Ensemble Learning for Hidden Markov Models

David J.C. MacKay
Cavendish Laboratory
Cambridge CB3 0HE. UK
mackay@mrao.cam.ac.uk

Abstract

The standard method for training Hidden Markov Models optimizes a point estimate of the model parameters. This estimate, which can be viewed as the maximum of a posterior probability density over the model parameters, may be susceptible to overfitting, and contains no indication of parameter uncertainty. Also, this maximum may be unrepresentative of the posterior probability distribution. In this paper we study a method in which we optimize an *ensemble* which approximates the entire posterior probability distribution. The ensemble learning algorithm requires the same resources as the traditional Baum–Welch algorithm.

The traditional training algorithm for hidden Markov models is an expectation–maximization (EM) algorithm (Dempster *et al.* 1977) known as the Baum–Welch algorithm. It is a maximum likelihood method, or, with a simple modification, a penalized maximum likelihood method, which can be viewed as maximizing a posterior probability density over the model parameters.

Recently, Hinton and van Camp (1993) developed a technique known as ensemble learning (see also MacKay (1995) for a review). Whereas maximum a posteriori methods optimize a point estimate of the parameters, in ensemble learning an *ensemble* is optimized, so that it approximates the entire posterior probability distribution over the parameters. The objective function that is optimized is a variational free energy (Feynman 1972) which measures the relative entropy between the approximating ensemble and the true distribution. In this paper we derive and test an ensemble learning algorithm for hidden Markov models, building on Neal

and Hinton's (1993) observation that expectation–maximization algorithms can be viewed as variational free energy minimization methods.

1 The Model

We use similar notation to Rabiner and Juang (1986).

- $S = \{s_1, s_2, ..., s_T\}$: hidden state sequence. $(s \in 1 \ldots I)$
- $X = \{x_1, x_2, ..., x_T\}$: observed sequence. $(x \in 1 \ldots M)$
- $\mathbf{A} = \{a_{ij}\}$, $a_{ij} = P(s_{t+1} = j | s_t = i)$: state transition probability matrix
- $\mathbf{B} = \{b_{im}\}$, $b_{im} = P(x_t = m | s_t = i)$: emission probabilities.
- $\pi = \{\pi_i\}$, $\pi_i = P(s_1 = i)$: initial state distribution.
- $\theta = \{\mathbf{A}, \mathbf{B}, \pi\}$: model's parameters.
- $\mathbf{U} = \{\mathbf{u}^{(A)}, \mathbf{u}^{(B)}, \mathbf{u}^{(\pi)}\}$: hyperparameters which define the prior over θ.

For given parameters θ, the probability of the hidden state sequence and the observed data is

$$P(X, S|\theta) = \left[\prod_{t=1}^{T-1} a_{s_t s_{t+1}}\right] \left[\prod_{t=1}^{T} b_{s_t x_t}\right] \pi_{s_1}. \tag{1}$$

The posterior probability of the hidden variables S given X and θ is given by

$$P(S|X, \theta) = \frac{1}{P(X|\theta)} \left[\prod_{t=1}^{T-1} a_{s_t s_{t+1}}\right] \left[\prod_{t=1}^{T} b_{s_t x_t}\right] \pi_{s_1}, \tag{2}$$

where

$$P(X|\theta) = \sum_S P(X, S|\theta). \tag{3}$$

We assume a prior probability over the parameters θ that is a product of Dirichlet distributions:

$$\text{Dirichlet}\,(\mathbf{p}; \mathbf{u}) \equiv \frac{1}{Z(\mathbf{u})} \prod_{i=1}^{I} p_i^{u_i - 1}\, \delta\left(\textstyle\sum_i p_i - 1\right) \tag{4}$$

where δ is a delta function which ensures $\mathbf{p}$ is normalized, and

$$Z(\mathbf{u}) = \frac{\prod_{i=1}^{I} \Gamma(u_i)}{\Gamma(u)}. \tag{5}$$

Here we have defined $u \equiv \sum u_i$. All the hyperparameters u_i are positive, with larger values of $\mathbf{u}$ corresponding to stronger priors. We set the prior on $\mathbf{A}$ to

$$P(\mathbf{A}|\mathbf{u}^{(A)}) = \prod_i \text{Dirichlet}\left(\{a_{i1} \ldots a_{iI}\}; \mathbf{u}^{(A)}\right), \tag{6}$$

with similar priors on $\mathbf{B}$ and π.

The reason for choosing these priors is that they give a direct correspondence between the standard penalized maximum likelihood method in which 'initial counts' or 'offsets' u_i are placed in the bins of the Baum–Welch algorithm and a maximum a posteriori method, if the posterior density is maximized in the 'softmax' basis (MacKay 1996) where each probability vector $\mathbf{p}$ is represented by parameters $\mathbf{a}$ such that

$$p_i(\mathbf{a}) = e^{a_i} / \sum_{i'} e^{a_{i'}} \ . \tag{7}$$

1.1 The standard Baum–Welch optimization

The Baum–Welch algorithm (with penalty terms $\mathbf{U}$) is an iterative algorithm that increases $P(\theta|X)$ at each iteration until a maximum is reached. In each iteration, a forward–backward step computes the probabilities of state sequences S conditioned on the current parameters θ, then an M–step updates the parameters θ.

The forward and backward probabilities $\alpha^{(t)}$ and $\beta^{(t)}$ are given by

$$\begin{array}{rclcrcl} \alpha_i^{(1)} & = & \pi_i b_{ix_1} & & \beta_i^{(T)} & = & b_{ix_T} \\ \alpha_j^{(t+1)} & = & \left[\sum_{i=1}^N \alpha_i^{(t)} a_{ij}\right] b_{jx_{t+1}} & & \beta_i^{(t)} & = & b_{ix_t}\left[\sum_{j=1}^N a_{ij}\beta_j^{(t+1)}\right] \end{array} \tag{8}$$

The M–step is expressed in terms of $n_{ij}^{(t)}$, the posterior probability that there was a transition between state i and state j at timestep t given X and θ,

$$n_{ij}^{(t)} = \sum_S P(S|X,\theta)\,\delta(s_t = i, s_{t+1} = j) \tag{9}$$

$$= \frac{1}{Z_n}\alpha_i^{(t)} a_{ij} \beta_j^{(t+1)} \tag{10}$$

where Z_n is a normalizing constant such that $\sum_{i,j=1}^I n_{ij}^{(t)} = 1$. Then the M–step is

$$a'_{ij} = \frac{u_j^{(A)} + \left[\sum_{t=1}^{T-1} n_{ij}^{(t)}\right]}{\sum_{j'=1}^I \left\{u_{j'}^{(A)} + \left[\sum_{t=1}^{T-1} n_{ij'}^{(t)}\right]\right\}}. \tag{11}$$

There are similar expressions for the updates of $\mathbf{B}$ and π.

2 Ensemble Learning for HMMs

The posterior distribution of the parameters $\theta = \{\mathbf{A}, \mathbf{B}, \pi\}$ and hidden state sequence S given an observation sequence X and fixed hyperparameters, $\mathbf{U}$, $P(S, \theta|X, \mathbf{U})$, is to be approximated by an ensemble $Q(S, \theta)$. We will constrain our approximating distribution to be separable, such that

$$Q(S, \theta) = Q_S(S) Q_A(\mathbf{A}) Q_B(\mathbf{B}) Q_\pi(\pi). \tag{12}$$

We will make no further constraining assumptions about the functional forms of the constituent distributions $Q_S(S)$, $Q_A(\mathbf{A})$ $Q_B(\mathbf{B})$, $Q_\pi(\pi)$.

Term	S	$\mathbf{A}$	$\mathbf{B}$	π	$\mathbf{U}$
$\sum_{i=1}^{N}\sum_{j=1}^{N}(u_{a_{ij}}-1)\log a_{ij}$		*			*
$\sum_{i=1}^{N}\sum_{j=1}^{M}(u_{b_{ij}}-1)\log b_{ij}$			*		*
$\sum_{i=1}^{N}(u_{\pi_i}-1)\log \pi_i$				*	*
$\sum_{t=1}^{T-1}\log a_{s_t s_{t+1}}$	*	*			
$\sum_{t=1}^{T}\log b_{s_t x_t}$	*		*		
$\log \pi_{s_1}$	*			*	

Table 1: Dependencies of the terms of equation (14).

To measure the closeness of the ensemble to the posterior, we define the free energy $F(Q(S,\theta))$:

$$F(Q) = -\int_{\mathbf{A}}\int_{\mathbf{B}}\int_{\pi}\sum_{S} Q(S,\theta)\log\left[\frac{P(X,S,\theta|\mathbf{U})}{Q(S,\theta)}\right]. \tag{13}$$

Our iterative strategy for optimizing Q is sequentially to optimize each of Q_A, Q_B, Q_π and Q_S, while keeping the other three distributions fixed. Looking ahead, it will turn out that the optimized distributions Q_A, Q_B and Q_π are all Dirichlet distributions, and the optimized distribution Q_S is a distribution similar to the posterior distribution in equation (2), so that these optimizations can be performed with the same computational resources as the Baum–Welch algorithm. Because $F(Q)$ is bounded below (by $-\log P(X|\mathbf{U})$), and each individual minimization decreases F, the ensemble learning algorithm is guaranteed to converge.

We first dissect the log–probability appearing in the free energy and note its dependence on the parameters and hyperparameters $S, \mathbf{A}, \mathbf{B}, \pi, \mathbf{U}$ in table 1.

$$\log P(X,S,\theta|\mathbf{U}) = \sum_{i,j=1}^{I}(u_j^{(A)}-1)\log a_{ij} + \sum_{i=1}^{I}\sum_{m=1}^{M}(u_m^{(B)}-1)\log b_{im}$$
$$+\sum_{i=1}^{I}(u_i^{(\pi)}-1)\log\pi_i + \sum_{t=1}^{T-1}\log a_{s_t s_{t+1}} + \sum_{t=1}^{T}\log b_{s_t x_t} + \log\pi_{s_1} + \text{const.} \tag{14}$$

We now derive the optimization steps for F over Q_A, Q_B, Q_π and Q_S respectively.

2.1 Optimization of Q_A

As a functional of Q_A, with Q_B, Q_π, Q_S fixed, $F(Q)$ can be expressed as follows:

$$F(Q_A) = -\int_{\mathbf{A}} Q_A(\mathbf{A})\left[-\sum_{i,j=1}^{I}(u_j^{(A)}-1)\log a_{ij} + \sum_{S} Q_S(S)\sum_{t=1}^{T-1}\log a_{s_t s_{t+1}}\right.$$
$$\left. -\log Q_A(\mathbf{A})\right] + \text{const.} \tag{15}$$

Now, defining a quantity w like n in the Baum–Welch algorithm,

$$w_{ij}^{(t)} = \sum_S Q_S(S)\,\delta(s_t = i, s_{t+1} = j), \tag{16}$$

we have

$$F(Q_A) \quad = \quad \int_{\mathbf{A}} Q_A(\mathbf{A}) \log \left[\frac{Q_A(\mathbf{A})}{\prod_{i,j} a_{ij}^{[W_{ij}-1]}} \right] + \text{const.}, \tag{17}$$

where

$$W_{ij} \equiv \sum_{t=1}^{T-1} w_{ij}^{(t)} + u_j^{(A)} \tag{18}$$

Now, by Gibbs's inequality, the expression $\int_x Q(x) \log \frac{Q(x)}{P^*(x)}$ is minimized with respect to $Q(x)$ by $Q(x) = \frac{1}{Z} P^*(x)$ where Z is the appropriate normalizing constant. So to minimize $F_A(Q_A))$, we choose the distribution Q_A to be a product of Dirichlet distributions:

$$Q_A(\mathbf{A}) = \prod_i \text{Dirichlet}\left(\{a_{ij}\}_{j=1}^I ; \{W_{ij}\}_{j=1}^I \right) \tag{19}$$

Similarly, the optimal distributions Q_B and Q_π are products of Dirichlet distributions defined in terms of

$$w_{im}^{(t)} = \sum_S Q_S(S)\,\delta(s_t = i, x_t = m) \;\text{ and }\; w_i^\pi = \sum_S Q_S(S)\,\delta(s_1 = i). \tag{20}$$

2.2 Optimization of $Q_S(S)$

As a functional of $Q_S(S)$, with Q_A, Q_B, Q_π fixed, $F(Q)$ can be expressed as follows:

$$\begin{aligned} F(Q_S(S)) \quad = \quad -\sum_S Q_S(S) \Bigg[& \int_{\mathbf{A}} Q_A(\mathbf{A}) \sum_{t=1}^{T-1} \log a_{s_t s_{t+1}} + \int_{\mathbf{B}} Q_B(\mathbf{B}) \sum_{t=1}^{T} \log b_{s_t x_t} \\ & + \int_\pi Q_\pi(\pi) \log \pi_{s_1} - \log Q_S(S) \Bigg] + \text{const.} \end{aligned} \tag{21}$$

Now, defining

$$a_{ij}^\star \equiv \exp\left[\int_{\mathbf{A}} Q_A(\mathbf{A}) \log a_{ij} \right], \quad b_{ik}^\star \equiv \exp\left[\textstyle\int_{\mathbf{B}} Q_B(\mathbf{B}) \log b_{ik} \right], \tag{22}$$

and $\pi_i^\star \equiv \exp\left[\int_\pi Q_\pi(\pi) \log \pi_i \right]$, we can write

$$F(Q_S(S)) \quad = \quad \sum_S Q_S(S) \log \left[\frac{Q_S(S)}{\left[\prod_{t=1}^{T-1} a_{s_t s_{t+1}}^\star \right] \left[\prod_{t=1}^{T} b_{s_t x_t}^\star \right] \pi_{s_1}^\star} \right] + \text{const.} \tag{23}$$

The optimal distribution $Q_S(S)$ which minimizes $F(Q_S(S))$ is thus:

$$Q_S(S) \quad = \quad \frac{1}{Z_S} \left[\prod_{t=1}^{T-1} a_{s_t s_{t+1}}^\star \right] \left[\prod_{t=1}^{T} b_{s_t x_t}^\star \right] \pi_{s_1}^\star \tag{24}$$

where Z_S is a normalizing constant. Note the resemblance between this distribution and the posterior distribution $P(S|X,\theta)$ in equation (2). The only difference is that here $a^\star$, $b^\star$ and $\pi^\star$ are not normalized probability vectors. In fact, since they are the geometric means of a_{ij}, b_{ik} and π_i under the distribution Q, they are subnormalized, *i.e.*, satisfy $\sum_i p_i^\star \leq 1$.

The circle is now complete. If, as in section 2.1, Q_A and Q_B have been set to products of Dirichlet distributions, and Q_π is a Dirichlet distribution (equation (19)) we can obtain $a^\star$, $b^\star$ and $\pi^\star$ using

$$\int_{\mathbf{p}} \text{Dirichlet}\,(\mathbf{p};\mathbf{u}) \log p_i = \psi\,(u_i) - \psi\,(u) \tag{25}$$

where

$$\psi(x) = \frac{\partial}{\partial x} \log \Gamma(x) \quad \text{and} \quad u = \sum_j u_j. \tag{26}$$

Then we can calculate the relevant properties of the optimal distribution $Q_S(S)$, the quantities w, using the forward–backward algorithm, just as n is obtained in equations (8) and (10). (The forward–backward algorithm is not affected by the fact that $a^\star$ etc. are subnormalized.)

Ensemble learning is thus a computationally inexpensive modification of the Baum–Welch algorithm.

3 Work in progress

The ensemble learning method is currently being applied to toy problems with hidden Markov models. We train two hidden Markov models as models of two distinct sources (for example, forwards English text and backwards English text) and then test the discriminative performance of these models on unseen test data from the two sources. We compare the results when both models are trained using the traditional Baum–Welch algorithm, and when they are trained by ensemble learning.

Work has still to be done on the following issues.

1. The question of how to get predictions from an optimized ensemble: a simple approach is to extract a single representative HMM by selecting the mean value of the parameters.
2. The option of simultaneous optimization of hyperparameters, following MacKay and Peto (1995): for simplicity, our initial investigations are using fixed hyperparameters.

4 Discussion

Will ensemble learning be a useful method for training hidden Markov models? The hope is that since it takes into account parameter uncertainty during optimization, there may be cases (for example, data–poor problems) where the optimized ensemble

gives a better representation of the posterior distribution than the mode of the posterior.

We can note that the standard penalized maximum likelihood method is a special case of ensemble learning in which we constrain the approximating distribution to be a product of a distribution $Q_S(S)$ and a product of adaptable delta functions,

$$Q^{\mathrm{MAP}}(S,\theta) \quad = \quad Q_S(S)\,\delta\Big(\theta - \hat{\theta}\Big)\,. \tag{27}$$

Acknowledgements

DJCM thanks Radford Neal, Geoff Hinton and Katriona Macphee for helpful discussions.

References

DEMPSTER, A., LAIRD, N., and RUBIN, D. (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B* **39**: 1–38.

FEYNMAN, R. P. (1972) *Statistical Mechanics*. W. A. Benjamin, Inc.

HINTON, G. E., and VAN CAMP, D. (1993) Keeping neural networks simple by minimizing the description length of the weights. In *Proc. 6th Annu. Workshop on Comput. Learning Theory*, pp. 5–13. ACM Press, New York, NY.

MACKAY, D. J. C. (1995) Developments in probabilistic modelling with neural networks—ensemble learning. In *Neural Networks: Artificial Intelligence and Industrial Applications. Proceedings of the 3rd Annual Symposium on Neural Networks, Nijmegen, Netherlands, 14-15 September 1995*, pp. 191–198, Berlin. Springer.

MACKAY, D. J. C., (1996) Choice of basis for Laplace approximation. Submitted to *Machine Learning*.

MACKAY, D. J. C., and PETO, L. (1995) A hierarchical Dirichlet language model. *Natural Language Engineering* **1** (3): 1–19.

NEAL, R. M., and HINTON, G. E. (1993) A new view of the EM algorithm that justifies incremental and other variants. *Biometrika*. submitted.

RABINER, L. R., and JUANG, B. H. (1986) An introduction to hidden Markov models. *IEEE ASSP Magazine* pp. 4–16.