Calculating Kullback-Leibler Distances Between Hidden Markov Models

Brent M. Hendricks
Department of Electrical and Computer Engineering
Rice University
brentmh@ece.rice.edu

Abstract

This paper describes a technique for estimating the Kullback-Leibler (KL) distance between two Hidden Markov Models (HMMs), and for measuring the quality of the estimator. It also provides some results based on applying the technique to wavelet domain Hidden Markov Tree (HMT) models used in image segmentation. The technique is easily applied, because in most situations the necessary tools (data generation and likelihood calculation) are already in place.

1 Introduction

Hidden Markov models (HMMs) are used extensively in the field of statistical signal processing for modeling various real-world phenomena such as speech and images. They provide an elegant, flexible way to model certain dependencies between states without having to compute a joint probability density of high dimension. In practice, we commonly use HMMs to classify signals or images based on which of several models would be most likely to produce a given data sequence. Stein's lemma tells us that the performance of such an optimal classifier is related to the Kullback-Leibler (KL) distance between the models.

For some models, however, the KL distance is impossible to determine analytically. For those cases, we would like to have a reliable procedure for estimating the distance, and for determining the accuracy of the estimator. This paper will layout a simple procedure for performing this estimation, and how to use the bootstrap technique for estimating statistical accuracy.

2 Background

2.1 Hidden Markov Models

HMMs are used to model data sequences where it is known (or believed) that there exists an underlying, or *hidden* state S_i that determines the probability distribution of the ith data element. Three probability measures come into play here:

1. The initial state distribution: $p_{S_1}(m) = \Pr(S_1 = m)$,

2. The transition distribution: $\epsilon_{i,i-1}^{mr} = p_{S_i|S_{i-1}}(S_i = m|S_{i-1} = r)$,

3. The observed data distribution, given the value of the corresponding state: $f(w_i|S_i = m)$,

where $m, r \in \mathcal{S}$, the alphabet on which S_i is defined.

Traditionally, there are three problems that are associated with HMMs [2].

1. *Calculating likelihood.* For a given set of model parameters, determine the probability of occurrence for an observed data sequence.

2. *State Estimation.* For a given set of model parameters, determine the most probable sequence of states for an observed data sequence. This is the set of states which, given a particular model, would be most likely to produce the observed data.

3. *Training.* Given one or more sets of data, determine the model parameters that maximize the probability of observing that data.

Calculating the KL distance between two HMMs depends only on the first problem, but the solutions to all three are well known [2]. Of particular interest to us is the Expectation Maximization (EM) algorithm [3], used to solve the first and third problems. We restrict ourselves here to the E step, since that is all that is necessary to solve the likelihood problem.

2.1.1 Hidden Markov Trees

Although the technique described here can be applied to any HMM, the remainder of this paper will focus on a specific HMM structure and application: the Hidden Markov Tree as applied to wavelet coefficients (see Figure 1. Note that we have drawn the tree in a top-down manner rather than bottom-up as is customary). For a full development of this, see [1]. In this case we have the observed variables w_i whose probability densities depend on their corresponding hidden state variables S_i with the property that each w_i is conditionally independent of all other random variables given its state S_i. The state variables exhibit a Markovian dependency structure down the tree such that each state variable depends on the state of its immediate ancestor in the tree. Expressed mathematically,

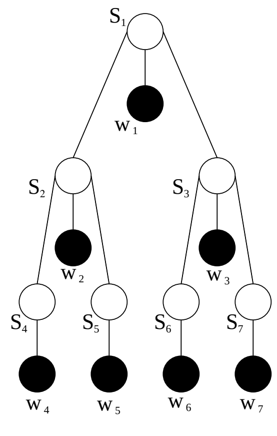


Figure 1: *Hidden Markov Tree structure. White nodes S_i represent hidden mixture state variables for observed variables W_i.*

$$p_{S_i}(s_i|\{S_j = s_j\}_{j\neq i}) = p_{S_i}(s_i|S_{\rho(i)} = r) \quad (1)$$

$$f_{W_i}(w_i|\{W_j\}_{j\neq i}, \{S_k = s_k\}_{\forall k}) = f_{W_i}(w_i|S_i = s_i) \quad (2)$$

where $\rho(i)$ represents the parent of state i. We will use an M-state Gaussian mixture model for each w_i, so that

$$f_{W_i|S_i}(w_i|S_i = m) \sim N(\mu_{i,m}, \sigma_{i,m}^2) \text{ for } m = 1 \ldots M \quad (3)$$

Let $p_{S_1}(m)$ be the probability mass function (pmf) of the root node S_1 and let $\epsilon_{i,\rho(i)}^{mr}$ specify the conditional probability of state S_i being in state m given that state $S_{\rho(i)}$ is in state r.

2.1.2 Computing likelihood using the EM algorithm

A complete application of the EM algorithm to wavelet-based HMTs appears in Appendix A of [1]. As mentioned above, we are only interested in the E step, to calculate the likelihood $f(\mathbf{w}|\theta)$ of a particular data sequence $\mathbf{w}$ given model parameters θ.

For simplicity of notation, we will define τ_i to be the the subtree that would contain w_i and all of its descendants if the observed variables were linked rather than the hidden variables. We also define

$$\beta_i(m) \equiv f(\tau_i|S_i = m, \theta) \tag{4}$$

$$\beta_{i,\rho(i)}(m) \equiv f(\tau_i|S_{\rho(i)} = m, \theta). \tag{5}$$

For the state variables at the lowest level, we set

$$\beta_i(m) = f(w_i|S_i = m) = N(\mu_{i,m}, \sigma_{i,m}^2). \tag{6}$$

The E step then consists of iterating up the tree, calculating $\beta_i(m)$ for each node, until we reach the root node.

$$\beta_{i,\rho(i)}(m) = \sum_{n=1}^{M} \epsilon_{i,\rho(i)}^{mr} \beta_i(n) \tag{7}$$

$$\beta_{\rho(i)}(m) = N(\mu_{\rho(i),m}, \sigma_{\rho(i),m}^2) \times \prod_{i \in c(\rho(i))} \beta_{i,\rho(i)}(m) \tag{8}$$

where $c(i)$ represents the set of children of node i. The likelihood of $\mathbf{w}$ is

$$f(\mathbf{w}|\theta) = f(\tau_1|\theta) = \sum_{m=1}^{M} p(S_1 = m, \tau_1|\theta) = \sum_{m=1}^{M} \beta_1(m) p_{S_1}(m). \tag{9}$$

This gives us an iterative procedure for calculating the likelihood function.

2.2 Kullback-Leibler Distances

The KL distance between the two probability densities $f^{(0)}(\mathbf{x})$ and $f^{(1)}(\mathbf{x})$ is expressed by

$$D(f^{(0)}||f^{(1)}) = \int_{\Re^n} f^{(0)}(\mathbf{x}) \ln \frac{f^{(0)}(\mathbf{x})}{f^{(1)}(\mathbf{x})} \, d\mathbf{x}. \tag{10}$$

Similarly, for two mass functions $p^{(0)}(\mathbf{x})$ and $p^{(1)}(\mathbf{x})$

$$D(p^{(0)}||p^{(1)}) = \sum_{\mathcal{X}^n} p^{(0)}(\mathbf{x}) \ln \frac{p^{(0)}(\mathbf{x})}{p^{(1)}(\mathbf{x})}. \tag{11}$$

While not a true distance metric (since it's not symmetric), it does often give a us sense of how "far" one density is from another. This intuitive sense of distance is made more rigorous by examining Stein's lemma, which states that for binary hypothesis testing between $p^{(1)}$ and $p^{(0)}$,

$$\text{If } P_M \leq \alpha, \text{ then } \lim_{L \to \infty} -\frac{\ln P_F}{L} = D(p^{(1)}||p^{(0)}) \tag{12}$$

where P_M is the probability of guessing hypothesis 0 when hypothesis 1 is true, and vice versa for P_F. This tells us that the error probability decays at best exponentially with the KL distance between the two hypotheses.

3 Approach

3.1 Analytical

For simplicity, let us first consider the degenerate case pictured in Figure 2, where we have two "trees", each with only one state S (but different state parameters), yielding coefficient of interest w. The probability densities for the two trees are

$$f^{(0)}(w) = \sum_{m=1}^{M} p_0(m) f^{(0)}_{W|S}(w|S=m) \tag{13}$$

$$f^{(1)}(w) = \sum_{m=1}^{M} p_1(m) f^{(1)}_{W|S}(w|S=m) \tag{14}$$

For our application, this results in a Gaussian mixture model. Unfortunately, the sum that appears in the likelihood formula for this model causes problems when trying to analytically compute the KL distance. The logarithm of a sum does not simplify, even if the sum is of Gaussian densities.

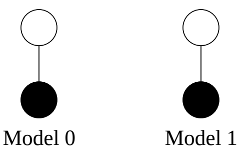


Figure 2: *Simplest HMT models*

Note that for higher order problems where the tree has more than one node, the problems compound. It might appear that since the states of the tree have a simple Markovian dependency, we could express the KL distance of an N-node tree in terms of simpler 1-node KL distances (N 1-dimensional integrals being easier to compute than one N-dimensional integral). This, however, is not the case because the Markov dependency exists between the *states* of the tree, whereas we are interested in the densities of the *coefficients*. There does not appear to be a way to separate the higher order integral. This shatters our hopes of an analytical solution and leaves us with numerical methods.

3.2 Numerical

Rather than use a numerical integration technique, we may approximate the KL distance by treating it as an expected value, since

$$D(f^{(0)}||f^{(1)}) = E_{f^{(0)}}\left[\ln \frac{f^{(0)}(\mathbf{x})}{f^{(1)}(\mathbf{x})}\right] \tag{15}$$

We now may apply known statistical techniques for mean estimation. Estimating the KL distance in this way consists of three separate steps.

1. *Generating data.* The goal of this step is to generate N realizations (data sequences) under model 0, where N is some large number. This step is accomplished in a straightforward manner by first assigning states down the the tree according to the state transitions $\epsilon^{nm}_{i,\rho(i)}$. Given these states, we may randomly draw the coefficients themselves from the correct distribution.

2. *Computing likelihoods.* As mentioned previously, the likelihood $f(\mathbf{w})$ is easily obtained by using the EM algorithm. In our case, we take the data generated by the first model, and compute the difference of log likelihoods of the data under each of the two models. This gives us $\ln \frac{f^{(0)}(\mathbf{w})}{f^{(1)}(\mathbf{w})}$, which we calculate for each of the N data sequences generated in step 1.

3. *Computing statistics.* The estimate $\hat{d}$ for the KL distance is obtained by averaging over the N log likelihood values found in the previous step. Here we invoke the law of large numbers (LLN) to claim that as the number of samples increases (note: this refers to the *number* of data sequences, not the *size* of the data sequences), the empirical average will approach the expected value. It should be noted, however, that the LLN says nothing about how quickly we will converge to the mean, or whether our estimator will be biased. We use the bootstrap technique to obtain these accuracy estimates for $\hat{d}$.

3.2.1 The Bootstrap

The bootstrap is a simple, though perhaps computationally intensive, technique for measuring the accuracy of a statistical estimate. The key feature of the bootstrap is that it utilizes the same set of data used to calculate the original estimator. The basic premise behind the bootstrap [5] is to create B bootstrap data sets by sampling with replacement from the original data. We compute the statistic of interest on each of these new data sets to obtain a set of B bootstrap statistics. We then measure the standard deviation of the set of bootstrap statistics to estimate the standard error of the original estimator.

In our problem, each data point is actually a sequence of wavelet coefficients $\mathbf{w}_j$, $j = 1 \ldots N$ where the subscript denotes the jth vector, rather than the jth element of the vector. Since these sequences are independent of each each other, we define $x_j = \ln \frac{f^{(0)}(\mathbf{w}_j)}{f^{(1)}(\mathbf{w}_j)}$, giving us a sequence of N independent datapoints $x_1, x_2, \ldots, x_N$. The statistic of interest $s(\mathbf{x})$ is then simply the mean of the datapoints.

The first step in the bootstrap process is to compute the B bootstrap data sequences by sampling the original data sequence with replacement. Call these new sequences $\mathbf{b}_i$ for $i = 1 \ldots B$. Efron [5] recommends a value for B of 50 to 200, so we chose B to be 200. Next we compute the sample statistic $s(\mathbf{b}_i)$ (in this case the mean) for each bootstrap sequence. This gives us a set of B bootstrap statistics, $s(\mathbf{b}_1) \ldots (\mathbf{b}_B)$. Let

$$m_{\text{boot}} = \frac{1}{B} \sum_{i=1}^{B} s(\mathbf{b}_i) \tag{16}$$

be the average of the bootstrap statistics.

The bootstrap estimate of standard error for $\hat{d}$ is the standard deviation of the set of bootstrap statistics,

$$\hat{se}_{\text{boot}} = \left(\sum_{i=1}^{B} \frac{[s(\mathbf{b}_i) - m_{\text{boot}}]^2}{B-1} \right)^{\frac{1}{2}} \tag{17}$$

We may also estimate the additive bias k of $\hat{d}$ using

$$\hat{k} = m_{\text{boot}} - \hat{d} \tag{18}$$

Ordinarily, using the bootstrap causes a vast increase in the amount of computation for a given problem. This is not true here, because we were able to deal with simple terms in x_i, rather than entire $\mathbf{w}_i$ sequences. The computation necessary to do the added processing on the bootstrap samples is relatively small when compared to generating the data sequences.

4 Tests and Results

As a sanity check, we would like for the distance between any HMT and itself to be zero. Using the approach described above, this will always be true since the likelihood is a deterministic function of the data sequence for a given model. This means that $f(\mathbf{w}|\theta_\mathbf{0}) = f(\mathbf{w}|\theta_\mathbf{1})$ when $\theta_\mathbf{0} = \theta_\mathbf{1}$ Therefore the estimator for the KL distance between a model and itself will be identically zero.

4.1 1D results

To further verify the validity of the estimates, we compute some simple 1D test cases whose KL distances can be computed "easily" through numeric integration. These models are pictured in Figure 3, and the estimation results are given in Table 1. In each case, we estimate the distance between $f_a(w)$ (the zero-mean model) and some other model with non-zero mean.

The first experiment estimated the KL distance between $f_a(w)$ and $f_b(w)$ as seen in Figure 3a,b. The results are all close to the numerically calculated value, being within ± 1.0 or 4%, a very encouraging result. Note that the bias values are quite small relative to the estimator, regardless of sample size.

In the second and third experiments (Figure 3a,c,d), we estimate $D(f_a(w)||f_c(w))$ and $D(f_a(w)||f_d(w))$ respectively. The estimated values are still close to the numerical calculation, but since the actual value is small, the percent error is larger.

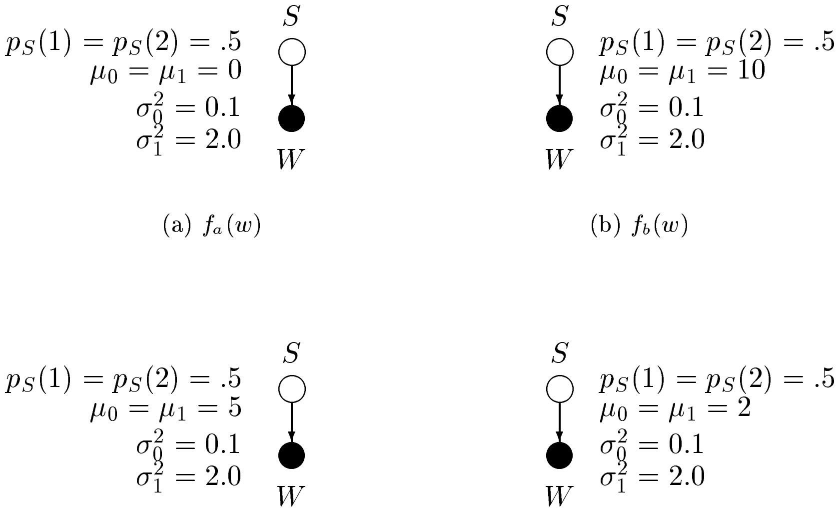
It appears to that the bias is about the same order of magnitude, regardless of the size of the estimate. This means that when estimating smaller means, the bias will be a larger percentage of the estimator.

$D(f_a\|\|f_b)$				
Samples	**10**	**100**	**1000**	**5000**
Mean	26.84	25.32	25.47	25.51
Bias	0.004	0.031	-0.001	0.005
Std. error	0.485	0.473	0.141	0.068
Numeric Calc.	26.284			

$D(f_a\|\|f_c)$				
Samples	**10**	**100**	**1000**	**5000**
Mean	7.117	7.307	7.276	7.174
Bias	-0.087	0.004	0.003	0.002
Std. error	1.247	0.232	0.085	0.039
Numeric Calc.	7.534			

$D(f_a\|\|f_d)$				
Samples	**10**	**100**	**1000**	**5000**
Mean	2.190	1.620	1.787	1.779
Bias	-0.047	0.005	0.002	-0.004
Std. error	0.298	0.152	0.047	0.020
Numeric Calc.	2.234			

Table 1: *Computed and estimated distances between 1D HMT models*



(a) $f_a(w)$ (b) $f_b(w)$

(c) $f_c(w)$ (d) $f_d(w)$

Figure 3: *1D HMT models. Each of these is a two state Gaussian mixture model. The only differences between the models are in the mean parameters. All other parameters are the same*

4.2 2D Results

This estimation approach for is easily extendible to 2D applications as well. There are effectively only two changes to be made. The first is that the HMT structures are no longer binary trees, but quad-trees so that each node has four children. The second change is that the wavelet coefficients are divided into three subbands(HH,LH,HL). Each of these subbands has a separate tree. The probabilistic relationship between these trees is not currently known, so for lack of a better approach we will assume that they are independent. This means that the likelihood for the whole tree is the product of the likelihoods for the three subband trees.

$D(f_{grass}\|\|f_{wood})$			
Samples	**10**	**100**	**1000**
Mean	17711	17610	17712
Bias	-9.725	-4.718	-0.581
Std. error	238.98	108.03	28.64

$D(f_{ground}\|\|f_{sea})$			
Samples	**10**	**100**	**1000**
Mean	15661	15492	15424
Bias	-7.721	-1.749	2.397
Std. error	219.97	76.221	23.40

$D(f_{forest}\|\|f_{scrub})$			
Samples	**10**	**100**	**1000**
Mean	552.95	546.36	548.5
Bias	0.682	-0.125	0.164
Std. error	17.55	5.590	1.878

Table 2: *Estimated distances between 2D HMT models*

Hyeokho Choi has done extensive work on image segmentation using 2D HMTs. His algorithm [4] trains HMTs based on training images, and then attempts to segment a given image by classifying its wavelet coefficients based on the trained models. As we mentioned earlier, Stein's lemma tells us that we will do better in classification problems where the KL distance is large. Let us look at the estimated KL distances between some the trained models in [4].

In the first example, we are comparing grass and wood textures. The images used to train the models are seen in Figure 4a,b. The second example compares models of ground and sea images trained on different regions of a satellite photograph (Figure 4c). The models for the third example were trained on forest and scrub regions of a radar image (Figure 4d). The estimation results for these three examples are in Table 2.

The first thing we notice about these results is the seemingly high distance estimates (at least, compared to the 1D results). This no doubt results from the large amount of data which is present since these images are 64x64 pixels each. Note that the biases and standard error values remain small relative to the estimated means. This confirms that these are good estimates of the actual KL distance between the models.

Something to note is the fact that the distance between the forest and scrub models is much lower than the distance in the either of the other two cases. This implies that these two models are much "closer together" than the others. After conferring with Hyeokho, it turns out that classification was more difficult with these two models than with the others, and met with poorer results as well.

$D(f_{figure} \| f_{background})$			
Samples	**10**	**100**	**1000**
Mean	8570	9014	9158
Bias	107.1	-47.25	-2.54
Std. error	916.7	348.3	108.6

$D(f_{text} \| f_{background})$			
Samples	**10**	**100**	**1000**
Mean	40696	41195	41021
Bias	2.75	16.40	5.044
Std. error	447.0	139.66	47.28

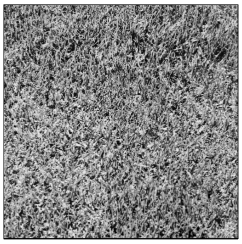
$D(f_{figure} \| f_{text})$			
Samples	**10**	**100**	**1000**
Mean	2056	2081	2093
Bias	1.78	0.456	-0.156
Std. error	29.97	9.386	3.204

Table 3: *Estimated distances between figure, text and background models*

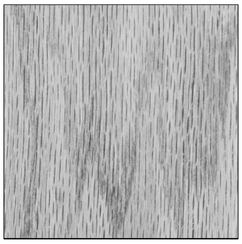
Another interesting example is shown in figure 4(e). In this case there are three models, one each for text, background, and figure. The results of KL distance estimation on these models are shown in Table 3. The relative sizes of the distance estimates fit what we would intuitively expect: a very large distance between text and background, a smaller distance between figure and background, and the smallest distance between figure and text. As before, the estimated biases are very small. The standard errors are relatively small, with the exception of those found when estimating the distance between the figure and the background. This means that there is a good deal of variation when measuring the distance between models for the figure and the background. It does decrease with increasing number of samples, but it would require more samples to reach the same level of confidence in the estimator as for the other cases.

5 Conclusions

In cases where the KL distance between two HMMs is difficult or impossible to solve analytically, we need a technique for distance estimation, and for calculating the accuracy of the estimator. The iterative approach for estimating KL distances presented here performs well for HMT classification

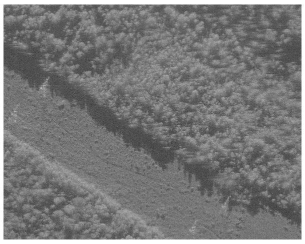


(a) Grass texture

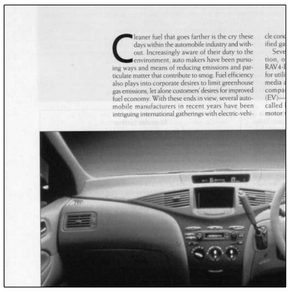


(b) Wood texture

(c) Forest and Sea



(d) Forest and Scrub



(e) Training

Figure 4: *Images used to train 2D HMT models. (a) and (b) represent individual images used to train separate models. In (c) (d), and (e) the models were trained on different regions of the image*

problems, appears to have low bias, and is straightforward to apply to existing models. Using the bootstrap method, we can easily put an accuracy measure on this distance estimate as well.

6 Future Work

There are two main avenues for future research which may stem from this project.

1. *Implementation improvements* One particular improvement would be to calculate confidence intervals [5] for the estimator $\hat{d}$. This will yield better accuracy estimates than relying solely on the standard error estimate. Other improvements might involve investigating more efficient methods for likelihood computation and data generation.

2. *Applications* There are are many possible ways to extend and use this technique for application to various HMT problems. One example might be to investigate the KL distance between HMTs before and after pruning(deleting subtrees). This might be useful in compression by minimizing the distance for a desired level of compression. Another possibility might be to examine KL distances at various levels down the tree. It may be that for certain classification problems, looking at a particular level would be optimal.

References

[1] M.S. Crouse, R.D. Nowak, and R.G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Proc.*, vol. 46, no. pp. 996-902, April 1998.

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

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

[4] H. Choi and R.G.Baraniuk, "Multiscale image segmentation using wavelet-domain classification," to appear in Proc. SPIE, Denver, July 1999.

[5] B. Efron and R.J. Tibshirani, *An introduction to the bootstrap.* New York: Chapman and Hall, 1993.