Denoising and Recognition using Hidden Markov Models with Observation Distributions Modeled by Hidden Markov Trees ⋆

Diego H. Milone [a,*], Leandro E. Di Persia [a], María E. Torres [a,b]

[a] *Laboratory for Signals and Computational Intelligence, Department of Informatics, National University of Litoral, Campus Santa Fe (3000), Argentina.*

[b] *Laboratory for Signals and non-Linear Dynamics, Department of Mathematics, National University of Entre Ríos, CC 47 Suc. 3 Paraná (3100), Argentina.*

Abstract

Hidden Markov models have been found very useful for a wide range of applications in machine learning and pattern recognition. The wavelet transform has emerged as a new tool for signal and image analysis. Learning models for wavelet coefficients have been mainly based on fixed-length sequences, but real applications often require to model variable-length, very long or real-time sequences. In this paper, we propose a new learning architecture for sequences analyzed on short-term basis, but not assuming stationarity within each frame. Long-term dependencies will be modeled with a hidden Markov model which, in each internal state, will deal with the local dynamics in the wavelet domain, using a hidden Markov tree. The training algorithms for all the parameters in the composite model are developed using the expectation-maximization framework. This novel learning architecture could be useful for a wide range of applications. We detail two experiments with artificial and

real data: model-based denoising and speech recognition. Denoising results indicate that the proposed model and learning algorithm are more effective than previous approaches based on isolated hidden Markov trees. In the case of the 'Doppler' benchmark sequence, with 1024 samples and additive white noise, the new method reduced the mean squared error from 1.0 to 0.0842. The proposed methods for feature extraction, modeling and learning, increased the phoneme recognition rates in 28.13%, with better convergence than models based on Gaussian mixtures.

Key words: Sequence Learning, EM Algorithm, Wavelets, Speech Recognition.

1 Introduction

Hidden Markov models (HMM) have been widely used in different areas of machine learning and pattern recognition, such as computer vision, bioinformatics, speech recognition, medical diagnosis and many others [1–4]. On the other hand, from its early applications, wavelet transform has shown to be a very interesting representation for signal and image analysis [5–7].

Learning algorithms for wavelet coefficients were initially based on the traditional assumptions of independence and Gaussianity. Statistical dependence at different scales and non-Gaussian statistics were considered in [8] with the introduction of the hidden Markov trees (HMT). Training algorithms for these models are based in a previous development of an expectation-maximization (EM) algorithm for dependence tree models [9]. In the last years, the HMT

⋆ This work is supported by the National Research Council for Science and Technology (CONICET), the National Agency for the Promotion of Science and Technology (ANPCyT-UNL PICT 11-25984, ANPCyT-UNL PAE-PICT 52 and ANPCyT-UNER PICT 11-12700), and the National University of Litoral (UNL CAID 012-72).
* Corresponding author. Tel.: +54 342 4575233 ext. 125.
Email address: dmilone@fich.unl.edu.ar (Diego H. Milone).

model has been improved in several ways, for example, using more states within each HMT node, developing more efficient algorithms for initialization and training, and extending the architecture to complex wavelets [10–12]. More recently, a variational formulation of the HMT parameter estimation has been developed in [13].

In relation to the applications, HMT has received considerable attention during the last years. For example, in computer vision we can find: object localization [14,15], texture segmentation [16–18], image denoising and inpainting [19–22], super resolution imaging [23], and writer identification [24]. Regarding signal processing applications we can mention: biomedical signal detection [25], modeling for compressive sensing [26] and audio coding [27,28].

While HMT cannot deal with long term dependencies and variable length sequences, continuous HMM provides a probability distribution over sequences of continuous data in $\mathbb{R}^N$. A general model for the continuous observation densities is the Gaussian mixture model (GMM) [29]. The HMM-GMM architecture has been widely used, for example, in speech recognition modelling [3,30]. The most important advantage of the HMM lies in that they can deal with sequences of variable length. However, if the whole sequence is analyzed with the standard discrete wavelet transform (DWT), like in the case of HMT, a representation whose structure is dependent on the sequence length is obtained. Therefore, the learning architecture must be trained and used only for this sequence length or, otherwise, a warping preprocessing is required (to fit the sequence length to the model structure). On the other hand, in HMM modeling, stationarity is generally assumed withing each observation in the sequence. This stationarity hypothesis can be removed when the observed features are extracted by the DWT, but a suitable statistical model for learning

these features in the wavelet domain is required.

In [31] a recursive hierarchical generalization of discrete HMM was proposed. The authors applied the model to learn the multiresolution structure of natural English text and cursive handwriting. Some years later, a simpler inference algorithm was developed by formulating a hierarchical HMM as a special kind of dynamic Bayesian networks [32]. A wide review about multiresolution Markov models was provided in [33], with special emphasis on applications to signal and image processing. A dual-Markov architecture was proposed in [34]. The model was trained by means of an iterative process, where the most probable sequence of states is identified, and each internal model is adapted with the selected observations. A similar Viterbi trainig, applied to image segmentation, was proposed in [35]. However, in both cases the model consists of two separated entities that work in a coupled way by means of a forced state alignment in the external model. In contrast, Weber et al. [36] proposed an EM algorithm for the full model, composed of an external HMM that for each state, an internal HMM provides the observation probability distribution.

Combining the advantages of the HMM to deal with variable length sequences and of the HMT to model the dependencies between the DWT coefficients, in this paper we propose an EM algorithm to train a composite model in which each state of the HMM uses the observation density provided by an HMT. In this HMM-HMT composite model, the HMM handles the long term dynamics of the sequence, while the local dynamics are appropriately captured, in the wavelet domain, by the set of HMT models. The proposed algorithm can be easy generalized to sequences in $\mathbb{R}^{N\times N}$ with 2-D HMTs like those used in [37].

Starting from preliminary studies [38], in this paper we provide a detailed and self-contained theoretical description of the proposed model and algorithms,

the complete proofs of the training equations, the application of the model in denoising, and an extended experimental section on phoneme recognition. In the next section we introduce the notation and the basic results for HMM and HMT. To develop the proposed model, in Section 3 we define the join likelihood and we deduce the training equations for single and multiple observation sequences. In Section 4, experiments for denoising and recognition are discussed, using artificial and real data. A detailed description of the methods used for the feature extraction and for the reconstruction is provided. In the last section we present the main conclusions and some opened ideas for future works using this novel learning architecture.

2 Preliminaries

The model proposed in this work is a composition of two Markovian models. In this section we introduce the notation and we briefly review the basic training equations for the uncoupled models. They will provide the foundations for the development of the model proposed in Section 3.

2.1 Hidden Markov Models

To model a sequence $\mathbf{W} = \mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^T$, with $\mathbf{w}^t \in \mathbb{R}^N$, a continuous HMM is defined with the structure $\vartheta = \langle \mathcal{Q}, \mathbf{A}, \boldsymbol{\pi}, \mathcal{B} \rangle$, where:

i) $\mathcal{Q} = \{Q\}$ is the set of states, where Q is a discrete random state variable taking values $q \in \{1, 2, \ldots N_Q\}$.

ii) $\mathbf{A} = [a_{ij}]$ is the matrix of transition probabilities, with $a_{ij} = \Pr(Q^t = j | Q^{t-1} = i)$, $\forall\, i, j \in \mathcal{Q}$, where $Q^t \in \mathcal{Q}$ is the model state at time $t \in$

$\{1, 2, \ldots T\}$, $a_{ij} \geq 0 \; \forall i, j$ and $\sum_j a_{ij} \triangleq 1 \; \forall i$.

iii) $\boldsymbol{\pi} = [\pi_j = \Pr(Q^1 = j)]$ is the initial state probability vector. In the case of left-to-right HMM this vector is $\boldsymbol{\pi} = \boldsymbol{\delta}_1$.

iv) $\mathcal{B} = \{b_k(\mathbf{w}^t)\}$ is the set of observation (or emission) probability distributions $b_k(\mathbf{w}^t) = \Pr(\mathbf{W}^t = \mathbf{w}^t | Q^t = k), \; \forall k \in \mathcal{Q}$.

With this definition in mind, in the next paragraphs the classic maximum likelihood estimate of an HMM will be briefly reviewed.

Assuming a first order Markov process and the statistical independence of the observations, the HMM likelihood can be defined using the probability of the observed data given the model:

$$\mathcal{L}_\vartheta(\mathbf{W}) = \sum_{\forall \mathbf{q}} \mathcal{L}_\vartheta(\mathbf{W}, \mathbf{q}) \triangleq \sum_{\forall \mathbf{q}} \prod_t a_{q^{t-1}q^t} b_{q^t}(\mathbf{w}^t), \tag{1}$$

where $\forall \mathbf{q}$ stands for over all possible state sequences $\mathbf{q} = q^1, q^2, \ldots, q^T \in \vartheta$ and $a_{01} = \pi_1 = 1$. To simplify the notation, in what follows we will indicate $\Pr(\mathbf{w}^t|q^t)$ as equivalent to $\Pr(\mathbf{W}^t = \mathbf{w}^t|Q^t = q^t)$ or, in a similar way, $\Pr(q^t|q^{t-1}) \equiv \Pr(Q^t = q^t|Q^{t-1} = q^{t-1})$.

The EM algorithm is the most widely used way to maximize this likelihood [39]. The forward-backward algorithm provides an efficient method for the expectation step [40]. The expected values for the state probabilities in ϑ can be calculated with the recursions

$$\begin{aligned} \alpha^t(j) &\triangleq \Pr\left(\mathbf{w}^1, \ldots, \mathbf{w}^t, q^t = j \,|\vartheta\right) \\ &= b_j(\mathbf{w}^t) \sum_i \alpha^{t-1}(i) a_{ij}, \end{aligned}$$

$$\begin{aligned}\beta^t(j) &\triangleq \Pr\left(\mathbf{w}^{t+1}, \ldots, \mathbf{w}^T, q^t = j \,|\vartheta\right) \\ &= \sum_k a_{jk} b_k(\mathbf{w}^{t+1}) \beta^{t+1}(k),\end{aligned}$$

initialized with $\alpha^1(i) = \pi_i b_i(\mathbf{w}^1)\ \forall i$ and $\beta^T(k) = 1\ \forall k$. Then, the probability of being in state i at time t is

$$\begin{aligned}\gamma^t(i) &\triangleq \Pr\left(q^t = i \,|\mathbf{W}, \vartheta\right) \\ &= \frac{\alpha^t(i)\beta^t(i)}{\sum_i \alpha^t(i)\beta^t(i)}\end{aligned} \tag{2}$$

and the probability of being in state i at time $t-1$, and in state j at time t is

$$\begin{aligned}\xi^t(i,j) &\triangleq \Pr\left(q^{t-1} = i, q^t = j \,|\mathbf{W}, \vartheta\right) \\ &= \frac{\alpha^{t-1}(i) a_{ij} b_j(\mathbf{w}^t) \beta^t(j)}{\sum_i \alpha^t(i)\beta^t(i)}.\end{aligned} \tag{3}$$

The learning rules can be obtained by maximizing the likelihood of the data as a function of the model parameters [41]. Thus, the transition probabilities can be estimated with

$$a_{ij} = \frac{\sum_t \xi^t(i,j)}{\sum_t \gamma^t(i)}.$$

These equations can be easily extended for training from multiple observation sequences [42]. The corresponding learning rules for the parameters of the observation distributions are dependent on the chosen model for $b_k(\mathbf{w}^t)$.

Let $\mathbf{w} = [w_1, w_2, \ldots, w_N]$ be the concatenation of the wavelet coefficients obtained after performing a DWT with J scales, without including w_0, the approximation coefficient at the coarsest scale. Therefore, $N = 2^J - 1$. The HMT can be defined with the structure $\theta = \langle \mathcal{U}, \mathcal{R}, \boldsymbol{\pi}, \boldsymbol{\epsilon}, \mathcal{F} \rangle$, where:

i) $\mathcal{U} = \{u\}$, with $u \in \{1, 2, \ldots N\}$, is the set of nodes in the tree.

ii) $\mathcal{R} = \bigcup_u \mathcal{R}_u$ is the set of states in all the nodes of the tree, denoting with $\mathcal{R}_u = \{R_u\}$ the set of discrete random state variables in the node u, and R_u taking values $r_u \in \{1, 2, \ldots M\}$.

iii) $\boldsymbol{\epsilon} = [\epsilon_{u,mn}]$, with $\epsilon_{u,mn} = \Pr(R_u = m | R_{\rho(u)} = n)$, $\forall m \in \mathcal{R}_u, \forall n \in \mathcal{R}_{\rho(u)}$, is the array whose elements hold the conditional probability of node u, being in state m, given that the state in its parent node $\rho(u)$ is n, and satisfy $\sum_m \epsilon_{u,mn} \stackrel{\circ}{=} 1$.

iv) $\boldsymbol{\pi} = [\pi_p]$, with $\pi_p = \Pr(R_1 = p)$, $\forall p \in \mathcal{R}_1$, the probabilities for the root node being on state p.

v) $\mathcal{F} = \{f_{u,m}(w_u)\}$, are the observation probability distributions, with $f_{u,m}(w_u) = \Pr(W_u = w_u | R_u = m)$ the probability of observing the wavelet coefficient w_u with the state m (in the node u).

Additionally, the following notation will be used:

- $\mathcal{C}(u) = \{c_1(u), \ldots, c_{N_u}(u)\}$ is the set of children of the node u.
- $\mathcal{T}_u$ is the subtree observed from the node u (including all its descendants).
- $\mathcal{T}_{u \setminus v}$ is the subtree from node u but excluding node v and all its descendants.

As in the sequence $\mathbf{q}$ for HMM, we will use the notation $\mathbf{r} = [r_1, r_2, \ldots, r_N]$ to refer a particular combination of hidden states in the HMT nodes.

Assuming that the following three basic properties in the HMT are true:

(1) $\Pr(r_u = m|\{r_v/v \neq u\}) = \Pr\left(r_u = m|r_{\rho(u)}, r_{c_1(u)}, r_{c_2(u)}, \ldots, r_{c_{N_u}(u)}\right)$, the Markovian dependencies for trees,

(2) $\Pr(\mathbf{w}|\mathbf{r}) = \prod_u \Pr(w_u|\mathbf{r})$, the statistical independence of the observed data given the hidden states,

(3) $\Pr(w_u|\mathbf{r}) = \Pr(w_u|r_u)$, the statistical independence of the observed coefficient in node u to the states in the other nodes of the tree,

and using the standard definition $\mathcal{L}_\theta(\mathbf{w}, \mathbf{r}) \triangleq \Pr(\mathbf{w}, \mathbf{r}|\theta)$, the HMT likelihood is

$$\mathcal{L}_\theta(\mathbf{w}) = \sum_{\forall \mathbf{r}} \mathcal{L}_\theta(\mathbf{w}, \mathbf{r}) = \sum_{\forall \mathbf{r}} \prod_u \epsilon_{u,r_u r_{\rho(u)}} f_{u,r_u}(w_u), \tag{4}$$

where $\forall \mathbf{r}$ means that we include all the possible combinations of hidden states in the tree nodes and $\epsilon_{1,r_1 r_{\rho(1)}} = \pi_{r_1}$.

For the computation of the expected values in the EM algorithm, the upward-downward recursions are used, in a similar way than the forward-backward ones in HMM. For this algorithm the following quantities are defined [9]:

$$\begin{aligned}
\alpha_u(n) &\triangleq \Pr\left(\mathcal{T}_{1 \setminus u}, r_u = n\,|\theta\right), \\
\beta_u(n) &\triangleq \Pr\left(\mathcal{T}_u\,|r_u = n, \theta\right), \\
\beta_{\rho(u),u}(n) &\triangleq \Pr\left(\mathcal{T}_u\,\middle|r_{\rho(u)} = n, \theta\right).
\end{aligned}$$

In the upward step the β quantities are computed as

$$\begin{aligned}
\beta_u(n) &= f_{u,n}(w_u) \prod_{v \in C(u)} \beta_{u,v}(n), \\
\beta_{\rho(u),u}(n) &= \sum_m \beta_u(m) \epsilon_{u,mn},
\end{aligned}$$

initialized with $\beta_u(n) = f_{u,n}(w_u)\ \forall u$ in the finest scale. Then, $\beta_{\rho(u),u}(n)$ is computed and the iterative process follows in the previous level, in an upward inductive tree traversal.

When the upward step reaches the root node, the downward step computes

$$\alpha_u(n) = \sum_m \frac{\epsilon_{u,nm}\beta_{\rho(u)}(m)\alpha_{\rho(u)}(m)}{\beta_{\rho(u),u}(m)},$$

starting from $\alpha_1(m) = \Pr(r_1 = m|\theta) = \pi_m$. The other two useful quantities are the probability of being in state m of node u,

$$\gamma_u(m) \triangleq \Pr(r_u = m|\mathbf{w}, \theta) = \frac{\alpha_u(m)\beta_u(m)}{\sum_n \alpha_u(n)\beta_u(n)}, \tag{5}$$

and the probability of being in state m at node u, and the state n at its parent node $\rho(u)$,

$$\begin{aligned}\xi_u(m,n) &\triangleq \Pr(r_u = m, r_{\rho(u)} = n|\mathbf{w}, \theta) \\ &= \frac{\beta_u(m)\epsilon_{u,mn}\alpha_{\rho(u)}(n)\beta_{\rho(u)}(n)/\beta_{\rho(u),u}(n)}{\sum_n \alpha_u(n)\beta_u(n)}.\end{aligned} \tag{6}$$

If we consider the maximization for multiple observations $\mathcal{W} = \left\{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^L\right\}$, with $\mathbf{w}^\ell \in \mathbb{R}^N$, the conditional probabilities $\epsilon_{u,mn}$ can be estimated from

$$\epsilon_{u,mn} = \frac{\sum_\ell \xi_u^\ell(m,n)}{\sum_\ell \gamma_{\rho(u)}^\ell(n)},$$

and using a normal distribution for the observation probability distributions,

$$f_{u,r_u}(w_u) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\frac{(w_u - \mu_{u,r_u})^2}{\sigma^2_{u,r_u}}\right), \tag{7}$$

we have [8]

$$\mu_{u,m} = \frac{\sum_\ell w_u^\ell \gamma_u^\ell(m)}{\sum_\ell \gamma_u^\ell(m)},$$

and

$$\sigma^2_{u,m} = \frac{\sum_\ell (w_u^\ell - \mu_{u,m})^2 \gamma_u^\ell(m)}{\sum_\ell \gamma_u^\ell(m)}.$$

3 The HMM-HMT Model

In this section we introduce the HMM-HMT architecture and we describe the training algorithms. Once the join likelihood for the composite model is obtained, the auxiliary function of current and new parameters will be defined. Then, this function will be maximized to obtain the learning rules. Training algorithms for single and multiple observation sequences will be presented in Subsections 3.1 and 3.2. We will revise the relations between the obtained training formulas and the well known equations of the HMM-GMM architecture.

Let be Θ an HMM like the one defined in Section 2.1 but using a set of HMTs to model the observation densities within each HMM state:

$$b_{q^t}(\mathbf{w}^t) = \sum_{\forall \mathbf{r}} \prod_{\forall u} \epsilon^{q^t}_{u,r_u r_{\rho(u)}} f^{q^t}_{u,r_u}(w_u^t). \tag{8}$$

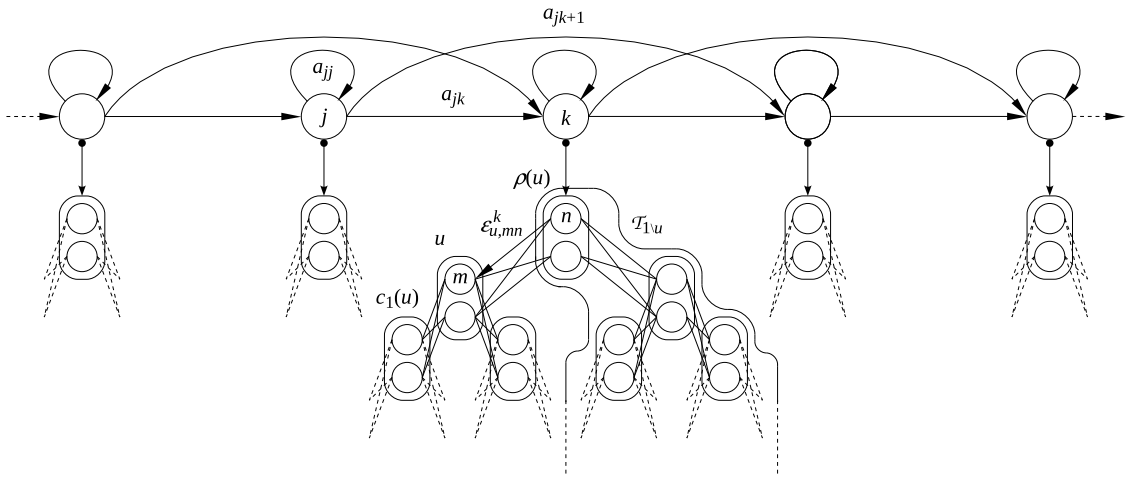

Fig. 1. Notation in the proposed HMM-HMT architecture. To simplify this figure, only two states and two children per node are shown in the tree.

To extend the notation in the composite model, we have added a superscript in the HMT variables to make reference to the state in the external HMM. For example, $\epsilon^k_{u,mn}$ will be the conditional probability that, in the state k of the external HMM, the node u is in state m given that the state of its parent node $\rho(u)$ is n (see Figure 1).

Thus, the complete join likelihood for the HMM-HMT can be computed as

$$\begin{aligned}\mathcal{L}_\Theta(\mathbf{W}) &= \sum_{\forall \mathbf{q}} \prod_t \left(a_{q^{t-1}q^t} \sum_{\forall \mathbf{r}} \prod_{\forall u} \epsilon^{q^t}_{u,r_u r_{\rho(u)}} f^{q^t}_{u,r_u}(w^t_u) \right) \\ &= \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \prod_t a_{q^{t-1}q^t} \prod_{\forall u} \epsilon^{q^t}_{u,r^t_u r^t_{\rho(u)}} f^{q^t}_{u,r^t_u}(w^t_u) \\ &\triangleq \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_\Theta(\mathbf{W}, \mathbf{q}, \mathbf{R}), \end{aligned} \tag{9}$$

where $\forall \mathbf{R}$ means all the possible sequences of all the possible combinations of hidden states $\mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^T$ in the nodes of each tree.

In this section we will obtain the maximum likelihood estimation of the model parameters. For the optimization, the auxiliary function can be defined as

$$\mathcal{D}(\Theta,\bar{\Theta}) \triangleq \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_{\Theta}(\mathbf{W},\mathbf{q},\mathbf{R}) \log\left(\mathcal{L}_{\bar{\Theta}}(\mathbf{W},\mathbf{q},\mathbf{R})\right), \tag{10}$$

and using (9)

$$\begin{aligned}\mathcal{D}(\Theta,\bar{\Theta}) = \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_{\Theta}(\mathbf{W},\mathbf{q},\mathbf{R}) \cdot \Bigg\{ \sum_{t} \log(a_{q^{t-1}q^{t}}) + \\ + \sum_{t} \sum_{\forall u} \left[\log\left(\epsilon^{q^t}_{u,r^t_u r^t_{\rho(u)}}\right) + \log\left(f^{q^t}_{u,r^t_u}(w^t_u)\right)\right] \Bigg\}.\end{aligned} \tag{11}$$

No changes are needed for the estimation of the transition probabilities in the HMM, a_{ij}. Intuitively, one would anticipate that, on each internal HMT, the estimation of the model parameters will be affected by the probability of being in the HMM state k at time t.

Let be $q^t = k$, $r^t_u = m$ and $r^t_{\rho(u)} = n$. To obtain the learning rule for $\epsilon^k_{u,mn}$ the restriction $\sum_m \epsilon^k_{u,mn} \stackrel{\circ}{=} 1$ should be satisfied. We can optimize

$$\hat{\mathcal{D}}(\Theta,\bar{\Theta}) \triangleq \mathcal{D}(\Theta,\bar{\Theta}) + \sum_{n} \lambda_n \left(\sum_{m} \epsilon^k_{u,mn} - 1 \right)$$

by means of $\nabla_{\epsilon^k_{u,mn}} \hat{\mathcal{D}}(\Theta,\bar{\Theta}) = 0$. That is

$$\sum_{t} \sum_{\substack{\forall \mathbf{q}/ \\ q^t = k}} \sum_{\substack{\forall \mathbf{R}/ \\ r^t_u = m \\ r^t_{\rho(u)} = n}} \left(\mathcal{L}_{\Theta}(\mathbf{W},\mathbf{q},\mathbf{R}) \frac{1}{\epsilon^k_{u,mn}} \right) + \lambda_n = 0. \tag{12}$$

Summing on each side and applying the restriction $\sum_m \epsilon^k_{u,mn} \stackrel{\circ}{=} 1$ we have

$$-\sum_m \epsilon^k_{u,mn}\lambda_n = \sum_m \sum_t \sum_{\substack{\forall \mathbf{q}/ \\ q^t=k}} \sum_{\substack{\forall \mathbf{R}/ \\ r^t_u=m \\ r^t_{\rho(u)}=n}} \mathcal{L}_\Theta(\mathbf{W},\mathbf{q},\mathbf{R})$$

and thus

$$\begin{aligned}\lambda_n &= -\sum_m \sum_t \mathcal{L}_\Theta(\mathbf{W}, q^t=k, r^t_u=m, r^t_{\rho(u)}=n) \\ &= -\sum_m \sum_t \Pr(q^t=k,\mathbf{W}|\Theta)\Pr(r^t_u=m, r^t_{\rho(u)}=n|\mathbf{w}^t,\theta^k).\end{aligned}$$

Using this in (12) the training equation is obtained as

$$\begin{aligned}\epsilon^k_{u,mn} &= \frac{\sum_t \Pr(q^t=k,\mathbf{W}|\Theta)\Pr(r^t_u=m, r^t_{\rho(u)}=n|\mathbf{w}^t,\theta^k)}{\sum_m\sum_t \Pr(q^t=k,\mathbf{W}|\Theta)\Pr(r^t_u=m, r^t_{\rho(u)}=n|\mathbf{w}^t,\theta^k)} \\ &= \frac{\sum_t \Pr(q^t=k|\mathbf{W},\Theta)\Pr(r^t_u=m, r^t_{\rho(u)}=n|\mathbf{w}^t,\theta^k)}{\sum_t \Pr(q^t=k|\mathbf{W},\Theta)\Pr(r^t_{\rho(u)}=n|\mathbf{w}^t,\theta^k)} \\ &= \frac{\sum_t \gamma^t(k)\xi^{tk}_u(m,n)}{\sum_t \gamma^t(k)\gamma^{tk}_{\rho(u)}(n)}. \end{aligned} \tag{13}$$

For the observation distributions we use $f^{q^t}_{u,r^t_u}(w^t_u) = \mathcal{N}\left(w^t_u, \mu^{q^t}_{u,r^t_u}, \sigma^{q^t}_{u,r^t_u}\right)$, as in (7). From (11) we have

$$\mathcal{D}(\Theta,\bar{\Theta}) = \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_{\Theta}(\mathbf{W},\mathbf{q},\mathbf{R}) \cdot \left[\sum_t \log(a_{q^{t-1}q^t}) + \right.$$
$$+ \sum_t \sum_{\forall u} \log\left(\epsilon^{q^t}_{u,r^t_u r^t_{\rho(u)}}\right) +$$
$$\left. + \sum_t \sum_{\forall u} \left(-\frac{\log(2\pi)}{2} - \log\left(\sigma^{q^t}_{u,r^t_u}\right) - \frac{\left(w^t_u - \mu^{q^t}_{u,r^t_u}\right)^2}{2\left(\sigma^{q^t}_{u,r^t_u}\right)^2} \right) \right]. \quad (14)$$

For the state $r^t_u = m$ in the tree of external state $q^t = k$, we obtain $\mu^k_{u,m}$ by optimizing

$$\nabla_{\mu^k_{u,m}} \hat{\mathcal{D}}(\Theta,\bar{\Theta}) = \sum_t \sum_{\substack{\forall \mathbf{q}/ \\ q^t = k}} \sum_{\substack{\forall \mathbf{R}/ \\ r^t_u = m}} \mathcal{L}_{\Theta}(\mathbf{W},\mathbf{q},\mathbf{R}) \left(w^t_u - \mu^{q^t}_{u,m}\right) = 0,$$

thus

$$\sum_t \mathcal{L}_{\Theta}(\mathbf{W}, q^t = k, r^t_u = m) \left(w^t_u - \mu^k_{u,m}\right) = 0$$

and we have

$$\mu^k_{u,m} = \frac{\sum_t \Pr(q^t = k, \mathbf{W}|\Theta) \Pr(r^t_u = m|\mathbf{w}^t, \theta^k) w^t_u}{\sum_t \Pr(q^t = k, \mathbf{W}|\Theta) \Pr(r^t_u = m|\mathbf{w}^t, \theta^k)}$$
$$= \frac{\sum_t \Pr(q^t = k|\mathbf{W}, \Theta) \Pr(r^t_u = m|\mathbf{w}^t, \theta^k) w^t_u}{\sum_t \Pr(q^t = k|\mathbf{W}, \Theta) \Pr(r^t_u = m|\mathbf{w}^t, \theta^k)}$$
$$= \frac{\sum_t \gamma^t(k) \gamma^{tk}_u(m) w^t_u}{\sum_t \gamma^t(k) \gamma^{tk}_u(m)}. \quad (15)$$

In a similar way for $\sigma^k_{u,m}$ we have

$$\nabla_{(\sigma^k_{u,m})^2}\hat{\mathcal{D}}(\Theta,\bar{\Theta}) = \sum_t \sum_{\substack{\forall \mathbf{q}/\\ q^t=k}} \sum_{\substack{\forall \mathbf{R}/\\ r^t_u=m}} \mathcal{L}_\Theta(\mathbf{W},\mathbf{q},\mathbf{R}) \cdot$$

$$\cdot \frac{1}{2(\sigma^k_{u,m})^2}\left(\frac{\left(w^t_u-\mu^{q^t}_{u,m}\right)^2}{(\sigma^k_{u,m})^2}-1\right)=0,$$

then

$$\sum_t \mathcal{L}_\Theta(\mathbf{W},q^t=k,r^t_u=m)\left(\frac{\left(w^t_u-\mu^k_{u,m}\right)^2}{(\sigma^k_{u,m})^2}-1\right)=0$$

and finally

$$\begin{aligned}(\sigma^k_{u,m})^2 &= \frac{\sum_t \Pr(q^t=k,\mathbf{W}|\Theta)\Pr(r^t_u=m|\mathbf{w}^t,\theta^k)\left(w^t_u-\mu^k_{u,m}\right)^2}{\sum_t \Pr(q^t=k,\mathbf{W}|\Theta)\Pr(r^t_u=m|\mathbf{w}^t,\theta^k)}\\ &= \frac{\sum_t \Pr(q^t=k|\mathbf{W},\Theta)\Pr(r^t_u=m|\mathbf{w}^t,\theta^k)\left(w^t_u-\mu^k_{u,m}\right)^2}{\sum_t \Pr(q^t=k|\mathbf{W},\Theta)\Pr(r^t_u=m|\mathbf{w}^t,\theta^k)}\\ &= \frac{\sum_t \gamma^t(k)\gamma^{tk}_u(m)\left(w^t_u-\mu^k_{u,m}\right)^2}{\sum_t \gamma^t(k)\gamma^{tk}_u(m)}.\end{aligned}$$

3.2 Multiple Observation Sequences

In practical situations we have a training set with a large number of observed data $\mathcal{W}=\left\{\mathbf{W^1},\mathbf{W^2},\ldots,\mathbf{W}^P\right\}$, where each observation consists of a sequence of evidences $\mathbf{W}^p=\mathbf{w}^{p,1},\mathbf{w}^{p,2},\ldots,\mathbf{w}^{p,T_p}$, with $\mathbf{w}^{p,t}\in\mathbb{R}^N$. In this case we define the auxiliary function

$$\mathcal{D}(\Theta,\bar{\Theta}) \triangleq \sum_{p=1}^{P}\frac{1}{\Pr\left(\mathbf{W}^p|\theta\right)}\sum_{\forall \mathbf{q}}\sum_{\forall \mathbf{R}}\mathcal{L}_\Theta(\mathbf{W}^p,\mathbf{q},\mathbf{R})\log\left(\mathcal{L}_{\bar{\Theta}}(\mathbf{W}^p,\mathbf{q},\mathbf{R})\right) \qquad (16)$$

and replacing with (9)

$$\mathcal{D}(\Theta,\bar{\Theta}) = \sum_{p=1}^{P} \frac{1}{\Pr\left(\mathbf{W}^p|\theta\right)} \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_{\Theta}(\mathbf{W}^p, \mathbf{q}, \mathbf{R}) \cdot \left\{ \sum_{t=1}^{T_p} \log(a_{q^{t-1}q^t}) + \right.$$
$$\left. + \sum_{t=1}^{T_p} \sum_{\forall u} \left[\log\left(\epsilon^{q^t}_{u,r^t_u r^t_{\rho(u)}} \right) + \log\left(f^{q^t}_{u,r^t_u}(w^{p,t}_u) \right) \right] \right\}. \tag{17}$$

The development of the training formulas follows the ones presented for a single sequence. We just summarize here the main results [1] :

- The training formula for the transition probabilities for the external HMM is

$$a_{ij} = \frac{\sum_{p=1}^{P} \sum_{t=1}^{T_p} \xi^{p,t}(i,j)}{\sum_{p=1}^{P} \sum_{t=1}^{T_p} \gamma^{p,t}(i)},$$

  where $\xi^{p,t}(i,j)$ and $\gamma^{p,t}(i)$ are computed from (3) and (2), but using each of the sequences $\mathbf{W}^p$.

- The training formula for the transition probabilities in the HMTs is

$$\epsilon^{k}_{u,mn} = \frac{\sum_{p=1}^{P} \sum_{t=1}^{T_p} \gamma^{p,t}(k) \xi^{p,tk}_u(m,n)}{\sum_{p=1}^{P} \sum_{t=1}^{T_p} \gamma^{p,t}(k) \gamma^{p,tk}_{\rho(u)}(n)}.$$

- The training formula for the parameters in the observation distributions of the HMTs are

$$\mu^{k}_{u,m} = \frac{\sum_{p=1}^{P} \sum_{t=1}^{T_p} \gamma^{p,t}(k) \gamma^{p,tk}_u(m) w^{p,t}_u}{\sum_{p=1}^{P} \sum_{t=1}^{T_p} \gamma^{p,t}(k) \gamma^{p,tk}_u(m)}$$

[1] Note that in this section we are introducing a new superscript p to denote the sequence used in the computation of the state probabilities.

and

$$(\sigma_{u,m}^k)^2 = \frac{\sum_{p=1}^{P}\sum_{t=1}^{T_p}\gamma^{p,t}(k)\gamma_u^{p,tk}(m)\left(w_u^{p,t}-\mu_{u,m}^k\right)^2}{\sum_{p=1}^{P}\sum_{t=1}^{T_p}\gamma^{p,t}(k)\gamma_u^{p,tk}(m)},$$

where $\gamma_u^{p,tk}(m)$ and $\xi_u^{p,tk}(m,n)$ are computed from (5) and (6) but using $\mathbf{w}^{p,t}$ on each θ^k.

Note that this training approach is different from the one proposed in [8] and other similar works. In these cases only one sequence or image is used to train a tied model. The latter consists of only one node and its associated transition probabilities. Our approach is more similar to that used in classification tasks, where we have many sequences to train and test the full model.

3.3 Links to HMM-GMM

When $b_k(\mathbf{w}^t)$ is a GMM we have

$$b_k\left(\mathbf{w}^t\right) = \sum_{m=1}^{M} c_m^k f_m^k\left(\mathbf{w}^t\right),$$

where $\int_{-\infty}^{+\infty} b_k(\mathbf{w}^t)\,\mathrm{d}\mathbf{w}^t \doteq 1 \;\; \forall j$, and

i) $f_m^k(\mathbf{w}^t)$ are normal distributions $\mathcal{N}\left(\mathbf{w}^t, \boldsymbol{\mu}_m^k, \mathbf{U}_m^k\right)$,

ii) $c_m^k \in \mathbb{R}^{+0}$ are the mixture weights, with $\sum_m c_m^k \doteq 1 \;\; \forall k$,

iii) $\boldsymbol{\mu}_m^k \in \mathbb{R}^N$ are the mean vectors, and

iv) $\mathbf{U}_m^k \in \mathbb{R}^{N\times N}$ are the covariance matrices.

For example, the means are estimated as [43]

$$\boldsymbol{\mu}_m^k = \frac{\sum_t \psi^t(k,m)\mathbf{w}^t}{\sum_t \psi^t(k,m)},$$

where

$$\psi^t(k,m) = \frac{\sum_j \alpha^{t-1}(j)a_{jk}c_m^k f_m^k(\mathbf{w}^t)\beta^t(k)}{\sum_i \alpha^t(i)\beta^t(i)}. \tag{18}$$

If we define

$$\gamma^{tk}(m) \triangleq \frac{c_m^k f_m^k(\mathbf{w}^t)}{b_k(\mathbf{w}^t)},$$

we can rewrite (18) as

$$\begin{aligned}\psi^t(k,m) &= \frac{c_m^k f_m^k(\mathbf{w}^t)}{b_k(\mathbf{w}^t)} \cdot \frac{\sum_j \alpha^{t-1}(j)a_{jk}b_k(\mathbf{w}^t)\beta^t(k)}{\sum_i \alpha^t(i)\beta^t(i)} \\ &= \gamma^{tk}(m)\sum_j \xi^t(j,k) \\ &= \gamma^{tk}(m)\gamma^t(k),\end{aligned}$$

and the training formulas can be written as

$$\boldsymbol{\mu}_m^k = \frac{\sum_t \gamma^t(k)\gamma^{tk}(m)\mathbf{w}^t}{\sum_t \gamma^t(k)\gamma^{tk}(m)}.$$

This equation has the same structure as (15), providing an important simplification for practical implementation of training algorithms. In the first step of the EM algorithm we can compute the conditional probability $\gamma^{tk}(m)$ for the GMMs or, in a similar way, $\gamma_u^{tk}(m)$ and $\xi_u^{tk}(m,n)$ for the HMTs. Then,

the observation probabilities for each state in the external HMM can be computed using $b_k(\mathbf{w}^t) = \sum_m c_m^k f_m^k(\mathbf{w}^t)$ for the GMMs or, after the upward-downward algorithm, using $b_k(\mathbf{w}^t) = \sum_n \alpha_u^{tk}(n)\beta_u^{tk}(n)$ for the HMTs. Finally, the forward-backward recursion in the external HMM allows to compute the quantities $\gamma^t(i)$ and $\xi^t(i,j)$ directly by using (2) and (3).

4 Experimental Results and Discussion

In this section we test the proposed model in two applications: a model-based denoising of artificially generated data and the automatic recognition of real speech. The first series of experiments was conducted with the Donoho's test series [44]. These data were widely used for benchmarking algorithms for signal estimation and denoising. Doppler and Heavisine data series were selected because they have similar characteristics to speech and we are interested in future applications of the model to robust speech recognition. These sequences can be generated artificially and sampled to get different lengths.

The TIMIT speech corpus [45] was used for the phoneme recognition experiments. TIMIT is a well known corpus that has been extensively used for research in automatic speech recognition. From this corpus five phonemes that are difficult to classify have been selected. We briefly describe here some of their characteristics. The voiced stops /*b*/ and /*d*/ have a very similar articulation (bilabial/alveolar) and different phonetic variants according to the context (allophones). Vowels /*eh*/ and /*ih*/ were selected because their formants are very close. Thus, these phonemes are very confusable. To complete the selected phonemes, the affricate phoneme /*jh*/ was added as representative of the voiceless group [46]. Table 1 shows the number of train and test

Table 1
Selected phonemes from TIMIT speech corpus. The duration values give an idea of the length variability in the sequences.

Phonemes		/b/	/d/	/eh/	/ih/	/jh/
Train patterns		2181	3548	3853	5051	1209
Test patterns		886	1245	1440	1709	372
Duration	min.	93	81	480	300	315
[samples]	ave.	292	363	1417	1234	942
	max.	1229	1468	3280	3595	2194

samples for each phoneme in the TIMIT corpus. The minimum, maximum and average duration of the phonemes are also indicated, to provide an idea of the length variability in the patterns to be recognized.

Regarding practical issues, the training equations have been implemented in logarithmic scale to perform a more efficient computation of products and to avoid underflow errors in the probability accumulators (about this problem see [11]). In addition, underflow errors have been reduced because in the HMM-HMT architecture each DWT is in a lower dimension than the one resulting from an unique HMT for the whole sequence. Recall that, in our model, the long temporal dependencies are modeled with the external HMM. All the learning algorithms and the transforms used in the experiments have been implemented in C++ from scratch [2].

The experimental details and results will be presented in what follows.

[2] The source code can be downloaded from:
`http://downloads.sourceforge.net/sourcesinc/po-0.3.7`

This application not only requires the extraction of features, but also the reconstruction of the cleaned data. Figure 2 shows the whole process of feature extraction (or analysis) and the reconstruction (or synthesis) after denoising with the HMM-HMT. Frame by frame, each local feature is extracted using a Hamming window of width N_w, shifted in steps of N_s samples [30]. To avoid the information loss at the beginning of the sequence, the first window begins N_o samples out (with zero padding). A similar procedure is used to avoid information loss at the end of the sequence.

In the next step a DWT is applied to each windowed frame. The DWT has been implemented by a fast pyramidal algorithm [5], using periodic convolutions and Daubechies-8 wavelet [47]. Preliminar tests have been carried out with other Daubechies and Spline wavelets, but no significant differences have been found in the results.

For model-based denoising, the proposed model has been applied in an empirical Wiener filter like the one used in [8]. Taking into account that in our case we allow the mean of the signal to be different from zero, it was subtracted before filtering and was added again after filtering:

$$\bar{w}_u^t = \sum_k \gamma^t(k) \sum_m \gamma_u^{tk}(m) \left(\frac{(\sigma_{u,m}^k)^2}{(h_u \tilde{\sigma}_w)^2 + (\sigma_{u,m}^k)^2} (w_u^t - \mu_{u,m}^k) + \mu_{u,m}^k \right), \qquad (19)$$

where w_u^t is the noisy wavelet coefficient and $\bar{w}_u^t$ the cleaned one. The state probabilities are: 1) $\gamma^t(k)$, the probability of being in state k (of the external HMM) at time t and 2) $\gamma_u^{tk}(m)$, the probability of being in state m of the node u, in the HMT corresponding to the state k in the HMM and at time t. The

model parameters are the estimated deviation ($\sigma^k_{u,m}$) and the estimated mean ($\mu^k_{u,m}$) of the coefficient at node u, according to the node state m of the HMT in the state k of the HMM. Note that the estimated noise deviation, $\tilde{\sigma}_w$, is multiplied by h_u, the corresponding attenuation introduced by the window in the frame analysis, subsampled as the wavelet coefficient in the node u. For the noise deviation we used the median estimate but, in this case, we have considered the median of the medians in all frames:

$$\tilde{\sigma}_w = \frac{1}{0.67} \operatorname*{med}_t \left\{ \frac{1}{0.54} \operatorname*{med}_{2^{J-1}<u\leq 2^J} \left\{ |w^t_u| \right\} \right\},$$

where 0.54 is the median of the Hamming window and 0.67 is a constant empirically determined from the data (as suggested in [44]). This is a simple and fast estimator for white noise, but not very accurate. We have verified that in most of the cases the estimation was below the real deviation (around 0.8). However, by setting $\tilde{\sigma}_w$ with arbitrary values, we have empirically verified that the filter (19) is robust to these estimation biases. Using white noise with variance 1.0, and fixing $\tilde{\sigma}_w \in [0.7, 1.3]$, we performed a series of 10 denoising experiments for each $\tilde{\sigma}_w$. All the obtained mean-squared errors (MSE) were around 5% of the ones obtained with real noise variance ($\tilde{\sigma}_w = 1.0$).

HMTs with 2 states per node have been used in all the experiments of this section. The external models have been left-to-right HMM, with transitions $a_{ij} = 0 \ \forall j > i + 1$. The only limitation of this architecture is that it is not possible to model sequences with less frames than states in the model, but there is not a limitation to the maximum number of frames in the sequence.

The synthesis consists of inverting each DWT for the processed trees and add each one with the corresponding shift. Then, the inverse of the sum of all used

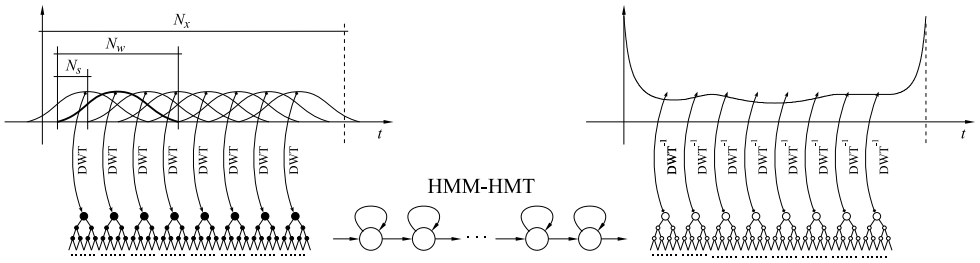

Fig. 2. Feature extraction, HMM denoising/classification and re-synthesis. In the left, each frame is multiplied by a Hamming window and transformed with the DWT. In the right, each denoised wavelet tree is inverted with the DWT^{-1} and added in their corresponding position to build the whole data. Then the inverse of the sum of all the applied windows is multiplied to the whole signal to revert the effect of the windowing. Note that the re-synthesis stage is not required in the classification experiments.

windows is applied (right part in the Figure 2). To avoid the border effects due to the periodic convolutions in the DWT, the first and last 8 samples in the inverted frame were not considered.

Several experiments have been conducted to evaluate the impact of the most important parameters regarding the feature extraction and the HMM-HMT architecture. Feature extraction has been tested for $N_w \in \{128, 256, 512\}$ and $N_s \in \{64, 128, 256\}$ (note that not all the combinations are possible, for example, the reconstruction would be impossible if $N_w = 128$ and $N_s = 128$). On each denoising test, the mean-squared error between the clean and denoised data has been computed.

The MSE as function of the number of states in the HMM, obtained in the case of Heavisine data with $N_x = 2048$, $N_w = 512$ and $N_s = 256$, is shown in Figure 3. All MSE are averages over the same 10 test sequences. It can be appreciated that the increment in the number of states in the external HMM (N_Q) reduces the MSE. Therefore, this is advantageous for the model until N_Q reaches the number of frames in the sequence. In this case, we have sequences of length $T = 8$, and then, given a left-to-right HMM, the maximum number

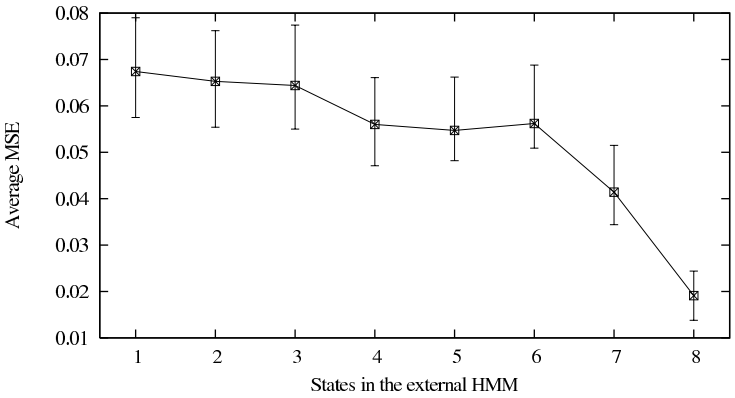

Fig. 3. Mean-squared errors for different number of states in the external HMM, N_Q. This example is for training with 30 realizations of noisy Heavisine with $N_x = 2048$, $N_w = 512$ and $N_s = 256$. Error bars show the maximum, minimum and average for testing with 10 realizations with noise variance 1.0.

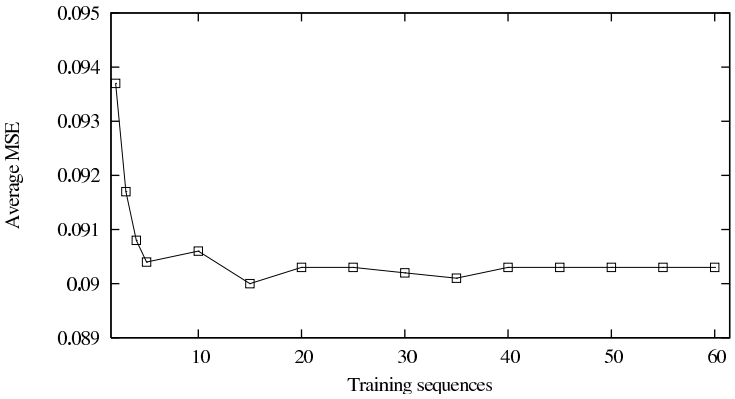

Fig. 4. Mean-squared errors for different number of training sequences. This example is for Heavisine with $N_x = 1024$, $N_w = 256$ and $N_s = 128$. First points are for 2, 3, 4 and 5 training sequences, the remaining ones are on increments of 5. All MSE are averages for testing with 10 realizations with noise variance 1.0.

of states is $N_Q = 8$.

In all these tests, the training set consisted of 30 sequences generated by adding a dither with variance 0.1. However, we have verified that the number of sequences in the training set is not so important because the learning algorithms can extract the relevant characteristics with just a few sequences (see Figure 4).

Table 2 presents a summary of results for different sequence lengths and additive noises. White noise was standard Gaussian noise of zero mean and variance

1.0, that is, a MSE of 1.0. The impulsive noise has been generated according to an ε mixture of zero-mean Gaussians

$$\mathcal{I}(\sigma_I, \sigma_B, \varepsilon) = \varepsilon \mathcal{N}(0, \sigma_I) + (1 - \varepsilon)\mathcal{N}(0, \sigma_B),$$

where σ_I is the standard deviation for the peaks of impulsive noise, σ_B is the standard deviation for the background noise ($\sigma_B \ll \sigma_I$) and ε is the weight or relative frequency of impulses occurrence [48,49]. In the present experiments, we have selected $\sigma_I = 7.5$, $\sigma_B = 0.75$ and $\varepsilon = 0.01$, so that the globlal MSE of the impulsive noise was similar to the one of the white noise.

For this type of noise, another interesting error measure is the normalized maximum amplitude error (NMAE), defined as

$$e_{\text{NMAE}} = \frac{\max(\mathbf{s} - \bar{\mathbf{s}})}{\max(\mathbf{s}) - \min(\mathbf{s})},$$

where $\mathbf{s}$ is the clean data and $\bar{\mathbf{s}}$ is the denoised data. NMAE emphasizes the maximum peak of the error instead of the average error over all the sequence. Therefore, this is an appropriate measure for impulsive noise and similar recording artifacts or outliers in the data [50].

Results on Table 2 have been obtained with the best parameters for feature extraction and modeling (N_w, N_s and N_Q), for each length N_x. All the experiments in this table have been carried out using 30 sequences for training and the error measures have been averaged over 30 different testing sequences. It can be seen that as N_x increases, it is convenient to extend the window size and step in the feature extraction. As in Figure 3, the number of states N_Q, in the external HMM, also grows to fit the frames in the sequence. It means

Table 2
Denoising results for artificial sequences corrupted with additive white and impulsive noise. For each sequence length N_x, the MSE and NMAE measures are sumarized for the best combination of parameters for feature extraction and modeling (N_w, N_s and N_Q).

	Parameters				White noise		Impulsive noise	
	N_x	N_w	N_s	N_Q	MSE	NMAE	MSE	NMAE
Doppler	1024	256	128	7	0.0842	0.0860	0.0830	0.0860
	2048	256	128	9	0.0751	0.0962	0.0727	0.0986
	4096	512	256	11	0.0559	0.1062	0.0533	0.0947
Heavisine	1024	512	256	3	0.0567	0.0580	0.0581	0.0709
	2048	512	256	7	0.0359	0.0741	0.0366	0.0832
	4096	512	256	15	0.0198	0.0572	0.0206	0.0581

that the number of states grows, avoiding to model two frames with the same HMT. At this point, it is worthwhile to recall that that both in the training and in the test sequences, the features are always synchronized in time. The performance of the same model, but with wrapped and shifted sequences in the test set, has been experimentally studied in [51]. Additionally, it can be verified, as in the original work of Donoho [44], that the MSE is reduced with N_x given that the model can provide a better smoothing for the same waveform with more samples. In [8] a tied version of HMT is compared with the standard ShureShrink and other methods, for the same Donoho's test series, $N = 1024$ and using averaged MSE. The approach proposed in our work provides a more complete model because it includes the relationships between parent and children nodes separately, at all levels of resolution and all time shifts. Our approach produced a lower MSE compared to the isolated and tied HMT due to two additional reasons: the flexibility provided for the external states in the HMM and the reduced depth in each HMT. The external states in the HMM provides more expression power for the modeling of long-term

dynamics, and provides the possibility of having a tree specialized in each region of the signal. In fact, in Table 2 we can see that the optimal number of states is always increased with the sequences length. The reduced depth is important not only in relation to the model complexity but also to improve the computation of the $\gamma_u^{tk}(m)$ probabilities, avoiding numerical issues in a simpler way than the method proposed in [11].

Concerning impulsive noise, it can be seen that the proposed method for model-based denoised is not sensitive to the peaks of the noise, and both the MSE as the NMAE are similar to those for white noise. This can be seen in detail in Figure 5, where the clean Doppler and Heavisine waveforms are shown in the middle (c-d). In the top of the figure (a-b), approximately six important peaks of impulsive noise can be distinguished in each sequence. Neither of the denoised waveform versions (e-f) appear to be affected by the peaks. In these figures, it has been used an example with a NMAE approximately equals to the average value corresponding to $N_x = 1024$ in Table 2. Note that for $N_x = 2048$ and $N_x = 4096$ the NMAE was slightly increased. This could be because increasing the number of samples increases the relative number of peaks (for the same waveform and fixed probability ε).

For further analysis of denoising capabilities, in Table 3 the experiments for white noise and $N_x = 1024$ have been replicated for 10 different values of the σ_ω. In this table MSE and NMAE are complemented with the signal to noise ratio (SNR), an error measure often used in the signal processing area. White noise realizations have been scaled to obtain standard deviations from 1.0 to 10.0. Then, these noise realizations have been added to the clean sequences. In this way, the resulting MSE measured in the corrupted data is independent of the clean sequence ($e_{\mathrm{MSE}} \approx \sigma_w^2$). Note that in this Table, for $\sigma_\omega = 1.0$, the

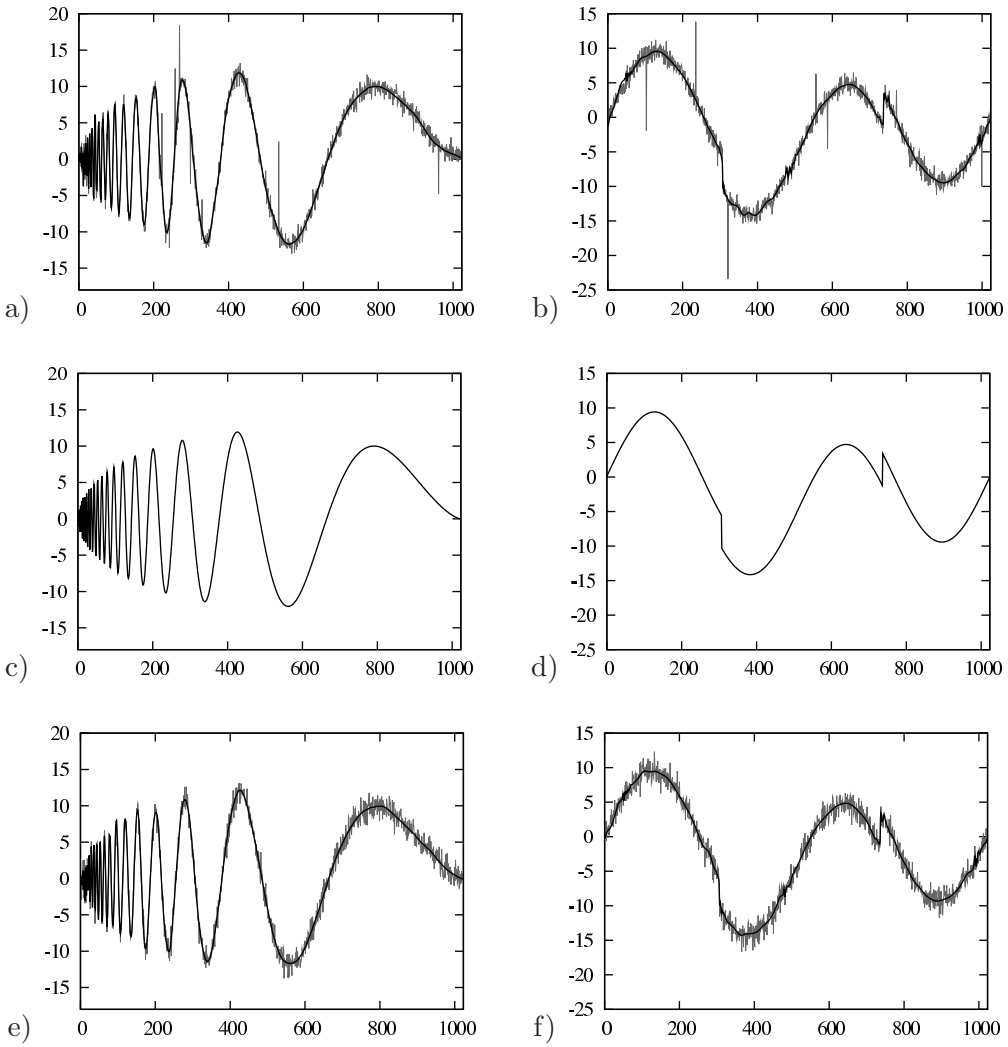

Fig. 5. Denoising examples with Doppler (left) and Heavisine (right) waveforms of length $N_x = 1024$. In the middle, the subfigures c) and d) are the original (clean) sequences. In the top, the sequences contaminated with impulsive noise are shown in gray and in the same plot the denoised sequences are shown in black solid lines. The bottom subfigures show the noisy and denoised sequencies for an example with approximately the same MSE as the corresponding average MSE in Table 2.

values of MSE and NMAE in each sequence (Doppler and Heavisine) are the same as in Table 2 for $N_x = 1024$. However, while the MSE of the corrupted sequences (quadratically) increases, the MSE of the cleaned sequences does not change significantly. In this range of σ_ω, while the value of MSE of corrupted data has increased 100 times, the MSE for Doppler denoised sequences has only increased 50 times and, in the case of Heavisine, the increment has been only about 30 times. This is mainly because the HMM-HMT model holds the

Table 3
Denoising results for artificial sequences corrupted with additive white noise of standard deviation ranging from 1.0 to 10.0. For these experiments $N_x = 1024$.

		MSE		NMAE		SNR	
	σ_w	noisy	denoised	noisy	denoised	noisy	denoised
Doppler	1.0	1.0040	0.0842	0.1411	0.0860	17.0029	27.7887
$N_w = 256$	2.0	4.0498	0.2676	0.2870	0.1317	11.5490	23.8126
$N_s = 128$	3.0	9.1717	0.5585	0.4330	0.1832	7.8454	20.5429
$N_Q = 7$	4.0	16.3711	0.9437	0.5790	0.2314	5.2232	18.1466
	5.0	25.6478	1.3337	0.7250	0.2661	3.1976	16.5707
	6.0	37.0020	1.7852	0.8710	0.2998	1.5464	15.2480
	7.0	50.4336	2.3233	1.0170	0.3349	0.1525	14.0609
	8.0	65.9426	2.9065	1.1630	0.3677	-1.0537	13.0509
	9.0	83.5291	3.5632	1.3091	0.3982	-2.1170	12.1388
	10.0	103.1930	4.2697	1.4551	0.4276	-3.0676	11.3285
Heavisine	1.0	1.0040	0.0567	0.1436	0.0580	17.2171	29.7523
$N_w = 512$	2.0	4.0498	0.1538	0.2920	0.0809	11.7704	26.4937
$N_s = 256$	3.0	9.1717	0.2812	0.4406	0.0963	8.0671	23.8297
$N_Q = 3$	4.0	16.3711	0.4346	0.5892	0.1096	5.4448	21.8896
	5.0	25.6478	0.6165	0.7379	0.1226	3.4192	20.3390
	6.0	37.0020	0.8277	0.8865	0.1347	1.7681	19.0344
	7.0	50.4336	1.0617	1.0351	0.1458	0.3741	17.9346
	8.0	65.9426	1.3198	1.1837	0.1572	-0.8321	16.9760
	9.0	83.5291	1.6003	1.3323	0.1696	-1.8954	16.1288
	10.0	103.1930	1.9030	1.4809	0.1824	-2.8460	15.3663

structure of the sequences in all the scales, with a detailed internal model for different regions in time. In the case of NMAE measures, the amplitude of peaks in corrupted data is increased about 10 times, while in the denoised sequences the NMAE increased just 5 times for Doppler and about 3 times for Heavisine. The most important discrepancies among the values of NMAE can be observed in the Doppler waveform. As Figure 5 suggests, this might be due

to the fact that the high frequency bursts in the first samples are difficult to model in these noisy conditions. The general Heavisine waveform is smoother than the previous one, and the two discontinuities are easier to model with the localizated HMTs, given that the training sequences are always synchronized.

As stated before, the main advantage of the HMM-HMT is that it provides a set of small models that fit each region of the data. In the SNR measures an average noise reduction of 13.2 dB for Doppler and 16.5 dB for Heavisine can be appreciated. Moreover, it can be noted that when more noise is added, more denoising effect is obtained, with reductions from 10 to 14 dB for Doppler and from 12 to 18 dB for Heavisine case. In the worst case (about -3 dB SNR), it can be seen that denoised sequence has a residual noise similar to the corrupted sequences with $\sigma_w < 2$, like the noisy data of Figure 5. This emphasizes the fact that the empirical Wiener filter can exploit the information, about the clean sequences structure, given by the HMM-HMT model.

4.2 Speech Recognition

In this section, we explore the application of the HMM-HMT in a traditional task in pattern recognition. Preliminary results for this task were presented in [38]. For these series of experiments, the analysis of the sequences is very similar to the presented in the previous section, but for classification, only the feature extraction is needed (see left part of the Figure 2). A separate model has been trained for each phoneme and the recognition has been performed by the maximum-likelihood classifier.

In a first study, different recognition architectures have been compared, but setting them to have the total number of trainable parameters (#TP) in the

Table 4
Recognition rates (RR%) for TIMIT phonemes using models with a similar number of trainable parameters (#TP) and the standard DWT. Each learning model was trained and tested with two frame sizes (N_w) in the feature extraction. The dialectical region 1 of TIMIT was used in this experiment.

Learning	Architecture		N_w=128		N_w = 256		Average
Model	N_Q	M	#TP	RR%	#TP	RR%	RR%
GMM	–	4	1280	28.99	2052	29.88	29.44
HMT	–	2	1280	31.36	2560	36.39	33.88
HMM-GMM	3	4	3849	35.21	6165	37.87	36.54
HMM-HMT	3	2	3849	47.34	7689	39.64	43.49

same order of magnitude. Table 4 shows the recognition rates (RR) for GMM, HMT, HMM-GMM and HMM-HMT [3]. For HMM-GMM and HMM-HMT the external HMM have connections $i \to i$, $i \to (i+1)$ and $i \to (i+2)$ (see Figure 1). The last link allows to model the shortest sequences, with less frames than states in the model. In both, GMM and HMM-GMM, the Gaussians in the mixture have been modeled with diagonal covariance matrices.

The maximum number of train iterations used for all experiments was 10, but also, as finalization criteria, the training process was stopped if the average (log) probability of the model given the training sequences was improved less than 1%. In most of the cases, the training converged after 4 to 6 iterations, but HMM-GMM models experienced several convergence problems with the DWT data. When a convergence problem was observed, the model corresponding to the last estimation with an improvement in the likelihood have been used for testing.

As can be appreciated in Table 4, the results always favor the HMM-HMT

[3] For the computation of the number of trainable parameters, recall that Gaussians in GMM and HMM-GMM are in $\mathbb{R}^{N_w}$ while Gaussians in HMT and HMM-HMT are in $\mathbb{R}^1$.

model. Comparing results of GMM against HMT, it can be seen that HMT always provides better recognition rates. As expected, this is mainly because GMM cannot adequately model the distribution of wavelet coefficients and the dependences between them. However, the incorporation of the GMM in an external HMM improves the results, even over those obtained with an isolated HMT. In this case, although the GMM capabilities are not enough to model the wavelet coefficients, the external HMM improves the recognition based on the modeling of the long-term dynamics of the phonems features. Thus, exploiting these other temporal features the HMM-GMM surpasses the RR obtained by the isolated HMT. As can be appreciated, the best combination of models is given by the HMM-HMT. It can adequately model the distribution of wavelet coefficients in the local frames, and uses the external HMM to capture the long-term dynamics in the phonemes. It must be noted that, unlike the denoising results, the phoneme models are trained and tested with sequences of highly variable lengths. In the case of the sequences for denoising experiments, a fixed length was used to train each model. As a consequence, the number of states with better results grew up to near the number of frames available in the sequence. In the classification of real sequences, like phonems on these experiments, the realizations often had a large variability in the length. Therefore, the best number of states in the model should be related to the real (long-term) dynamic of the sequences. For example, for future applications in speech recognition, the models may be adaptive either to short and long realizations of the same word.

The next experiments were focused to compare the two main models related with this work, that is, HMM using observation probabilities provided either by GMMs or HMTs. In this context, the best relative scenario for HMM-GMM is using $N_w = 256$ (see Table 4).

In Figure 6, two realizations of the English phoneme /*eh*/ are analyzed frame by frame. This phoneme, like all vowels, is a quasi-stationary signal. Therefore, its statistical properties vary very little through time. To simplify this example we have selected two realizations with the same number of frames. However, remember that, in general, these sequences have different lengths. Subfigures 6.a) and 6.b) display the result of applying the DWT frame by frame to the two realizations of /*eh*/. As it can be seen, along the scales there is a pattern slidding from a frame to the next one. It is clear that this artifact is not related with the identity of the phoneme. It has to do with the fact that DWT is not a shift-invariant representation. The largest component of this slidding pattern is due to the relation between the fundamental frequency of the phoneme and the window step (N_s) used in the feature extraction. Furthermore, other secondary components are also present, for example, those related with the formants in all voiced phonemes. Each one of these components have its own phase and frequency. Therefore, different slidding patterns are generated when the convolution is computed at the corresponding scale. Certainly, these artifacts make training data too confusable for a recognition architecture without translation-invariance.

To overcome this problem, two approaches can be applied: to modify the feature extraction in order to get invariant patterns, or to modify the recognition architecture to be robust to translation-variant representations (we are working in a robust to translation HMM-HMT). A wavelet representation for pseudo-periodic signals has been proposed in [52]. However, a simpler idea can be applied: to use, at each Scale, the Module of the Spectrum (SMS) of the wavelet coefficients, instead of the wavelet coefficients themselves. That is, given the wavelet coefficients $\mathbf{w}^t$, at frame t, obtained from a DWT with J scales (as in Section 2.2), let be $\mathbf{w}_j^t = [w_{2^J-j}, w_{2^J-j+1}, \ldots, w_{2^J-j+1-2}, w_{2^J-j+1-1}]$

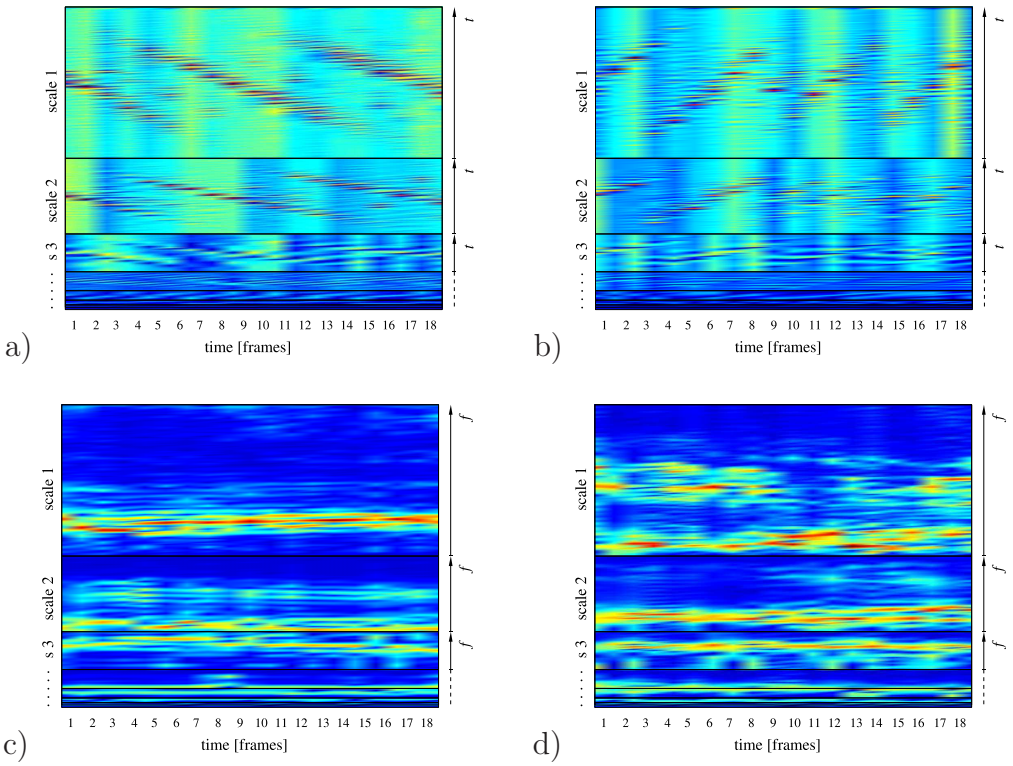

Fig. 6. Two realizations of the phoneme /*eh*/ in TIMIT corpus. In a) and b) sequences are analyzed frame by frame with the DWT and with exactly the same parameters. However, it can be seen that a slidding pattern in each scale is moving frame to frame. In c) and d) the spectrum module, computed scale by scale (SMS-DWT) and frame by frame, is shown. On each scale the coefficients were normalized to have standard deviation 1 (to use the full color scale in the image).

the detail coefficients at scale j, with $j \in [1 \dots J-1]$. Then, the SMS is defined for each scale j as $\boldsymbol{\varsigma}_j^t = |\mathcal{F}(\mathbf{w}_j^t)|$, where $\mathcal{F}$ is the discrete Fourier transform. Thus, SMS of the whole frame is the concatenation $\boldsymbol{\varsigma}^t = [w_0^t, w_1^t, \boldsymbol{\varsigma}_{J-1}^t, \boldsymbol{\varsigma}_{J-2}^t, \dots, \boldsymbol{\varsigma}_1^t]$.

Using this scale-by-scale transformation, not only the largest component is considered, but also the secondary ones. The comparative results of this approach can be appreciated in subfigures 6.c) and 6.d). In what follows we will refer to this method for feature extraction as SMS-DWT.

In Table 5 we present a fine tuning for the HMM-GMM model, trained by the standard forward backward algorithm. To obtain a better balance of classes in cross-validation, in these experiments we used 600 patterns for training and

Table 5
Recognition results for TIMIT phonemes applying the SMS post-processing to the standard DWT. HMM-GMM and HMM-HMT were trained with the forward-backward algorithm in the HMM. HMM-HMT$^{\mathrm{V}}$ was trained with the forced-alignment algorithm (see details in text).

Model	M	#TP	RR%
HMM-GMM	2	3087	60.87
	4	6165	63.80
	8	12321	61.73
	16	24633	57.20
	32	49257	61.40
	64	98505	63.73
	128	197001	62.53
	256	393993	52.93
HMM-HMT$^{\mathrm{V}}$	2	7689	56.53
HMM-HMT	2	7689	66.00

300 for testing, for each phoneme. Note that, given the number of trainable parameters, comparable architectures for HMM-HMT are HMM-GMM with 4 or 8 Gaussians in the mixtures. The first remark is that the RR have been improved in at least 20% using the SMS-DWT feature extraction in comparison with standard DWT (see Table 4). Results for 4 and 64 Gaussians in the mixture are the best for the HMM-GMM model. It can be seen that the RR do not follow a simple law with the number of Gaussians in the mixture. This fluctuating behaviour could be related to the limited modeling capability of the GMM for wavelet coefficients and the poor convergence properties of the whole HMM-GMM model in this domain. Despite these considerations, from the experimental point of view, the increasing RR for 64 Gaussians motivates us to perform experiments with more Gaussians in the mixture. As it can be seen in the same table, test for $M = 128$ and $M = 256$ have been conducted with no improvements in RR. For these model architectures, note that the

number of parameters to train is about two orders of magnitude greater than the required for HMM-HMT. Therefore, the available data in the training set could be not enough to estimate such amount of model parameters.

In this table we also compare results against two methods for training the proposed model. HMM-HMT is the model trained with the algorithm proposed in this work and HMM-HMT^{V} is the same model but trained with a forced-alignment in the external HMM. That is, for each training iteration, the Viterbi algorithm is used to compute the most probable sequence of states in the HMM and then train separately each HMT with the selected observations in its corresponding HMM state (as proposed for similar architectures in [34] and [35]). In relation to the HMM architectures and training algorithms, the HMM-HMT proposed in this work is still providing the best recognition rates. It must be noted that the training algorithm developed in this work, in addition to the formalization of the fully integrated HMM-HMT model, provides important improvements in RR with respect to a simple Viterbi alignment in the external HMM model.

In order to provide an idea of the computational costs, results reported in Table 4 demanded 30.20 s of training for the HMM-GMM, whereas the same training set demanded 240.89 s in the HMM-HMT [4] . This is mainly because the upward-downward recursions in the HMTs. For comparison, the time demanded to train the HMM-HMT with the proposed algorithm was equivalent to the time demanded to train an standard HMM-GMM with 64 Gaussians in the mixture. Furthermore, the time required to train the HMM-HMT^{V} (with the forced-aligment algorithm) was similar to the one demanded to train an standard HMM-GMM with 16 Gaussians in the mixture. Future works include

[4] Using an Intel Core 2 Duo E6600 processor.

the optimization of the proposed training algorithms.

5 Conclusions and Future Work

A novel Markov architecture for learning sequences in the wavelet domain was presented. The proposed architecture is a composite of an HMM in which the observation probabilities are provided by a set of HMTs. With this structure, the HMM captures the long-term dependencies and the HMTs deal with the local dynamics. The HMM-HMT allows learning from long or variable-length sequences, with potential applicability to real-time processing. The training algorithms were obtained using the EM framework, resulting in a set of learning rules with a simple structure. The resulting training algorithm is a Baum-Welch scheme, that takes into account all interrelations between the two structures. This yields in fact an unique full model, different to the previously proposed architectures in which two independent models were forced to work in a coupled way by means of a Viterbi-like training.

Empirical results were obtained concerning the applications of model-based denoising and speech recognition, with artificial and real data. For model-based denoising, the simulated sequences were contaminated with white and impulsive noise. In the denoised sequences an important qualitative and quantitative improvement was appreciated. The recognition rates obtained for speech recognition are very competitive, even in comparison with the state-of-the-art technologies in this application domain. The development of the full Baum-Welch algorithm for the composite model was the key which allowed the formulation of the model-based denoising and provided a clear improvement of the recognition rates in comparison with the HMM-HMT trained by forced-alignment

and the standard HMM-GMM. Recognition rate was improved from 37.87%, the best result for HMM-GMM and the standard DWT (Table 4), to 66.00% for HMM-HMT and the proposed SMS-DWT (Table 5).

From this novel architecture, we considere that many topics can be addressed in future works. For example, alternative architectures can be developed with direct links between the HMT nodes (without the external HMM). Moreover, different tying schemes can be used to reduce the total number of trainable parameters, reducing the computational cost and improving the generalization capabilities. More tests would be necessary for the HMT model, with different numbers of states per node and using other observation models within the states (for example, GMM or Laplacian distributions). It would be also interesting to go forward in the study of the translation invariance, both in the feature extraction and in the recognition model itself. Concerning to the experiments, we are planning to extend our work to continuous speech recognition, using speech contaminated by real non-stationary noises. With these experiments we expect to be able to exploit, jointly in a same HMM-HMT, the two applications presented in this paper.

References

[1] N. Sebe, I. Cohen, A. Garg, T. Huang, Machine Learning in Computer Vision, Springer, 2005.

[2] M. Gollery, Handbook of Hidden Markov Models in Bioinformatics, Chapman & Hall/CRC, 2008.

[3] M. Gales, S. Young, The Application of Hidden Markov Models in Speech Recognition, Now Publishers, 2008.

[4] S. Kim, P. Smyth, Segmental hidden Markov models with random effects for waveform modeling, Journal of Machine Learning Research 7 (2006) 945–969.

[5] S. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (7) (1989) 674–693.

[6] T. Chan, J. Shen, Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods, Society for Industrial Mathematics, 2005.

[7] S. Broughton, K. Bryan, Discrete Fourier Analysis and Wavelets: Applications to Signal and Image Processing, WileyBlackwell, 2008.

[8] M. Crouse, R. Nowak, R. Baraniuk, Wavelet-based statistical signal processing using hidden Markov models, IEEE Transactions on Signal Processing 46 (4) (1998) 886–902.

[9] O. Ronen, J. Rohlicek, M. Ostendorf, Parameter estimation of dependance tree models using the EM algorithm, IEEE Signal Processing Letters 2 (8) (1995) 157–159.

[10] G. Fan, X.-G. Xia, Improved hidden Markov models in the wavelet-domain, IEEE Transactions on Signal Processing 49 (1) (2001) 115–120.

[11] J.-B. Durand, P. Gonçalvès, Y. Guédon, Computational methods for hidden Markov trees, IEEE Transactions on Signal Processing 52 (9) (2004) 2551–2560.

[12] I. Selesnick, R. Baraniuk, N. Kingsbury, The dual-tree complex wavelet transform, IEEE Signal Processing Magazine 22 (6) (2005) 123–151.

[13] N. Dasgupta, L. Carin, Texture analysis with variational hidden Markov trees, IEEE Transactions on Signal Processing 54 (6) (2006) 2353–2356.

[14] Y. Zhang, Y. Zhang, Z. He, X. Tang, Multiscale fusion of wavelet-domain hidden Markov tree through graph cut, Image and Vision Computing In Press, Corrected Proof (2009) –.

[15] R. Ferrari, H. Zhang, C. Kube, Real-time detection of steam in video images, Pattern Recognition 40 (3) (2007) 1148 – 1159.

[16] E. Mor, M. Aladjem, Boundary refinements for wavelet-domain multiscale texture segmentation, Image and Vision Computing 23 (13) (2005) 1150 – 1158.

[17] V. R. Rallabandi, V. S. Rallabandi, Rotation-invariant texture retrieval using wavelet-based hidden Markov trees, Signal Processing 88 (10) (2008) 2593 – 2598.

[18] J. Sun, D. Gu, H. Cai, G. Liu, G. Chen, Bayesian document segmentation based on complex wavelet domain hidden Markov tree models, in: International Conference on Information and Automation (ICIA 2008), 2008, pp. 493–498.

[19] Y. Tian, J. Wang, J. Zhang, Y. Ma, A contextual hidden Markov tree model image denoising using a new nonuniform quincunx directional filter banks, in: Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2007), Vol. 1, 2007, pp. 151–154.

[20] L. Ying, C. Li, Based adaptive wavelet hidden Markov tree for microarray image enhancement, in: The 2nd International Conference on Bioinformatics and Biomedical Engineering (ICBBE 2008), 2008, pp. 314–317.

[21] S. Lefkimmiatis, G. Papandreou, P. Maragos, Photon-limited image denoising by inference on multiscale models, in: Proc. IEEE Int. Conf. on Image Processing (ICIP-08), San Diego, CA, 2008, pp. 2332–2335.

[22] G. Papandreou, P. Maragos, A. Kokaram, Image inpainting with a wavelet domain hidden Markov tree model, in: Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-2008), Las Vegas, Nevada, 2008, pp. 773–776.

[23] F. Li, X. Jia, D. Fraser, Universal HMT based super resolution for remote sensing images, in: 15th IEEE International Conference on Image Processing (ICIP 2008), 2008, pp. 333–336.

[24] Z. He, X. You, Y. Y. Tang, Writer identification of chinese handwriting documents using hidden Markov tree model, Pattern Recognition 41 (4) (2008) 1295 – 1307.

[25] S. Graja, J.-M. Boucher, Hidden Markov tree model applied to ECG delineation, IEEE Transactions on Instrumentation and Measurement 54 (6) (2005) 2163–2168.

[26] M. Duarte, M. Wakin, R. Baraniuk, Wavelet-domain compressive signal reconstruction using a hidden Markov tree model, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008), 2008, pp. 5137–5140.

[27] S. Molla, B. Torresani, Hidden Markov tree based transient estimation for audio coding, in: Proc. IEEE International Conference on Multimedia and Expo, 2002. (ICME 2002), Vol. 1, 2002, pp. 489–492 vol.1.

[28] C. Tantibundhit, J. Boston, C. Li, J. Durrant, S. Shaiman, K. Kovacyk, A. El-Jaroudi, New signal decomposition method based speech enhancement, Signal Processing 87 (11) (2007) 2607 – 2628.

[29] C. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.

[30] L. Rabiner, B. Juang, Fundamentals of Speech Recognition, Prentice-Hall, New Jersey, 1993.

[31] S. Fine, Y. Singer, N. Tishby, The hierarchical hidden Markov model: Analysis and applications, Machine Learning 32 (1) (1998) 41–62.

[32] K. Murphy, M. Paskin, Linear time inference in hierarchical HMMs, in: T. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14, Vol. 14, MIT Press, Cambridge, MA, 2002.

[33] A. Willsky, Multiresolution Markov models for signal and image processing, Proceedings of the IEEE 90 (8) (2002) 1396–1458.

[34] N. Dasgupta, P. Runkle, L. Couchman, L. Carin, Dual hidden Markov model for characterizing wavelet coefficients from multi-aspect scattering data, Signal Processing 81 (6) (2001) 1303–1316.

[35] J. Lu, L. Carin, HMM-based multiresolution image segmentation, in: Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 4, Orlando, FL, 2002, pp. 3357–3360.

[36] K. Weber, S. Ikbal, S. Bengio, H. Bourlard, Robust speech recognition and feature extraction using HMM2, Computer Speech & Language 17 (2-3) (2003) 195–211.

[37] M. Ichir, A. Mohammad-Djafari, Hidden Markov models for wavelet-based blind source separation, IEEE Transactions on Image Processing 15 (7) (2006) 1887–1899.

[38] D. H. Milone, L. Di Persia, An EM algorithm to learn sequences in the wavelet domain, in: A. F. Gelbukh, A. F. Kuri Morales (Eds.), Advances in Artificial Intelligence (MICAI 2007), Vol. 4827, Springer, Aguascalientes, Mexico, 2007, pp. 518–528.

[39] O. Duda, P. Hart, D. Stork, Pattern Classification, 2nd Edition, John Wiley and Sons, New York, 2001.

[40] L. Baum, T. Petric, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, Annals Mathematical Statistics 41 (1970) 164–171.

[41] X. Huang, Y. Ariki, M. Jack, Hidden Markov Models for Speech Recognition, Edinburgh University Press, Edinburgh, 1990.

[42] L. Liporace, Maximum likelihood estimation for multivariate stochastic observations of Markov chains, IEEE Trans. Information Theory 28 (5).

[43] B.-H. Juang, Maximum-likekihood estimation for mixture multivariate stochastic observations of Markov chains, AT&T Technical Journal 64 (6) (1985) 1235–1249.

[44] D. Donoho, I. Johnstone, Adapting to unknown smoothness by wavelet shrinkage, Journal of the American Statistical Association 90 (432) (1995) 1200–1224.

[45] V. Zue, S. Sneff, J. Glass, Speech database development: TIMIT and beyond, Speech Communication 9 (4) (1990) 351–356.

[46] K. Stevens, Acoustic phonetics, MIT Press, Cambrige, Masachussets, 1998.

[47] I. Daubechies, Ten Lectures on Wavelets, No. 61 in CBMS-NSF Series in Applied Mathematics, SIAM, Philadelphia, 1992.

[48] S. Krishnan, K. Keong, S. Yan, C. Luk, Advances in Cardiac Signal Processing, 2007, Ch. 13.

[49] C.-H. Chu, E. Delp, Impulsive noise suppression and background normalization of electrocardiogram signals using morphological operators, IEEE Transactions on Biomedical Engineering 36 (2) (1989) 262–273.

[50] A. Ramakrishnan, S. Saha, ECG coding by wavelet-based linear prediction, IEEE Transactions on Biomedical Engineering 44 (12) (1997) 1253–1261.

[51] D. H. Milone, L. E. Di Persia, D. R. Tomassi, Signal denoising with hidden Markov models using hidden Markov trees as observation densities, in: IEEE Workshop on Machine Learning for Signal Processing (MLSP 2008), Cancun, Mexico, 2008, pp. 374–379.

[52] G. Evangelista, Pitch-synchronous wavelet representations of speech and music signals, IEEE Transactions on Signal Processing 41 (12) (1993) 3313–3330.