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: rnowak@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)
January 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). The framework enables us to concisely model the statistical dependencies and nonGaussian statistics encountered with 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 no. MIP 94–57438, 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. Other work has been aimed at modeling correlations between wavelet coefficients, but these approaches usually assume Gaussian signal models (see [6], for example). The goal of this paper is to develop new wavelet-domain data models that match the statistical dependencies and nonGaussian statistics often encountered in practice. These new data models lead to sophisticated processing techniques that coordinate the nonlinear processing amongst coefficients to outperform current techniques. The new models are designed with the intrinsic properties of the wavelet transform in mind.

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 $\psi(t)$ [8,9]. For special choices of the wavelet, the atoms

$$\psi_{J,K}(t) \;\equiv\; 2^{-J/2}\,\psi\Big(2^{-J}t - K\Big), \qquad J,K \in \mathbb{Z} \tag{1}$$

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

$$z(t) \;=\; \sum_{J,K} w_{J,K}\,\psi_{J,K}(t), \qquad w_{J,K} \;\equiv\; \int z(t)\,\psi^*_{J,K}(t)\,dt. \tag{2}$$

For a wavelet centered at time zero and frequency f_0, the *wavelet coefficient* $w_{J,K}$ measures the content of the signal around the time $2^J K$ and frequency $2^{-J}f_0$ (see Figure 1). To analyze images, we employ two-dimensional wavelet systems [8,9]. (To keep the notation manageable in the sequel, we will adopt an abstract single index system for wavelet atoms and coefficients: $\psi_{J,K} \to \psi_i$, $w_{J,K} \to w_i$.) In wavelet-based signal processing, we process the signal $z(t)$ by operating on its wavelet coefficients $\{w_i\}$.

In this paper, we adopt a statistical approach to wavelet-based signal processing. We assume that the observational data are measurements of a signal waveform in additive, random noise. We regard the signal as random realization from a family or distribution of signals; hence, the signal component of the data is random as well. We will not explicitly specify prior signal distributions, rather we will deduce plausible models based on the properties of the wavelet transform itself. Our objective is to develop probability models for the wavelet transform of signals and images 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.

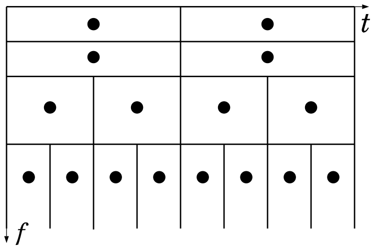

Figure 1: *Tiling of the time-frequency plane by the atoms of the wavelet transform. Each box depicts the idealized support of an atom ψ_i in time-frequency; the solid dot at the center corresponds to the wavelet coefficient w_i. Each different row of wavelet atoms corresponds to a different* scale *or*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

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. Therefore, wavelets can match a wide range of different signal components, from transients to harmonics.

Multiresolution: Wavelet atoms compress and dilate to analyze at a nested set of scales. This allows the transform to match both short-duration and long-duration signal structures.

Compression: The wavelet transforms of real-world signals and images tend to be sparse. As a result, the wavelet coefficient distributions of real-world signals and images have *nonGaussian* statistics, with heavy tails.

To date, wavelet coefficients have been modeled either as jointly Gaussian [4,6,10,11], or as nonGaussian, but independent [2, 3, 12–14]. Jointly Gaussian models can efficiently capture linear correlations between wavelet coefficients; however, any jointly Gaussian model is in conflict with the nonGaussian statistics implied by the Compression property. NonGaussian models have been formulated, with independence between the coefficients assumed for tractability reasons. 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 and images — 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 [15].

Persistence across Scale: Large/small values of wavelet coefficients tend to propagate across scales [16, 17].

Both of these empirical observations have been exploited with tremendous success by the compression community [5, 15]. Our goal is to do the same for signal processing.

Completely modeling the joint probability density function of all the coefficients, $f(\mathbf{w})$ with $\mathbf{w} = \{w_i\}$, would characterize the dependencies between wavelet coefficients. Oftentimes, however, the complete joint probability density is both 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 [18]. Our models owe their richness and flexibilty 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 2(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 2(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 [19, 20] and artificial intelligence [21].

The different state-to-state connectivities illustrated in Figure 2(b) yield fundamentally different models, each appropriate for certain applications. In this paper, we will emphasize three models. 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 refer to these models collectively as *wavelet-domain HMMs.*

After specifying a modeling framework, we can train a model by adjusting its parameters (the mixture density parameters and the probabilistic graph state transition probabilities) to best match our data in the maximum likelihood (ML) sense. In this way, we do not impose an artificial model on the data; rather we let the data itself dictate the exact form of the model. We will see that

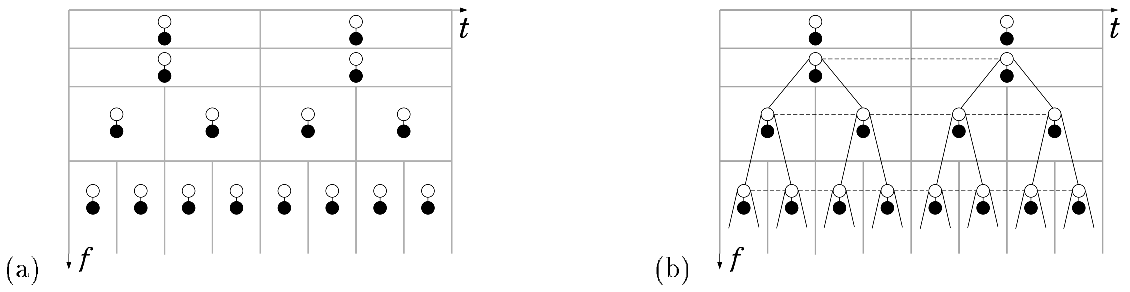

Figure 2: *Statistical models for the wavelet transform. (a) Independent Mixture (IM) model. To match the nonGaussian nature of the wavelet coefficients, we model each coefficient as a mixture with a hidden state variable. Each black node represents a continuous wavelet coefficient W_i. Each white node 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.*

probabilistic graphs allow us to develop a natural Markovian structure that enables rapid and robust iterative parameter estimators like the Expectation Maximization (EM) algorithm.

Once trained, HMMs provide an excellent approximation to the full joint probability of the wavelet coefficients $f(\mathbf{w})$. Furthermore, the statistical framework provides a natural setting for exploiting the structure inherent in real-world signals and images for estimation, detection, classification, prediction and filtering, and even synthesis. For example, 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 7 and Table 1). In Section 5.2, we will apply our models to two difficult problems in detection and classification.

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 [2,3,12–14]. 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 zero-tree coder of Shapiro [5] has revolutionized wavelet image compression, significantly improving compression performance by exploiting dependencies between wavelet coefficients. This work has spurred a number of new, improved image coders, too numerous to mention here. Our work differs from most current compression research in that we focus on an underlying probability model rather than just efficient data structures (such as the zero-tree). Thus, our framework is well-suited not only for compression, but for a host of problems in statistical signal processing.

Wavelet-domain HMMs also differ distinctly 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, in [6], Basseville et al. emphasize linear Gaussian models. Wavelet-domain HMMs are nonlinear, nonGaussian, and do not constrain the wavelet coefficients to be strictly Markov. Furthermore, processing with wavelet-domain HMMs remains simple due to the Markov structure of the hidden states.

Though similar in spirit to wavelet-domain HMMs, the multiscale models formulated in [22] tackle an entirely different problem. Developed for image segmentation, these models provide an efficient multiscale model that is applied directly to the signal. They do not apply to a multiresolution or tree-structured representation of the signal and, hence, do not provide feasible wavelet-domain models.

The wavelet-domain HMM framework developed in this paper also offers several advantages over traditional hidden Markov chain models from time series analysis. While these latter models have been successfully applied in situations such as speech processing, where the model does not provide a realistic model of the data generation mechanisms [23], in the wavelet context, HMMs are completely natural and in fact are evoked by the primary and secondary properties of the wavelet transform. For instance, the joint time-scale localization of the wavelet transform enables our models to concisely represent both short duration and long duration signal structure. In addition, marginal mixture models for individual wavelet coefficients have proven very effective in practice (primarily because of the Compression property) [2,3]. Mixture models are intimately related to the underlying signal spaces that are mathematically appropriate for the representation of real-world signals [1,12]. By contrast, in most traditional HMM applications, mixture models are chosen as a convenience and are not necessarily a realistic model of the underlying physical phenomena [23].

After dispensing with definitions and notation in Section 2, we turn to wavelet transform modeling using HMMs in Section 3. We derive a new EM algorithm for training the models on real data in Section 4. In Section 5, we apply this powerful machinery to several problems in signal estimation and detection and classification. We close in Section 6 with a discussion and conclusions.

2 Preliminaries

Before we launch headlong into hidden Markov wavelet modeling, we introduce further terminology of wavelet transforms, graphs and trees.

By introducing a set of lowpass *scaling functions* $\phi_{J_0,k}(t) \equiv 2^{-J_0}\,\phi(2^{-J_0}t - K)$ into (2) we obtain the alternate wavelet representation [8]

$$z(t) \;=\; \sum_K u_K\,\phi_{J_0,K}(t) \;+\; \sum_{J=-\infty}^{J_0}\sum_K w_{J,K}\,\psi_{J,K}(t), \qquad (3)$$

with $u_K = \int z(t)\,\phi^*_{J_0,K}(t)\,dt$ and $w_{J,K}$ as defined in (2). Note the semi-infinite range of the scale parameter J. The wavelet and scaling coefficients of sampled signals can be computed extremely

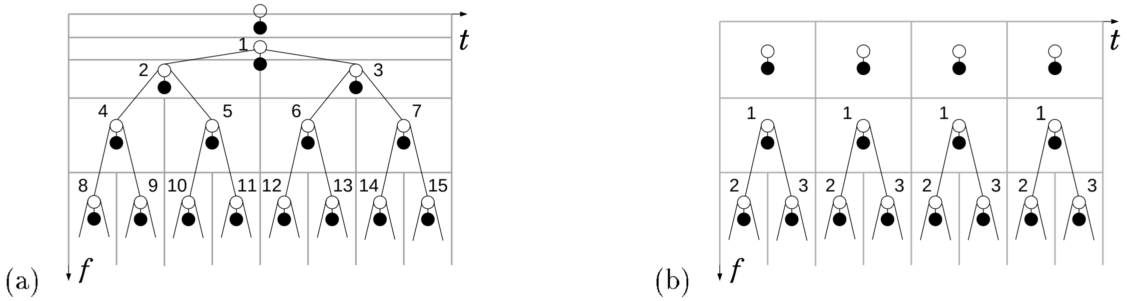

Figure 3: *Organization of a wavelet transform as a forest of binary trees. Tilings of the time-frequency plane and tree structures for (a) full decomposition (one tree), (b) decomposition with two fewer scale bands (four trees). A scaling coefficient sits above the root of each tree. Associated with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

efficiently using filter banks [9]. Wavelet transforms of images can be computed using combinations of one-dimensional wavelets and scaling functions. For more information on wavelet systems and their construction, see [8, 9].

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. Each node in a *binary* tree 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 Figure 3). In our abstract indexing scheme, we will denote the i-th wavelet coefficient from the k-th tree as w_i^k. Wavelet transforms of two-dimensional images have a similar organization, but in terms of *quad trees* [5].

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. We will denote vectors with boldface letters.

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

3 Wavelet Domain Probability Models for Observational Data

Recall that our objective is to develop probability models for the wavelet transform of signals and images 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 and images. We show that a two-state Gaussian mixture model is appropriate for the marginal distribution of individual wavelet coefficients.

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 the secondary properties of the wavelet transform: Clustering and Persistence across Scale. 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 hidden Markov models 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 or image consists of a small number of large coefficients and a large number of small coefficients. This property, combined with our view of the signal as a random realization from a family or distribution of signals, leads to the following simple model for an individual wavelet coefficient. Most wavelet coefficients contain very little signal information and hence these coefficients have small, random values. A few wavelet coefficients have large values that represent significant signal information. Thus we can model each coefficient as being in one of two states: "high," corresponding to a wavelet component containing significant dominant 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 4, this simple model is completely parameterized by the pmf of the state variable S_i, $p_{S_i}(1), 1 - p_{S_i}(1)$, and the variances of the Gaussian pdfs corresponding to each state, $\sigma^2_{i,j}$, $j = 1, 2$. Substantial evidence, both empirical and theoretical, shows that this simple two-state, zero-mean Gaussian mixture model can approximate the marginal densities of wavelet coefficients quite well. Empirically, this two-state zero-mean mixture model has proven both effective and convenient [2,3]. Our

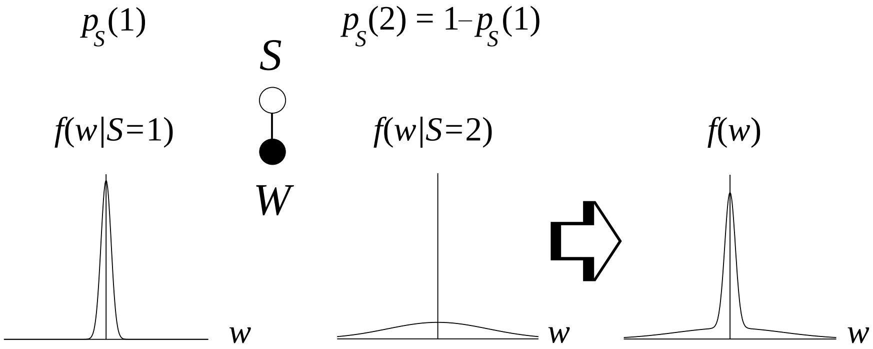

Figure 4: *A two-state, zero-mean Gaussian mixture model for a random variable W. We denote the state variable S with a white dot, the random variable W with a closed dot. Illustrated are the Gaussian conditional pdf's for $W|S$ as well as the overall mixture pdf for W. In our application, we model each wavelet coefficient W_i (each black dot in Figure 1) in this way.*

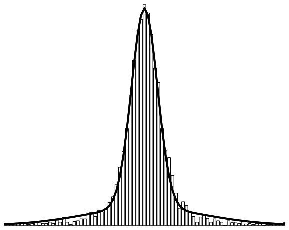

Figure 5: *A two-state, zero-mean Gaussian mixture model can closely fit real wavelet coefficient data. Here we compare the model pdf to the histogram of one scale of the wavelet transform of an image of fruit.*

experience corroborates these results; in Figure 5 we demonstrate the fit that a two-state, zero-mean Gaussian mixture provides for an actual signal. Theoretically, zero-mean Gaussian mixtures have been shown to naturally characterize the wavelet-domain statistics of signals from Besov spaces [12], and Besov spaces have proven to be extremely useful signal classes for real-world data [1].

Although we have illustrated and motivated a two-state, zero-mean Gaussian mixture model, we will develop machinery to handle a wider class of Gaussian mixture models with $M \geq 2$ states and non-zero means.[2] This increased flexibility allows us to better fit the marginal densities encountered in certain applications. In fact, 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 [26].

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

[2] In fact, similar machinery can be developed for mixtures of conditional densities from an exponential family of distributions [25]. However, the two-state Gaussian mixture model is simple, robust, and easy-to-use — attractive features for practical applications.

$p_S(s)$ and then draw an observation w according to $f_{W|S}(w|S = s)$. The pdf of W is given by

$$f_W(w) = \sum_{m=1}^{M} p_S(m)\, f_{W|S}(w|S = m). \tag{4}$$

In most applications of mixture models, the value w is observed, but the value of the state variable S is not; we say that the value of S is *hidden*. Notice that mixture models can be used for the scaling coefficients (which are definitely not zero-mean), but in this paper we do not model nor process the scaling coefficients.

3.2 Probabilistic Models for a Wavelet Tree

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*. 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 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 [5,15]. 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 the neighbor is very likely also 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. In the next sections, we apply probabilistic graph theory [21, 23, 27] to develop these models.

3.2.1 Graph models for wavelet transforms

Probabilistic graphs are useful tools for modeling the local dependencies between a set of random variables [21, 23, 27]. 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 2, we illustrate these three simple graphs.

We are by no means limited to just these three graphs. More complex dependencies can be modeled by placing additional connections between the states. Furthermore, if these more complex graphs satisfy a "chordality" property, we can formulate efficient algorithms for training and applying them [21, 23]. However, to keep our presentation and analysis simple, we will concentrate on the three special cases described in Figure 2, which we elaborate on here.

Independent Mixture (IM) Model: A mixture model with no connections, as in Figure 2(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 2(b) specifies a Markov-1 chain dependency between the state variables *within each scale* [19]. This new 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 scales* in Figure 2(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 emphasize the IM and HMT models in the sequel.

The HMT model matches both the Clustering and Persistence across Scale 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 3(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. Also, this simple tree-structure is capable of approximating the joint parent-child wavelet coefficient pdf to arbitrary precision. To see this, consider what happens as we increase the number of mixture components, M, used to model the marginal parent and child distributions.

Recall the components of the parent-child model. The parent is modeled using M Gaussian univariate mixing densities and an M-vector of probabilities for the densities. Conditioned on the parent state variable, the child wavelet coefficient is modeled using its own M Gaussian univariate densities and an M^2 matrix of probabilities for transitions from the parent's state to the child's state.

The joint model for parent and child is therefore an M^2 component bivariate Gaussian mixture. The mixing densities are the Cartesian products of the univariate mixing densities. The mixing probabilities are products of the parent state probabilities and the M^2 matrix of transition probabilities. Appealing to the approximation capabilities of Gaussian sums for [26], it is easily shown that this bivariate mixture model can approximate with arbitrary accuracy any bivariate parent-child pdf that has a finite number of disconinuities. The proof is analogous to the one for the universal approximation capabilities of radial basis function networks [28].

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. $\epsilon_{i,\rho(i)}^{mr} = p_{S_i|S_{\rho(i)}}(m|S_{\rho(i)} = r)$, 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 $\boldsymbol{\theta}$.

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

$$f_{W_i}\left(w_i|\{W_j\}_{j\neq i},\{S_j=s_j\}_{j\neq i},S_i=s_i\right) \;=\; f_{W_i}\left(w_i|S_i=s_i\right). \tag{5}$$

In words, W_i is conditionally independent of all other random variables given its state S_i. Hence, the independence properties for the states also lead to 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 $\{S_j\}_{j\in c(i)}$.

Using Figure 3(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 the coefficients themselves. This is an important distinction between our model and other multiscale Markov signal representations such as those considered in [6,10]. In general, our HMM framework does not imply a Markov structure on the wavelet coefficients directly. Let $l(i)$ denote that the scale of W_i (and S_i), and assume that scales are ordered $l=1$, finest, to $l=L$, coarsest. In our model

$$f_{W_i}\Big(w_i|\{W_j\}_{l(j)>l(i)}\Big) \;\neq\; f_{W_i}\Big(w_i|W_{\rho(i)}\Big). \tag{6}$$

However, observe that 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.

3.2.2 Three standard problems of HMMs

There are three canonical problems associated with the wavelet-domain HMMs we have described [19]:

Training: Given one or more sets of observed wavelet coefficients, determine the wavelet-domain HMM parameters that best characterize the wavelet coefficients. Before we process signals using wavelet-domain HMMs, we first must train the HMMs to capture the wavelet-domain properties of our signals of interest. This standard HMM training problem can be efficiently accomplished using the Expectation Maximization (EM) algorithm described in the next Section.

Likelihood Determination: Given a fixed wavelet-domain HMM, determine the likelihood of an observed set of wavelet coefficients. In other words, determine how well the wavelet-domain HMM describes the signal's transform. Besides its use in training, likelihood determination is vital for applications such as classification and detection, as we will see in Section 5.

State estimation: Given a fixed wavelet-domain HMM, determine the most likely sequence of hidden states for an observed set of wavelet coefficients. This is useful for problems such as segmentation (see [22]), where the hidden states represent a physically meaningful quantity. The Viterbi algorithm [19, 23] efficiently performs this optimization.

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

4 Model Training and Likelihood Determination

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 a set of observed signals or images; the model parameters $\boldsymbol{\theta}$ consist of the mixture state probabilities and mixture Gaussian means and variances. For parameter estimation, we apply the maximum likelihood (ML) principle: We choose the model parameters that maximize the probability of the observed wavelet data. 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). 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 [29], which jointly estimates both the model parameters $\boldsymbol{\theta}$ and the hidden states $\mathbf{S}$ given the observed wavelet coefficients $\mathbf{W}$. In the context of HMMs, the EM algorithm is often known as the *Baum-Welch* algorithm.

4.1 EM algorithms for Model Training

Our discussion of EM algorithms focuses on the specific problem of parameter estimation in wavelet-based HMMs; for a more general treatment, see [29]. By $(\mathbf{W}, \mathbf{S})$ we denote the complete-data vector consisting of the observed wavelet coefficients $\mathbf{W}$ and their unobserved states $\mathbf{S}$. Our goal is to maximize the incomplete log-likelihood function $\ln f(\mathbf{W}|\boldsymbol{\theta})$, with $\boldsymbol{\theta}$ the parameters of our HMM.

The premise behind the EM algorithm is that iteratively maximizing the expected value of the complete log-likelihood function $\ln f(\mathbf{W}, \mathbf{S}|\boldsymbol{\theta})$ leads to a local maximum of the incomplete log-likelihood function $f(\mathbf{W}|\boldsymbol{\theta})$. At the p-th iteration, let $\boldsymbol{\theta}^{(p)}$ be the parameter estimates and $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(p)})$ the expected value of the complete log-likelihood function, with the expectation performed over the unknown states, conditioned on the observed data and current parameter estimates. Then, we have

$$\text{Expectation:} \qquad Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(p)}) = E\left[\ln f(\mathbf{W}, \mathbf{S}|\boldsymbol{\theta})|\mathbf{W}, \boldsymbol{\theta}^{(p)}\right] \tag{7}$$

$$\text{Maximization:} \qquad \boldsymbol{\theta}^{(p+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(p)}). \tag{8}$$

Under mild conditions, this iteration converges to a local maximum of the likelihood function.

In [23], Lucke demonstrates that EM training algorithms exist for any HMM based on a probabilistic graph with a *chordal* structure. Since all acyclic graphs (trees) are trivially chordal, the three HMMs introduced in this paper admit efficient EM training. More complicated models, such as those linking states across both time and scale (using both solid and dashed connections in Figure 2(b)), are not necessarily chordal. To perform efficient EM training, we can modify nonchordal graphs to make them chordal by adding links between states (see [21, 23]).

For an HMM, the complexity of each iteration of the EM algorithm is linear in the number of observations [19, 23]. However, as the dependencies between the states become more complicated, the EM algorithm becomes less efficient (still linear complexity, but with a large constant factor). In particular, the Expectation (E) step becomes more difficult, due to the increased interplay between the states. (The Maximization step remains fairly simple.) Although approximate E steps, such as those developed via Mean Field Theory [30], have been developed to reduce complexity, even these approximations can be computationally intense. Thus, when designing an HMM, one must focus on characterizing only the essential dependencies, so that the model remains simple and efficiently trainable.

The specific EM steps for the IM and hidden Markov chain models have been developed thoroughly in [19, 25], 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 [31]. Since the observed wavelet data in the HMT model is continuous-valued, we provide the exact EM steps for this model in the Appendix.

4.2 Likelihood Determination

The E step of (7) is useful in its own right, since it provides us the by-product $\ln f(\mathbf{W}|\boldsymbol{\theta})$, the likelihood of the observed data given the model. This function measure how well the model $\boldsymbol{\theta}$ describes the data $\mathbf{W}$.

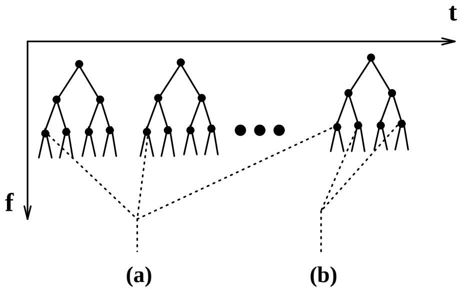

Figure 6: *Tying in the HMT model. (a) Tying across wavelet trees. (b) Tying within a wavelet tree.*

Likelihood determination is extremely useful for detection or classification applications, as we will see in Section 5.2, and for prediction and estimation. The E step is often referred to as the forward-backward algorithm in the HMM literature [19] and as the upward-downward or inward-outward algorithm in the artificial intelligence literature [21, 27, 31].

4.3 Robust Training and 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. Which brings us to a key problem: If limited training data are available, how can we make our modeling more robust? For HMMs we can improve robustness by *tying* together random variables — modeling random variables with similar statistical properties using a common Gaussian mixture density. (Which random variables are similar enough to be modeled with a common density is up to the model designer.) Practically speaking, we characterize tied random variables with a common parameter or set of parameters, such as the mixture means and variances or transition probabilities. Tying makes our estimates of these common parameters more robust, since we increase the amount of training data associated with each parameter.

In Figure 6, we distinguish between two different types of tying, tying between wavelet trees and tying within wavelet trees. Recall from Section 2 (Figure 3) that in general, the wavelet decomposition of even a single signal observation will 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

The development of wavelet-domain HMMs is 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 the classical quadratic detector, which is based on a Gaussian signal model. 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 [1,3]. The Compression property indicates that the wavelet transform typically compresses signals into a few coefficients of large magnitude, and because the wavelet transform is orthogonal it leaves 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.

Existing denoising methods usually ignore possible dependencies between signal wavelet coefficients, and hence these methods do not exploit key Clustering and Persistence Across Scale 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 co-ordinates the noise removal among the wavelet coefficients and automatically adjusts to subtle structure within the signal [18].

Consider the problem of estimating a length-N signal in zero-mean white Gaussian noise with power σ_n^2. Taking the wavelet transform of the noisy signal, we obtain $K \geq 1$ trees of noisy wavelet coefficients w_i^k, where k indexes the tree number and i indexes the location in the tree (Figure 3(b). Since the orthonormal 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

$$w_i^k \;=\; y_i^k + n_i^k, \tag{9}$$

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 estimate a HMT model for the y_i^k's from the noisy data and then use this estimate 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, since we estimate our Bayesian prior from the data itself.

To estimate an HMM for the noisy wavelet coefficients, we use the EM algorithm from the Appendix. We begin by estimating the parameters $\{p_{S_1}(m),\ \epsilon_{i,\rho(i)}^{mr},\ \sigma_{i,m}^2\}$ for the *signal* wavelet coefficients using the *noisy* signal observation.[3] The additive zero-mean white Gaussian noise n_i^k increases each mixture model variance $\sigma_{i,m}^2$ by σ_n^2. The other parameters are unchanged by the additive noise component. 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). Hence, we can easily obtain the signal wavelet model from the noisy signal by training a model for the noisy signal wavelet coefficients and then subtracting the added variance due to noise. Of course, we typically have only a single noisy signal observation at hand. Therefore, in order to insure reliable parameter estimation we must "share" similar statistical information between related wavelet coefficients. This is accomplished by tying wavelet states across trees (for example, S_i^1 and S_i^2 are tied) and within each scale, since the statistical characteristics of these coefficients are likely to be similar.

Once a trained HMM is obtained, estimation of the true signal wavelet coefficients (denoising) is very 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 estimation problems of estimating zero-mean Gaussian random variables in zero-mean additive Gaussian noise. The optimal conditional mean estimate of y_i^k, given w_i^k and the state s_i^k, is

$$E\left[Y_i^k | W_i^k = w_i^k, S_i^k = m\right] \;=\; \frac{\sigma_{i,m}^2}{\sigma_n^2 + \sigma_{i,m}^2}\, w_i^k. \tag{10}$$

Now recall that by-products of the EM algorithm are the hidden state probabilities $p(S_i^k|\mathbf{W}^k, \boldsymbol{\theta})$ 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

$$E\left[y_i^k|\mathbf{W}^k, \boldsymbol{\theta}\right] \;=\; \sum_m p(S_i^k = m|\mathbf{W}^k, \boldsymbol{\theta})\, \frac{\sigma_{m,i}^2}{\sigma_n^2 + \sigma_{m,i}^2}\, w_i^k. \tag{11}$$

The final signal estimate (denoised signal) is computed as the inverse wavelet transform of these estimates of the signal wavelet coefficients. Note that only the wavelet coefficients are processed. The original scaling coefficients are used in the inverse transform.

[3] 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.

Table 1: *Denoising results for Donoho and Johnstone's benchmark 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

We next compare our "empirical" Bayesian denoising procedure using the IM and HMT with current state-of-the-art wavelet denoising algorithms.[4] 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 soft 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. MSE results are tabulated for denoising Donoho and Johnstone's standard 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 in this case is the Heavisine signal.

Figure 7 illustrates the subjective improvement of the HMT model for denoising the Doppler signal in white Gaussian noise of power $\sigma_n^2 = 2.25$. The HMT denoising method offers two significant advantages over the other methods: (1) HMT denoising is smoother than both SureShrink and IM, and (2) HMT denoising preserves the high-frequency components at the beginning of the signal much better than the other two 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

Our marriage of wavelet transforms and HMMs yields a flexible framework for 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,

[4] 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 SureShrink algorithm and the Bayesian algorithm of Chipman et al. use the maximum possible number of wavelet decomposition levels (within the resolution limits of the wavelet filter). The IM and HMT algorithms used a seven-level wavelet decomposition. For Table 1, error results for SureShrink and the Bayesian algorithm of Chipman et al. were quoted from [3]. Error results for IM and HMT were obtained by averaging over 1000 trials. For Figure 7, SureShrink was implemented using the "hybrid" shrinkage estimator in the WaveLab software. The Bayesian mixture algorithm [3] was not implemented for Figure 7, but is similar to IM both in its Bayesian formulation and MSE performance.

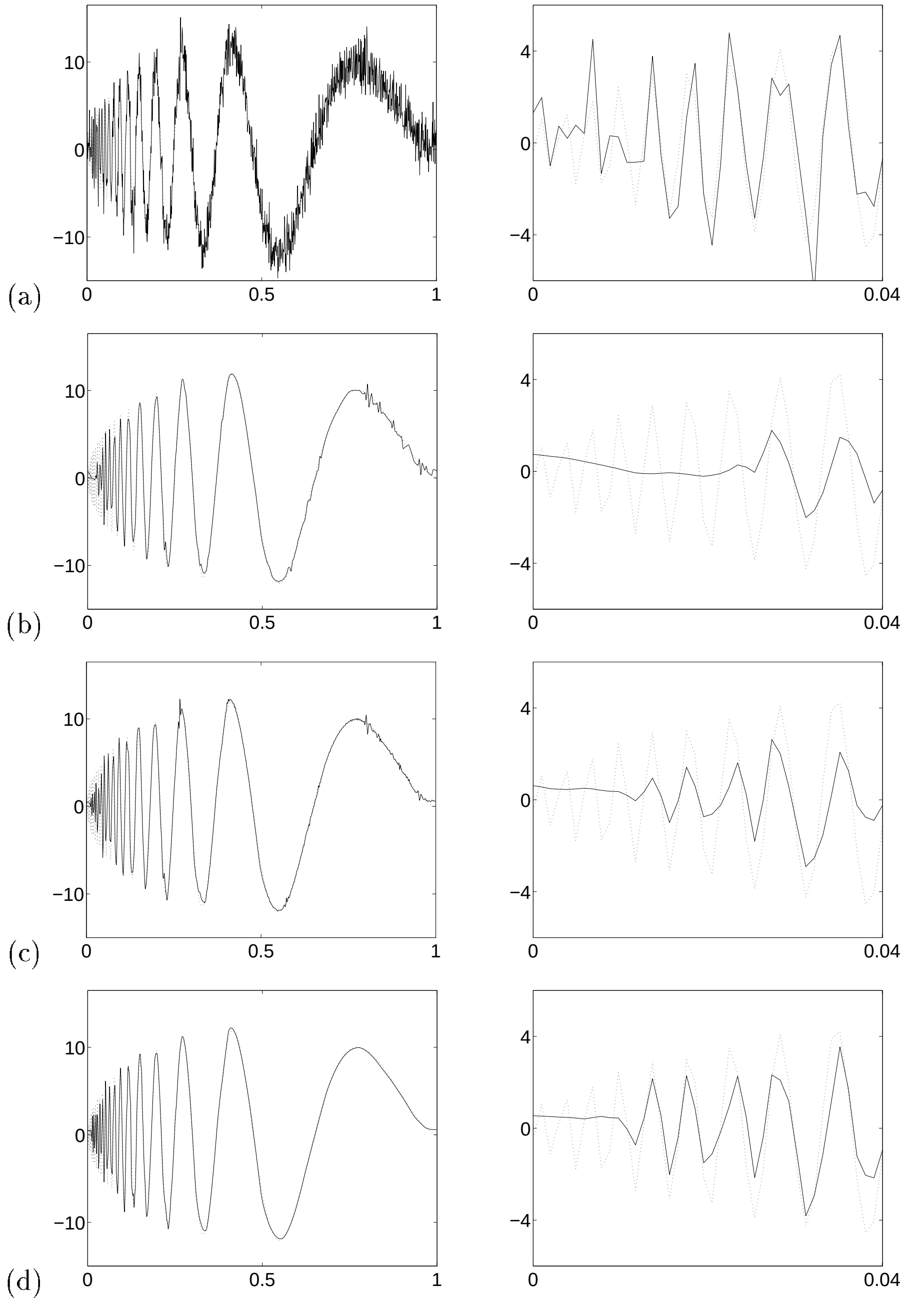

Figure 7: *Denoising the Doppler test signal in white Gaussian noise, $\sigma_n^2 = 2.25$. On each plot a dotted line is used to depict the original signal, a solid line the noisy or denoised signal. The leftmost plots depict the signals entirely; the rightmost plots depict the signals "zoomed" to the 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.*

we can train HMMs for each class, resulting in a probability model for each signal class. We use the trained HMMs to detect or classify a new signal observation by determining which probability model describes the new observation best. This task boils down to computing the likelihood of the new signal observation for each HMM and then selecting the class whose HMM provides the greatest likelihood. This approach is analogous to the use of HMMs for speech recognition [20], 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 [32].

Our approach is quite different from existing wavelet-based detection and classification schemes [4,24]. A common theme in existing wavelet-based methods is to first extract or select key wavelet coefficients that represent discriminating signal features, then use this select subset for detection/classification using classical techniques. In contrast, our approach is based on the principle of maximum likelihood. Both training and testing firmly rest on this principle, and in this way our entire methodology is co-ordinated to maximize performance.

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.

5.2.1 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:

$$\text{Class I:} \qquad x_1(t) = ax_1(t-1) + n_1(t) \tag{12}$$

$$\text{Class II:} \qquad x_2(t) = y_2(t) + 0.2y_2^3(t), \quad \text{with } y_2(t) = by_2(t-1) + n_2(t) \tag{13}$$

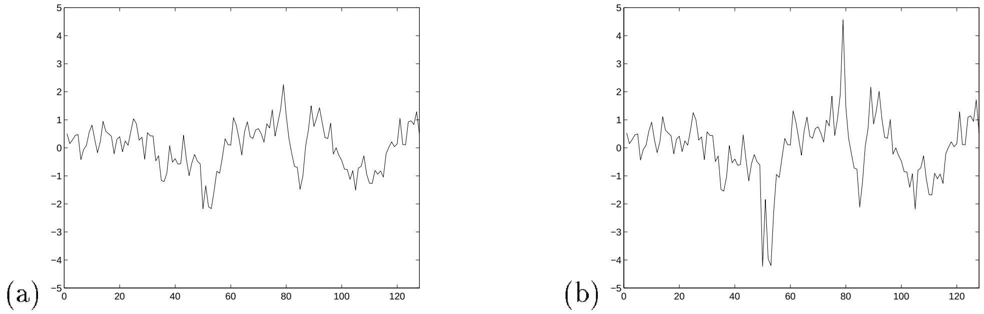

Figure 8: *Typical autoregressive (AR) signals used in nonlinear classification experiment. (a) Linear AR process (Class I). (b) Linear AR process passed through a mild cubic nonlinearity (Class II).*

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 8 (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 varies independently for each realization.) We trained an IM model and an HMT model for each class using the discrete Haar wavelet transform, with each wavelet coefficient modeled using a two-state mixture with nonzero means. The training was unconstrained (no tying). For comparison, we constructed an optimal quadratic detector under the assumption that the two classes have Gaussian distributions with different means and covariances [33]. In cases where the number of training observations N_T was smaller than the dimension of the observations, we formed the optimal 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 9.

Given a limited amount of training data, the quadratic detector has a difficult time distinguishing the classes and thus offers very poor performance. In contrast, the HMM wavelet models 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

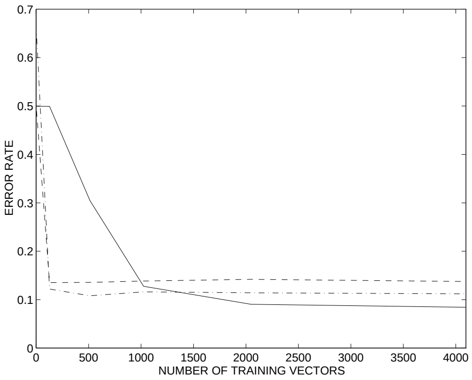

Figure 9: *Minimum probability of error rates for quadratic classifier (solid), wavelet-domain IM model classifier (dash), and wavelet-domain HMT model classifier (dash-dot).*

that with 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 performance 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.

5.2.2 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 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:

$$\text{Class I:} \qquad x_1(t) = a_1 + n_1(t) \tag{14}$$

$$\text{Class II:} \qquad y_2(t) = a_2 I_{t\in\{1,\ldots,\tau\}} + b_2 I_{t\in\{\tau+1,\ldots,128\}} + n_2(t) \tag{15}$$

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 $\{16, \ldots, 112\}$. An excellent treatment of many existing methods for the detection of abrupt changes in given in [34]. 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. Examples of signals from each class are shown in Figure 10.

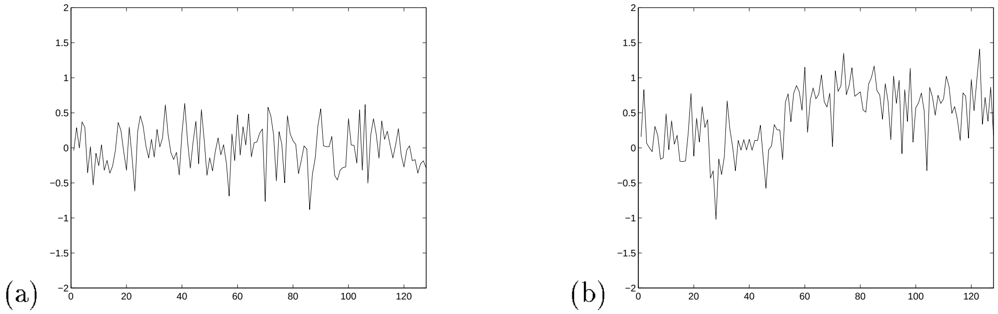

Figure 10: *Typical signals for the abrupt change detection experiment. (a) Gaussian white noise added to constant signal (Class I). (b) Gaussian white noise added to signal with abrupt change (Class II).*

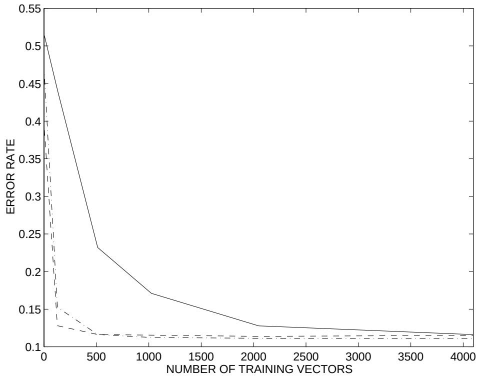

Figure 11: *Detection of an abrupt change. Minimum probability of error rates for quadratic classifier (solid), wavelet-domain IM model classifier (dash), and wavelet-domain HMT model classifier (dash-dot).*

In this example, we again designed the classifiers (quadratic, 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, HMT model, and quadratic detector, as a function of the number of training vectors M from each class, are shown in Figure 11. Again we see the fast convergence of the wavelet-domain HMM detectors with just a small number of training observations. The quadratic detector requires far more data to provide the similar performance, and in this case the wavelet-based HMMs even asymptotically outperform the quadratic detector. Keep in mind that we are not claiming that the HMMs are the optimal detector in this problem. With precise knowledge of the problem at hand, more efficient detectors could be designed. However, this experiment again demonstrates the utility of the HMM wavelet models 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 generally 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 across Scale, 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 image and signal processing based on wavelet-domain Hidden Markov models (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, two-dimensional wavelet domain HMMs are easily derived from our results, since the models and training algorithms apply to quad trees as well as binary 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, 31], with work primarily focusing on trees of discrete-valued random variables. Following [31], we develop an EM algorithm for HMTs generalized to handle continuous-valued wavelet coefficients and specialized to the tree structure provided by the wavelet transform.

In applying the EM algorithm for HMTs, our task is to fit an M-state HMT model, parameterized via $\boldsymbol{\theta} = \{p_{S_i}(m), \epsilon_{i,\rho(i)}^{nm}, \mu_{i,m}, \sigma_{i,m}^2 \;|i = 1, \ldots, P;\;\; n, m = 1, \ldots, M\}$, 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.[5] We can 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

[5] We could model the scaling coefficient as an independent mixture, as shown in Figure 3(a), or connect its state variable to the state variable of the coarsest wavelet coefficient. Either extension is straightforward.

different wavelet trees as shown in Figure 3(b). In the later case, we actually tie across trees — model 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 for HMTs converges to a locally-optimal ML fit. The iterative structure in this case is as follows:

Initialization: Select an initial model estimate $\boldsymbol{\theta}$.

1. E step (upward-downward algorithm): Estimate probabilities for the hidden state variables of the wavelet coefficients.

a. Up step: Propagate hidden state information up the tree.

b. Down step: Propagate hidden state information down the tree.

2. M step: Update the model $\boldsymbol{\theta}$ to maximize the expected likelihood function.

3. Convergence test: Iterate between the E step and M step until converged.

For HMTs, the M step is simple — the key step is the E step, also known as the upward-downward algorithm. To keep things clear and simple, we first develop the E step for a single tree. We then develop the EM steps for multiple trees.[6] We finish by incorporating into the EM steps the notion of tying within trees from Section 4.3.

A.1 E step for a single tree

We first focus on processing a single size-P wavelet tree, with 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 hidden state probabilities $p(S_i = m|\mathbf{W}, \boldsymbol{\theta})$ and $p(S_i = m, S_{\rho(i)} = n|\mathbf{W}, \boldsymbol{\theta})$ given the model $\boldsymbol{\theta}$. To obtain these probabilities, we introduce a number of intermediate variables.

We first introduce notation for trees of observed wavelet coefficients. Similar in structure to the trees of Figure 3, 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 its descendants. We also define $\mathcal{T}_{i\backslash j,\ j>i}$ to be the set of observed wavelet coefficients obtained by removing the subtree $\mathcal{T}_j$ from $\mathcal{T}_i$, with $\mathcal{T}_{i\backslash i}$ the null tree. 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

[6] Note that with no tying, the M step for a single tree is meaningless, since it entails fitting Gaussian mixtures to single-coefficient histograms.

wavelet coefficients (a tree-structured version of the vector $\mathbf{W}$), so in our probability expressions we interchange $\mathcal{T}_1$ and $\mathbf{W}$ when convenient.

For each subtree T_i, we define the conditional likelihoods

$$\begin{aligned} \beta_i(m) &= f(\mathcal{T}_i | S_i = m, \boldsymbol{\theta}) && (16) \\ \beta_{i,\rho(i)}(m) &= f(\mathcal{T}_i | S_{\rho(i)} = m, \boldsymbol{\theta}) && (17) \\ \beta_{\rho(i)\backslash i}(m) &= f(\mathcal{T}_{\rho(i)\backslash i} | S_{\rho(i)} = m, \boldsymbol{\theta}), && (18) \end{aligned}$$

and the joint probability functions

$$\alpha_i(m) = p(S_i = m, \mathcal{T}_{1\backslash i} | \boldsymbol{\theta}), \tag{19}$$

with S_i taking discrete values and the elements of $\mathcal{T}_{1\backslash i}$ 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 probabilitiy calculus, leads to the desired state probabilities in terms of the α's and β's. First we obtain

$$\begin{aligned} p(S_i = m, \mathcal{T}_1 | \boldsymbol{\theta}) &= \alpha_i(m)\ \beta_i(m) && (20) \\ p(S_i = m, S_{\rho(i)} = n, \mathcal{T}_1 | \boldsymbol{\theta}) &= \alpha_{\rho(i)}(n)\ \beta_{\rho(i)\backslash i}(n)\ \beta_i(m)\ \epsilon_{i,\rho(i)}^{nm}. && (21) \end{aligned}$$

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

$$f(\mathbf{W}|\boldsymbol{\theta}) = f(\mathcal{T}_1|\boldsymbol{\theta}) = \sum_{m=1}^{M} p(S_i = m, \mathcal{T}_1 | \boldsymbol{\theta}) = \sum_{m=1}^{M} \beta_i(m)\ \alpha_i(m). \tag{22}$$

Bayes rule applied to equations (20)-(22) leads to the desired conditional probabilities

$$\begin{aligned} p(S_i = m | \mathbf{W}, \boldsymbol{\theta}) &= \frac{\alpha_i(m)\ \beta_i(m)}{\sum_{n=1}^{M} \alpha_i(n)\ \beta_i(n)} && (23) \\ p(S_i = m, S_{\rho(i)} = n | \mathbf{W}, \boldsymbol{\theta}) &= \frac{\alpha_{\rho(i)}(n)\ \beta_{\rho(i)\backslash i}(n)\ \beta_i(m)\ \epsilon_{i,\rho(i)}^{nm}}{\sum_{n=1}^{M} \alpha_i(n)\ \beta_i(n)}. && (24) \end{aligned}$$

E step (Upward-Downward Algorithm)

All state variables within our HMT model are inter-dependent; in determining probabilities for these 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 propagating information from the leaves to the root; the down step calculates the α's by propagating information from the root to the leaves. Combining information from the α's and β's via equations (23)-(24), we obtain probabilities for each hidden state in the tree.

For our derivation, we focus on models with mixing components that are Gaussian with probability density function.

$$g\left(w;\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(w-\mu)^2}{2\sigma^2}\right). \tag{25}$$

More general densities can also be treated. Recall that we assign to each node i in the tree a level $l(i) \in \{1,\ldots,L\}$, with L the depth of the wavelet decomposition, $l = 1$ the finest scale and $l = L$ 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 coarsest level $l = 1$, calculate

$$\beta_i(m) = g(w_i;\mu_{i,m},\sigma^2_{i,m}), \qquad m = 1,\ldots,M.$$

1. For all state variables S_i at level l, compute for $m = 1,\ldots,M$,

$$\begin{aligned}
\beta_{i,\rho(i)}(m) &= \sum_{n=1}^{M} \epsilon^{nm}_{i,\rho(i)}\ \beta_i(n) \\
\beta_{\rho(i)}(m) &= g(w_{\rho(i)};\mu_{\rho(i),m},\sigma^2_{\rho(i),m}) \prod_{i\in c(\rho(i))} \beta_{\rho(i),i}(m) \\
\beta_{\rho(i)\backslash i}(m) &= \frac{\beta_{\rho(i)}(m)}{\beta_{i,\rho(i)}(m)}.
\end{aligned}$$

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

3. If $l = L$ then quit. Else return to step 1.

Down Step

Initialize: For state variable S_1 at level $l = L$, calculate

$$\alpha_1(m) = p_{S_1}(m), \qquad m = 1,\ldots,M.$$

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

2. For all state variables S_i at level l, compute

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

3. If $l = 1$, then quit. Else return to step 1.

A.2 EM steps for multiple trees

To handle $K > 1$ trees, we add superscript k for tree number. We denote the observed wavelet coefficients $\mathbf{W} = [\mathbf{W}^1\ \mathbf{W}^2\ \ldots\ \mathbf{W}^K]$ and the hidden states $\mathbf{S} = [\mathbf{S}^1\ \mathbf{S}^2\ \ldots\ \mathbf{S}^K]$. $\mathbf{W}^k = \left[W_1^k\ W_2^k\ \ldots\ W_{N-1}^k\right]$ and $\mathbf{S}^k = \left[S_1^k\ S_2^k \ldots\ S_{N-1}^k\right]$ are vectors containing the wavelet coefficient and states of the kth tree, respectively.

To implement the E step, we apply the upward-downward algorithm independently to each of the K wavelet trees. This allows us to calculate the probabilities $p(S_i^k = m|\mathbf{W}^k, \boldsymbol{\theta})$ and $p(S_i^k = m, S_{\rho(i)}^k = n|\mathbf{W}^k, \boldsymbol{\theta})$ for each tree via equations (23) and (24).

Once probabilities for the hidden states are known, the ML parameter updates of M step are relatively simple:

$$p_{S_i}(m) = \frac{1}{K}\sum_{k=1}^{K} p(S_i^k = m|\mathbf{W}^k, \boldsymbol{\theta}) \tag{26}$$

$$\epsilon_{i,\rho(i)}^{nm} = \frac{1}{K\ p_{S_{\rho(i)}}(m)}\sum_{k=1}^{K} p(S_i^k = n, S_{\rho(i)}^k = m|\mathbf{W}^k, \boldsymbol{\theta}). \tag{27}$$

$$\mu_{i,m} = \frac{1}{K\ p_{S_i}(m)}\sum_{k=1}^{K} w_i^k\ p(S_i^k = m|\mathbf{W}^k, \boldsymbol{\theta}) \tag{28}$$

$$\sigma_{i,m}^2 = \frac{1}{K\ p_{S_i}(m)}\sum_{k=1}^{K}(w_i^k - \mu_{i,m})^2\ p(S_i^k = m|\mathbf{W}^k, \boldsymbol{\theta}) \tag{29}$$

The updates for the state probabilities $p_{S_i}(m)$ and $\epsilon_{i,\rho(i)}^{nm}$ are performed by summing the individual state probabilities and then normalizing so the probabilities sum to one. Just as for the IM model [25] or the hidden Markov (chain) model [19], the updates for the Gaussian mixture means and variances are performed by a weighted averaging of the empirical means and variances, with weights chosen in proportion to the probabilities of each mixture.

As should be clear from the E and M steps, the complexity of the EM algorithm is only linear in the number of observed wavelet coefficients. The linear 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.2.1 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 observation $\mathbf{W}^k$ with wavelet coefficients w_i^k, we write $i \sim j$ if w_i^k and w_j^k (and their states) are tied, modeled with the same underlying density. The set $[i] = \{j | w_j^k \sim w_i^k\}$ 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, coefficients in $\mathbf{W}^k$ are tied in the same way as those in $\mathbf{W}^j$) according to a collection of equivalence classes given by the $[i]$'s. In this scenario, the M step becomes:

$$p_{S_i}(m) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|[i]|} \sum_{j \in [i]} p(S_j^k = m | \mathbf{W}^k, \boldsymbol{\theta}) \tag{30}$$

$$\epsilon_{i,\rho(i)}^{nm} = \frac{1}{K \, p_{S_{\rho(i)}}(m)} \sum_{k=1}^{K} \frac{1}{|[i]|} \sum_{j \in [i]} p(S_j^k = n, S_{\rho(j)}^k = m | \mathbf{W}^k, \boldsymbol{\theta}) \tag{31}$$

$$\mu_{i,m} = \frac{1}{K \, p_{S_i}(m)} \sum_{k=1}^{K} \frac{1}{|[i]|} \sum_{j \in [i]} w_j^k \; p(S_j^k = m | \mathbf{W}^k, \boldsymbol{\theta}) \tag{32}$$

$$\sigma_{i,m}^2 = \frac{1}{K \, p_{S_i}(m)} \sum_{k=1}^{K} \frac{1}{|[i]|} \sum_{j \in [i]} (w_j^k - \mu_{j,m})^2 \; p(S_j^k = m | \mathbf{W}^k, \boldsymbol{\theta}). \tag{33}$$

Although (30)-(33) appear more computationally intensive than (26)-(29), the complexity remains the same since the common parameters for each equivalence class $[i]$ need only be calculated 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. Willsky, “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. Malfait, M. Jansen, and D. Roose, “Bayesian approach to wavelet-based image processing,” in *Joint Statistical Meetings*, (Chicago), August 1996.

[15] 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.

[16] 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.

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

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

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

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

[21] P. Smyth, D. Heckerman, and M. Jordan, "Probabilistic independence networks for hidden Markov probability models," *Neural Comp.*, vol. 9, no. 1, To appear.

[22] 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.

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

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

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

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

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

[29] 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.

[30] J. Zhang, "The mean field theory in em procedures for Markov random fields," *IEEE Trans. Signal Proc.*, vol. 40, pp. 2570–2583, October 1992.

[31] 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.

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

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

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