Entropically Constrained Parameter Estimation for Hidden Markov Models

Graham Grindlay
grindlay@soe.ucsc.edu

June 11, 2004

Abstract

In this paper we examine alternative parameter updates for hidden Markov models. The main focus will be on entropically constrained updates where we attempt to maximize a log-likelihood subject to an entropy-based constraint. In the case of the Joint-Entropy update [3], this may take the form of a relative entropy which measures the distance between the current parameters and those which we are trying to maximize. We will make the connection between the Joint-Entropy update and the entropic maximum *a posteriori* (MAP) update [1] for multinomials, deriving both updates as well as a hybrid of the two. While we only consider discrete observation HMMs here, it should be possible to derive similar updates for HMMs with Gaussian observations.

1 Notation & Preliminaries

Typically, we are given a set of data sequences Ω, where each sequence, $X \in \Omega$, is comprised of a set of observations such that $X = \{x_1, x_2, ..., x_T\}$. A discrete-observation HMM has a set of states, $Q = \{q_1, q_2, ..., q_N\}$, a set of observation symbols, $O = \{o_1, o_2, ..., o_M\}$, a vector of priors, π, where π_i gives the probability of starting a state sequence in state q_i, a transition matrix, a, where $a_{i,j}$ gives the probability of transitioning from state q_i to state q_j, and an observation matrix, b, where $b_{i,k}$ gives the probability of observation o_k when in state q_i. Because the number of states and observation symbols are implicit in the probability matrices, an HMM, θ, can be parameterized by: $\theta = \{\pi, a, b\}$.

In the following sections we refer to the generic parameter, θ_i where i ranges from 1 to the total number of parameters in the HMM (ie. the number of entries in π, a, and b combined). We use subscripts to index multi-dimentional parameters and superscripts to denote membership. For example, q_i refers to the i^{th} state, while q^i refers to the state of which parameter i is a member and θ^i refers to the vector of probabilities (multinomial) of which parameter i is a member.

We can think of an HMM as a probabilistic model which describes the joint probability of a collection of random variables, $\{O_1, ..., O_T, Q_1, ..., Q_T\}$. For a given state sequence, S, we define $n_{q_i}(S)$ to be the number of times that state q_i appears in sequence S and $n_i(X, S)$ to be the number of times that parameter i is used in a state sequence, S, with data, X. It is important to keep in mind that this quantity does *not* depend on a particular instance of θ. We also define several expectation terms. First, let $\hat{n}_{\theta i}(X|\theta) = \sum_S n_i(X, S)P(S|X, \theta)$ be the expected usage of parameter θ_i over all state sequences, S, that produce, X, given the HMM with parameter set, θ. Note that this quantity is efficiently calculated by the forward-backward algorithm [2]. Second, let us also define $\hat{n}_{\theta_i}(\theta) = \sum_{X,S} n_i(X, S)P(S, X|\theta)$ to be the expected usage of θ_i, based on all possible state sequences, S, and all possible observation sequences, X. Finally, let $\hat{n}_{q_i}(\theta) = n_{q_i}(S)P(S|\theta)$ be the expected usage of state q_i by an HMM with parameter set, θ. Note that this quantity is also equal to $\sum_{j \in q_i} \hat{n}_{\theta_j}(\theta)$. Because this quantity can be expensive to calculate in practice, we could also use an approximation based on the data available:

$$\hat{n}_{q_i}(\theta) \approx \frac{1}{|\Omega|} \sum_{X \in \Omega} \sum_t P(q_t = i|X) \tag{1}$$

where $P(q_t = i|X)$ is the probability of being in state q_i at time t of

sequence X and is often called $\gamma_i(t, X)$ in the speech literature. It can also be efficiently calculated by using the forward-backward algorithm.

2 The Joint-Entropy Update

In the Joint-Entropy framework [3, 4], our goal is to re-estimate parameters by maximizing an objective function which is composed of a weighted combination of log-likelihood and model divergence. Intuitively, we want to find the set of parameters that best models the data while staying close to our old parameter which encapsulate all that we have learned so far. We can formulate this as follows:

$$U(\tilde{\theta}) = \mathcal{LL}(\tilde{\theta}|\Omega) - \frac{1}{\eta}\Delta(\tilde{\theta}, \theta) \tag{2}$$

Because the log-likelihood depends on usage statistics, this equation can be difficult to maximize. Therefore, we typically approximate with a Taylor expansion around the log-likelihood of the current parameter estimates:

$$U(\tilde{\theta}) = \Big(\mathcal{LL}(\theta|\Omega) + (\tilde{\theta} - \theta)\nabla_\theta(\mathcal{LL}(\theta|\Omega)\Big) - \frac{1}{\eta}\Delta(\tilde{\theta}, \theta) \tag{3}$$

First, let us define the log-likelihood term. Given that we are considering only discrete HMMs with multinomial parameter vectors, we can use our above notation to define the complete-data likelihood for an observation sequence, X and state sequence, S, as: $P(X, S|\theta) = \prod_i \theta_i^{n_i(X,S)}$. This lets us also define the likelihood of an observation sequence, X, by summing over all possible state sequences, S: $P(X|\theta) = \sum_S \prod_i \theta_i^{n_i(X,S)}$. In the batch case, we define the log-likelihood as the average of the individual data sequence log-likelihoods:

$$\mathcal{LL}(\theta|\Omega) = \frac{1}{|\Omega|}\sum_{X\in\Omega} \log(P(X|\theta)) \tag{4}$$

We take derivatives with respect to each parameter as follows:

$$\begin{aligned}\frac{\partial}{\partial\theta_i}P(X, S|\theta) &= \prod_{j=1}^{i-1}\theta_j^{n_{\theta_j}(X,S)} n_i(X, S)\theta_i^{n_i(X,S)-1}\prod_{k=i+1}^{|q^i|}\theta_k^{n_{\theta_k}(X,S)} \\ &= \frac{n_i(X, S)}{\theta_i}\prod_i \theta_i^{n_i(X,S)}\end{aligned} \tag{5}$$

$$
\begin{aligned}
\frac{\partial}{\partial \theta_i}\mathcal{LL}(\theta|\Omega) &= \frac{1}{|\Omega|}\sum_{X\in\Omega}\sum_{S}\frac{\frac{\partial}{\partial\theta_i}P(X,S|\theta)}{\sum_S P(X,S|\theta)} \\
&= \frac{1}{|\Omega|}\sum_{X\in\Omega}\sum_{S}\frac{\frac{\partial}{\partial\theta_i}P(X,S|\theta)}{P(X|\theta)} \\
&= \frac{1}{|\Omega|}\sum_{X\in\Omega}\sum_{S}\frac{n_i(X,S)}{\theta_i}\frac{P(X,S|\theta)}{P(X|\theta)} \\
&= \frac{1}{|\Omega|}\sum_{X\in\Omega}\sum_{S}\frac{n_i(X,S)}{\theta_i}P(S|X,\theta) \\
&= \frac{\sum_{X\in\Omega}\hat{n}_{\theta i}(X|\theta)}{|\Omega|\theta_i}
\end{aligned} \tag{6}
$$

Now we need to define an appropriate divergence function, $\Delta(\tilde{\theta},\theta)$. In the Joint-Entropy framework, we use a relative entropy between models. For HMMs, this is a divergence between the distributions induced by two HMMs over all possible observation sequences, X, and all possible hidden state sequences, S:

$$
\Delta(\tilde{\theta},\theta) = \sum_X\left(\sum_S P(X,S|\tilde{\theta})\right)\log\frac{\sum_S P(X,S|\tilde{\theta})}{\sum_S P(X,S|\theta)} \tag{7}
$$

Using our previously defined notation and the log-sum inequality, we can approximate the above by:

$$
\begin{aligned}
\hat{\Delta}(\tilde{\theta}, \theta) &= \sum_{X,S} P(X, S|\tilde{\theta}) \log \frac{P(X, S|\tilde{\theta})}{P(X, S|\theta)} \\
&= \sum_{X,S} P(X, S|\tilde{\theta}) \log \frac{\prod_i \tilde{\theta}_i^{\, n_i(X,S)}}{\prod_i \theta_i^{n_i(X,S)}} \\
&= \sum_{X,S} P(X, S|\tilde{\theta}) \sum_i n_i(X, S) \log \frac{\tilde{\theta}_i}{\theta_i} \\
&= \sum_i \sum_{X,S} P(X, S|\tilde{\theta}) n_i(X, S) \log \frac{\tilde{\theta}_i}{\theta_i} \\
&= \sum_i \hat{n}_{\theta_i}(\tilde{\theta}) \log \frac{\tilde{\theta}_i}{\theta_i} \\
&= \sum_i \hat{n}_{q^i}(\tilde{\theta}) \left(\tilde{\theta}_i \log \frac{\tilde{\theta}_i}{\theta_i} \right) \qquad (8) \\
&= \sum_i \hat{n}_{q_i}(\tilde{\theta}) \sum_{j \in q_i} \tilde{\theta}_j \log \frac{\tilde{\theta}_j}{\theta_j} \qquad (9)
\end{aligned}
$$

The equality in (8) is justified by the property of absorbing HMMs that, for any parameter $\theta_i \in \theta$, $\hat{n}_{\theta_i}(\theta) = \theta_i \hat{n}_{q_i}(\theta)$.

Because (9) contains the expected state usage in the new parameters, $\tilde{\theta}$, this expression is very difficult to solve for. Therefore, we further approximate, by replacing the usage term, $\hat{n}_{q_i}(\tilde{\theta})$ with $\hat{n}_{q_i}(\theta)$, yielding:

$$
\hat{\tilde{\Delta}}(\tilde{\theta}, \theta) = \sum_i \hat{n}_{q_i}(\theta) \sum_{j \in q_i} \tilde{\theta}_j \log \frac{\tilde{\theta}_j}{\theta_j} \qquad (10)
$$

Taking derivatives, we get:

$$
\frac{\partial}{\partial \tilde{\theta}_i} \hat{\tilde{\Delta}}(\tilde{\theta}, \theta) = \hat{n}_{q^i}(\theta) \left(\log \frac{\tilde{\theta}_i}{\theta_i} + 1 \right) \qquad (11)
$$

To obtain the updated parameters, we set the derivative of $U(\tilde{\theta})$ to 0 and solve for θ_i. The derivative of $U(\tilde{\theta})$ is readily formed by plugging the derivatives of the log-likelihood and divergence function and adding a Lagrangian term to constrain each of the parameter vectors to sum to 1. This gives:

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \tilde{\theta}_i} \mathcal{LL}(\theta|\Omega) - \frac{1}{\eta} \frac{\partial}{\partial \tilde{\theta}_i} \hat{\Delta}(\tilde{\theta}, \theta) + \frac{\partial}{\partial \tilde{\theta}_i} \lambda_{\tilde{\theta}^i} \left(\sum_i^{|\tilde{\theta}^i|} \tilde{\theta}_i - 1 \right) \\
&= \frac{\sum_{X \in \Omega} \hat{n}_{\theta i}(X|\theta)}{|\Omega|\theta_i} - \frac{1}{\eta} \hat{n}_{q_i}(\theta) \left(\log \frac{\tilde{\theta}_j}{\theta_j} - 1 \right) + \lambda_{\tilde{\theta}^i} \\
&= \frac{\sum_{X \in \Omega} \hat{n}_{\theta i}(X|\theta)}{\hat{n}_{q_i}(\theta)|\Omega|\theta_i} - \frac{1}{\eta} \log \frac{\tilde{\theta}_j}{\theta_j} + \frac{\lambda_{\tilde{\theta}^i}}{\hat{n}_{q_i}(\theta)} - 1 \\
&= \frac{\sum_{X \in \Omega} \hat{n}_{\theta i}(X|\theta)}{\hat{n}_{q_i}(\theta)|\Omega|\theta_i} - \frac{1}{\eta} \log \frac{\tilde{\theta}_j}{\theta_j} + \lambda'_{\tilde{\theta}^i} \qquad (12)
\end{aligned}
$$

where $\lambda'_{\tilde{\theta}^i} = \frac{\lambda_{\tilde{\theta}^i}}{\hat{n}_{q_i}(\theta)} - 1$

Solving for $\tilde{\theta}_i$ and replacing $\lambda'_{\tilde{\theta}^i}$ with a normalizing term, we arrive at the batch *Joint-Entropy update*:

$$
\tilde{\theta}_i = \frac{\theta_i \exp\left(\frac{\eta}{\hat{n}_{q_i}(\theta)} \frac{\sum_{X \in \Omega} \hat{n}_{\theta i}(X|\theta)}{|\Omega|\theta_i} \right)}{\sum_{j \in \theta^i} \theta_j \exp\left(\frac{\eta}{\hat{n}_{q_j}(\theta)} \frac{\sum_{X \in \Omega} \hat{n}_{\theta j}(X|\theta)}{|\Omega|\theta_j} \right)} \qquad (13)
$$

Note that if we use the approximation to the state usage term, $\hat{n}_{q_j}(\theta)$, as given in (1), then this update is becomes the *approximated Joint-Entropy update*.

3 The Entropic MAP Estimator

In the Joint-Entropy update, we would ideally like to solve for the posterior with respect to the complete set of model parameters while using the log-likelihood of the parameters for which we are trying to maximize. This is an extremely difficult problem in the case of HMMs or other models where we have latent variables and because of this, we were forced to use several approximations in our update.

To see how we might be able to circumvent at least some of our approximations, let us return to the objective. Looking back at $U(\tilde{\theta})$, we see that it can be interpreted as a log-posterior. Thus, the posterior can be viewed using Bayes' Rule as:

$$P(\tilde{\theta}|\omega) = \frac{P(\tilde{\theta})\mathcal{L}(\tilde{\theta}|\omega)}{P(\omega)} \tag{14}$$

$$= \frac{e^{-\frac{1}{\eta}\Delta(\tilde{\theta},\theta)}\mathcal{L}(\tilde{\theta}|\omega)}{P(\omega)} \tag{15}$$

$$\propto e^{-\frac{1}{\eta}\Delta(\tilde{\theta},\theta)}\mathcal{L}(\tilde{\theta}|\omega) \tag{16}$$

Since $P(\omega)$ does not depend on $\tilde{\theta}$, we can safely discard it. The reason why we have replaced Ω with ω is to reinforce the idea that we are now interpreting the objective function in Bayesian terms and thus trying to optimize $\tilde{\theta}$ given the *evidence*, ω (which for our purposes, will represent counts or parameter usage). Although this abstraction may seem purely philosophical, it will prove useful in our discussion.

In order for this notion of evidence to be meaningful, we need an appropriate definition for the log-likelihood. Recall that the EM algorithm constructs a local lower bound to the log-likelihood by working to maximize the expected complete-data log-likelihood (referred to as the Q function) rather than trying to maximize the incomplete log-likelihood directly. In this spirit, we define the expected complete log-likelihood as follows using our notation defined above:

$$
\begin{aligned}
Q(\tilde{\theta}, \theta) &= E\left[\log(P(\Omega, S|\tilde{\theta})) \mid \Omega, \theta\right] \\
&= \sum_S \log(P(\Omega, S|\tilde{\theta}))P(S|\Omega, \theta) \\
&= \sum_S \frac{1}{|\Omega|} \sum_{X \in \Omega} \log(P(X, S|\tilde{\theta}))P(S|X, \theta) \\
&= \sum_S \frac{1}{|\Omega|} \sum_{X \in \Omega} \log(\prod_i \tilde{\theta}_i^{n_i(X,S)})P(S|X, \theta) \\
&= \sum_S \frac{1}{|\Omega|} \sum_{X \in \Omega} \sum_i \log({\tilde{\theta}_i}^{n_i(X,S)})P(S|X, \theta) \\
&= \sum_S \frac{1}{|\Omega|} \sum_{X \in \Omega} \sum_i n_i(X, S) \log(\tilde{\theta}_i)P(S|X, \theta) \\
&= \frac{1}{|\Omega|} \sum_{X \in \Omega} \sum_i \log(\tilde{\theta}_i) \sum_S n_i(X, S)P(S|X, \theta) \\
&= \frac{1}{|\Omega|} \sum_{X \in \Omega} \sum_i \log(\tilde{\theta}_i)\, \hat{n}_{\theta i}(X|\theta) \\
&= \frac{1}{|\Omega|} \sum_{X \in \Omega} \sum_i \log({\tilde{\theta}_i}^{\hat{n}_{\theta i}(X|\theta)}) \\
&= \frac{1}{|\Omega|} \sum_{X \in \Omega} \log(\prod_i {\tilde{\theta}_i}^{\hat{n}_{\theta i}(X|\theta)}) \qquad (17)
\end{aligned}
$$

To keep our notation consistent, from now on we will now refer to $Q(\tilde{\theta}, \theta)$ as $\hat{\mathcal{LL}}(\tilde{\theta}|\Omega)$. We can now easily take derivatives of $\hat{\mathcal{LL}}(\tilde{\theta}|\Omega)$ with respect to the target parameter, $\tilde{\theta}_i$:

$$
\begin{aligned}
\frac{\partial}{\partial \theta_i} \hat{\mathcal{LL}}(\tilde{\theta}|\Omega) &= \frac{1}{|\Omega|} \sum_{X \in \Omega} \left(\frac{\frac{\partial}{\partial \tilde{\theta}_i} \prod_i {\tilde{\theta}_i}^{\hat{n}_{\theta i}(X|\theta)}}{\prod_i {\tilde{\theta}_i}^{\hat{n}_{\theta i}(X|\theta)}} \right) \\
&= \frac{1}{|\Omega|} \sum_{X \in \Omega} \left(\frac{\hat{n}_{\theta i}(X|\theta) \prod_i {\tilde{\theta}_i}^{\hat{n}_{\theta i}(X|\theta)}}{\tilde{\theta}_i \prod_i {\tilde{\theta}_i}^{\hat{n}_{\theta i}(X|\theta)}} \right) \\
&= \frac{1}{|\Omega|} \sum_{X \in \Omega} \left(\frac{\hat{n}_{\theta i}(X|\theta)}{\tilde{\theta}_i} \right) \\
&= \frac{\sum_{X \in \Omega} \hat{n}_{\theta i}(X|\theta)}{|\Omega| \tilde{\theta}_i} \qquad (18)
\end{aligned}
$$

Note that we have arrived at a form very similar to that of (6), only now we have $\tilde{\theta}_i$ in the denominator as opposed to θ_i. To make the following derivations more clear, we collect what corresponds to the evidence terms in (18) into a single variable, ω_i:

$$\omega_i = \frac{\sum_{X \in \Omega} \hat{n}_{\theta i}(X|\theta)}{|\Omega|} \tag{19}$$

With this definition of evidence we have achieved a powerful result: the log-likelihood depends only on the evidence term rather than the data and hidden variables. In the case of HMMs, this effectively decouples the model parameters meaning that we can decompose the model into a set of independent multinomial distributions and solve each independently.

3.1 A Relative Entropy MAP estimator (REMAP) for Multinomials

It is possible to find MAP solutions to $\tilde{\theta}$ for a multinomial distribution given some evidence, ω. Brand provides such a solution in [1] but using an entropy constraint on the new model parameters rather than a relative entropy between the new and old parameters. In his solution, the goal is to bias parameter estimation towards sparse, simple models (ie. minimum entropy). It is worth noting that this minimum entropy bias comes as a special case of the relative entropy bias when we use the uniform distribution instead of the old parameters. We will develop the relative-entropy MAP estimator here.

Let us begin our derivation, by considering the likelihood function for a multinomial: $P(\omega|\theta) = \prod_i {\theta_i}^{\omega_i}$. Now consider the relative entropy. In the multinomial MAP estimate, we can use an exact relative entropy between new and old distributions; there is no need to approximate as in the HMM case. Therefore, the relative entropy is: $\Delta(\tilde{\theta}, \theta) = \sum_i \tilde{\theta}_i \log \frac{\tilde{\theta}_i}{\theta_i}$ and our prior becomes: $P(\tilde{\theta}) = \exp\left(-\frac{1}{\eta}\sum_i \tilde{\theta}_i \log \frac{\tilde{\theta}_i}{\theta_i}\right)$ which we can rewrite as: $P(\tilde{\theta}) = \prod_i \left(\frac{\tilde{\theta}_i}{\theta_i}\right)^{-\left(\frac{\tilde{\theta}_i}{\eta}\right)}$.

Now consider the posterior:

$$\begin{aligned} P(\tilde{\theta}|\omega) &\propto \mathcal{L}(\tilde{\theta}|\omega)\ P(\tilde{\theta}) \\ &= \prod_i {\tilde{\theta}_i}^{\omega_i} \prod_i \left(\frac{\tilde{\theta}_i}{\theta_i}\right)^{-\left(\frac{\tilde{\theta}_i}{\eta}\right)} \\ &= \prod_i {\tilde{\theta}_i}^{\omega_i} \left(\frac{\tilde{\theta}_i}{\theta_i}\right)^{-\left(\frac{\tilde{\theta}_i}{\eta}\right)} \end{aligned} \tag{20}$$

To solve for parameter $\tilde{\theta}_i$, we add Lagrange multiplier to ensure that each multinomial sums to 1, take derivatives of the log-posterior, set to 0, and solve:

$$\begin{aligned} 0 &= \frac{\partial}{\partial\tilde{\theta}_i}\left[\log\left(\prod_i {\tilde{\theta}_i}^{\omega_i}\left(\frac{\tilde{\theta}_i}{\theta_i}\right)^{-\left(\frac{\tilde{\theta}_i}{\eta}\right)}\right) + \lambda\left(\sum_i \tilde{\theta}_i - 1\right)\right] \\ &= \frac{\partial}{\partial\tilde{\theta}_i}\left[\sum_i \log\left({\tilde{\theta}_i}^{\omega_i}\left(\frac{\tilde{\theta}_i}{\theta_i}\right)^{-\left(\frac{\tilde{\theta}_i}{\eta}\right)}\right) + \lambda\left(\sum_i \tilde{\theta}_i - 1\right)\right] \\ &= \frac{\partial}{\partial\tilde{\theta}_i}\left[\sum_i \left(\log\left({\tilde{\theta}_i}^{\omega_i}\right) + \log\left(\left(\frac{\tilde{\theta}_i}{\theta_i}\right)^{-\left(\frac{\tilde{\theta}_i}{\eta}\right)}\right)\right) + \lambda\left(\sum_i \tilde{\theta}_i - 1\right)\right] \\ &= \frac{\partial}{\partial\tilde{\theta}_i}\left[\sum_i \left(\omega_i \log\tilde{\theta}_i - \frac{\tilde{\theta}_i}{\eta}\log\frac{\tilde{\theta}_i}{\theta_i}\right) + \lambda\left(\sum_i \tilde{\theta}_i - 1\right)\right] \\ &= \frac{\omega_i}{\tilde{\theta}_i} - \frac{1}{\eta}\log\frac{\tilde{\theta}_i}{\theta_i} - \frac{1}{\eta} + \lambda \end{aligned} \tag{21}$$

We can easily solve (21) for λ:

$$\lambda = -\frac{\omega_i}{\tilde{\theta}_i} + \frac{1}{\eta}\log\left(\frac{\tilde{\theta}_i}{\theta_i}\right) + \frac{1}{\eta} \tag{22}$$

Solving for $\tilde{\theta}_i$ is a bit more tricky due to the mixed polynomial and logarithmic terms. However, we can do so by making use of the Lambert W

function: $W(y)e^{W(y)} = y$ (see Figure 1). There are a variety of techniques for efficient calculation of the W function. For details, see [1].

We will forgo the derivation of $\tilde{\theta}_i$ in the interest of brevity and because we provide a full derivation for the complete HMM case in the follow section which is quite similar. The update for $\tilde{\theta}_i$ is:

$$\tilde{\theta}_i = \frac{\eta\omega_i}{W\left(\frac{\eta\omega_i}{\theta_i}e^{1-\eta\lambda}\right)} \tag{23}$$

This solution, along with (22) form a fix-point solution for λ and therefore $\tilde{\theta}_i$. We can iteratively update the *current update's* estimate of $\tilde{\theta}$ by first solving for $\tilde{\theta}$ given λ, normalizing $\tilde{\theta}$, and then calculating λ given $\tilde{\theta}$. Convergence is fast (2-5 iterations). Brand [1] provides ideas for how to initialize λ.

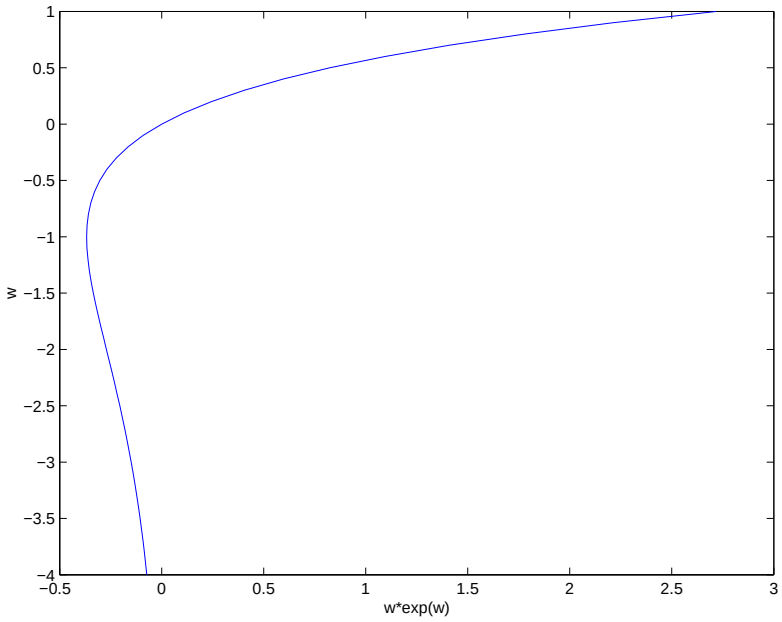

Figure 1: This plot shows the real-valued branches of the Lambert W function. The W function is the multivalued inverse of $w \rightarrow we^w$ and so we can plot the real branches indirectly, by plotting $w \rightarrow we^w$ with the axes swapped.

3.1.1 Interpreting the Entropic Posterior

Brand [1] provides an interesting interpretation of the solution to the log-posterior purely as an entropy minimization problem. In our framework, there is a similar interpretation for (14):

$$
\begin{aligned}
-\max_{\tilde{\theta}} \log(P(\tilde{\theta}|\omega)) &= \min_{\tilde{\theta}} -\log\left(\prod_i {\tilde{\theta}_i}^{\omega_i}\left(\frac{\tilde{\theta}_i}{\theta_i}\right)^{-\left(\frac{\tilde{\theta}_i}{\eta}\right)}\right) \\
&= \min_{\tilde{\theta}} -\sum_i \log\left({\tilde{\theta}_i}^{\omega_i}\left(\frac{\tilde{\theta}_i}{\theta_i}\right)^{-\left(\frac{\tilde{\theta}_i}{\eta}\right)}\right) \\
&= \min_{\tilde{\theta}} -\sum_i \left(\omega_i \log\tilde{\theta}_i - \frac{\tilde{\theta}_i}{\eta}\log\frac{\tilde{\theta}_i}{\theta_i}\right) \\
&= \min_{\tilde{\theta}} -\sum_i \left(\omega_i \log\tilde{\theta}_i - \omega_i\log\omega_i + \omega_i\log\omega_i - \frac{\tilde{\theta}_i}{\eta}\log\frac{\tilde{\theta}_i}{\theta_i}\right) \\
&= \min_{\tilde{\theta}} \left(\sum_i \omega_i \log\frac{\omega_i}{\tilde{\theta}_i} - \sum_i \omega_i\log\omega_i + \frac{1}{\eta}\sum_i \tilde{\theta}_i\log\frac{\tilde{\theta}_i}{\theta_i}\right) \\
&= \min_{\tilde{\theta}} \left(\Delta(\omega,\tilde{\theta}) + H(\omega) + \frac{1}{\eta}\Delta(\tilde{\theta},\theta)\right) \qquad (24) \\
& \qquad (25)
\end{aligned}
$$

We can see that the posterior works to minimize a weighted sum of entropies (H is the shannon entropy). $\frac{1}{\eta}\Delta(\tilde{\theta},\theta)$ keeps our new and old parameters close while $\Delta(\omega,\tilde{\theta})$ describes how much the evidence and model parameters disagree. Because the evidence, ω, is derived from the expected complete log-likelihood, which is in turn based on the model structure, $H(\omega)$ can be thought of as a lower-bound on the code length necessary to describe which of the data variations encoded by the model structure is actually instantiated in the dataset.

3.2 The Joint-Entropy MAP (JEMAP) Estimator

Let us now return to the Joint-Entropy framework and see if we can derive an entropic MAP estimator with respect to the full set of HMM

parameters. The difference between the multinomial case and the full model case, is that we need to use our approximated HMM divergence (10).

We construct the derivatives of the log-posterior using (11) and (18), add Lagrange multipliers, set to 0, and solve for $\tilde{\theta}_i$ and $\lambda_{\tilde{\theta}^i}$:

$$\begin{aligned}
0 &= \frac{\partial}{\partial \theta_i}\hat{\mathcal{L}\mathcal{L}}(\tilde{\theta}|\Omega) - \frac{\partial}{\partial \tilde{\theta}_i}\frac{1}{\eta}\hat{\tilde{\Delta}}(\tilde{\theta},\theta) + \frac{\partial}{\partial \tilde{\theta}_i}\lambda_{\tilde{\theta}^i}\left(\sum_i^{|\tilde{\theta}^i|}\tilde{\theta}_i - 1\right) \\
&= \frac{\omega_i}{\tilde{\theta}_i} - \frac{\hat{n}_{q^i}(\theta)}{\eta}\log\frac{\tilde{\theta}_i}{\theta_i} - \frac{\hat{n}_{q^i}(\theta)}{\eta} + \frac{\partial}{\partial \tilde{\theta}_i}\lambda_{\tilde{\theta}^i}\left(\sum_i^{|\tilde{\theta}^i|}\tilde{\theta}_i - 1\right) \\
&= \frac{\omega_i}{\tilde{\theta}_i} - \frac{\hat{n}_{q^i}(\theta)}{\eta}\log\frac{\tilde{\theta}_i}{\theta_i} - \frac{\hat{n}_{q^i}(\theta)}{\eta} + \lambda_{\tilde{\theta}^i} \qquad (26) \\
&= \frac{\hat{n}_{q^i}(\theta)}{\eta}\left[\frac{\eta\omega_i}{\hat{n}_{q^i}(\theta)\tilde{\theta}_i} - \log\frac{\tilde{\theta}_i}{\theta_i} - 1 + \frac{\eta\lambda_{\tilde{\theta}^i}}{\hat{n}_{q^i}(\theta)}\right] \\
&= a\left[\frac{\omega_i}{a\tilde{\theta}_i} - \log\frac{\tilde{\theta}_i}{\theta_i} - 1 + \frac{\lambda_{\tilde{\theta}^i}}{a}\right] \qquad (27)
\end{aligned}$$

where $a = \frac{\hat{n}_{q^i}(\theta)}{\eta}$. We can easily solve for $\lambda_{\tilde{\theta}^i}$:

$$\lambda_{\tilde{\theta}^i} = -\frac{\omega_i}{\tilde{\theta}_i} + \frac{\hat{n}_{q^i}(\theta)}{\eta}\log\frac{\tilde{\theta}_i}{\theta_i} + \frac{\hat{n}_{q^i}(\theta)}{\eta} \qquad (28)$$

Now we show how to work backwards from the W function and arrive at the bracketed expression in (27). During this derivation, we will drop the subscripting on $\lambda_{\tilde{\theta}^i}$ for the sake of clarity. Once we have derived an expression for $\tilde{\theta}_i$ we will return λ to its full form. To begin, note that the W function can be re-written as: $W(y) + \log(W(y)) = \log(y)$. Now let $y = e^m$.

$$\begin{aligned}
0 &= W(e^m) + \log(W(e^m)) - m \\
&= \frac{z}{z/W(e^m)} + \log(W(e^m)) - m + \log z - \log z \\
&= \frac{z}{z/W(e^m)} + \log\left(\frac{W(e^m)}{z}\right) - m + \log z \qquad (29)
\end{aligned}$$

Now, let $m = 1 - \frac{\lambda}{a} + \log z - \log\theta_i$. Substituting in for m and continuing, we have:

$$
\begin{aligned}
0 &= \frac{z}{z/W\left(e^{1-\frac{\lambda}{a}+\log z-\log\theta_i}\right)} + \log\left(\frac{W\left(e^{1-\frac{\lambda}{a}+\log z-\log\theta_i}\right)}{z}\right) - 1 + \frac{\lambda}{a} + \log\theta_i \\
&= \frac{z}{z/W\left(\frac{z}{\theta_i}e^{1-\frac{\lambda}{a}}\right)} + \log\left(\frac{W\left(\frac{z}{\theta_i}e^{1-\frac{\lambda}{a}}\right)}{z}\right) - 1 + \frac{\lambda}{a} + \log\theta_i \\
&= \frac{z}{z/W\left(\frac{z}{\theta_i}e^{1-\frac{\lambda}{a}}\right)} - \log\left(\frac{z}{\theta_i\, W\left(\frac{z}{\theta_i}e^{1-\frac{\lambda}{a}}\right)}\right) - 1 + \frac{\lambda}{a} \\
&= \frac{\frac{\omega_i}{a}}{\frac{\omega_i}{a}/W\left(\frac{\omega_i}{a\theta_i}e^{1-\frac{\lambda}{a}}\right)} - \log\left(\frac{\frac{\omega_i}{a}}{\theta_i\, W\left(\frac{\omega_i}{a\theta_i}e^{1-\frac{\lambda}{a}}\right)}\right) - 1 + \frac{\lambda}{a} \qquad (30)
\end{aligned}
$$

Where we have let $z = \frac{\omega_i}{a}$. This implies that:

$$
\tilde{\theta}_i = \frac{\omega_i/a}{W\left(\frac{\omega_i}{a\theta_i}e^{1-\frac{\lambda}{a}}\right)} \qquad (31)
$$

Substituting (31) into (30), we get:

$$
0 = \frac{\omega_i/a}{\tilde{\theta}_i} - \log\left(\frac{\tilde{\theta}_i}{\theta_i}\right) - 1 + \frac{\lambda}{a} \qquad (32)
$$

Which is the derivative of the log-posterior (27) divided by the constant a. Multiplying by a and returning the subscripts to λ:

$$
\begin{aligned}
0 &= a\left[\frac{\omega_i/a}{\tilde{\theta}_i} - \log\left(\frac{\tilde{\theta}_i}{\theta_i}\right) - 1 + \frac{\lambda}{a}\right] \\
&= \frac{\omega_i}{\tilde{\theta}_i} - a\log\left(\frac{\tilde{\theta}_i}{\theta_i}\right) - a + \lambda \\
&= \frac{\omega_i}{\tilde{\theta}_i} - \frac{\hat{n}_{q^i}(\theta)}{\eta}\log\left(\frac{\tilde{\theta}_i}{\theta_i}\right) - \frac{\hat{n}_{q^i}(\theta)}{\eta} + \lambda_{\tilde{\theta}^i} \qquad (33)
\end{aligned}
$$

We have arrived back at (26) and therefore derived an expression for $\tilde{\theta}_i$. Now we can substitute back in for ω_i and a:

$$\begin{aligned}\tilde{\theta}_i &= \frac{\omega_i/a}{W\left(\frac{\omega_i}{a\theta_i}e^{1-\frac{\lambda}{a}}\right)} \\ &= \frac{\frac{\eta\sum_{X\in\Omega}\hat{n}_{\theta i}(X|\theta)}{|\Omega|\hat{n}_{q^i}(\theta)}}{W\left(\frac{\eta\sum_{X\in\Omega}\hat{n}_{\theta i}(X|\theta)}{\theta_i|\Omega|\hat{n}_{q^i}(\theta)}\exp(1-\frac{\eta\lambda}{\hat{n}_{q^i}(\theta)})\right)}\end{aligned} \tag{34}$$

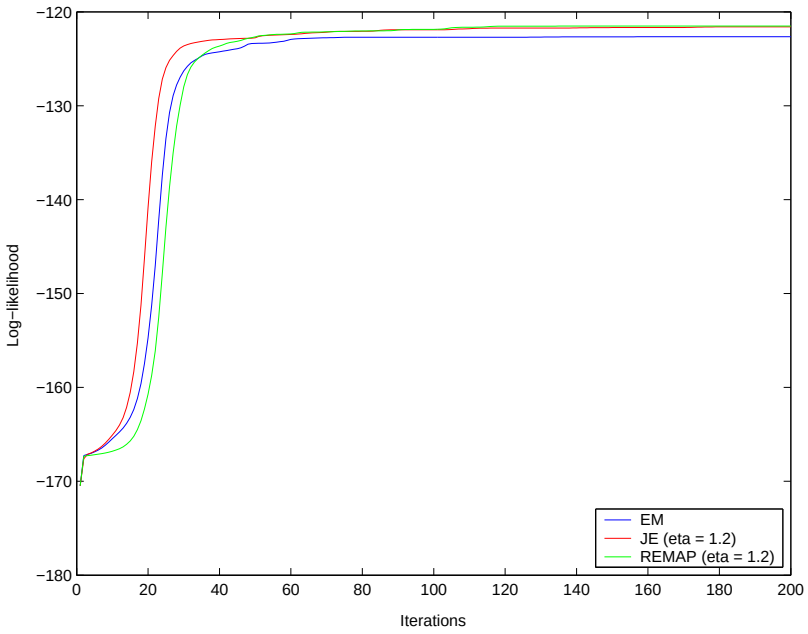

Figure 2: Log-likelihoods of the three different updates training on data generated by a sparse model (approx. 60% zeroed-out parameters). Both the data-generating model and the training models used 10 states and 10 observations symbols. The models were initialized randomly and then trained on 10 examples of 50 observations each.

4 Experiments

In order to test how well our new algorithms performed, we devised three sets of experiments. The first made use of synthetic data generated by a

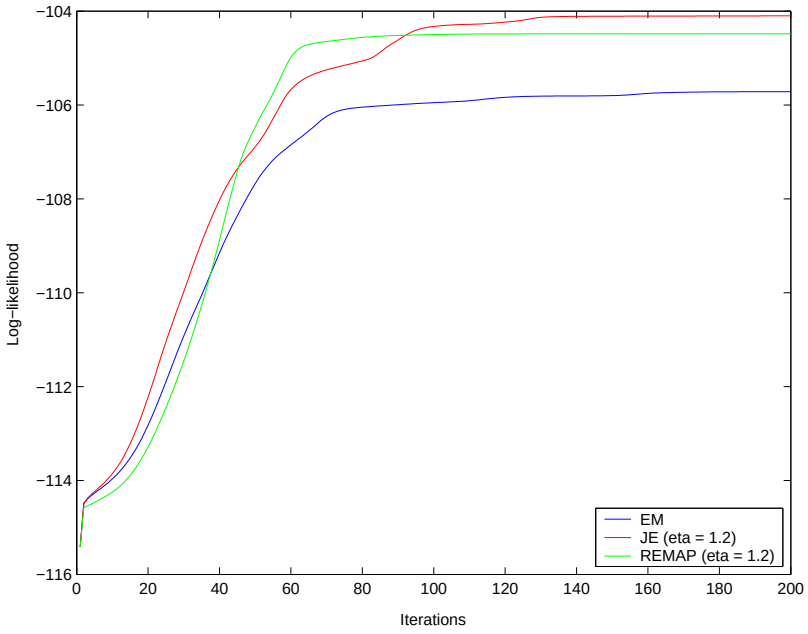

Figure 3: Log-likelihoods of the three different updates training on data generated by a dense model (no zeroed-out parameters). Both the data-generating model and the training models used 10 states and 10 observations symbols. The models were initialized randomly and then trained on 10 examples of 50 observations each.

sparsely parameterized HMM. This source model was generated randomly, but had approximately 60% of its parameters set to 0. The second set of experiments also used synthetically generated data from a randomly parameterized HMM, but this time we allowed the model to remain dense. Finally, we also conducted a set of experiments using speech data from the TIMIT data set. In all three cases, we compared the batch version of the Joint-Entropy update, the REMAP estimator, and EM. The JEMAP estimator was left out of the experiments as we had difficulty getting stable updates from it.

In figure 2 we see that in the sparse-model experiment, EM clearly performed the worst. The REMAP estimator and the Joint-Entropy update performed somewhat similarly. Although the REMAP estimator achieved a slightly higher log-likelihood, the Joint-Entropy update was able to plateau somewhat sooner than either REMAP or EM.

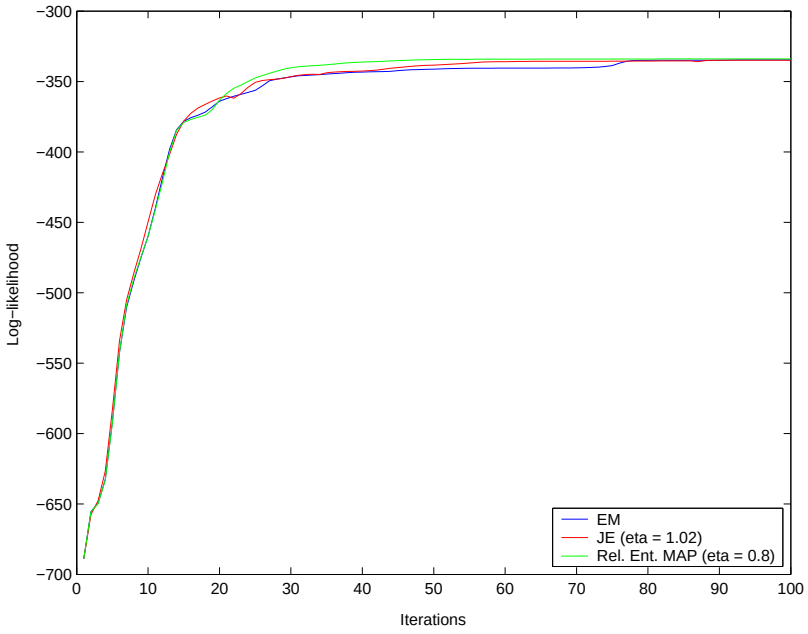

Figure 4: Log-likelihoods of the three different updates training on TIMIT speech data. The data was composed of 20 examples of a male speaker saying the word 'one'. Each utterance contained roughly 200 observations. 12 states and 32 observations symbols which represented discretized cepstral coefficients, were used. The training models were initialized randomly.

Figure 3 shows the results from the dense-model experiment. Here we can see that, while the REMAP estimator achieved a high log-likelihood value fairly quickly, it hit a local minima which prevented it from doing as well as the Joint-Entropy update. Both updates were clearly superior to EM.

For the speech-data experiment, we found that the Joint-Entropy and REMAP updates were somewhat more sensitive to their η values than in the synthetic-data experiments. Interestingly, the REMAP estimator seemed most stable with $\eta < 1$, while the Joint-Entropy update became unstable with $\eta > 1.01$. Figure 4 shows the results from the speech-data experiment. We can see that, although all three updates performed somewhat similarly, the REMAP update was slightly better than the other two.

Finally, it should be noted, that evaluating the W function required some overhead in our setup (Matlab uses the Maple kernel to evaluate it). While

a more efficient implementation would have certainly made a difference, this computational overhead should be taken into consideration when evaluating the results of the updates.

5 Conclusion and Ideas for Future Work

We have derived several entropy-based updates for hidden Markov models and their constituent distributions. The Joint-Entropy update [3] seeks to find the best set of model parameters subject to the constraint that they stay close to the current set. This update is with respect to the entire HMM and approximates the log-likelihood of the new parameters with a first-order Taylor expansion around that of the current set. The model divergence is approximated using usage statistics from the current parameter settings.

The relative-entropy MAP estimator (REMAP) for multinomials uses the expected complete-data log-likelihood to form an evidence term which, in conjunction with the model divergence, can then be used to derive an update using the Lambert W function. This update can be used with HMMs by updating each multinomial distribution in the model separately.

The Joint-Entropy MAP estimator (JEMAP) combines the evidence term formed from the expected complete-data log-likelihood, with the approximate model divergence, to form a maximum a posteriori update with respect to the complete HMM parameter set.

It should be a fairly easy matter to extend either the relative-entropy MAP estimator or the full Joint-Entropy MAP estimator by adding additional constraints. One idea is to add a weighted negative-entropy term, $-\alpha H(\tilde{\theta})$, to the log-posterior such that we constrain the updated parameters to be both sparse and close to the previous set. This would effectively fuse our current work with that of Brand [1].

As the focus of this paper has been on updates which converge quickly and to a high log-likelihood value, no effort was made to compare other properties of the HMMs produced by the updates, such as generalization. This would be an interesting area to explore in the future. Another open ended question, is how to best set the weight parameter(s). Working out connections between variations in the tradeoff parameter and annealing techniques, is another topic of future research.

References

[1] Matt Brand. *Structure learning in conditional probability models via an entropic prior and parameter extinction.* Technical Report, MERL, 1998.

[2] Lawrence R. Rabiner. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* IEEE Proceedings, 1989.

[3] Yoram Singer and Manfred K. Warmuth. *Training Algorithms for Hidden Markov Models Using Entropy Based Distance Functions.* 1996.

[4] Yoram Singer and Manfred K. Warmuth. *Batch and On-line Parameter Estimation of Gaussian Mixtures Based on the Joint Entropy.* Advances in Neural Information Processing Systems 11, 1998.