf{S}|\boldsymbol{\theta}\right)\middle|\mathbf{w}, \boldsymbol{\theta}^l\right]$).

2. M step:
Set $\boldsymbol{\theta}^{l+1} = \arg\max_{\boldsymbol{\theta}} E_{\mathbf{S}}\left[\ln f\left(\mathbf{w}, \mathbf{S}|\boldsymbol{\theta}\right)\middle|\mathbf{w}, \boldsymbol{\theta}^l\right]$.

3. Set $l = l+1$. If converged, then stop; else return to E step.

We will suppress the superscript l when it is clear from the context.

Since the wavelets coefficients are conditionally Gaussian given the hidden states, the M step involves simple closed-form updates for the means, variances, and transition probabilities as a function of the wavelet coefficients $\mathbf{w}$, marginal state pmfs $p\left(\mathbf{S}_i = m\middle|\mathbf{w}, \boldsymbol{\theta}^l\right)$, and parent-child pmfs $p\left(\mathbf{S}_i = m, \mathbf{S}_\rho(i) = n\middle|\mathbf{w}, \boldsymbol{\theta}^l\right)$. To perform the M step updates, in the E step we calculate these pmfs. In the E step we can also calculate likelihoods, such as $f\left(\mathbf{w}|\boldsymbol{\theta}\right)$, that can be used for classification and other tasks.

To keep things clear and simple, we will first develop the E step for a single tree. (The E step is often referred to as the forward-backward algorithm in the HMM literature [18] and as the upward-downward or inward-outward algorithm in the artificial intelligence literature [20, 27, 32].) We will then develop the E step for multiple trees and the M step. We will finish by incorporating into the EM steps the notion of tying within trees from Section 4.3.

We first focus on processing a single size-P wavelet tree containing observed wavelet coefficients $\mathbf{w} = [w_1\ w_2\ \ldots\ w_P]$ having hidden states $\mathbf{S} = [S_1\ S_2\ \ldots\ S_P]$ that take on values $m = 1, \ldots, M$. The primary task of the E step is to calculate the hidden state probabilities $p\left(S_i = m|\mathbf{w}, \boldsymbol{\theta}\right)$ and $p\left(S_i = m, S_{\rho(i)} = n\middle|\mathbf{w}, \boldsymbol{\theta}\right)$. To obtain these probabilities, we introduce a number of intermediate variables.

A.1 Set-up

We now introduce some notation for trees of observed wavelet coefficients. Similar in structure to the trees of Figure 5, these trees are formed by linking the wavelet coefficients rather than the hidden states. We define $\mathcal{T}_i$ to be the subtree of observed wavelet coefficients with root at node i, so that the subtree $\mathcal{T}_i$ contains coefficient w_i and all of its descendants. Now if $\mathcal{T}_j$ is a subtree of $\mathcal{T}_i$ (*i.e.*, W_j and all its descendants are members of $\mathcal{T}_i$), then we define $\mathcal{T}_{i\backslash j}$ to be the set of wavelet coefficients obtained by removing the subtree $\mathcal{T}_j$ from $\mathcal{T}_i$. Without loss of generality, we order $\mathbf{w}$ so that w_1 is at the root of the entire tree. Thus, $\mathcal{T}_1$ is the entire tree of observed wavelet coefficients (a tree-structured version of the vector $\mathbf{w}$). In our probability expressions, we will interchange $\mathcal{T}_1$ and $\mathbf{w}$ when convenient.

For each subtree $\mathcal{T}_i$, we define the conditional likelihoods

$$\beta_i(m) \equiv f(\mathcal{T}_i | S_i = m, \boldsymbol{\theta}) \tag{10}$$

$$\beta_{i,\rho(i)}(m) \equiv f\left(\mathcal{T}_i \middle| S_{\rho(i)} = m, \boldsymbol{\theta}\right) \tag{11}$$

$$\beta_{\rho(i)\backslash i}(m) \equiv f\left(\mathcal{T}_{\rho(i)\backslash i} \middle| S_{\rho(i)} = m, \boldsymbol{\theta}\right), \tag{12}$$

and the joint probability functions

$$\alpha_i(m) \equiv p\left(S_i = m, \mathcal{T}_{1\backslash i} \middle| \boldsymbol{\theta}\right), \tag{13}$$

with S_i taking discrete values and the coefficients in $\mathcal{T}_{1\backslash i}$ taking continuous values.

Based on the HMT properties from Section 3.2, the trees $\mathcal{T}_i$ and $\mathcal{T}_{1\backslash i}$ are independent given the state variable S_i. This fact, along with the chain rule of probability calculus, leads to the desired state probabilities in terms of the α's and β's. First we obtain

$$p(S_i = m, \mathcal{T}_1 | \boldsymbol{\theta}) = \alpha_i(m)\ \beta_i(m) \tag{14}$$

and

$$p\left(S_i = m, S_{\rho(i)} = n, \mathcal{T}_1 \middle| \boldsymbol{\theta}\right) = \beta_i(m)\ \epsilon_{i,\rho(i)}^{mn}\ \alpha_{\rho(i)}(n)\ \beta_{\rho(i)\backslash i}(n). \tag{15}$$

The likelihood of $\mathbf{w}$ is then

$$f(\mathbf{w}|\boldsymbol{\theta}) = f(\mathcal{T}_1|\boldsymbol{\theta}) = \sum_{m=1}^{M} p(S_i = m, \mathcal{T}_1 | \boldsymbol{\theta}) = \sum_{m=1}^{M} \beta_i(m)\ \alpha_i(m). \tag{16}$$

Bayes rule applied to equations (14)–(16) leads to the desired conditional probabilities

$$p(S_i = m | \mathbf{w}, \boldsymbol{\theta}) = \frac{\alpha_i(m)\ \beta_i(m)}{\sum_{n=1}^{M} \alpha_i(n)\ \beta_i(n)} \tag{17}$$

and

$$p\left(S_i = m, S_{\rho(i)} = n \middle| \mathbf{w}, \boldsymbol{\theta}\right) = \frac{\beta_i(m)\ \epsilon_{i,\rho(i)}^{mn}\ \alpha_{\rho(i)}(n)\ \beta_{\rho(i)\backslash i}(n)}{\sum_{n=1}^{M} \alpha_i(n)\ \beta_i(n)}. \tag{18}$$

A.2 E step for a single wavelet tree (upward-downward algorithm)

All state variables within our HMT model are interdependent; in determining probabilities for the state variables, we must propagate state information throughout the tree. The upward-downward algorithm is an efficient method for propagating this information. The Up step calculates the β's by transmitting information about the fine-scale wavelet coefficients to the states of the coarse-scale wavelet coefficients; the Down step calculates the α's by propagating information about the coarse-scale wavelet coefficients down to the states of the fine-scale wavelet coefficients. Combining the information from the α's and β's via equations (17)–(18), we obtain conditional pmfs for the state of each wavelet coefficient in the tree.

For our derivation, we will focus on models with mixing components that are Gaussian, with density

$$g\left(w; \mu, \sigma^2\right) \equiv \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(w-\mu)^2}{2\sigma^2}\right). \tag{19}$$

More general densities can also be treated. Recall that we assign to each node i in the tree a scale $J(i) \in \{1, \ldots, L\}$, with $J = 1$ the finest scale and $J = L$ the coarsest scale. Also recall that $\rho(i)$ is the parent of node i and $c(i)$ the set of children to node i.

Up step

Initialize: For all state variables S_i at the finest scale $J = 1$, calculate for $m = 1, \ldots, M$:

$$\beta_i(m) = g\left(w_i; \mu_{i,m}, \sigma^2_{i,m}\right). \quad (20)$$

1. For all state variables S_i at scale J, compute for $m = 1, \ldots, M$:

$$\beta_{i,\rho(i)}(m) = \sum_{n=1}^{M} \epsilon^{mn}_{i,\rho(i)}\ \beta_i(n), \quad (21)$$

$$\beta_{\rho(i)}(m) = g\left(w_{\rho(i)}; \mu_{\rho(i),m}, \sigma^2_{\rho(i),m}\right) \times \prod_{i \in c(\rho(i))} \beta_{i,\rho(i)}(m), \quad (22)$$

$$\beta_{\rho(i)\backslash i}(m) = \frac{\beta_{\rho(i)}(m)}{\beta_{i,\rho(i)}(m)}. \quad (23)$$

2. Set $J = J + 1$ (move up the tree one scale).

3. If $J = L$, then stop; else return to step 1.

Down step

Initialize: For state variable S_1 at the coarsest scale $J = L$, set for $m = 1, \ldots, M$:

$$\alpha_1(m) = p_{S_1}(m). \quad (24)$$

1. Set $J = J - 1$ (move down the tree one scale).

2. For all state variables S_i at scale J, compute for $m = 1, \ldots, M$:

$$\alpha_i(m) = \sum_{n=1}^{M} \epsilon^{mn}_{i,\rho(i)}\ \alpha_{\rho(i)}(n)\ \beta_{\rho(i)\backslash i}(n). \quad (25)$$

3. If $J = 1$, then stop; else return to step 1.

A.3 E step for multiple wavelet trees

To handle $K > 1$ wavelet trees, we add a superscript k to denote the tree number. We denote the observed wavelet coefficients as $\mathbf{w} = [\mathbf{w}^1\ \mathbf{w}^2\ \ldots\ \mathbf{w}^K]$ and the hidden states as $\mathbf{S} = [\mathbf{S}^1\ \mathbf{S}^2\ \ldots\ \mathbf{S}^K]$. The vectors $\mathbf{w}^k = \left[w^k_1\ w^k_2\ \ldots\ w^k_P\right]$ and $\mathbf{S}^k = \left[S^k_1\ S^k_2 \ldots\ S^k_P\right]$ contain the wavelet coefficients and states of the kth tree, respectively.

To implement the E step at iteration l of the EM algorithm, we apply the upward-downward algorithm independently to each of the K wavelet trees. Using the parameter estimates $\boldsymbol{\theta} = \boldsymbol{\theta}^l$, we calculate the probabilities $p\left(S^k_i = m \middle| \mathbf{w}^k, \boldsymbol{\theta}^l\right)$ and $p\left(S^k_i = m, S^k_{\rho(i)} = n \middle| \mathbf{w}^k, \boldsymbol{\theta}^l\right)$ for each tree via equations (17) and (18).

A.4 M step

Once the probabilities for the hidden states are known, the M step is straightforward. We update the entries of $\boldsymbol{\theta}^{l+1}$ as follows:

$$p_{S_i}(m) = \frac{1}{K} \sum_{k=1}^{K} p\left(S^k_i = m \middle| \mathbf{w}^k, \boldsymbol{\theta}^l\right), \quad (26)$$

$$\epsilon^{mn}_{i,\rho(i)} = \frac{\sum_{k=1}^{K} p\left(S^k_i = m, S^k_{\rho(i)} = n \middle| \mathbf{w}^k, \boldsymbol{\theta}^l\right)}{K\ p_{S_{\rho(i)}}(n)}, \quad (27)$$

$$\mu_{i,m} = \frac{\sum_{k=1}^{K} w^k_i\ p\left(S^k_i = m \middle| \mathbf{w}^k, \boldsymbol{\theta}^l\right)}{K\ p_{S_i}(m)}, \quad (28)$$

$$\sigma^2_{i,m} = \frac{\sum_{k=1}^{K} (w^k_i - \mu_{i,m})^2\ p\left(S^k_i = m \middle| \mathbf{w}^k, \boldsymbol{\theta}^l\right)}{K\ p_{S_i}(m)}. \quad (29)$$

The updates for the state probabilities $p_{S_i}(m)$ and $\epsilon^{mn}_{i,\rho(i)}$ are performed by summing the individual state probabilities and then normalizing so that the probabilities sum to one. Just as for the IM model [26] and the hidden Markov chain model [18], updates for the Gaussian mixture means and variances are performed by a weighted averaging of the empirical means and variances, with the weights chosen in proportion to the probabilities of each mixture.

As should be clear from the E and M steps, the per iteration computational complexity of the

EM algorithm is linear in the number of observed wavelet coefficients. The overall complexity may involve a large multiplicative constant depending on the number of hidden states used and the number of iterations required to converge. However, as shown throughout this paper, even the simplest two-state HMT model can capture many densities quite well.

A.5 Tying within trees

The M step changes slightly when tying is performed within trees, such as tying wavelet coefficients and their states within a certain subband or scale. (See Section 4.3 for the basic idea behind tying.) With tying, we perform extra statistical averaging over coefficients that are tied together within each tree. For the kth tree $\mathbf{w}^k$ with wavelet coefficients w_i^k, we write $i \sim j$ if w_i^k and w_j^k (and their states) are tied, which means that they are modeled with the same underlying density parameters. The set $[i] = \left\{ j \middle| w_j^k \sim w_i^k \right\}$ is the equivalence class of i, with $|[i]|$ the number of elements in the class.

For simplicity, we assume that all trees are tied in the same fashion (that is, the coefficients in the trees $\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^K$ are tied in the same manner) according to the collection of equivalence classes given by the $[i]$'s. In this scenario, the M step becomes:

$$p_{S_i}(m) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|[i]|} \sum_{j \in [i]} p\left(S_j^k = m \middle| \mathbf{w}^k, \boldsymbol{\theta}^l\right), \quad (30)$$

$$\begin{aligned} \epsilon_{i,\rho(i)}^{mn} &= \frac{1}{K \; p_{S_{\rho(i)}}(m)} \sum_{k=1}^{K} \frac{1}{|[i]|} \sum_{j \in [i]} \\ & \qquad p\left(S_j^k = n, S_{\rho(j)}^k = m \middle| \mathbf{w}^k, \boldsymbol{\theta}^l\right), \end{aligned} \quad (31)$$

$$\begin{aligned} \mu_{i,m} &= \frac{1}{K \; p_{S_i}(m)} \sum_{k=1}^{K} \frac{1}{|[i]|} \sum_{j \in [i]} \\ & \qquad w_j^k \; p\left(S_j^k = m \middle| \mathbf{w}^k, \boldsymbol{\theta}^l\right), \end{aligned} \quad (32)$$

$$\begin{aligned} \sigma_{i,m}^2 &= \frac{1}{K \; p_{S_i}(m)} \sum_{k=1}^{K} \frac{1}{|[i]|} \sum_{j \in [i]} \\ & \qquad (w_j^k - \mu_{j,m})^2 \; p\left(S_j^k = m \middle| \mathbf{w}^k, \boldsymbol{\theta}^l\right). \end{aligned} \quad (33)$$

Although (30)-(33) appear more computationally intensive than (26)-(29), the computational complexity remains the same, since the common parameters for each equivalence class $[i]$ are calculated only once.

References

[1] D. Donoho and I. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Stat. Assoc.*, vol. 90, pp. 1200–1224, Dec. 1995.

[2] J.-C. Pesquet, H. Krim, and E. Hamman, "Bayesian approach to best basis selection," in *IEEE Int. Conf. on Acoust., Speech, Signal Proc. — ICASSP '96*, (Atlanta), pp. 2634–2637, 1996.

[3] H. Chipman, E. Kolaczyk, and R. McCulloch, "Adaptive Bayesian wavelet shrinkage," *Journal of the American Statistical Association*, vol. 92, 1997.

[4] N. Lee, Q. Huynh, and S. Schwarz, "New methods of linear time-frequency analysis for signal detection," in *IEEE Int. Symp. Time-frequency and Time-scale Analysis*, 1996.

[5] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Proc.*, vol. 41, pp. 3445–3462, Dec. 1993.

[6] M. Basseville, A. Benveniste, K. C. Chou, S. A. Golden, R. Nikoukhah, and A. S. Will-

sky, "Modeling and estimation of multiresolution stochastic processes," *IEEE Trans. Inform. Theory*, vol. 38, pp. 766–784, Mar. 1992.

[7] P. Flandrin, "Wavelet analysis and synthesis of fractional Brownian motion," *IEEE Trans. Inform. Theory*, vol. 38, pp. 910–916, Mar. 1992.

[8] I. Daubechies, *Ten Lectures on Wavelets*. New York: SIAM, 1992.

[9] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*. Prentice Hall, 1995.

[10] M. R. Luettgen, W. C. Karl, A. S. Willsky, and R. R. Tenney, "Multiscale representations of Markov random fields," *IEEE Trans. Signal Proc.*, vol. 41, pp. 3377–3395, Dec. 1993.

[11] K. C. Chou and L. P. Heck, "A multiscale stochastic modeling approach to the monitoring of mechanical systems," in *IEEE Int. Symp. Time-frequency and Time-scale Analysis*, 1994.

[12] F. Abramovich, T. Sapatinas, and B. W. Silverman, "Wavelet thresholding via a Bayesian approach," tech. rep., Math Dept., Univ. of Bristol, November 1996.

[13] E. P. Simoncelli and E. H. Adelson, "Noise removal via Bayesian wavelet coring," in *IEEE Int. Conf. on Image Proc. — ICIP 1996*, (Switzerland), September 1996.

[14] M. T. Orchard and K. Ramchandran, "An investigation of wavelet-based image coding using an entropy-constrained quantization framework," in *Data Compression Conference '94*, (Snowbird, Utah), pp. 341–350, 1994.

[15] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 710–732, July 1992.

[16] S. Mallat and W. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 617–643, 1992.

[17] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Hidden Markov models for wavelet-based signal processing," in *Proc. 30th Asilomar Conf.*, (Pacific Grove, CA), Nov. 1996.

[18] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–285, Feb. 1989.

[19] J. Deller, J. Proakis, and J. Hanson, *Discrete-time processing of speech signals*. New Jersey: Prentice Hall, 1993.

[20] P. Smyth, D. Heckerman, and M. Jordan, "Probabilistic independence networks for hidden Markov probability models," *Neural Comp.*, vol. 9, no. 2, pp. 227–269, 1997.

[21] S. LoPresto, K. Ramchandran, and M. T. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," in *Data Compression Conference '97*, (Snowbird, Utah), pp. 221–230, 1997.

[22] R. W. Buccigrossi and E. P. Simoncelli, "Progressive wavelet image coding based on a conditional probability model," in *IEEE Int. Conf. on Acoust., Speech, Signal Proc. —*

ICASSP '97, (Germany), pp. 2957–2960, April 1997.

[23] C. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Proc.*, vol. 3, pp. 162–177, March 1994.

[24] N. Saito and R. R. Coifman, "Local discriminant bases," in *Mathematical Imaging: Wavelet Applications in Signal and Image Processing*, pp. 2–14, Proc. SPIE 2303, 1994.

[25] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, pp. 465–479, 1971.

[26] R. Redner and H. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, pp. 195–239, Apr. 1994.

[27] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco: Morgan Kaufmann, 1988.

[28] H. Lucke, "Which stochastic models allow Baum-Welch training?," *IEEE Trans. Signal Proc.*, vol. 11, pp. 2746–2756, Nov. 1996.

[29] M. S. Crouse and R. G. Baraniuk, "Simplified wavelet-domain hidden Markov models using contexts," in *Proc. 31st Asilomar Conf.*, (Pacific Grove, CA), Nov. 1997.

[30] J. Park and I. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 13, pp. 246–257, 1991.

[31] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc.*, vol. 39, pp. 1–38, 1977.

[32] O. Ronen, J. Rohlicek, and M. Ostendorf, "Parameter estimation of dependence tree models using the EM algorithm," *IEEE Signal Proc. Lett.*, vol. 2, pp. 157–159, Aug. 1995.

[33] N. Merhav, "Universal classification for hidden Markov models," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1586–1594, November 1991.

[34] Y. Wang, "Jump and sharp cusp detection by wavelets," *Biometrika*, vol. 82, pp. 385–397, 1995.

[35] R. T. Ogden and E. Parzen, "Data dependent wavelet thresholding in nonparametric regression with change-point applications," *Computational Statistics and Data Analysis*, vol. 22, pp. 53–70, 1996.

[36] L. L. Scharf, *Statistical Signal Processing. Detection, Estimation, and Time Series Analysis.* Reading, MA: Addison-Wesley, 1991.

[37] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes.* New Jersey: Prentice Hall, 1993.

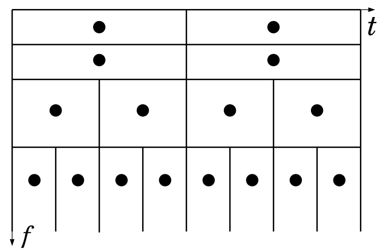

Figure 1: *Tiling of the time-frequency plane by the atoms of the wavelet transform. Each box depicts the idealized support of a scaling atom ϕ_K (top row) or a wavelet atom ψ_i (other rows) in time-frequency; the solid dot at the center corresponds to the scaling coefficient u_K or wavelet coefficient w_i. Each different row of wavelet atoms corresponds to a different* scale *or* frequency band. *(We run the frequency axis down rather than up for later convenience.)*

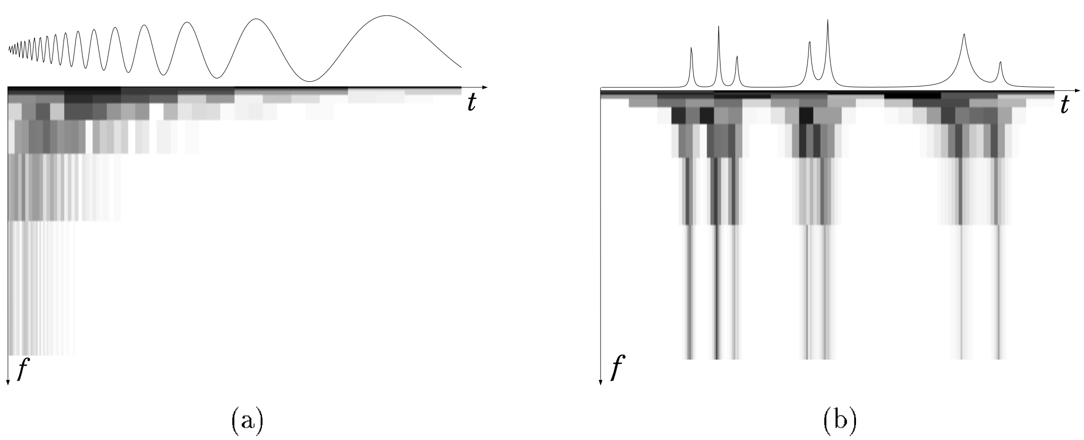

Figure 2: *Clustering and Persistence illustrated, respectively, in Donoho and Johnstone's (a) Doppler and (b) Bumps test signals [1]. The signals lie atop the time-frequency tiling (Figure 1) provided by a seven-scale wavelet transform. Each tile is colored as a monotonic function of the wavelet coefficient energy w_i^2, with darker tiles indicating greater energy.*

Table 1: *Denoising results for Donoho and Johnstone's length-1024 test signals [1]. Noise variance* $\sigma_n^2 = 1$.

Method	**Mean-squared error**			
	Bumps	Blocks	Doppler	Heavisine
SureShrink [1]	0.683	0.222	0.228	0.095
Bayesian [3]	0.350	0.099	0.165	0.087
IM	0.335	0.105	0.170	0.080
HMT	0.268	0.079	0.132	0.081

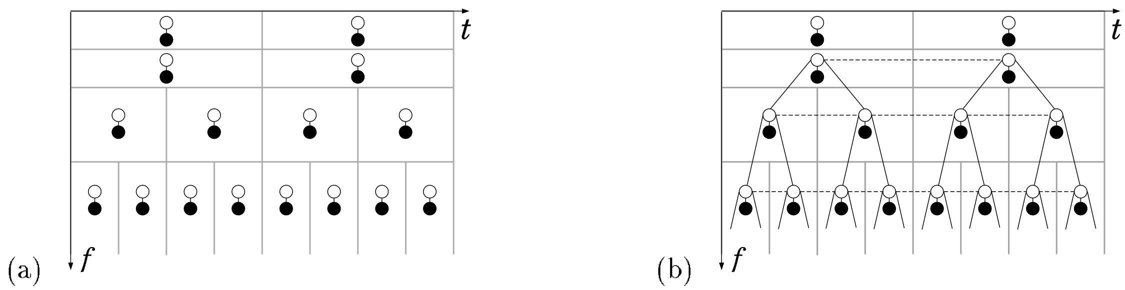

Figure 3: *Statistical models for the wavelet transform. (a) Independent Mixture (IM) model. To match the nonGaussian nature of the wavelet coefficients, we model each coefficient as a mixture with a hidden state variable. Each black node represents a continuous wavelet coefficient W_i. Each white node represents the mixture state variable S_i for W_i. (b) To match the inter-coefficient dependencies, we link the hidden states. Connecting discrete nodes horizontally across time (dashed links) yields the Hidden Markov Chain model. Connecting discrete nodes vertically across scale (solid links) yields the Hidden Markov Tree (HMT) model.*

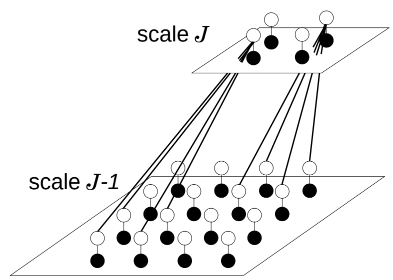

Figure 4: HMT for an image quadtree. As shown for two parent states, each parent hidden state is connected to its four child states. (The other parent-child connections are omitted for visual clarity. The two remaining parent states would have similar connections to their child states.) The two fields of nodes depict the wavelet coefficients at scales J and $J-1$, respectively, and correspond to 2-d wavelet basis functions with a specific spatial orientation (horizontal, vertical, or diagonal). For more details on 2-d wavelet systems see [5, 8, 9].

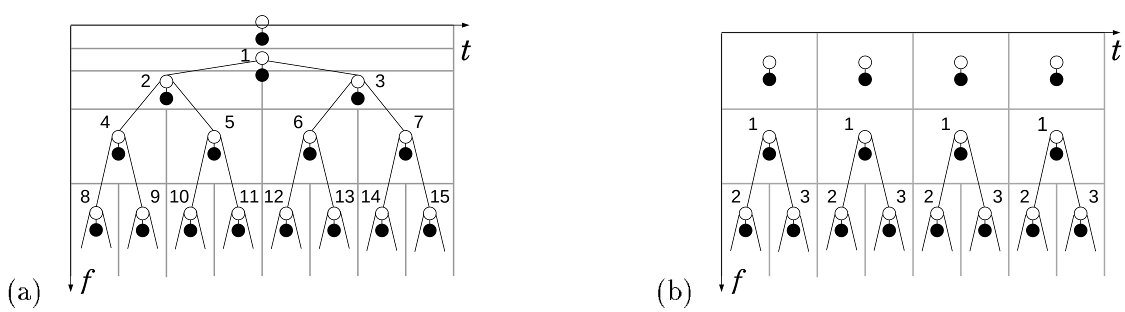

Figure 5: *Organization of a wavelet transform as a forest of binary trees. Tilings of the time-frequency plane and tree structures for (a) full decomposition (one tree), (b) decomposition with two fewer scale bands (four trees). A scaling coefficient sits above the root of each tree. Associated with each index i is a pair of nodes representing the wavelet coefficient W_i (black node) and its state variable S_i (white node). See Figure 3(b) for a wavelet decomposition with two trees.*

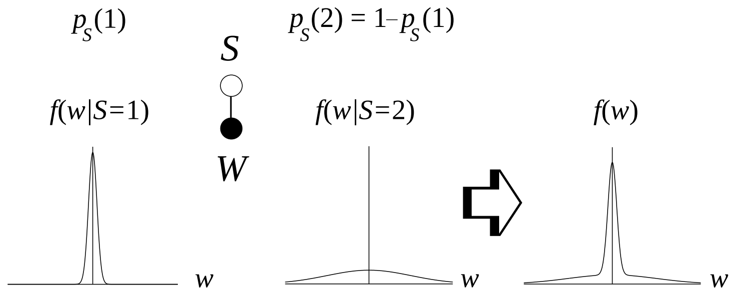

Figure 6: *A two-state, zero-mean Gaussian mixture model for a random variable W. We denote the state variable S with a white node, the random variable W with a black node. Illustrated are the Gaussian conditional pdf's for $W|S$ as well as the overall nonGaussian pdf for W. $S = 1$ corresponds to a low-variance Gaussian pdf, and $S = 2$ corresponds to a high-variance Gaussian pdf. In our application, we model each wavelet coefficient W_i (each black node in Figure 1) in this way.*

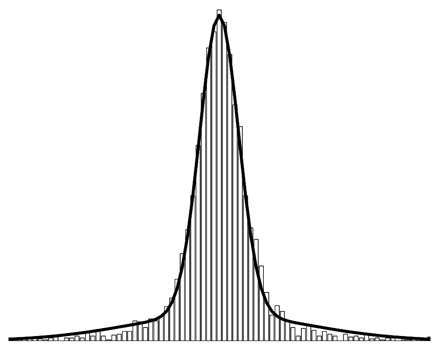

Figure 7: *A two-state, zero-mean Gaussian mixture model can closely fit real wavelet coefficient data. Here we compare the model pdf to a histogram of one scale of the Daubechies-4 wavelet transform of the "fruit" image in the Georgia Tech database.*

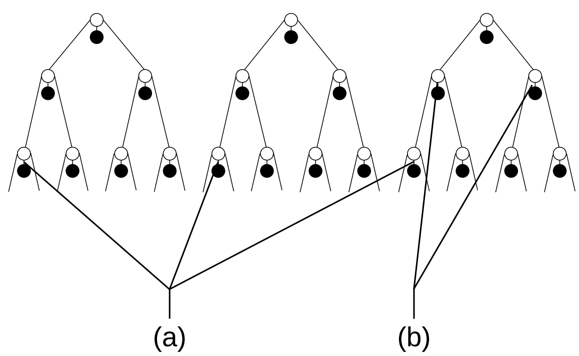

Figure 8: *Tying in the HMT model. Pictured above are three wavelet trees. (a) Tying across the wavelet trees. (b) Tying within a wavelet tree.*

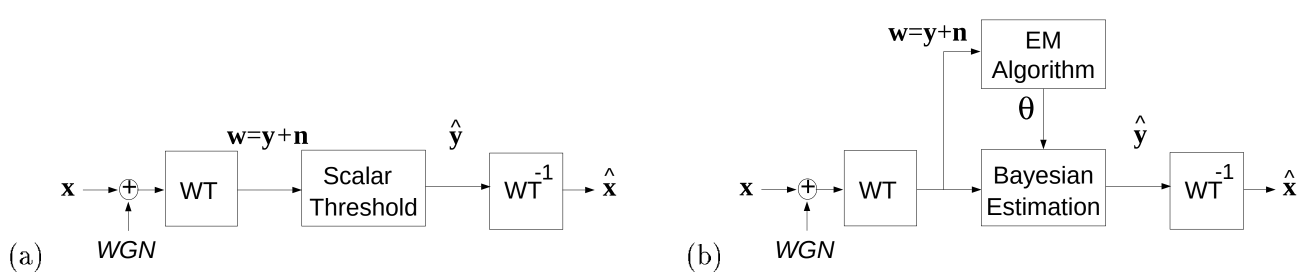

Figure 9: *Block diagrams for wavelet-based denoising. (a) Standard wavelet-based denoising. (b) Our empirical Bayesian wavelet-based denoising. In each case, the signal in additive white Gaussian noise (WGN) is wavelet-transformed, passed through threshold-like nonlinearities, and inverse transformed to get the denoised signal. However, the empirical Bayesian approach attempts to learn and exploit the signal's wavelet-domain structure to better separate signal from noise.*

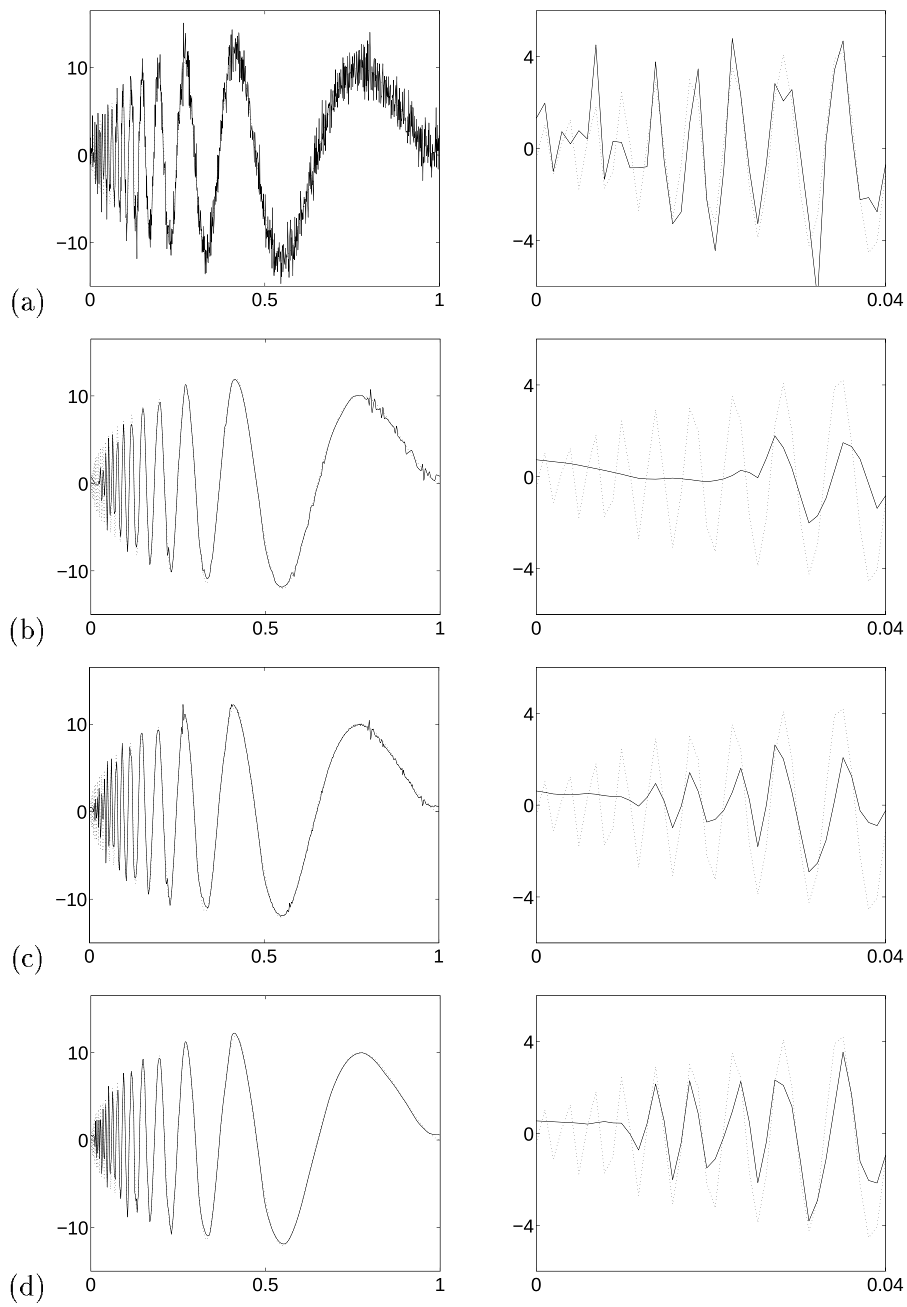

Figure 10: *Denoising the Doppler test signal in white Gaussian noise, $\sigma_n^2 = 2.25$. On each plot a dotted line is used to depict the original signal, a solid line the noisy or denoised signal. The leftmost plots depict the entire signals; the rightmost plots depict the signals "zoomed" to the interval [0,0.04], where it is difficult to distinguish high-frequency signal from noise. (a) Noisy length-1024 Doppler signal, MSE = 2.42. (b) Denoised via SureShrink [1], MSE = 0.43. (c) Denoised via wavelet-based Bayesian IM model, MSE = 0.34 (d) Denoised via wavelet-based Bayesian HMT model, MSE = 0.26.*

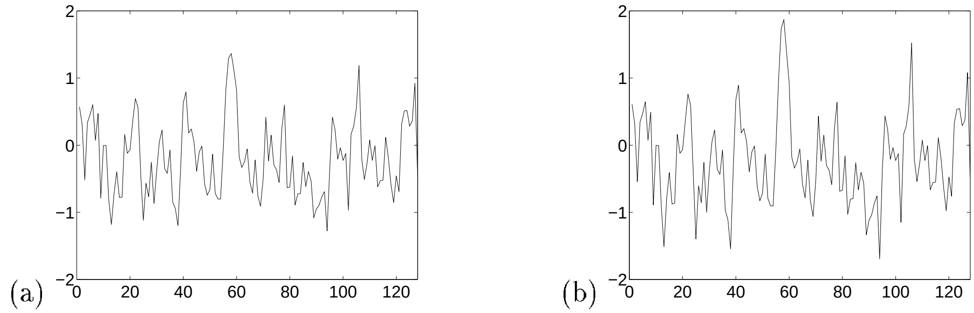

Figure 11: *Typical autoregressive (AR) signals used in nonlinear classification experiment. (a) Linear AR process (Class I). (b) Linear AR process passed through a mild cubic nonlinearity (Class II).*

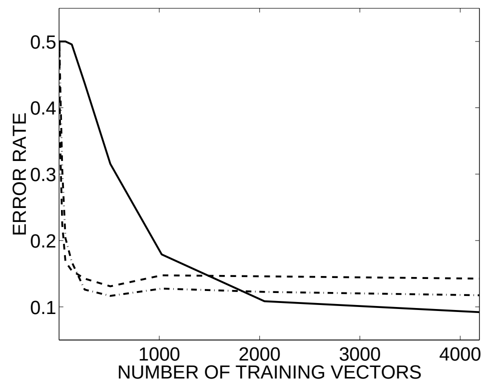

Figure 12: *Minimum probability of error rates for quadratic classifier (solid), wavelet-domain IM model classifier (dash), and wavelet-domain HMT model classifier (dash-dot).*

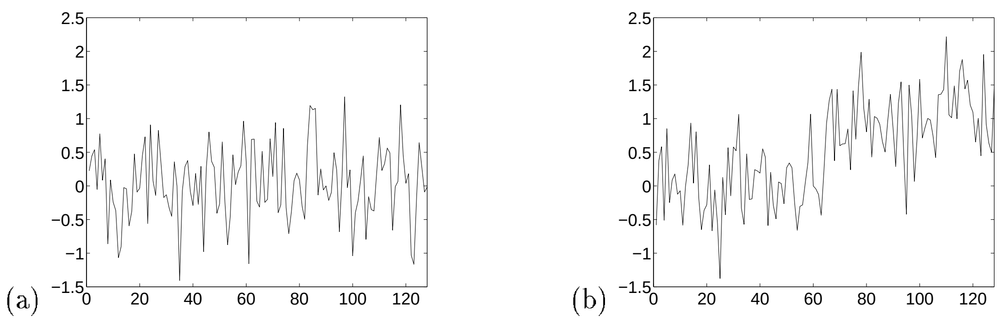

Figure 13: *Typical signals for the abrupt change detection experiment. (a) Gaussian white noise added to constant signal (Class I). (b) Gaussian white noise added to signal with abrupt change (Class II).*

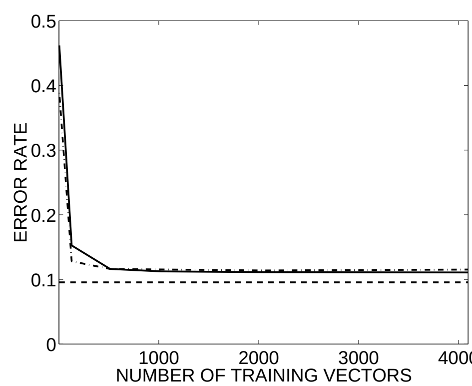

Figure 14: *Detection of an abrupt change. Minimum probability of error rates for wavelet-domain HMT model detector (solid), wavelet-domain IM model detector (dash-dot), and GLRT detector (dash-dash).*