Fast Approximation of Kullback-Leibler Distance for Dependence Trees and Hidden Markov Models

Minh N. Do

Abstract— **We present a fast algorithm to approximate the Kullback-Leibler distance (KLD) between two dependence tree models. The algorithm uses the "upward" (or "forward") procedure to compute an upper bound for the KLD. For hidden Markov models, this algorithm is reduced to a simple expression. Numerical experiments show that for a similar accuracy, the proposed algorithm offers a saving of hundreds of times in computational complexity compared to the commonly used Monte-Carlo method. This makes the proposed algorithm important for real-time applications, like image retrieval.**

Keywords— **Kullback-Leibler distance, dependence tree models, hidden Markov models**

I. Introduction

Hidden Markov models (HMM) and their generalized versions on dependence tree structures [1] have become powerful tools in speech recognition [2] and signal processing [3]. Their successes come from their effectiveness in modeling large classes of natural measurements using a small set of parameters. Furthermore, there are fast algorithms to evaluate and train these models for given data sets.

In certain problems, we would like to measure the distance between two statistical models. For example, this distance can be used in evaluating the training algorithm or classifying the estimated models [4]. In an image retrieval application, where each texture pattern is efficiently represented by a wavelet-domain hidden Markov tree model [5], the search is performed by comparing the distances between the model of the query image and the model of each candidate image. The *Kullback-Leibler distance* (KLD) or the *relative entropy* arises in many contexts as an appropriate measurement of the distance between two models. The KLD between the two probability density functions f and $\tilde{f}$ is defined as [6]:

$$D(f\|\tilde{f}) = \int f \log \frac{f}{\tilde{f}}. \tag{1}$$

For dependence trees and hidden Markov models, the probability function is very complex, and practically it can be only computed via a recursive procedure – the "forward/backward" or "upward/downward" algorithms [2], [1]. Thus there is no simple closed form expression for the KLD for these models. Commonly, the Monte-Carlo method is used to numerically approximate the integral in (1). This is done by rewriting (1) as

$$D(f\|\tilde{f}) = E_f[\log f(X) - \log \tilde{f}(X)].$$

M. N. Do is with the Department of Electrical and Computer Engineering and the Beckman Institute, University of Illinois at Urbana-Champaign, Urbana IL 61801 (email: minhdo@uiuc.edu).

With this, one can randomly and independently generate a set of sample data $x_1, x_2, \ldots, x_N$ based on the model density $f(X)$ and then approximate the KLD by:

$$D(f\|\tilde{f}) \approx \frac{1}{N}\sum_{n=1}^{N}[\log f(x_n) - \log \tilde{f}(x_n)]. \tag{2}$$

Typically, for an accurate approximation of $D(f\|\tilde{f})$, N has to be large, which can be prohibitively expensive in certain applications. Furthermore, due to the "random" nature of the Monte-Carlo method, the approximations of the distance could vary in different computations.

In this paper, we propose a fast algorithm to approximate the KLD between two dependence tree models or two hidden Markov models. In fact, the algorithm computes an upper bound for the KLD. For general dependence trees, the proposed algorithm has computational complexity similar to computing *one* density in (2), and thus it is much faster compared with the Monte-Carlo method. For hidden Markov models – a special case of dependence trees, the algorithm is reduced to a simple expression.

II. KLD between Dependence Tree Models

Denote $D(\boldsymbol{w}\|\tilde{\boldsymbol{w}})$ as the KLD between two probability mass functions $\boldsymbol{w} = (w_1, \ldots, w_J)$ and $\tilde{\boldsymbol{w}} = (\tilde{w}_1, \ldots, \tilde{w}_J)$

$$D(\boldsymbol{w}\|\tilde{\boldsymbol{w}}) = \sum_{i=1}^{J} w_i \log \frac{w_i}{\tilde{w}_i}. \tag{3}$$

Our results are based on the following key lemma which was stated for mixture of Gaussians in [7].

Lemma 1: The KLD between two mixture densities $\sum_{i=1}^{J} w_i f_i$ and $\sum_{i=1}^{J} \tilde{w}_i \tilde{f}_i$ is upper bounded by

$$D\left(\sum_{i=1}^{J} w_i f_i \| \sum_{i=1}^{J} \tilde{w}_i \tilde{f}_i\right) \le D(\boldsymbol{w}\|\tilde{\boldsymbol{w}}) + \sum_{i=1}^{J} w_i D(f_i\|\tilde{f}_i), \tag{4}$$

with equality if and only if $\frac{w_i f_i}{\sum_i w_i f_i} = \frac{\tilde{w}_i \tilde{f}_i(x)}{\sum_i \tilde{w}_i \tilde{f}_i(x)}$, for all i.

Proof: Using the *log-sum* inequality [6] (p.29)

$$\begin{aligned} D\left(\sum_i w_i f_i \| \sum_i \tilde{w}_i \tilde{f}_i\right) &= \int \left(\sum_i w_i f_i\right) \log \frac{\sum_i w_i f_i}{\sum_i \tilde{w}_i \tilde{f}_i} \\ &\le \int \sum_i w_i f_i \log \frac{w_i f_i}{\tilde{w}_i \tilde{f}_i} \\ &= \sum_i w_i \log \frac{w_i}{\tilde{w}_i} + \sum_i w_i \int f_i \log \frac{f_i}{\tilde{f}_i} \\ &= D(\boldsymbol{w}\|\tilde{\boldsymbol{w}}) + \sum_i w_i D(f_i\|\tilde{f}_i), \end{aligned}$$

■

Consider a statistical dependence tree T, where at each node n in the tree there is a hidden state variable S_n and an observation variable O_n (Figure 1). Denote $\rho(n)$ to be the parent of the node n and $C(n)$ to be the set of children of the node n. Furthermore, denote T_n to be the subtree of all nodes with the root at n and O_{T_n} to be the set of all observation variables attached to these nodes. Node 1 is assigned to the root of T, and thus $T_1 = T$.

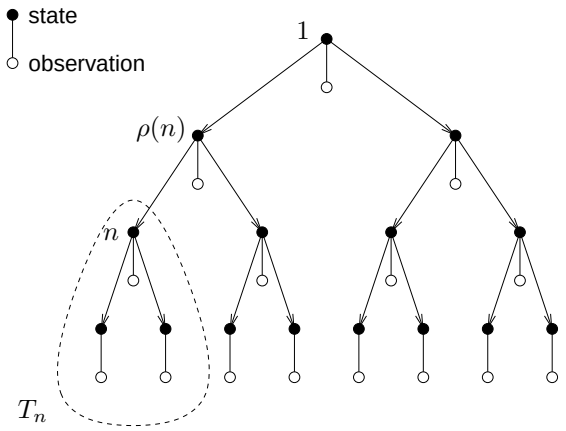

Fig. 1. A dependence tree model.

The state variables S_n have discrete value in the set $\{1, 2, \ldots, J\}$ and follow a Markov model, where the state transition probability is defined on the edges of T as

$$P(S_1 = i) = \pi_i,$$
$$P(S_n = j \mid S_{\rho(n)} = i) = a_{ij}^{(n)}. \quad (5)$$

The observation variables have emission probabilities conditioned only on the state of the same node

$$P(O_n = o \mid S_n = i) = b_i^{(n)}(o), \quad (6)$$

where $b_i^{(n)}(.)$ can be either a probability mass function (pmf) for discrete models or a parameterized probability density function (pdf), usually a mixture of Gaussians, for continuous models. Therefore, the set of parameters $\theta = \left\{\pi_i, a_{ij}^{(n)}, b_i^{(n)}(.)\right\}_{1\le i,j\le J,\ n\in T}$ completely specifies a dependence tree or hidden Markov tree model.

At each node n, we define $\beta_i^{(n)}$ to be the conditional probability function of the subtree observation data which has root at the node n given its state is i, that is

$$\beta_i^{(n)}(o_{T_n}) = P(O_{T_n} = o_{T_n} \mid S_n = i, \theta), \quad i = 1, \ldots, J. \quad (7)$$

For a leaf node n, we have

$$\beta_i^{(n)}(o_n) = b_i^{(n)}(o_n). \quad (8)$$

From the definition of the dependence tree model, if we fix the state $S_n = i$ of a node n, then the observation O_n and its subtrees O_{T_m} for each $m \in C(n)$ are independent (refer to Figure 1). This leads to the following key induction relation

$$\beta_i^{(n)}(o_{T_n}) = b_i^{(n)}(o_n) \prod_{m\in C(n)} \sum_{j=1}^{J} a_{ij}^{(m)} \beta_j^{(m)}(o_{T_m}) \quad (9)$$

This equation is the heart of the "forward" or "upward" algorithm [1] in which the probabilities β's are computed iteratively up the tree to the root where the probability of the whole observation tree is computed as

$$P(O_T = o_T|\theta) = \sum_{j=1}^{J} \pi_j \beta_j^{(1)}(o_T). \quad (10)$$

Based on this induction relation, we propose an efficient algorithm to approximate the KLD between two dependence tree models θ and $\tilde{\theta}$.

1. **Initialization:** At each leaf node n of T, using (8) we have

$$D(\beta_i^{(n)} \| \tilde{\beta}_i^{(n)}) = D(b_i^{(n)} \| \tilde{b}_i^{(n)}). \quad (11)$$

For discrete models, $D(b_i^{(n)} \| \tilde{b}_i^{(n)})$ can be computed directly as shown in (3) for the KLD between two pmf's. For continuous models, where $b_i^{(n)}$ and $\tilde{b}_i^{(n)}$ are mixtures of Gaussians, we can upper bound their KLD using Lemma 1 and the following closed form expression for the KLD between two d-dimensional Gaussians [7]:

$$D(\mathcal{N}(.;\boldsymbol{\mu},\mathsf{C}) \| \mathcal{N}(.;\tilde{\boldsymbol{\mu}},\tilde{\mathsf{C}})) = \frac{1}{2}[\log \frac{\det \tilde{\mathsf{C}}}{\det \mathsf{C}} - d + \mathrm{trace}(\tilde{\mathsf{C}}^{-1}\mathsf{C}) + (\boldsymbol{\mu}-\tilde{\boldsymbol{\mu}})^T \tilde{\mathsf{C}}^{-1}(\boldsymbol{\mu}-\tilde{\boldsymbol{\mu}})]. \quad (12)$$

2. **Induction:** Since given $S_n = i$, O_n and O_{T_m} are independent for all $m \in C(n)$, applying the chain rule for KLD [6] (p.23) to (9), we have

$$D(\beta_i^{(n)} \| \tilde{\beta}_i^{(n)}) = D(b_i^{(n)} \| \tilde{b}_i^{(n)}) + \sum_{m\in C(n)} D\left(\sum_{j=1}^{J} a_{ij}^{(m)} \beta_j^{(m)} \| \sum_{j=1}^{J} \tilde{a}_{ij}^{(m)} \tilde{\beta}_j^{(m)}\right).$$

Applying Lemma 1 to the last term in the above equation, we obtain

$$D(\beta_i^{(n)} \| \tilde{\beta}_i^{(n)}) \le D(b_i^{(n)} \| \tilde{b}_i^{(n)}) + \sum_{m\in C(n)} \left(D(\boldsymbol{a}_i^{(m)} \| \tilde{\boldsymbol{a}}_i^{(m)}) + \sum_{j=1}^{J} a_{ij}^{(m)} D(\beta_j^{(m)} \| \tilde{\beta}_j^{(m)}) \right). \quad (13)$$

Here we denote $\boldsymbol{a}_i^{(m)} = (a_{i1}^{(m)}, \ldots, a_{iJ}^{(m)})$, which is the pmf for the state of child note S_m given its parent state $S_n = i$.

3. **Termination:** Finally, applying Lemma 1 to (10)

$$D(\theta \| \tilde{\theta}) \le D(\boldsymbol{\pi} \| \tilde{\boldsymbol{\pi}}) + \sum_{j=1}^{J} \pi_j D(\beta_j^{(1)} \| \tilde{\beta}_j^{(1)}). \quad (14)$$

Tracing through the chain of inequalities, in effect, we have a fast algorithm to compute an upper bound for the KLD between two dependence tree models. The proposed algorithm has the same computational complexity as computing *one* density for a dependence tree model using the "upward" algorithm [1]. Thus comparing with the Monte-Carlo method (2), the proposed algorithm offers a saving of N times, where N is the number of randomly generated trees used in the Monte-Carlo method.

It can be proved[1] that the above algorithm is equivalent to applying Lemma 1 to the dependence tree models as they were expressed as mixture of densities

$$P(O_T = o_T|\theta) = \sum_{s_T} P(S_T = s_T) \left(\prod_{n \in T} b_{S_n}^{(n)}(o_n) \right), \quad (15)$$

where the sum is taken over all $J^{|T|}$ combinations of states s_T on the tree T ($|T|$ is the number of nodes on T). Such direct application of Lemma 1 to (15) is infeasible in practice due to the typically huge number of densities in (15). Nevertheless, this equivalence provides an interpretation of the result from the proposed algorithm. In particular, applying the Bayes' theorem to the equality condition of Lemma 1, we conclude that the resulting upper bound of the proposed algorithm is tight if and only if the *posteriori* state probabilities of two models are the same

$$P(S_T = s_T|o_T, \theta) = P(\tilde{S}_T = s_T|o_T, \tilde{\theta}), \quad \text{for all } s_T, o_T. \quad (16)$$

III. KLD between Hidden Markov Models

A special case of dependence tree models is the hidden Markov model (HMM) which was shown to be useful in many applications, especially speech recognition [2]. In an HMM, the dependence tree T becomes a chain; that is, except the last one, each node has exactly one child. Furthermore, all the nodes share the same statistics: $a_{ij}^{(n)} = a_{ij}$, $b_i^{(n)} = b_i$, for all n. Number the nodes in the chain in the obvious way: start from 1 and end at N for the last node.[2] The inductive relation (13) becomes

$$D(\beta_i^{(n)} \| \tilde{\beta}_i^{(n)}) \le D(b_i \| \tilde{b}_i) + D(\boldsymbol{a}_i \| \tilde{\boldsymbol{a}}_i) + \sum_{j=1}^{J} a_{ij} D(\beta_j^{(n+1)} \| \tilde{\beta}_j^{(n+1)}). \quad (17)$$

Denote $d_i = D(\boldsymbol{a}_i \| \tilde{\boldsymbol{a}}_i) + D(b_i \| \tilde{b}_i)$, $D_i^{(n)} = D(\beta_i^{(n)} \| \tilde{\beta}_i^{(n)})$, $\boldsymbol{d} = (d_1, \ldots, d_J)^T$, $\boldsymbol{D}^{(n)} = (D_1^{(n)}, \ldots, D_J^{(n)})^T$, and $\mathsf{A} = \{a_{ij}\}$ to be the state-transition probability matrix. Then (17) can be written in a compact form as

$$\boldsymbol{D}^{(n)} \le \boldsymbol{d} + \mathsf{A}\boldsymbol{D}^{(n+1)}. \quad (18)$$

[1]The proof is omitted due to lack of space.

[2]Using the notation for dependence trees defined previously, for HMM's the quantity $\beta_i^{(n)}$ becomes $P(O_{n:N} = o_{n:N}|S_n = i, \theta)$. This is slightly different with conventional notation in HMM literature, which defines $\beta_i^{(n)}$ to be $P(O_{n+1:N} = o_{n+1:N}|S_n = i, \theta)$.

The initialization step (11) becomes

$$\boldsymbol{D}^{(N)} = (D(b_1 \| \tilde{b}_1), \ldots, D(b_J \| \tilde{b}_J))^T \stackrel{\text{def}}{=} \boldsymbol{e}.$$

And the termination step (14) becomes

$$D(\theta \| \tilde{\theta}) \le D(\boldsymbol{\pi} \| \tilde{\boldsymbol{\pi}}) + \boldsymbol{\pi}^T \boldsymbol{D}^{(1)}.$$

By applying (18) iteratively, we obtain

$$D(\theta \| \tilde{\theta}) \le D(\boldsymbol{\pi} \| \tilde{\boldsymbol{\pi}}) + \boldsymbol{\pi}^T \left(\sum_{n=1}^{N-1} \mathsf{A}^{n-1} \boldsymbol{d} + \mathsf{A}^{N-1} \boldsymbol{e} \right). \quad (19)$$

Note that the KLD between two HMM's depends on the length N of the observation sequence. Therefore, typically the following *Kullback-Leibler divergence rate* (KLDR) between two HMM's is used:

$$\bar{D}(\theta \| \tilde{\theta}) = \lim_{N \to \infty} \frac{1}{N} D(\theta \| \tilde{\theta}). \quad (20)$$

If we assume that the model θ is stationary, that is there exists a stationary distribution vector $\boldsymbol{\nu}$ such that $\boldsymbol{\nu}^T \mathsf{A} = \boldsymbol{\nu}^T$ and

$$\lim_{n \to \infty} \boldsymbol{\pi}^T \mathsf{A}^n = \boldsymbol{\nu}^T,$$

then by substituting (19) into (20) and taking the limit as a Cesáro mean, we obtain

$$\bar{D}(\theta \| \tilde{\theta}) \le \boldsymbol{\nu}^T \boldsymbol{d} = \sum_{j=1}^{J} \nu_j \left(D(\boldsymbol{a}_j \| \tilde{\boldsymbol{a}}_j) + D(b_j \| \tilde{b}_j) \right). \quad (21)$$

The upper bound in (21) is a very simple expression that can be computed directly on the model parameters and requires about J^2 operations.

IV. Numerical Experiments

We use numerical experiments to evaluate the tightness and the computational saving of the proposed algorithm in comparison with the commonly used Monte-Carlo method for computing the KLD. Due to the random nature of the Monte-Carlo method, we run it for 1000 independent trials to estimate the variation of the Monte-Carlo restuls and the true KLD – chosen as the mean of these results.

First, we experiment with discrete HMM's (DHMM's), where the pmf's b_i is represented by a stochastic matrix B. Consider the following two DHMM's:

$$\begin{aligned} \boldsymbol{\pi} &= \begin{pmatrix} 0.5 & 0.5 \end{pmatrix} & \tilde{\boldsymbol{\pi}} &= \begin{pmatrix} 0.5 & 0.5 \end{pmatrix} \\ \mathsf{A} &= \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix} & \tilde{\mathsf{A}} &= \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix} \\ \mathsf{B} &= \begin{pmatrix} 0.1 & 0.3 & 0.6 \\ 0.2 & 0.1 & 0.7 \end{pmatrix} & \tilde{\mathsf{B}} &= \begin{pmatrix} 0.3 & 0.5 & 0.2 \\ 0.6 & 0.2 & 0.2 \end{pmatrix} \end{aligned} \quad (22)$$

Figure 2 shows the approximation results of the KLDR $\bar{D}(\theta \| \tilde{\theta})$ of the above two DHMM's using two different methods. The results from the Monte-Carlo method vary significantly unless the length N of the randomly generated

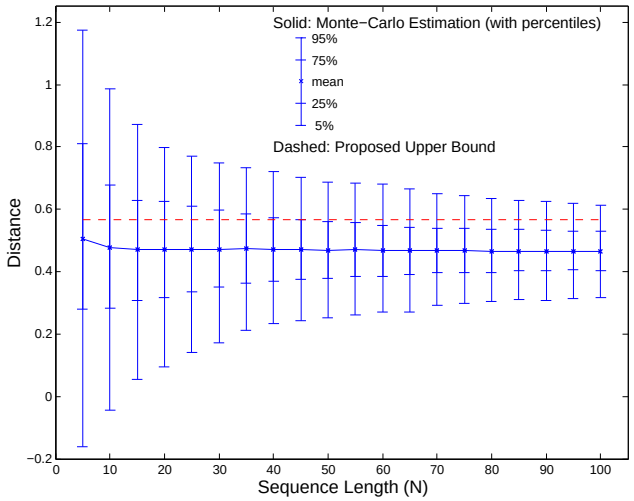

Fig. 2. Approximation results of the KLDR between two DHMM's in (22).

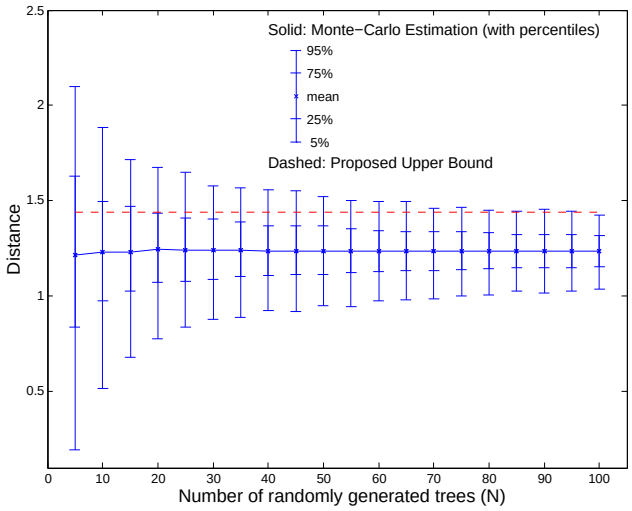

Fig. 3. Approximation results of the KLD between two wavelet-domain hidden Markov tree models in (23).

sequence is large. In this case, we see that the proposed algorithm has similar accuracy with the Monte-Carlo method that uses sequence of length $N \approx 100$; and thus it offers a saving of hundreds of times in computational complexity.

Next, we experiment with wavelet-domain hidden Markov tree models [3]. In these models, wavelet coefficients are organized into trees where edges link parent and child coefficients across scales. The marginal distribution of each wavelet coefficient is modeled by a mixture of two zero-mean Gaussian densities, corresponding to two states of being "small" or "large" in magnitude. The transition state probabilities model the persistent property where large/small values of wavelet coefficients tend to propagate through scales. To keep the model size manageable, we assume model parameters are the same at each scale. Following are examples of trained model parameters for the vertical wavelet subbands of the "Lena" (θ) and "Barbara" ($\tilde{\theta}$) images using Daubechies' 8-tap filters and 3 decomposition levels.

$$
\begin{aligned}
\boldsymbol{\pi} &= \begin{pmatrix} 0.69 & 0.31 \end{pmatrix} & \tilde{\boldsymbol{\pi}} &= \begin{pmatrix} 0.63 & 0.37 \end{pmatrix} \\
\mathsf{A}^{(2)} &= \begin{pmatrix} 0.99 & 0.01 \\ 0.22 & 0.78 \end{pmatrix} & \tilde{\mathsf{A}}^{(2)} &= \begin{pmatrix} 0.98 & 0.02 \\ 0.20 & 0.80 \end{pmatrix} \\
\mathsf{A}^{(3)} &= \begin{pmatrix} 0.99 & 0.01 \\ 0.32 & 0.68 \end{pmatrix} & \tilde{\mathsf{A}}^{(3)} &= \begin{pmatrix} 0.99 & 0.01 \\ 0.22 & 0.78 \end{pmatrix} \\
\sigma_1^{(1)} &= 11.8,\ \sigma_2^{(1)} = 67.1 & \tilde{\sigma}_1^{(1)} &= 24.6,\ \tilde{\sigma}_2^{(1)} = 74.8 \\
\sigma_1^{(2)} &= 4.1,\ \sigma_2^{(2)} = 29.3 & \tilde{\sigma}_1^{(2)} &= 6.9,\ \tilde{\sigma}_2^{(2)} = 31.9 \\
\sigma_1^{(3)} &= 2.8,\ \sigma_2^{(3)} = 10.3 & \tilde{\sigma}_1^{(3)} &= 3.1,\ \tilde{\sigma}_2^{(3)} = 14.8
\end{aligned}
\tag{23}
$$

Figure 3 shows the approximation results of the KLD $D(\theta\|\tilde{\theta})$ of the above two models. Again we see that the proposed algorithm offers comparable accuracy with the Monte-Carlo method that uses $N \approx 100$ randomly generated trees. Furthermore, the tying of dependence tree model parameters at each level greatly simplifies the computational complexity of the proposed algorithm, to about $15L$ multiplications and $8L$ additions, where L is the number of wavelet decomposition levels [5]. This makes the proposed algorithm crucial for the retrieval application.

V. Conclusion

We present a fast algorithm to approximate the Kullback-Leibler distance between two general dependence tree models. The algorithm uses the "upward" procedure to compute an upper bound for the KLD and has computational complexity similar to computing one density for a dependence tree model. For hidden Markov models, this algorithm is reduced to a simple expression, which is evaluated directly on the model parameters. Unlike the commonly used Monte-Carlo method, the proposed approximation is deterministic. Numerical experiments show that for the same accuracy, the proposed algorithm offers a computational saving of hundreds of times compared to the Monte-Carlo method.

References

[1] O. Ronen, J. R. Rohlicek, and M. Ostendorf, "Parameter estimation of dependence tree models using the EM algorithm," *IEEE Signal Proc. Letters*, vol. 2, no. 8, pp. 157–159, Aug. 1995.

[2] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[3] M. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based signal processing using hidden Markov models," *IEEE Trans. Signal Proc. (Special Issue on Wavelets and Filterbanks)*, pp. 886–902, Apr. 1998.

[4] B. H. Juang and L. R. Rabiner, "A probabilistic distance measure for hidden Markov models," *AT & T Tech. J.*, vol. 64, no. 2, pp. 391–408, Feb. 1985.

[5] M. N. Do and M. Vetterli, "Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models," *IEEE Trans. Multimedia*, 2002, to appear, `http://www.ifp.uiuc.edu/~minhdo/publications`.

[6] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Interscience, New York, NY, 1991.

[7] Y. Singer and M. K. Warmuth, "Batch and on-line parameter estimation of Gaussian mixtures based on the joint entropy," *Advances in Neural Information Processing Systems 11 (NIPS'98)*, pp. 578–584, 1998.