Minimum Classification Error Learning for Sequential Data in the Wavelet Domain

D.R. Tomassi[a,c,*], D.H. Milone[a], L.M. Forzani[b,c]

[a] *Laboratory for Signals and Computational Intelligence, Department of Informatics, Faculty of Engineering and Water Sciences, National University of Litoral - CONICET, Argentina*
[b] *Department of Mathematics, Faculty of Chemical Engineering, National University of Litoral, Argentina*
[c] *Institute of Applied Mathematics of Litoral - CONICET, Argentina*

Abstract

Wavelet analysis has found widespread use in signal processing and many classification tasks. Nevertheless, its use in dynamic pattern recognition have been much more restricted since most of wavelet models cannot handle variable length sequences properly. Recently, composite hidden Markov models which observe structured data in the wavelet domain were proposed to deal with this kind of sequences. In these models, hidden Markov trees account for local dynamics in a multiresolution framework, while standard hidden Markov models capture longer correlations in time. Despite these models have shown promising results in simple applications, only generative approaches have been used so far for parameter estimation. The goal of this work is to take a step forward in the development of dynamic pattern recognizers using wavelet features by introducing a new discriminative training method for this Markov models. The learning strategy relies on the minimum classification error approach and provides re-estimation formulas for fully non-tied models. Numerical experiments on phoneme recognition show important improvement over the recognition rate achieved by the same models trained using maximum likelihood estimation.

Keywords: Hidden Markov models; hidden Markov trees; discriminative

*Corresponding author. Ciudad Universitaria CC 217, Ruta Nacional No 168 Km 472.4, TE: +54 342 4575233 ext 125, FAX: +54 342 4575224, Santa Fe (3000), Argentina.
Email address: diegotomassi@gmail.com (D.R. Tomassi)

training; minimum classification error; wavelet transform.

1. Introduction

Multiscale analysis using wavelets is a well-established tool for signal and image representation. The multiresolution property of the wavelet transform and its flexibility to deal with local features simultaneously in time/space and frequency provide a suitable scenario for many signal processing and pattern recognition tasks. Initial interest in these representations was largely driven by powerful non-linear methods which relied on simple scalar transformations of coefficients. Many posterior developments kept in mind the idea of some decorrelation property of the wavelet transform or assumed very simple statistical models for the coefficients. Nevertheless, in practical applications signals and images usually show sparse representations and some dependence structure between coefficients which cannot be described with such models. Simply speaking, coefficients typically are not normally distributed and large ones tend to form clusters along scales and to propagate across scales.

As both coefficients magnitude and statistical dependencies between them carry relevant information about signals and their underlying distribution, an ideal approach to exploit these features for pattern recognition would be to know the joint distribution of the coefficients. Nevertheless, complete knowledge of this probability is infeasible, so that we should replace it with some suitable model. A nice example of such model was introduced by Crouse et al. [1]. It aims at providing a concise statistical description of the wavelet transform and its main properties. In their framework, the marginal probability of each coefficient is modeled as a Gaussian mixture driven by a hidden state variable in order to account for sparseness. Then, markovian dependencies between hidden states allow to account for the dependencies between coefficients; they give rise to a probabilistic graph which takes advantage of the natural tree structure of the wavelet transform. The resulting structure is a hidden Markov model (HMM) on the wavelet domain which is usually referred to as hidden Markov

tree (HMT).

In the last years this model has received considerable attention for several applications, including signal processing [2–4], image processing [5–8], texture classification [9, 10], computer vision [11, 12] and writer identification [13]. For classification tasks, however, it can deal only with static patterns. This limitation arises from the use of the discrete wavelet transform (DWT), which makes the structure of representations depend on the size of signals or images. To overcome this we could think of tying parameters along scales, but it would come at the price of reducing modeling power. In a typical scenario we have multiple observations available and we would want to use the whole information in order to train a full model. In these cases, the HMT should be trained and used only with signals or images with the same size; otherwise, a warping preprocessing would be required to match different sizes and that would be difficult to achieve on-line.

A different approach to deal with variable length signals in the wavelet domain was introduced by Milone et al. [14]. They exploit the probabilistic nature of the HMT to embed it as the observation model for a standard HMM. An adapted version of the expectation-maximization (EM) algorithm[1] was derived to drive the parameter estimation of fully coupled models. The resulting structure is a composite hidden Markov model in which the HMT accounts for local features in a multiresolution framework while the external HMM handles dependencies in a larger time scale and adds flexibility to deal with sequential data. The HMM-HMT model was shown to achieve promising results both for pattern recognition and for denoising tasks [14, 16]. Nevertheless, it is worth noting that training algorithms used so far provide maximum likelihood (ML) estimates of model parameters. For classifier design, it is well-known that this learning approach minimizes the expected classification error only when models can accurately describe true class distributions and when the training set

[1] The specific formultation of the EM algorithm to deal with hidden Markov models is also known as the Baum-Welch algorithm [15].

is large enough to achieve asymptotic optimality of the estimators. However, these assumptions hardly ever hold in many pattern classification applications. Models posteriors usually cannot be expected to match the true class posteriors and sample availability for parameter estimation often is too small to account for large variability in data. Thus, this approach to classifier design becomes suboptimal and minimization of expected classification error cannot be guaranteed.

These limitations are common to all classifiers based on HMMs. To overcome them, in recent years there has been a growing interest in discriminative training of hidden Markov models [17]. Unlike the previous approach that focused on the generative power of the model for each class considered independently, these methods aim to exploit the dissimilarity between models using training samples from all classes simultaneously. Several criteria have been proposed under this framework to drive the learning process, giving rise to different methods. As examples, Maximum mutual information [18] seeks to maximize the mutual information between the observations and their labels. This criterion inherits several properties from information theory, but cannot guarantee, a priori, to achieve the least error rate. On the other hand, Minimum classification error (MCE) [19] sets minimization of the error rate explicitely as the optimization task, allowing for a more direct link between the design stage of the classifier and its expected performance. Minimum phone error (MPE) [20] is another criterion widely known in the speech recognition community. It is conceptually similar to MCE, but when the data is structured at several hierarchical levels it allows to consider smaller units of the sequences to account for the classification error. For example, sentences in speech contain words and words contain phonemes. MCE would account for errors at the sentence level regardless of how many errors occured within the sentence, whereas MPE would account for errors at the phoneme level.

In this paper we will focus in MCE training. It is a discriminant analysis approach that relies in a soft approximation of the decision risk of the classifier. The learning problem becomes an optimization problem which aims to find the

parameter estimates for the set of HMMs that minimize that risk, and it is frequently solved using a gradient-based method known as generalized probabilistic descent (GPD) [21]. MCE training has shown to outperform the conventional maximum likelihood approach in many applications. This success has also triggered several efforts both to ground the method on a more principled basis [22, 23] and to improve its efficiency in large-scale applications [24]. Nevertheless, most of these works deal only with standard hidden Markov models and are not suitable for wavelet representations.

The goal of this paper is to take a step forward in the development of sequential pattern recognizers in the wavelet domain by extending the MCE/GPD discriminative learning approach to this different scenario in which data is observed in the transformed domain. Direct application of standard procedures used with Gaussian mixture-HMMs is shown not to be effective for the HMM-HMT model, requiring a modification of the way rival candidates are weighted during the classification process. To deal with this, we propose a new approximation to the misclassification loss that penalizes differences in the order of magnitude of model likelihoods rather than in their values. The advantage of the proposed learning approach over the fully generative one is assessed in a phoneme recognition experiment with highly confusable phonemes from the TIMIT speech corpus [25]. Recognition rates show important improvements on performance compared to the same models trained using the traditional maximum likelihood approach.

Related work. Many multiresolution Markov models are reviewed in [26], with special emphasis on signal and image processing. The HMT model [1], used as a building block for the model in this work, has been further improved in several ways since its introduction, for example, enlarging the state space [27], considering more general multiscale transforms [28, 29], and developing more efficient algorithms for initialization and training [30]. Hidden Markov models are widely used to model sequential data, due to their capability to handle both correlations in time and data with different sizes. The most common

observation densities used with HMM are Gaussian mixtures, but many other models have been proposed as well [31]. In [32], an EM algorithm was derived for fully coupled Markov models whose observation densities are Gaussian mixtures-HMMs. A dual Markov model for wavelet-domain data was also proposed by Dasgupta et al [33]. Nevertheless, the learning algorithm proposed by the authors considers the external HMM independently of the HMMs that serve as observation models. In contrast, the learning algorithm developed in [14] takes the external HMM and the conditional HMTs in a trully coupled way. We use this composite model as the starting point for this work and contribute here a new discriminative learning strategy for parameter estimation based on the minimum classification error approach.

The paper is organized as follows. We start by reviewing the basics of the MCE approach for classifier design in Section 2 and the definition of the composite HMM-HMT model in Section 3. We then introduce the proposed algorithm and give re-estimation formulas for all the parameters in Section 4. Experimental results for phoneme recognition are shown in Section 5 and main conclusions and future works are outlined in Section 6.

2. MCE approach for classifier design

Pattern recognition usually involves a feature extraction stage to give a suitable representation for data and a classification stage to decide the class where an unlabeled observation comes from. Such decision depends on a parameterized set of functions or models, one for each class, to measure the degree of membership of an observation to that class. Let $\{g_j(\mathbf{W};\Theta)\}_{j=1}^{\mathcal{M}}$ be that parameterized set of functions for a classification task comprising $\mathcal{M}$ classes $c_1, c_2, \ldots, c_{\mathcal{M}}$, $\mathbf{W}$ be an observation, and $\Theta = \{\vartheta_j\}_{j=1}^{\mathcal{M}}$ be the whole parameter set. An unlabeled observation $\mathbf{W}$ will be said to belong to class c_i when the decision of the classifier is

$$C(\mathbf{W};\Theta) \triangleq \arg\max_j \{g_j(\mathbf{W};\Theta)\} = i \ . \tag{1}$$

The classifier design involves the estimation of an optimum parameter set Θ^*

that minimizes the expected classification error over all the observation space. In traditional generative learning, the model for each class is trained independently of the others, using training samples for that class only in order to maximize the likelihood of the observations. Unlike this approach, in a framework of discriminative learning all models are updated simultaneously in a competitive way. This process aims to exploit differences between classes that can lead to a reduction in the error rate of the classifier. In MCE training in particular, minimization of the classification error is set formally as a goal. We now summarize the main topics of the method and provide simulation examples with a simple Gaussian model in order to motivate our developments.

2.1. Derivation of the MCE criterion

The main ingredient of the MCE approach for classifier design is a soft approximation of the misclassification risk over the set of samples available for training. Although in advance we would not guarantee minimum expected error over all possible observations working just on a finite (possibly small) training set, the method has shown to generalize well over validation sets [34, 35]. Recent works have also explained the generalization property of MCE methods by linking them with large margin estimation [23, 36]. For an observation $\mathbf{W}$, the conditional risk of misclassification is given by

$$\mathcal{R}(\Theta|\mathbf{W}) = \sum_{j=1}^{\mathcal{M}} \ell(C(\mathbf{W};\Theta), c_j)\mathrm{P}(c_j|\mathbf{W}),$$

where $\ell(C(\mathbf{W};\Theta), c_j)$ is a loss function which penalizes a wrong decision when classifying an observation $\mathbf{W}_\tau$ from class c_j. The usual choice for the loss function is the zero-one loss which assigns $\ell(C(\mathbf{W}), c_j; \Theta) = 1$ for $C(\mathbf{W}) \neq c_j$ and zero for correct classification [37]. In the training process, we look for a parameter set Θ^* that minimizes the risk

$$\mathcal{R}(\Theta) = \int \sum_{j=1}^{\mathcal{M}} \ell(C(\mathbf{W};\Theta), c_j)\mathrm{P}(c_j|\mathbf{W})d\mathrm{P}(\mathbf{W}),$$

where the integral extends over the entire signal space. Nevertheless, when designing a classifier we only have the labeled observations in the training set.

Let Ω_j stand for the subset of observations in this set which belong to class c_j. The expectation above can be replaced with an average of the loss with all the observations given equal probability mass

$$\tilde{\mathcal{R}}(\Theta) = \frac{1}{T}\sum_{\tau=1}^{T}\sum_{j=1}^{\mathcal{M}} \ell(C(\mathbf{W}_\tau;\Theta), c_j)\mathcal{I}(\mathbf{W}_\tau \in \Omega_j).$$

In the equation above $\mathcal{I}(\cdot)$ is the indicator function and T is the size of the training set.

The MCE approach minimizes a smoothed version of this empirical risk which is differentiable respect to model parameters [19]. Let us write this approximation $\ell(C(\mathbf{W};\Theta), c_j) = \ell(d_j(\mathbf{W};\Theta))$, where function $d_j(\mathbf{W};\Theta)$ simulates the decision of the classifier. Consider the current training observation comes from class c_i. A common choice for $\ell(d_i(\mathbf{W};\Theta))$ is the sigmoid [19, 38]

$$\ell(d_i(\mathbf{W};\Theta)) = \ell_i(\mathbf{W};\Theta) = \{1 + \exp\left(-\gamma d_i(\mathbf{W};\Theta) + \beta\right)\}^{-1} . \tag{2}$$

Parameter γ controls the sharpness of the sigmoid and the bias β is usually set to zero. To complete the picture we must specify function $d_i(\mathbf{W};\Theta)$, which is often referred to as the misclassification function [19, 21, 38]. In order to allow $\ell(d_i(\mathbf{W};\Theta))$ to behave close to the zero-one loss, it must give a large enough positive value for strongly misclassified observations and a small negative value when the decision is right. In addition, very confusing samples should give a value close to zero so that their related loss fall in the raising segment of the sigmoid. Remembering (1), an obvious candidate for $d_i(\mathbf{W};\Theta)$ is

$$d_i(\mathbf{W};\Theta) = \max_{j\neq i}\{g_j(\mathbf{W};\Theta)\} - g_i(\mathbf{W};\Theta) .$$

However, the maximum operation is not differentiable. As we are looking for a smoothed version of the risk, what is used in practice is a soft approximation like an ℓ_p-norm with p large. However, different selections of the misclassification function are possible (see, for example, [21]) and they can have important effects on the performance of the algorithm as we will see below.

2.2. GPD optimization

In the preceding section we have described the approximation of the empirical risk which serves as the optimization criterion for MCE learning. The simplest approach to find the parameter estimates is a gradient-based optimization technique often known as Generalized Probabilistic Descent (GPD), which is a special case of stochastic approximation [21, 38, 39]. This is simply an on-line scheme which aims at minimizing the smoothed approximation of the classification risk by updating the whole set of parameters Θ in the steepest-descent direction of the loss. Starting from an initial estimate $\hat{\Theta}_0$, the τ-th iteration of the algorithm can be summarized as

$$\hat{\Theta} \longleftarrow \hat{\Theta} - \alpha_\tau \, \nabla_\Theta \ell(\mathbf{W}_\tau; \Theta)|_{\Theta=\hat{\Theta}_\tau} \ , \tag{3}$$

where α_τ is the learning rate which decreases gradually as iterations proceed in order to assure convergence [21]. Usually, $\hat{\Theta}_0$ is chosen to be the maximum likelihood estimate of Θ and the updating process is carried out for each training signal [38]. Batch implementations can also be used to exploit parallelization [34, 36]. It is important to see that the strength of the update depends on how confusing the training observation is to the classifier and not on the correctness of the decision. This way, patterns that are similarly likely to belong to different classes induce the update of the parameter set, even if they are well classified.

2.3. An example with Gaussian models

In order to show the potential of discriminative learning over traditional maximum likelihood estimation of model parameters, let us consider a simulation example for a binary classification problem. We assume Gaussian models for both classes, but allow data from one of them, say class A, to be drawn actually from a two-component Gaussian mixture with weights 0.9 and 0.1, respectively. This is a simple example of a model not fitting the real distribution of observed data. To make the decision task more difficult, suppose also that the real distribution of class B data is a Gaussian with mean and variance very close to the global mean and variance for class A. Figure 1 illustrates the proposed

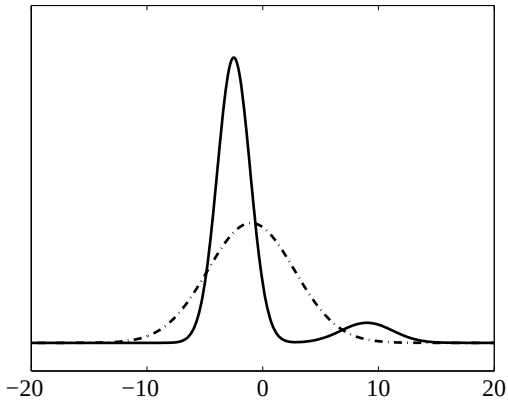


Figure 1: Distribution of the data for the proposed experiment. The solid line shows the distribution of class A while the dotted line shows the one of class B.

situation. It is clear that this is a very demanding task for a quadratic classifier based on maximum likelihood estimation. In fact, we expect it to discriminate very poorly and we are interested in seeing how much improvement can the MCE approach achieve.

Obtained results varying the number of MCE iterations are shown in Figure 2. Ten runs were carried out for each tested condition. For every run, data was generated randomly for class A first and its sample estimates for mean and variance were used to generate data from class B. A thousand samples from each class were used in both the training set and a separate testing set. Maximum likelihood estimates were used as initial guesses for the discriminative training, and standard settings were used for the MCE criterion [22]. Figure shows important improvement in recognition rate with only a few iterations of the algorithm. After five iterations, the discriminative approach reduces the error rate from 38% to 31%. Further iterations do not seem to provide significant improvements for this case.

Figure 3 compares the trained models obtained with maximum likelihood only against those estimated discriminatingly. The competitive updating pro-

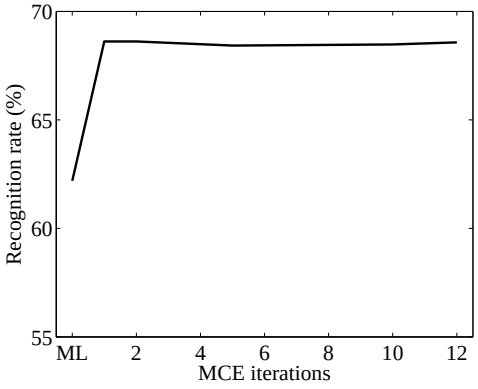
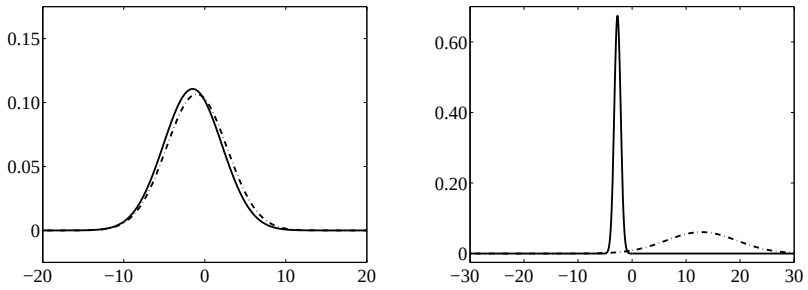


Figure 2: Recognition rates over the testing set as a function of the number of MCE iterations. Shown scores are averages over ten runs for each tested condition.

cess modifies initial model parameters so that the Gaussian for class A concentrates around the mean for the most likely component in the original mixture. On the other hand, the model for class B widens a lot to account for all other values in data. The final models used for classification are very different from the real data distributions. Thus, unlike with the maximum likelihood approach, obtained parameter estimates do not try to explain the data but only to improve the classifier performance emphasizing differences between distributions.

3. The HMM-HMT model

The HMM-HMT is a composite hidden Markov model proposed to allow modeling wavelet representations of signals with different lengths [14, 40]. In this model, signals are seen as realizations of a random process which emits wavelet coefficients in a short term basis driven by a hidden Markov chain. The emitted coefficients are not independent, but obey probabilistic dependencies structured as a tree. To make following sections more clear, we review the definition of the model along with notation and its likelihood next.



(a) (b)

Figure 3: Comparison of the trained classifiers, showing the models they use for classification. a) Models obtained with maximum likelihood estimation. b) Models obtained with MCE training after five iterations over the whole training set. Solid lines show the model for class A and dotted lines show the one for class B.

3.1. Model definition and notation

Let $\mathbf{w}^t \in \mathbb{R}^N$ be the set of coefficients emitted at time t and $\mathbf{W} = \{\mathbf{w}^1, \ldots, \mathbf{w}^T\}$ be the entire wavelet sequence. The observation is modeled by a continuous HMM defined with the structure $\vartheta = \langle \mathcal{Q}, \mathbf{A}, \boldsymbol{\pi}, \mathcal{B} \rangle$. $\mathcal{Q}$ is the set of states, which takes values $q \in \{1, 2, \ldots, N_Q\}$; $\mathbf{A} = \left\{a_{ij} = \mathrm{P}(q^t = i | q^{t-1} = j)\right\}$ is the matrix of transition probabilities from some state j to state i; $\boldsymbol{\pi}$ is the initial state probability vector; and $\mathcal{B} = \{b_k(\mathbf{w}^t) = \mathrm{P}(\mathbf{w}^t | q = k)\}$ is the set of observation (or emission) densities.

In the assumed model, for every state k of the chain, observed coefficients are drawn from a hidden Markov tree, so that $b_k(\mathbf{w}^t)$ is itself a hidden Markov structure. Figure 4 shows a sketch of the full model. We regard hidden variables in this observation model as nodes and denote with $\mathcal{U}$ the set of nodes in the tree. For future references, the set of states in all the nodes of the tree will be denoted with $\mathcal{R}$, and $\mathcal{R}_u$ will denote the set of states in the node u, taking values $r_u \in \{1, 2, \ldots, M\}$. A main assumption in the HMT is that the state in

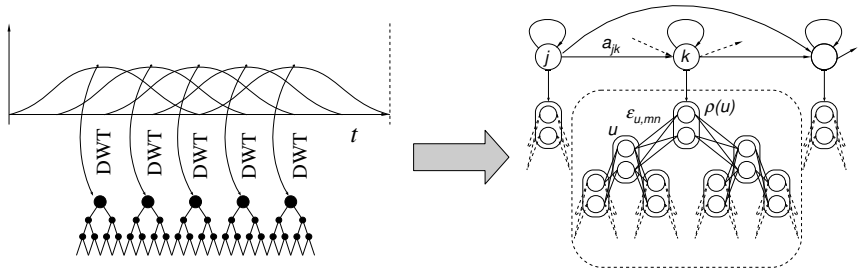


Figure 4: The HMM-HMT model. A left-to-right hidden Markov model uses hidden Markov trees as models for the observed data in the wavelet domain.

a given node depends strongly on the state in its parent node in the tree. We will denote $\epsilon_{u,mn} = \mathrm{P}(r_u = m|\rho(u) = n)$ the conditional probability of node u being in state m given that the state in its parent node $\rho(u)$ is n. The whole set of these probabilities will be denoted by $\boldsymbol{\epsilon}$ and $\boldsymbol{\kappa}$ will denote the initial state probabilities in the root node. We recall that the observed coefficients w_u are drawn from an observation model $f_{u,m}(w_u)$ conditioned on the state m of the node. We assume scalar Gaussian models $\mathcal{N}(w_u^t; \mu_{u,m}, \sigma_{u,m})$ for all of them and denote the set of all observation densities with $\mathcal{F}$. Finally, we will use superscript k to sign parameters of $b_k\left(\mathbf{w}^t\right)$, so that it is completely defined with the structure $\theta^k = \langle \mathcal{U}^k, \mathcal{R}^k, \boldsymbol{\kappa}^k, \boldsymbol{\epsilon}^k, \mathcal{F}^k \rangle$.

3.2. Likelihood of the observations

The likelihood of the first order HMM for conditionally independent observations is given by [41]

$$\mathcal{L}_\vartheta(\mathbf{W}) = \sum_{\forall \mathbf{q}} \prod_t a_{q^{t-1}q^t} b_{q^t}(\mathbf{w}^t) \ . \tag{4}$$

Similarly, for the HMT is usual to assume first order dependencies on the states of the nodes and conditionally independent observations for all of them too, so that the observation density for each HMM state reads (see [1])

$$b_{q^t}(\mathbf{w}^t) = \sum_{\forall \mathbf{r}} \prod_{\forall u} \epsilon_{u,r_u r_{\rho(u)}}^{q^t} f_{u,r_u}^{q^t}(w_u^t) \ , \tag{5}$$

with $\mathbf{r} = [r_1, r_2, \ldots, r_N]$ a combination of hidden states in the HMT nodes. At a first glance, this expression for the likelihood of the HMT may resemble that for the standard HMM. Nevertheless, we must keep in mind that transition probabilities in the time-domain HMM have very different meaning than time-scale transitions in the HMT which are either supposed to be the same across the tree. Thus, any analogy between wavelet nodes and time instants fall short despite the similar expressions. Replacing (5) in (4), the complete likelihood for the joint HMM-HMT model is:

$$\begin{aligned}
\mathcal{L}_\vartheta(\mathbf{W}) &= \sum_{\forall \mathbf{q}} \prod_t a_{q^{t-1}q^t} b_{q^t}(\mathbf{w}^t), \\
&= \sum_{\forall \mathbf{q}} \prod_t a_{q^{t-1}q^t} \sum_{\forall \mathbf{r}} \prod_{\forall u} \epsilon^{q^t}_{u,r^t_u r^t_{\rho(u)}} f^{q^t}_{u,r^t_u}(w^t_u) \\
&= \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \prod_t a_{q^{t-1}q^t} \prod_{\forall u} \epsilon^{q^t}_{u,r^t_u r^t_{\rho(u)}} f^{q^t}_{u,r^t_u}(w^t_u) \\
&\triangleq \sum_{\forall \mathbf{q}} \sum_{\forall \mathbf{R}} \mathcal{L}_\vartheta(\mathbf{W}, \mathbf{q}, \mathbf{R}) \ , \qquad (6)
\end{aligned}$$

where $a_{01} = \pi_1 = 1$. In these expressions, $\forall \mathbf{q}$ denotes that the sum is over all possible state sequences $\mathbf{q} = q^1, q^2, \ldots, q^T$ and $\forall \mathbf{R}$ accounts for all possible sequences of all possible combinations of hidden states $\mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^T$ in the nodes of each tree. See [14] for further details about the HMM-HMT model. In the following we will refer to $\mathcal{L}_\vartheta(\mathbf{W}, \mathbf{q}, \mathbf{R})$ as the joint likelihood of the observations and the states of the model.

4. Algorithm formulation

It is clear from our discussion of the general aspects of the MCE/GPD approach in Section 2 that the key points to be defined when designing a classifier under this framework are: i) the parametrized form for the discriminant functions; ii) suitable transformations of the parameters in order to account for constraints; and iii) the misclassification function $d_i(\mathbf{W}; \Theta)$. We will follow rather conventional choices for i) and ii) in Section 4.1, but we will go apart from the mainstream when considering iii) in Section 4.2. Updating formulas

are outlined in Section 4.3, while details about their derivation are left to the appendix.

4.1. Discriminant functions and parameter transformations

For a HMM-based discriminant function approach to pattern recognition, it is a usual practice to define $g_j(\mathbf{W};\Theta)$ as a function of the joint likelihood $\mathcal{L}_{\vartheta_j}(\mathbf{W},\mathbf{q},\mathbf{R})$ [38]. In particular, due to the efficiency of Viterbi's decoding algorithm for both HMM and HMT [30], it is attractive to define

$$\begin{aligned} g_j(\mathbf{W};\Theta) &= \left|\log\left(\max_{\mathbf{q},\mathbf{R}}\left\{\mathcal{L}_{\vartheta_j}(\mathbf{W},\mathbf{q},\mathbf{R})\right\}\right)\right| \\ &= -\sum_t \log a_{\bar{q}^{t-1}\bar{q}^t} - \sum_t\sum_{\forall u}\log \epsilon^{\bar{q}^t}_{u,\bar{r}^t_u \bar{r}^t_{\rho(u)}} - \sum_t\sum_{\forall u}\log f^{\bar{q}^t}_{u,\bar{r}^t_u}(w^t_u)\ , \end{aligned} \tag{7}$$

where, $\bar{q}^t$ and $\bar{r}^t$ refer to states in the external HMM and the corresponding HMT model, respectively, that achieve maximum joint likelihood. It should be noticed that this definition involves a little change in what we have said about the decision of the classifier in (1). Now this decision is ruled by the minimum (rather than the maximum) of the discriminant functions, valued at the unlabeled observation.

Despite of discriminant functions using standard model parameters, we must introduce some parameter transformations to account for restrictions if we are to use a gradient-based optimization technique such as GPD [19, 38]. To constrain a_{ij} to be a probability, we define $\tilde{a}_{ij}$ so that

$$a_{sj} = \frac{\exp\tilde{a}_{sj}}{\sum_m \exp\tilde{a}_{sm}}. \tag{8}$$

Exponentiation assures a_{ij} is non-negative and normalization makes it less or equal to one. A similar transformation is needed for the transition probabilities in the internal HMTs. With analogous arguments, we define $\tilde{\epsilon}^k_{u,mn}$ so that

$$\epsilon^k_{u,mn} = \frac{\exp\tilde{\epsilon}^k_{u,mn}}{\sum_p \exp\tilde{\epsilon}^k_{u,pn}}. \tag{9}$$

We also need to constrain the Gaussian variances to be positive-valued. To do so, we define $\tilde{\sigma}^k_{u,m}$ so that $\tilde{\sigma}^k_{u,m} = \log\sigma^k_{u,m}$. In addition, we scale the means of the Gaussian distributions as $\tilde{\mu}^k_{u,m} = \mu^k_{u,m}/\sigma^k_{u,m}$. Note that these transformations are rather standard in the literature [19, 38].

4.2. Misclassification function

For HMMs with Gaussian mixture observations and discriminant functions defined as the negative of those stated above, the frequent choice for MCE training has been simulating the decision of the classifier with the function [38]

$$\tilde{d}_i(\mathbf{W};\Theta) = -\tilde{g}_i(\mathbf{W};\Theta) + \log\left[\frac{1}{\mathcal{M}-1}\sum_{j\neq i} e^{\tilde{g}_j(\mathbf{W};\Theta)\eta}\right]^{1/\eta} . \tag{10}$$

As η becomes arbitrarily large the term in brackets approximates, up to a constant, the supremum of $\{\tilde{g}_j(\mathbf{W};\Theta)\}$ for all j different than i. This definition of the misclassification function, composed with a zero-bias approximation to the zero-one loss, penalizes confussing patterns rather than a wrong classifcation. Thus, a strong decision of the classifier implies no update of the parameter set, whether this decision is right or not. Despite it can look counterintuitive at first, it is in fact a conservative statement which avoids modifying parameter estimates due to bad data.

Nevertheless, likelihoods for the HMT model are tipically much smaller than those found for Gaussian mixtures in standard feature spaces. We can expect this noting that the joint likelihood for the HMM-HMT model involves many products which are probabilities often being very small. As a result, $g_j(\mathbf{W};\Theta)$ takes extremely low values for $\mathbf{W} \notin \Omega_j$ and the exponentiation leads to numerical underflow. A natural option to look for a similar behaviour of the misclassification function but avoiding those numerical issues is to define it as

$$\bar{d}_i(\mathbf{W};\Theta) = g_i(\mathbf{W};\Theta) - \left[\frac{1}{\mathcal{M}-1}\sum_{j\neq i} g_j(\mathbf{W};\Theta)^{-\eta}\right]^{-1/\eta} . \tag{11}$$

Roughly speaking, both of these functions account for the decision margin between the true model and the best competing ones. They weight rival candidates, but do not introduce any special corrective penalty in case of a wrong classification. Because of this, we will refer to them as symmetric misclassification functions and will use the acronym SMF to refer to (11) in what follows.

Due to the behaviour of the likelihoods for the HMM-HMT model discussed above, also their dispersion is much larger than in the Gaussian mixture-HMM

case. In this situation, similarity could be better measured comparing the order of magnitude between discriminant functions rather than their difference. To do so, we define an alternative form for discriminant functions as

$$d_i(\mathbf{W};\Theta) = 1 - \frac{\left[\frac{1}{\mathcal{M}-1}\sum_{j\neq i} g_j(\mathbf{W};\Theta)^{-\eta}\right]^{-1/\eta}}{g_i(\mathbf{W};\Theta)} \ . \tag{12}$$

As above, η is supposed to be a large positive scalar so that the sum in the numerator approaches the minimum of the terms as η grows. When the classifier takes a right decision, this minimum will be larger than $g_i(\mathbf{W};\Theta)$ and $d_i(\mathbf{W};\Theta)$ will take a negative value as required. If the observation makes decision hard for the classifier, $d_i(\mathbf{W};\Theta)$ will be close to zero. However, it must be noticed that $d_i(\mathbf{W};\Theta)$ will take no value larger than one. This implies that all misclassified observations will fall in the raising segment of the approximation to the zero-one loss if it is not too sharp. This simple fact has a very important effect in practice because it determines that every misclassified observation in the training set induces an update of the parameter set. To stress this lack of symmetry in dealing with correct and wrong classifications, we will refer to (12) as a no-symmetric misclassification function and will use the acronym nSMF to denote it in the following.

Now, we can go back to the simple Gaussian model to explore whether this always-updating feature of $d_i(\mathbf{W};\Theta)$ for misclassified sequences could affect convergence of the algorithm. We repeated the simulation experiment in Section 2.3, just replacing the standard definition for the misclassification function with nSMF.

In this experiment, the recognition rate improves a bit more slowly than in the previous case. Nevertheless, after five iterations of the whole training set performance for both choices of the misclassification function does not show significant differences. Thus, convergence is not hammered by the proposed alternative for the misclassification function. Furthermore, the variance of the error rates remains fairly the same with both methods. Therefore, at least as far as this simple experiment concerns, there is no evidence against using nSMF

in the MCE formulation. Though it would be meaningless in the Gaussian example, we will find it very important for the models we are interested in.

4.3. Updating formulas

In the following, let assume that the τ-th training sequence $\mathbf{W}_\tau$ belongs to Ω_i. To simplify notation, allow ℓ_i, d_j and g_j stand for $\ell_i(\mathbf{W};\Theta)$, $d_j(\mathbf{W};\Theta)$ and $g_j(\mathbf{W};\Theta)$, respectively. For convenience, define also

$$\zeta_{ii} \triangleq \frac{\mathrm{d}\ell_i(\mathbf{W};\Theta)}{\mathrm{d}d_i(\mathbf{W};\Theta)} \frac{\partial d_i(\mathbf{W};\Theta)}{\partial g_i(\mathbf{W};\Theta)} \ ,$$

and

$$\zeta_{ij} \triangleq \frac{\mathrm{d}\ell_i(\mathbf{W};\Theta)}{\mathrm{d}d_i(\mathbf{W};\Theta)} \frac{\partial d_i(\mathbf{W};\Theta)}{\partial g_j(\mathbf{W};\Theta)} \ ,$$

where in the last expression we assume $i \neq j$. For the misclassification function SMF, these quantities take values

$$\zeta_{ii} = \gamma\ell_i(1-\ell_i) \tag{13}$$

$$\zeta_{ij} = \gamma\ell_i(1-\ell_i)(d_i - g_i)\frac{g_j^{-\eta-1}}{\sum_{k\neq i} g_k^{-\eta}} \ . \tag{14}$$

Note that $\zeta_{ij} = -\zeta_{ii}$ for a binary classification problem. For the misclassification function nSMF, we have

$$\zeta_{ii} = \gamma\ell_i(1-\ell_i)\frac{d_i - 1}{g_i} \tag{15}$$

$$\zeta_{ij} = \gamma\ell_i(1-\ell_i)(1-d_i)\frac{g_j^{-\eta-1}}{\sum_{k\neq i} g_k^{-\eta}}. \tag{16}$$

Again, ζ_{ii} and ζ_{ij} always have opposite sign, but their absolute value it is not the same even for a two-classes only task.

The updating process works upon the transformed parameters to assure the original ones remain in their feasibility range. For the Gaussian mean associated to the state m in the node u of the HMT linked to the state k of the HMM for class c_j, the updating step is given by

$$\tilde{\mu}_{u,m}^{(j)k} \longleftarrow \tilde{\mu}_{u,m}^{(j)k} - \alpha_\tau \left.\frac{\partial \ell_i(\mathbf{W}_\tau;\Theta)}{\partial \tilde{\mu}_{u,m}^{(j)k}}\right|_{\Theta=\hat{\Theta}_\tau} \ , \tag{17}$$

where $\hat{\Theta}_\tau$ refers to the estimates of parameters obtained in the previous iteration. Applying the chain rule of differentiation and using the variables defined above, we get (see details in Appendix A):

$$\tilde{\mu}_{u,m}^{(j)k} \longleftarrow \tilde{\mu}_{u,m}^{(j)k} - \alpha_\tau \zeta \sum_t \delta(\bar{q}^t - k, \bar{r}_u^t - m) \left[\frac{w_u^t - \hat{\mu}_{u,m}^{(j)k}}{\hat{\sigma}_{u,m}^{(j)k}} \right], \tag{18}$$

where ζ takes the value ζ_{ii} or ζ_{ij} depending on whether we are dealing with a training pattern from the same class as the model or not. The delta function $\delta(\cdot,\cdot)$ is typical of Viterbi decoding. As the factor in brackets depends on the time frame through w_u^t, this function states that we only consider for the updating process the standardized observed coefficient for the node in those frames when the most likely state in the external model is k and the most likely state in the node is m. Then, to restore the original parameters we just compute $\mu_{u,m}^{(j)k}(\tau+1) = \sigma_{u,m}^{(j)k}(\tau)\tilde{\mu}_{u,m}^{(j)k}(\tau+1)$. The updating process for Gaussian variances is completely analogous to the one shown above for the means. The working expression for training reads:

$$\tilde{\sigma}_{u,m}^{(j)k} \longleftarrow \tilde{\sigma}_{u,m}^{(j)k} - \alpha_\tau \zeta \sum_t \delta(\bar{q}^t - k, \bar{r}_u^t - m) \left[\left(\frac{w_u^t - \hat{\mu}_{u,m}^{(j)k}}{\hat{\sigma}_{u,m}^{(j)k}} \right)^2 - 1 \right], \tag{19}$$

where ζ and $\delta(\cdot,\cdot)$ have the same meaning as above. Once again, Viterbi decoding acting on the Markovian dependencies decouples all the nodes and the final formula resembles just the derivative of a log-normal on its standard deviation. Then, original variances are restored doing $\sigma_{u,m}^{(j)k}(\tau+1) = \exp(\tilde{\sigma}_{u,m}^{(j)k}(\tau+1))$.

The above strategy works for updating the transition probabilities too. It is shown in the Appendix B that the updating formula for the transformed probaility $\tilde{\epsilon}_{u,mn}^{(j)k}$ reads:

$$\begin{aligned} \tilde{\epsilon}_{u,mn}^{(j)k} \longleftarrow & \tilde{\epsilon}_{u,mn}^{(j)k} - \alpha_\tau \zeta \left\{ \sum_t \delta(\bar{q}^t - k, \bar{r}_u^t - m, \bar{r}_{\rho(u)}^t - n) - \right. \\ & \left. - \sum_t \sum_p \delta(\bar{q}^t - k, \bar{r}_u^t - p, \bar{r}_{\rho(u)}^t - n) \hat{\epsilon}_{u,mn}^{(j)k} \right\}. \end{aligned} \tag{20}$$

The first sum in brackets counts how many times the most likely state in the node is m given that the most likely state in its parent node is n and the state

in the HMM is most likely to be k. For the double sum, note that $\hat{\epsilon}_{u,mn}^{(i)k}$ is a common factor and the sum actually counts all the frames when the most likely state in the parent of the given node is n and the most likely state in the external HMM is that related to the corresponding HMT, k in this case. Restoration of the original parameters is straightforward from the definition of $\tilde{\epsilon}_{u,mn}^{(j)k}$.

Finally, following identical procedures we find the updating formulas for the transformed state transition probabilities $\tilde{a}_{sj}^{(j)}$ given by:

$$\tilde{a}_{sj}^{(i)} \longleftarrow \tilde{a}_{sj}^{(i)} - \alpha_\tau \zeta \left\{ \sum_{t=1}^{T} \delta(\bar{q}_{t-1} - s, \bar{q}_t - j) - \sum_{t=1}^{T} \delta(\bar{q}_{t-1} - s) \hat{a}_{sj}^{(i)} \right\} . \tag{21}$$

Once again, we can interprete the sumations in the above formula as counters acting on the sequence of most likely states in the external HMM, as given by Viterbi decoding. Original parameters $a_{sj}^{(j)}(\tau + 1)$ are easily restored using the definition of $\tilde{a}_{sj}^{(j)}$.

5. Experimental results

In order to assess the proposed training method, we carry out automatic speech recognition tests using phonemes from the TIMIT database [25]. This is a well known corpus in the field and it has already been used in previous works dealing with similar schemes [14, 40]. In particular, we use samples of phonemes /b/, /d/, /eh/, /ih/ and /jh/. The voiced stops /b/ and /d/ have a very similar articulation and different phonetic variants according to the context. Vowels /eh/ and /ih/ were selected because their formants are very close. Thus, these pairs of phonemes are very confusable. The affricate phoneme /jh/ was added as representative of the voiceless group to complete the set. It must be remarked that this signals are not spoken isolatedly but extracted from continuous speech. As a result, there is a large variability in both acoustic features and duration in the dataset. All of these contribute to a very demanding task for a classifier.

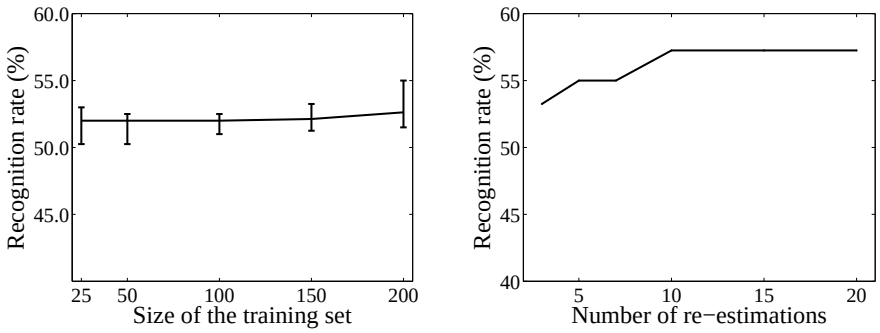
As a measure of performance, we compare recognition rates achieved with the proposed method against those for the same models trained only using the

EM algorithm. In all the experiments we model each phoneme with a left-to-right hidden Markov model with three states ($N_Q = 3$). The observation density for each state is given by an HMT with two states per node. This is the standard setting for the state space in most HMT applications [1]. The sequence analysis is performed on a short-term basis using Hamming windows 256-samples long, with 50% overlap between consecutive frames. On each frame, a full dyadic discrete wavelet decomposition is carried out using Daubechies wavelets with four vanishing moments [14, 42].

In a first set of experiments, we show numerically that the recognition rate achieved with the EM algorithm attains an upper bound for the given models and dataset. This bound is shown not to be surpassed neither increasing the number of reestimations of the algorithm nor enlarging the training set. We next carry out a two-phoneme recognition task using the approach developed in Section 4. The re-estimation formulas are reduced to much simpler expressions in this case, allowing to get further insight into the discriminative training process. It also serves us to compare the misclassification functions proposed in Section 4.2. Finally, we carry out a multiclass speech recognition experiment to assess the error rate reduction after adding a discriminative stage to the training process.

5.1. Limits on performance for ML estimators

Discriminative training methods usually use maximum-likelihood estimates computed via the EM algorithm as initial values for model parameters [36, 38]. Thus, it is fair to ask if better performance could be achieved just using more training sequences in the pure ML approach or increasing the number of re-estimations in the EM algorithm, without adding a discriminative stage. To answer this question empirically for our data and our particular model, we first perform a two-phoneme recognition task using models trained with the EM algorithm proposed in [14, 40]. We ran the experiment using training sets of increasing sizes, from 25 sequences to 200. Each training set was picked at random from the whole training partition of the dataset. A separate testing set



(a) (b)

Figure 5: Recognition rates for EM training. a) Increasing the size of the training set. Shown results are the median over ten runs for each tested condition. Error-bars are given by the first and third quartiles of the obtained scores. b) Increasing the number of reestimations. The {/b/,/d/} pair was used in both experiments.

with 200 sequences remained fixed for all trials. Each tested condition was run ten times and the number of re-estimations used for the EM algorithm was fixed at 6 in all of them. Obtained results for the {/b/,/d/} pair are given in Figure 5.a). It is clear from the figure that increasing the number of training samples does not lead to a significant improvement in the recognition rate when only the EM algorithm is used for training. In fact, analysis of results shows that the p-value for the {/b/,/d/} pair is 0.4476, which is far from the critical value to reject the null hypothesis of all means being statistically the same. Similar comments apply for the {/eh/,/ih/} pair.

On the other hand, the effect of fixing the size of the training set and increasing the number of re-estimations used in the EM algorithm is shown in Figure 5.b). Given values correspond to training sets with 200 sequences. It can be seen that recognition rates remain fairly the same with the increase in the number of re-estimations. For the {/b/,/d/} pair and the specific set of sequences used in the experiment, there is a slowly improvement in performance up to ten

re-estimations. Beyond that there is no benefit in adding re-estimations steps in the EM algorithm. For the {/eh/,/ih/} pair of phonemes there is a little improvement up to five re-estimations but no further improvement is seen either adding more re-estimations.

Observed results in this experiment reproduce a typical scenario when working with "real" data. Always the proposed model it is obviously not the true model for the data in that case. Increasing the training set or adding re-estimations to the EM algorithm can only contribute to find better estimates for the parameters in those models. If models were the true ones, this would help for classification. But as models do not give the true distribution of data, we cannot expect this to translate into better discrimination. Note that this is not a statement on the goodness of fit of the model itself. For complex real data (like speech, in this case), hardly any model we propose would suffer from this. Here is when discriminative training becomes so important.

5.2. MCE training for two-class phoneme recognition

In order to get some insight into the learning process, we first consider a classification task comprising only two phonemes. In this case, for a training sequence $\mathbf{W} \in \Omega_1$, the misclassification function SMF reduces to

$$\bar{d}_1(\mathbf{W};\Theta) = g_1(\mathbf{W};\Theta) - g_2(\mathbf{W};\Theta) \ .$$

Aside from the change in sign to account for the different definition of the discriminant functions we made in (8), this is the same as the frequently used function (10) for a binary classification problem [22]. When the classifier decision is right, $g_1(\mathbf{W};\Theta) < g_2(\mathbf{W};\Theta)$ and the misclassification function takes a negative value. As this decision is stronger, $\bar{d}_1(\mathbf{W};\Theta)$ becomes more negative and the resulting loss (2) goes to zero. We then see from the updating formulas in Section 4.3 that no updating is performed in such a case. So, the algorithm preserves model parameters that do well when classifying the current training signal. Furthermore, for strongly confused patterns $\bar{d}_1(\mathbf{W};\Theta)$ becomes a large positive value and no update is introduced either.

On the other hand, the missclassification function nSMF reduces to

$$d_1(\mathbf{W};\Theta) = 1 - \frac{g_2(\mathbf{W};\Theta)}{g_1(\mathbf{W};\Theta)} \ .$$

When the classifier decision is right, it behaves closely to $\bar{d}_1(\mathbf{W};\Theta)$. Nevertheless, if the current training sequence is strongly misclassified, $d_1(\mathbf{W};\Theta)$ will tend to 1. Unlike the previous case, parameters will be updated unless γ is too large. Therefore, this definition of the misclassification function adds a corrective feature to the learning process. In both cases, parameter update takes place when models are confusable and it is the strongest when the current training sequence is equally likely for both of them. With the second definition, however, we can also expect an updating step even for strongly misclassified patterns.

We can get an idea of the strength of the updating steps looking at the distribution of $\gamma\ell_i(1-\ell_i)$. For a given pattern, this factor scales the gradients in the re-estimation formulas according to how confusable the pattern is for the classifier, as told by the misclassification function. Figure 6 compares the distribution of this factor at the beginning of the iterative process, obtained for the same training set but choosing a different training method in each case. Figure 6.a) corresponds to standard MCE training for HMMs with Gaussian mixtures as observation densities on a cepstral-based feature space. Figure 6.b) comes from a classifier based on HMM-HMTs, using the misclassification function SMF to derive the MCE criterion; and Figure 6.c) comes from a classifier based on HMM-HMTs, but using nSMF as the misclassification function. In these later histograms, the bin that includes the value $\gamma\ell_i(1-\ell_i) = 0$ was removed to keep figures at a similar scale. It is interesting to see that despite of (10) and SMF sharing the same misclassification function for a binary problem like this, it is the criterion based on the misclassification function nSMF which generates the distribution of factors more similar to the standard case shown in plot a) when using the HMM-HMT. This finding remains valid for a wide range of useful values of γ chosen to adapt the sigmoid to each case. Therefore, changing the feature space used to represent the data can induce important modifications in the way the updating process is driven by a given approximation of the loss.

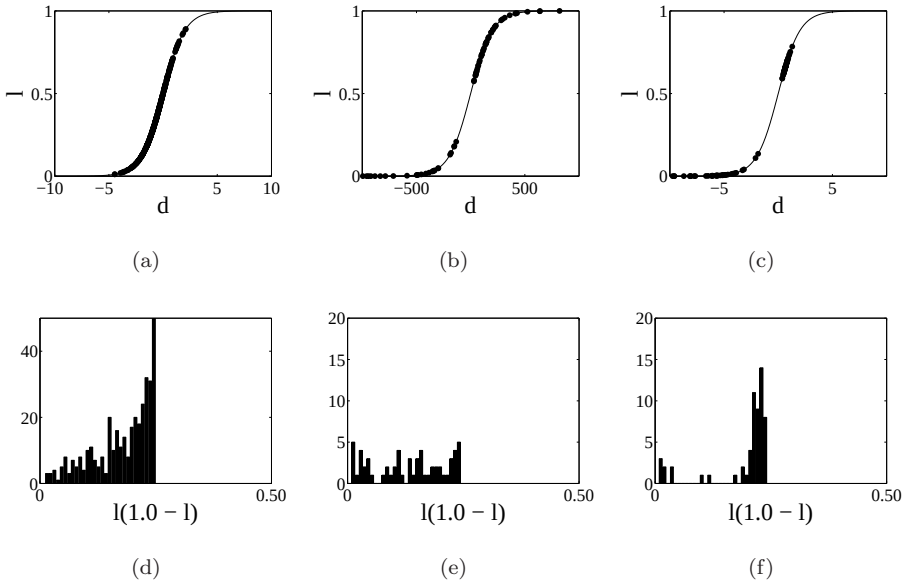
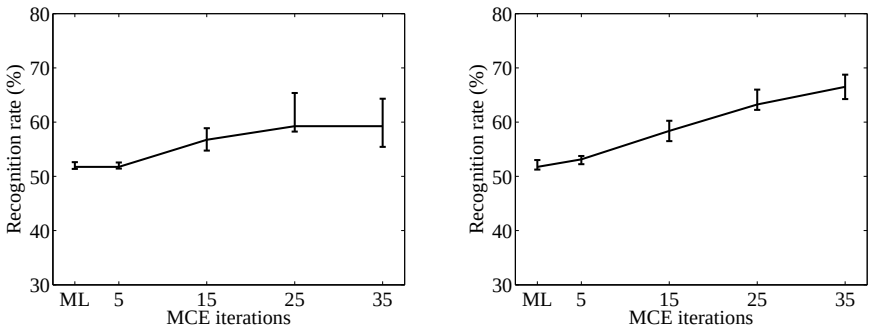


Figure 6: Distribution of the loss and the factor $\gamma\ell_i(1-\ell_i)$ at the beginning of different settings of the MCE training. Upper figures show the location of the loss for each sequence in the training set, while figures at the bottom show the resulting histogram for the factor $\gamma\ell_i(1-\ell_i)$. a) and d) using cepstral features and Gaussian mixture-HMMs along with a standard misclassification function as in (10); b) and d) using the HMM-HMT and the misclassification function SMF; c) and e) using the HMM-HMT and the misclassification function nSMF.

To compare the performance achieved by the proposed misclassification functions, we carried out numerical experiments with phonemes {/b/,/d/} and {/eh/,/ih/}, which are the most confused pairs in the set. Two hundred sequences from each class were used for training and another set of two hundred sequences from each class were used for testing. Five re-estimation steps were used in the EM algorithm, along with Viterbi flat start [41]. Parameters for the MCE learning stage were set following informal tests on a validation test, aimed to find the values that give better performance for each pair of phonemes and for each choice of misclassification function. When using SMF we set $\alpha_0 = 2.5$



(a) (b)

Figure 7: Recognition rates for phonemes /b/ and /d/: a) using SMF; b) using nSMF. Shown results are the median over ten runs for each tested condition. Error-bars are given by the first and third quartiles of the obtained scores.

and $\gamma = 0.01$, while we set $\alpha_0 = 0.5$ and $\gamma = 1$ for the algorithm derived using nSMF. In all cases, the learning rate was decreased from $\alpha_\tau = \alpha_0$ at the beginning of the run to $\alpha_\tau = 0$ at its end. The number of iterations of the MCE algorithm through the whole training set was varied as $5, 15, 25$ and 35. Ten runs were performed for each tested condition, varying the training set in each one but keeping fixed the set for testing.

Obtained results for each pair of phonemes and each choice of the misclassification function are shown in Figure 7 and Figure 8. Figure 7 shows the achieved recognition rates for the pair {/b/,/d/}. Performance for zero iterations of the MCE algorithm refers to the case when the classifier is trained using ML estimation and serves as the baseline for comparison. It can be seen that the scores using discriminative steps are significantly higher than the baseline with both MCE criteria for all tested conditions with more than five iterations. For five MCE iterations there is no significant improvement on the average. Figure also shows that the training method using the misclassification function nSMF outperforms that based on SMF. With 35 iterations of the algorithm, the former achieves an average reduction of about 30% in the error rate, whereas the later

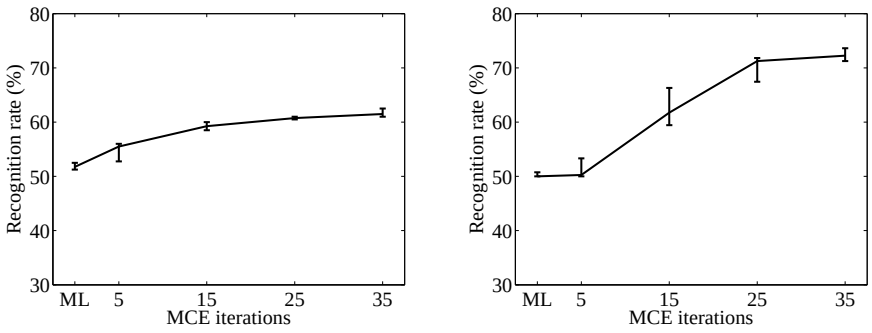


Figure 8: Recognition rates for phonemes /eh/ and /ih/: a) using the misclassification function SMF; b) using nSMF. Shown results are the median over ten runs for each tested condition. Error-bars are given by the first and third quartiles of the obtained scores.

does a 14%. In addition, there seems to be a trend to continue rising the recognition rate in Figure 7.b), while in 7.a) improvements appear to have reached a bound. Furthermore, the variance of the obtained scores remain very similar as they go better for the method using the misclassification function nSMF, while it increases significantly for the method using SMF.

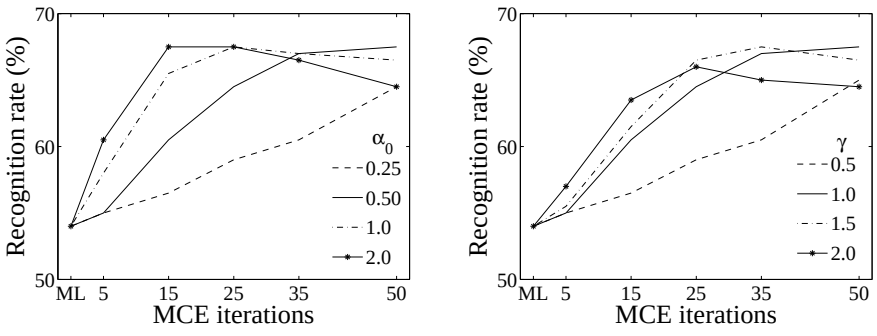
The difference in performance achieved with a different choice of the misclassification function is stressed in the results for phonemes {/eh/,/ih/} shown in Figure 8. Scores obtained here with the method based on nSMF are markedly better than those achieved using SMF. For the former the average improvement in the error rate is around 45%, whereas for the latter it is about 20%. A possible explanation of these results relies on the wide dispersion of discriminant function values. As SMF is based just on a difference between these values, it also has a large variability that makes it very difficult to choose a suitable sigmoid to capture many confusable samples to drive the competitive update without picking too much of them. The selected value for γ becomes conservative and then only a small subset of confusable samples are used to trigger the

updates, which results in a poorer performance. It must be noticed that this effect is expected to be emphasized as the duration of sequences increases, so that is natural to have better results for the shorter samples from {/b/,/d/}. On the other hand, the misclassification function nSMF introduces a scaling that avoids it to have so much variation in its values, which makes it easier to find a suitable sigmoid to drive the selection of confusable patterns.

5.3. Sensitivity to parameters of the algorithm

It is interesting to see the effect on the recognition rate when changing the parameters of the MCE/GPD algorithm. Consider the problem of classifying phonemes {/eh/,/ih/}. We first carried out a simple experiment setting $\eta = 4$ and $\gamma = 1$ as in previous tests, and changed α_0 to take values {0.25, 0.50, 1.0, 2.0}. Obtained results are shown in Figure 9.a). It can be seen that for this dataset recognition rates attain a bound at 67.5% for all conditions, but they differ in the speed they do it with. The smaller learning rate shows the lowest increase in recognition rate when increasing the number of iterations of the learning algorithm. Increasing α_0 speeds up the process, but it can be seen also that it can lead to overfitting. This situation is common to all gradient-based techniques as the one proposed here. The optimal value of α_0 depends on the data and the size of the training sample. Some rough guidelines to choose this parameter are stated in [34], taking into account the variability of the sample.

A similar effect can be seen in Figure 9.b), but varying γ and letting α_0 and η fixed. Nevertheless, the reason is quite different. Parameter γ determines the rate of change of the loss aproximation. For small values of γ, the signoid grows slowly from $\ell = 0$ to $\ell = 1$ and much of the training samples result in values of the misclassification function that fall in the raising segment of the sigmoid. In this case, even well classified sequences trigger strong updates. As γ becomes large, the raising segment of the sigmoid gets sharper and less cases fall in this region. Thus, well classified observations introduce a much weaker change on the parameters. At the same time, when nSMF is used as the misclassification function, small values of γ make misclassified cases fall in a narrow segment



(a) (b)

Figure 9: Sensitivity of recognition rate to changes on the parameters of the MCE/GPD algorithm. a) Varying α_0, with γ and η fixed. b) Varying γ, with α_0 and η fixed.

of the sigmoid, as seen in Figure 10. They give rise to updates with similar strength regardless the confusability of the training sequence. As γ becomes larger, misclassified cases occupy a broader region of the sigmoid, triggering updates that depend more on confusability.

5.4. Multiclass phoneme recognition

To further assess the proposed discriminative training method for the HMM-HMT model, a new speech recognition task including the whole set of phonemes was carried out. In this experiment, only the MCE approach based on the misclassification function nSMF was taken into account, as consistently better results were found for this choice in the previous task. Ten training sets picked at random were considered and a replicate of the experiment was run for each of them. The testing set remained fixed for all runs. Both the training sets and the testing set were build randomly taking 200 sequences from each class. The same learning rate was used for all the parameters in the models. The initial rate α_0 was chosen to be the largest value that gave a monotonic improvement in recognition rate as a function of the number of iterations of the MCE algorithm,

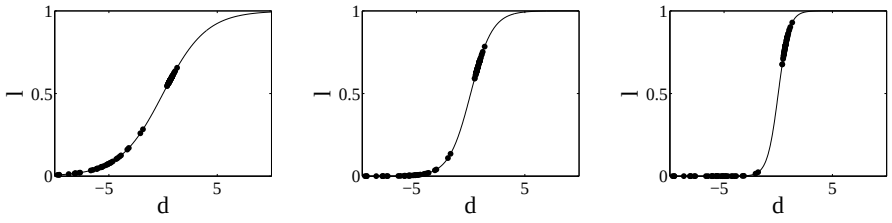


Figure 10: Location of the training sequences on the loss function for different values of parameter γ, using nSMF: a) $\gamma = 0.5$, b) $\gamma = 1$; and c) $\gamma = 2$.

when using a separate set of sequences both for training and testing. This was checked in preliminary runs. During the experiments, this learning rate was linearly decreased from $\alpha_\tau = \alpha_0$ at the first iteration to $\alpha_\tau = 0$ at the end of the training process.

Obtained results are shown in Figure 11. A monotonic improvement in the error rate is achieved as more iterations over the whole training set are added to the discriminative training process. After 35 iterations, the average error rate reduction is about 18%. Most of the improvement, however, occurs up to 25 iterations of the MCE algorithm, reducing the error rate around a 17.25% at this level. The variance in the obtained rates remains fairly the same with the increased number of iterations. Analysis of individual runs reveals that for some training sets performance degrades with the first iterations of the algorithm and then starts to improve as more iterations are carried out. Furthermore, three of the ten runs show that the achieved score starts to decrease slowly at 35 iterations, suggesting that overfitting could be taking place after this point.

This difficult classification task show a consistent improvement in recognition rate using discriminative parameter estimation for the HMM-HMT model.

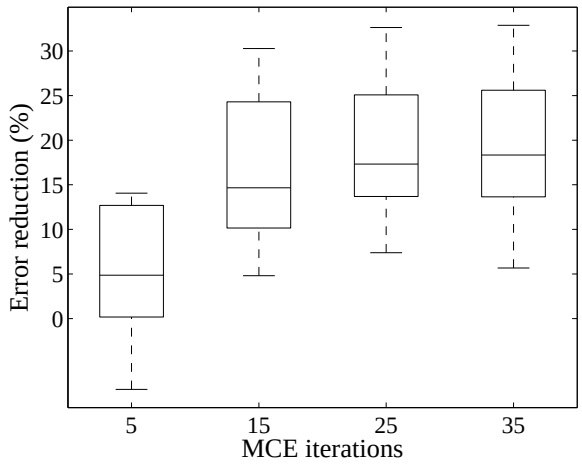


Figure 11: Error rate improvement over standard ML training using the proposed MCE approach to train the classifier for the set of five phonemes. The misclassification function nSMF was used in this experiment. Initial recognition rates using maximum likelihood estimates are around 37% for the considered phoneme set.

6. Conclusions

This paper introduces a new method for discriminative training of hidden Markov models whose observations are sequences in the wavelet domain. The algorithm is based on the MCE/GPD approach and it allows for training fully non-tied HMM-HMT models. This observation model and feature space required special considerations. It was shown that standard procedures were numerically unfeasible in this scenario, and alternative choices were needed to simulate the classifier decision when the MCE criterion was derived. Assessment of proposed misclassification functions in a simple phoneme recognition task showed that comparing the order of magnitude of the log-likelihoods for competing models was more appealing to this context than simple comparison of their value. This important modification results in a stronger penalty for mis-

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
D. R. Tomassi, D. H. Milone & L. Forzani; "Minimum Classification Error Learning for Sequential Data in the Wavelet Domain"
Pattern Recognition, Vol. 43, No. 12, pp. 3998-4010, 2010.

classified patterns, giving rise to a corrective characteristic that works well in this context. Speech recognition experiments show that the proposed method achieves consistent improvements on recognition rates over training with the standard EM algorithm only. The average error rate reduction for this task was found to be around 18% using 35 iterations of the algorithm.

Obtained results are promising and there is plenty of room for further improvements. In particular, only plain HMM-HMTs were used in this work. Strategies to account for shift invariance has shown to be effective when working with HMTs and could also be introduced in this conext. Future works will address this issue along with more efficient optimization approaches in order to speed up convergence.

AppendixA. Updating of Gaussian observations

In this section, we review the main steps for deriving the updating formulas for the Gaussian distributions in the observation model for each HMT of the composite HMM-HMT. Let us consider the training formulas for the Gaussian means first. We begin noting that the discriminant functions read:

$$\begin{aligned} g_j(\mathbf{W};\Theta) &= \left|\log\left(\max_{\mathbf{q},\mathbf{R}}\left\{\mathcal{L}_{\vartheta_j}(\mathbf{W},\mathbf{q},\mathbf{R})\right\}\right)\right| \\ &= -\log\left(\max_{\mathbf{q},\mathbf{R}}\left\{\prod_{t=1}^{T} a_{q^{t-1}q^t}\prod_{\forall u}\epsilon^{q^t}_{u,r^t_u r^t_{\rho(u)}} f^{q^t}_{u,r^t_u}(w^t_u)\right\}\right) \\ &= -\sum_t \log a_{\bar{q}^{t-1}\bar{q}^t} - \sum_t\sum_{\forall u}\log\epsilon^{\bar{q}^t}_{u,\bar{r}^t_u\bar{r}^t_{\rho(u)}} - \sum_t\sum_{\forall u}\log f^{\bar{q}^t}_{u,\bar{r}^t_u}(w^t_u)\ , \end{aligned}$$

where, $\bar{q}^t$ and $\bar{r}^t$ refer to states in the external HMM and the corresponding HMT model, respectively, that achieve the maximum joint likelihood. To find (18), we know from (20) that we need

$$\begin{aligned} \frac{\partial \ell_i(\mathbf{W};\Theta)}{\partial \tilde{\mu}^{(j)k}_{u,m}} &= \frac{\mathrm{d}\ell_i(\mathbf{W};\Theta)}{\mathrm{d}d_i(\mathbf{W};\Theta)}\frac{\partial d_i(\mathbf{W};\Theta)}{\partial g_i(\mathbf{W};\Theta)}\frac{\partial g_i(\mathbf{W};\Theta)}{\partial \tilde{\mu}^{(j)k}_{u,m}} \\ &= -\zeta\frac{\partial g_i(\mathbf{W};\Theta)}{\partial \tilde{\mu}^{(j)k}_{u,m}} \\ &= -\zeta\frac{\partial \sum_t\sum_{\forall u}\log f^{\bar{q}^t}_{u,\bar{r}^t_u}(w^t_u)}{\partial \tilde{\mu}^{(j)k}_{u,m}}\ . \end{aligned}$$

In the expression above, we used ζ defined in Section 4.3. As observations in a node depends only on the state of that node, we have

$$\frac{\partial \ell_i(\mathbf{W};\Theta)}{\partial \tilde{\mu}_{u,m}^{(j)k}} = -\zeta \frac{\partial \sum_t \log f_{u,\bar{r}_u^t}^{\bar{q}^t}(w_u^t)}{\partial \tilde{\mu}_{u,m}^{(j)k}}.$$

As the sum takes into account only the most likely states in the node of the HMT related to the most likely state of the HMM in a given frame, we write

$$\begin{aligned}\frac{\partial \ell_i(\mathbf{W};\Theta)}{\partial \tilde{\mu}_{u,m}^{(j)k}} &= -\zeta \sum_t \delta(\bar{q}^t - k, \bar{r}_u^t - m) \frac{\partial \log f_{u,\bar{r}_u^t}^{\bar{q}^t}(w_u^t)}{\partial \tilde{\mu}_{u,m}^{(j)k}}. \\ &= -\zeta \sum_t \delta(\bar{q}^t - k, \bar{r}_u^t - m) \frac{\partial \mu_{u,m}^{(j)k}}{\partial \tilde{\mu}_{u,m}^{(j)k}} \frac{\partial \log f_{u,\bar{r}_u^t}^{\bar{q}^t}(w_u^t)}{\partial \mu_{u,m}^{(j)k}}.\end{aligned}$$

Noting that $\partial \mu_{u,m}^{(j)k} / \partial \tilde{\mu}_{u,m}^{(j)k} = \sigma_{u,m}^{(j)k}$ and that we are using an univariate Gaussian distribution for $f_{u,\bar{r}_u^t}^{\bar{q}^t}(w_u^t)$, we get (18).

The steps to derive the updating formulas for the Gaussian variances are completely analogous.

AppendixB. Updating of transition probabilities

The procedure applied above also works well for transition probabilities, both in each HMT and in the external HMM of the whole HMM-HMT. Let us consider the estimation of the transition probabilities in the internal HMT. Reasoning as above, we just need

$$\frac{\partial \ell_i(\mathbf{W};\Theta)}{\partial \tilde{\epsilon}_{u,mn}^{(i)k}} = -\zeta \frac{\sum_t \log \epsilon_{u,\bar{r}_u^t \bar{r}_{\rho(u)}^t}^{\bar{q}^t}}{\partial \tilde{\epsilon}_{u,mn}^{(i)k}}.$$

Remembering of the transformation for this transition probabilities and proceeding as before to account for the most likely states in each frame, we get

$$\begin{aligned}\frac{\partial \ell_i(\mathbf{W};\Theta)}{\partial \tilde{\epsilon}_{u,mn}^{(i)k}} &= -\zeta \sum_t \sum_p \frac{\partial \epsilon_{u,pn}^{(i)k}}{\partial \tilde{\epsilon}_{u,mn}^{(i)k}} \frac{\partial \log \epsilon_{u,\bar{r}_u^t \bar{r}_{\rho(u)}^t}^{\bar{q}^t}}{\partial \epsilon_{u,pn}^{(i)k}} \\ &= -\zeta \sum_t \sum_p \delta(\bar{q}^t - k, \bar{r}_u^t - p, \bar{r}_{\rho(u)}^t - n) \frac{\partial \epsilon_{u,pn}^{(i)k}}{\partial \tilde{\epsilon}_{u,mn}^{(i)k}} \frac{\partial \log \epsilon_{u,pn}^{k}}{\partial \epsilon_{u,pn}^{(i)k}}.\end{aligned}$$

We now see that for $p \neq m$, we have $\partial \epsilon_{u,pn}^{(i)k} / \partial \tilde{\epsilon}_{u,mn}^{(i)k} = -\epsilon_{u,pn}^{(i)k} \epsilon_{u,mn}^{(i)k}$ and for $p = m$ we have $\partial \epsilon_{u,pn}^{(i)k} / \partial \tilde{\epsilon}_{u,mn}^{(i)k} = \epsilon_{u,mn}^{(i)k}(1 - \epsilon_{u,mn}^{(i)k})$. Replacing these results in the formula for the gradient and reordering, we get (20). An analogous procedure applies to derive the updating formulas for transition probabilities in the external HMM.

Acknowledgements

This work was carried out with financial support from National University of Litoral (CAI+D-012-72), National Agency for the Promotion of Science and Technology (PAE-PICT-2007-00052), Lòreal, and the National Scientific and Technical Research Council (CONICET).

References

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

[2] 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. doi:10.1109/ICASSP.2008.4518815.

[3] S. Graja, J.-M. Boucher, Hidden Markov tree model applied to ECG delineation, IEEE Transactions on Instrumentation and Measurement 54 (6) (2005) 2163–2168. doi:10.1109/TIM.2005.858568.

[4] 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. doi:10.1016/j.sigpro.2007.04.014.

[5] 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. doi:10.1109/ICIP.2008.4712259.

[6] 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. doi:10.1109/ICIP.2008.4711759.

[7] 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. doi:10.1109/ICASSP.2008.4517724.

[8] 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. doi:10.1109/IIH-MSP.2007.1.

[9] E. Mor, M. Aladjem, Boundary refinements for wavelet-domain multiscale texture segmentation, Image and Vision Computing 23 (13) (2005) 1150 – 1158. doi:10.1016/j.imavis.2005.07.011.

[10] V. R. Rallabandi, V. S. Rallabandi, Rotation-invariant texture retrieval using wavelet-based hidden Markov trees, Signal Processing 88 (10) (2008) 2593 – 2598. doi:10.1016/j.sigpro.2008.04.019.

[11] R. Ferrari, H. Zhang, C. Kube, Real-time detection of steam in video images, Pattern Recognition 40 (3) (2007) 1148 – 1159. doi:DOI: 10.1016/j.patcog.2006.07.007.

[12] 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) –. doi:10.1016/j.imavis.2008.12.005.

[13] 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. doi:10.1016/j.patcog.2007.08.017.

[14] D. H. Milone, L. E. D. Persia, M. E. Torres, Denoising and recognition using hidden Markov models with observation distributions modeled by hidden Markov trees, Pattern Recognition, in press-doi:10.1016/j.patcog.2009.11.010.

[15] 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.

[16] D. Milone, L. D. Persia, D. Tomassi, Signal denoising with hidden Markov models using hidden Markov trees as observation densities, in: Proc. of the IEEE MLSP08 Workshop, 2008, pp. 374–379.

[17] X. He, L. Deng, W. Chou, Discriminative learning in sequential pattern recognition: a unifying review for optimization-based speech recognition, IEEE Signal Processing Magazine 25 (2008) 14–36.

[18] L. Bahl, P. Brown, P. D. Souza, R. Mercer, Maximum mutual information estimation of HMM parameters for speech recognition, in: Proc. of the Int. Conf. on Audio, Speech, and Signal processing (ICASSP86), 1986, pp. 49–52.

[19] B.-H. Juang, W. Chou, C.-H. Lee, Minimum classification error rate methods for speech recognition, IEEE Transactions on Speech and Audio Processing 5 (1997) 257–265.

[20] D. Povey, Discriminative training for large vocabulary speech recognition, Ph.D. thesis, Cambridge University, Cambridge, UK (2004).

[21] S. Katagiri, B.-H. Juang, C. Lee, Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method, Proceedings of the IEEE 86 (1998) 2345–2373.

[22] M. Afify, X. Li, H. Jiang, Statistical analysis of minimum classification error learning for gaussian and hidden Markov model classifiers, IEEE Transactions on Audio, Speech, and Language Processing 15 (2007) 2405–2417.

[23] E. McDermott, S. Katagiri, A derivation of minimum classification error from the theoretical classification risk using Parzen estimation, Computers, Speech and Language 18 (2004) 102–122.

[24] X. He, L. Deng, A new look at discriminative training for hidden Markov models, Pattern Recognition Letters 28 (2007) 1285–1294.

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

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

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

[28] D.-Y. Po, M. Do, Directional multiscale modeling of images using the contourlet transform, IEEE Transactions on Image Processing 15 (2) (2006) 651–664.

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

[30] J. Durand, P. Goncalves, Y. Guédon, Computational methods for hidden Markov tree models - an application to wavelet trees, IEEE Transactions on Signal Processing 52 (2004) 2551–2560.

[31] Y. Bengio, Markovian models for sequential data, Neural Computing Surveys 2 (1999) 129–162.

[32] S. Bengio, H. Bourlard, K. Weber, An EM algorithm for HMMs with emission distributions represented by HMMs, Technical Report IDIAP-

RR 11, Institut Dalle Molle d'Intelligence Artificielle Perceptive, Martigny, Switzerland (2000).

[33] 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.

[34] E. McDermott, T. Hazen, J. L. Roux, A. Nakamura, S. Katagiri, Discriminative training for large-vocabulary speech recognition using minimum classification error, IEEE Transactions on Audio, Speech, and Language Processing 15 (2007) 203–223.

[35] P. Woodland, D. Povey, Large scale discriminative training of hidden Markov models for speech recognition, Computer, Speech and Language 16 (2002) 25–47.

[36] H. Jiang, Discriminative training of HMMs for automatic speech recognition: A survey, Computer, Speech and Language, in press-doi:10.1016/j.csl.2009.08.002.

[37] R. Duda, P. Hart, D. Stork, Pattern Classification, Second Edition, Wiley, 2000.

[38] W. Chou, Minimum classification error rate (MCE) approach in pattern recognition, in: W. Chou, B. Juang (Eds.), Pattern Recognition in Speech and Language Processing, CRC Press, 2003, pp. 1–49.

[39] W. Chou, B.-H. Huang, C.-H. Lee, Segmental GPD training for HMM based speech recognition, in: Proc. of the Int. Conf. on Audio, Speech, and Signal processing (ICASSP92), Vol. 1, 1992, pp. 473–476.

[40] D. H. Milone, L. E. D. Persia, An EM algorithm to learn sequences in the wavelet domain, Lecture Notes in Computer Science 4827 (2007) 518–528.

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

[42] S. Mallat, A Wavelet Tour of Signal Processing. Second Edition, Academic Press, 1999.