Hidden Markov models with extended observation densities

William D. Penny and Stephen J. Roberts

Technical Report,
Neural Systems Research Group,
Department of Electrical and Electronic Engineering,
Imperial College of Science, Technology and Medicine,
London SW7 2BT., U.K.
w.penny@ic.ac.uk, s.j.roberts@ic.ac.uk

October 6, 1998

Abstract

The standard Hidden Markov Model (HMM), as employed for example in speech recognition, assumes that each state generates observations according to a Gaussian or Gaussian Mixture model. In this paper we consider HMMs in which the observations are generated according to a number of different models; (i) autoregressive, (ii) multivariate autoregressive, (iii) principal components, (iv) independent components. We show how learning can be achieved using EM algorithms. The models perform non-stationary spectral and cross-spectral analysis and non-stationary factor analysis.

1 Introduction

The standard Hidden Markov Model (HMM), as employed for example in speech recognition [5], assumes that each state generates observations according to a Gaussian or Gaussian Mixture model. In this paper we consider HMMs for continuous observations where the observations are generated according to a number of different models; (i) autoregressive, (ii) multivariate autoregressive and (iii) principal components.

We consider HMMs with N discrete hidden states, S_i, which are equal to q_t at time t. The initial state distribution is

$$\pi_j = P(q_1 = S_j) \tag{1}$$

State transition probabilities are defined by the matrix $\boldsymbol{A}$ having entries

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \tag{2}$$

The probability of observing the vector $\boldsymbol{y}_t$, given that the model is in state j, is defined as

$$b_j(\boldsymbol{y}_t) = P(\boldsymbol{y}_t | q_t = S_j) \tag{3}$$

The parameters of the observation model (which we will define later) are concatenated into the parameter set $\boldsymbol{B}$. The parameters of the HMM, as a whole, are concatenated into the parameter set $\boldsymbol{\theta} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$.

To train an HMM it is necessary to be able to calculate the probability of being in state S_i at time t given a particular observation sequence

$$\gamma_t(i) = P(q_t = S_i | \boldsymbol{y}_1^T) \tag{4}$$

It is also necessary to calculate the probability of state S_j succeding state S_i given a particular observation sequence

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | \boldsymbol{y}_1^T) \tag{5}$$

The two variables $\gamma_t(i)$ and $\xi_t(i,j)$ can be calculated using the forward-backward algorithm described in Appendix B. The HMM can then be trained using an EM algorithm. After the models have been trained, they can be applied to novel data sets and the most likely sequence of hidden states calculated using the Viterbi algorithm (see Appendix C).

2 The EM algorithm for HMMs

In an HMM parameterised by some vector $\hat{\boldsymbol{\theta}}$ the observed variable y_t depends *only* on the current state q_t (and not previous states) and the current state depends *only* on the previous state (and not states before that). This allows the joint likelihood of an observation sequence $\boldsymbol{y} = [y_1, y_t, ..., y_T]$ and hidden variable (state) sequence $\boldsymbol{q} = [q_1, q_2, ..., q_T]$ to be written as

$$p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{q}, \boldsymbol{y}) = p_{\hat{\boldsymbol{\theta}}}(q_1) \prod_{t=2}^{T} p_{\hat{\boldsymbol{\theta}}}(q_t \mid q_{t-1}) \prod_{t=1}^{T} p_{\hat{\boldsymbol{\theta}}}(y_t \mid q_t) \tag{6}$$

The joint log likelihood may be written

$$\log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{q}, \boldsymbol{y}) = \log p_{\hat{\boldsymbol{\theta}}}(q_1) + \sum_{t=2}^{T} \log p_{\hat{\boldsymbol{\theta}}}(q_t \mid q_{t-1}) + \sum_{t=1}^{T} \log p_{\hat{\boldsymbol{\theta}}}(y_t \mid q_t) \tag{7}$$

The EM algorithm (see Appendix A) requires us to maximise, $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$, the expectation of the joint log likelihood where the expectation is taken relative to the old distribution of hidden (state) variables, $p_{\boldsymbol{\theta}}(\boldsymbol{q}|\boldsymbol{y})$. Hence,

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \sum_{\boldsymbol{q}} p_{\boldsymbol{\theta}}(\boldsymbol{q}|\boldsymbol{y}) \log p_{\hat{\boldsymbol{\theta}}}(q_1) + \sum_{\boldsymbol{q}} p_{\boldsymbol{\theta}}(\boldsymbol{q}|\boldsymbol{y}) \sum_{t=2}^{T} \log p_{\hat{\boldsymbol{\theta}}}(q_t \mid q_{t-1}) + \sum_{\boldsymbol{q}} p_{\boldsymbol{\theta}}(\boldsymbol{q}|\boldsymbol{y}) \sum_{t=1}^{T} \log p_{\hat{\boldsymbol{\theta}}}(y_t \mid q_t) \tag{8}$$

where $\sum_{\boldsymbol{q}}$ denotes a sum over all possible hidden state sequences. The above equation consists of three terms; the first for the initial probability parameters, π_i, the second for the state transition probability paramaters, $\boldsymbol{A}$ and the third for the observation model parameters $\boldsymbol{B}$

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = Q(\pi, \hat{\pi}) + Q(\boldsymbol{A}, \hat{\boldsymbol{A}}) + Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) \tag{9}$$

These can be maximised separately giving rise to parameter update equations for each different part of the model. The old distribution of hidden state variables, $p_{\boldsymbol{\theta}}(\boldsymbol{q}|\boldsymbol{y})$, can be calculated from the forward-backward algorithm.

2.1 Initial state probabilities

An update formula for the initial state distribution parameters, π_i, can be found by maximising

$$Q(\pi, \hat{\pi}) = \sum_{\boldsymbol{q}} p_{\boldsymbol{\theta}}(\boldsymbol{q}|\boldsymbol{y}) \log p_{\hat{\boldsymbol{\theta}}}(q_1) \tag{10}$$

This sum can be re-arranged as

$$Q(\pi, \hat{\pi}) = \sum_{i}^{N} p_{\boldsymbol{\theta}}(q_1 = i|\boldsymbol{y}) \log p_{\hat{\boldsymbol{\theta}}}(i) \tag{11}$$

Using the notation introduced earlier this gives

$$Q(\pi, \hat{\pi}) = \sum_{i=1}^{N} \gamma_1(i) \log \hat{\pi}_i \tag{12}$$

By maximising this subject to the constraint that $\sum_i \pi_i = 1$ and noting also that $\sum_i \gamma_1(i) = 1$ we obtain the solution

$$\hat{\pi}_i = \gamma_1(i) \tag{13}$$

The $\gamma_1(i)$ values are calculated using the old $\boldsymbol{\theta}$ values (as part of the E-step) using the forward-backward algorithm and the new π_i values are calculated using the above update formula (as part of the M-step).

2.2 State transition probabilities

An update formula for the state transition matrix $\boldsymbol{A}$, having elements a_{ij}, can be found by maximising

$$Q(\boldsymbol{A}, \hat{\boldsymbol{A}}) = \sum_{\boldsymbol{q}} p_{\boldsymbol{\theta}}(\boldsymbol{q}|\boldsymbol{y}) \sum_{t=2}^{T} \log p_{\hat{\boldsymbol{\theta}}}(q_t \mid q_{t-1}) \tag{14}$$

This sum can be re-arranged as

$$Q(\boldsymbol{A}, \hat{\boldsymbol{A}}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=2}^{T} p_{\boldsymbol{\theta}}(q_t = j, q_{t-1} = i|\boldsymbol{y}) \log p_{\hat{\boldsymbol{\theta}}}(q_t \mid q_{t-1}) \tag{15}$$

Writing this in more familiar notation gives

$$Q(\boldsymbol{A}, \hat{\boldsymbol{A}}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\sum_{t} \xi_t(i, j) \right) \log(a_{ij}) \tag{16}$$

Maximising this subject to the necessary constraints on the probabilities gives the update formula

$$a_{ij} = \frac{\sum_t \xi_t(i, j)}{\sum_t \gamma_t(i)} \tag{17}$$

3 Observation models

An update formula for the observation model parameters can be found by maximising

$$Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) = \sum_{\boldsymbol{q}} p_{\boldsymbol{\theta}}(\boldsymbol{q}|\boldsymbol{y}) \sum_{t=1}^{T} \log p_{\hat{\boldsymbol{\theta}}}(y_t \mid q_t) \tag{18}$$

This sum can be re-arranged as

$$Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) = \sum_i \sum_t p_{\boldsymbol{\theta}}(q_t = i \mid \boldsymbol{y}) \log p_{\hat{\boldsymbol{\theta}}}(y_t \mid q_t) \tag{19}$$

In principle, the observation model can be anything. In speech recognition, for example, a Gaussian observation model or Gaussian Mixture model is often used. In this paper, we consider a number of alternatives.

3.1 Gaussian observations

For a Gaussian observation model the probability of observing time-series value y_t given that the HMM is in state q_t is given by

$$p_{\hat{\boldsymbol{\theta}}}(y_t \mid q_t) = \frac{1}{\hat{\sigma}_{q_t}\sqrt{2\pi}} \exp\left[\frac{-(y_t - \hat{\mu}_{q_t})^2}{2\hat{\sigma}^2_{q_t}}\right] \tag{20}$$

By substituting this into equation 19 we get

$$Q_{GAUSS} = \sum_i \sum_t \gamma_t(i) \left[-(y_t - \hat{\mu}_i)^2/2\hat{\sigma}_i^2 - \log \hat{\sigma}_i\right] \tag{21}$$

Maximising this with respect to $\hat{\mu}_i$ gives the update formula

$$\hat{\mu}_j = \frac{\sum_t \gamma_t(j) y_t}{\sum_t \gamma_t(j)} \tag{22}$$

Maximising this with respect to $\hat{\sigma}_i$ gives the update formula

$$\hat{\sigma}_j^2 = \frac{\sum_t \gamma_t(j)(y_t - \hat{\mu}_i)^2}{\sum_t \gamma_t(j)} \tag{23}$$

For d time series, a multivariate Gaussian is appropriate

$$p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{y}_t \mid q_t) = \frac{1}{(2\pi)^{d/2} \mid \hat{\boldsymbol{C}}_{\boldsymbol{q_t}} \mid^{-1/2}} \exp[-\frac{1}{2}(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_{q_t})^T \hat{\boldsymbol{C}}_{q_t}(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_{q_t})] \tag{24}$$

The update formulae become

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_t \gamma_t(j) \boldsymbol{y}_t}{\sum_t \gamma_t(j)} \tag{25}$$

and

$$\hat{\boldsymbol{C}}_j = \frac{\sum_t \gamma_t(j)(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_j)(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_j)^T}{\sum_t \gamma_t(j)} \tag{26}$$

3.2 Autoregressive observations

An autoregressive model is a linear predictor used for modelling time series. It is a time-series version of linear regression. An $AR(p)$ model predicts the next value in a time series as a linear combination of the p previous values

$$y_t = -\sum_i^p a_i y_{t-i} + e_t \tag{27}$$

where a_i are the AR coefficients and e_t is a IID Gaussian noise with zero mean and variance σ^2. The probability of observing time-series value y_t given that the HMM is in state q_t is given by

$$p_{\hat{\boldsymbol{\theta}}}(y_t \mid q_t) = \frac{1}{\hat{\sigma}_{q_t}\sqrt{2\pi}} \exp\left[\frac{-(y_t - \hat{\mu}_{q_t} - \sum_k^p \hat{a}_{q_t,k} y_{t-k})^2}{2\hat{\sigma}_{q_t}^2}\right] \tag{28}$$

By substituting this into equation 19 we get

$$Q_{AR} = \sum_i \sum_t \gamma_t(i) \left[-(y_t - \hat{\mu}_i - \sum_k^p \hat{a}_{i,k} y_{t-k})^2 / 2\hat{\sigma}_i^2 - \log \hat{\sigma}_i\right] \tag{29}$$

For each state the AR coefficients and the mean value can be found by minimising

$$E_{AR} = \sum_t \gamma_t(i)(y_t - \hat{\mu}_i - \sum_k^p \hat{a}_{i,k} y_{t-k})^2 \tag{30}$$

Minimisation of this function is a *weighted* least squares problem and is readily solved by Singular Value Decomposition (SVD) (see Press [4] ; page 662). Each 'weight', $\gamma_t(i)$, is the probability that the HMM is in state i at time t. The variances are then calculated as

$$\hat{\sigma}_j^2 = \frac{\sum_t \gamma_t(j)(y_t - \hat{\mu}_i - \sum_k^p \hat{a}_{i,k} y_{t-k})^2}{\sum_t \gamma_t(j)} \tag{31}$$

3.3 Multivariate autoregressive observations

A multivariate autoregressive (MAR) model is a linear predictor used for modelling multiple time series. An $MAR(p)$ model predicts the next vector value in a d-dimensional time series, $\boldsymbol{y}_t$ as a linear combination of the p previous vector values of the time series

$$\boldsymbol{y}_t = -\sum_i^p \boldsymbol{a}_i \boldsymbol{y}_{t-i} + \boldsymbol{e}_t \tag{32}$$

where each $\boldsymbol{a}_i$ is a $d - by - d$ matrix of AR coefficients and $\boldsymbol{e}_t$ is an IID Gaussian noise vector with zero mean and covariance $\boldsymbol{C}$. Thus, the MAR model has the Gaussian observation model as a special case (where only the noise term is included). The observation density is

$$p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{y}_t \mid q_t) = \frac{1}{(2\pi)^{d/2} \mid \hat{\boldsymbol{C}}_{q_t} \mid^{-1/2}} \exp\left[-\frac{1}{2}\boldsymbol{x}_t(q_t)^T \hat{\boldsymbol{C}}_{q_t} \boldsymbol{x}_t(q_t)\right] \tag{33}$$

where

$$\boldsymbol{x}_t(i) = \left(\boldsymbol{y}_t - \hat{\boldsymbol{\mu}}_i - \sum_k^p \hat{\boldsymbol{a}}_{i,k} \boldsymbol{y}_{t-k}\right) \tag{34}$$

and $\hat{\boldsymbol{a}}_{i,k}$ is the $d - by - d$ matrix of AR coefficients for state i at lag k. The MAR coefficients can then be found by minimising a multidimensional weighted least squares problem. The minimisation process is complicated by the fact that the residuals are correlated. But standard algorithms for estimating MAR coefficients can be used if the weighting factors, $\gamma_t(i)$, are introduced. The covariance matrix is then calculated as

$$\hat{\boldsymbol{C}}_j = \frac{\sum_t \gamma_t(j)\boldsymbol{x}_t(j)\boldsymbol{x}_t(j)^T}{\sum_t \gamma_t(j)} \tag{35}$$

3.4 Principal components observations

Bishop and Tipping [7] derive EM algorithms for a probabilistic version of principal components analysis in which the observed variable is modelled as

$$\boldsymbol{y}_t = \boldsymbol{W}_j \boldsymbol{x}_t + \boldsymbol{\mu}_j + \boldsymbol{e}_t \tag{36}$$

where $\boldsymbol{x}_t$ are the underlying factors, $\boldsymbol{W}_j$ is the factor loading matrix, $\boldsymbol{\mu}_j$ is the mean observed vector and $\boldsymbol{e}_t$ is the observation noise. The variable j indexes the HMM state. The Gaussian observation model is a special case of this model, in which the observations are modelled entirely as noise ($\boldsymbol{W}_j = \boldsymbol{0}$). In principal components analysis the noise is considered to be drawn from a Gaussian with an isotropic covariance matrix $\sigma_j^2 \boldsymbol{I}$. The factor loading matrix is then chosen so as to minimise σ_j^2. In probabilistic principal components analysis the noise covariance matrix is again isotropic but $\boldsymbol{W}_j$ and σ_j^2 are chosen so as to maximise the likelihood of the data. This can be achieved with the following iterative update scheme [7]

$$\hat{\boldsymbol{W}}_j = \boldsymbol{C}_j \boldsymbol{W}_j \left(\sigma_j^2 \boldsymbol{I} + \boldsymbol{M}_j^{-1} \boldsymbol{W}_j^T \boldsymbol{C}_j \boldsymbol{W}_j \right)^{-1} \tag{37}$$

and

$$\hat{\sigma}_j^2 = \frac{1}{d} Tr \left(\boldsymbol{C}_j - \boldsymbol{C}_j \boldsymbol{W}_j \boldsymbol{M}_j^{-1} \hat{\boldsymbol{W}}_j^{T} \right) \tag{38}$$

where

$$\boldsymbol{M}_j = \sigma_j^2 \boldsymbol{I} + \boldsymbol{W}_j^T \boldsymbol{W}_j \tag{39}$$

The matrix $\boldsymbol{C}$ is the sample covariance matrix. For the jth state in an HMM the covariance matrix is given by

$$\boldsymbol{C}_j = \frac{\sum_t \gamma_t(j)(\boldsymbol{y}_t - \boldsymbol{\mu}_j)(\boldsymbol{y}_t - \boldsymbol{\mu}_j)^T}{\sum_t \gamma_t(j)} \tag{40}$$

where

$$\boldsymbol{\mu}_j = \frac{\sum_t \gamma_t(j) \boldsymbol{y}_t}{\sum_t \gamma_t(j)} \tag{41}$$

3.5 Independent components observations

In Independent Components Analysis (ICA) the observed variable is modelled as

$$\boldsymbol{y}_t = \boldsymbol{W}_s \boldsymbol{x}_t + \boldsymbol{\mu}_s + \boldsymbol{e}_t \tag{42}$$

where the underlying factors, $\boldsymbol{x}_t$, are independent of each other. The variable s indexes the HMM state. For the moment, we consider the noiseless ($\boldsymbol{e}_t = \mathbf{0}$), zero mean ($\boldsymbol{\mu}_s = \mathbf{0}$) case. Mackay [3] (equation 11) notes that the log likelihood of data vector y_t is given by

$$\log p_{\hat{\boldsymbol{\theta}}}(y_t \mid q_t) = -\log \mid \boldsymbol{W}_{q_t} \mid + \sum_i \log p_i((\boldsymbol{W}_{q_t}^{-1})_{ij} \boldsymbol{y}_{t,j}) \tag{43}$$

where $\boldsymbol{W}_{q_t}$ refers to the factor loading matrix for HMM state q_t, $()_{ij}$ refers to the i,jth element of a matrix and $p_i(x)$ refers to the univariate density of the ith independent source component (the density of all components is often chosen to be an inverse cosh distribution). By substituting this into equation 19 we get

$$Q_{ICA} = \sum_s Q_{ICA_s} \tag{44}$$

where

$$Q_{ICA_s} = -\gamma(s) \log \mid \boldsymbol{W}_s \mid + \sum_t \gamma_t(s) \left(\sum_i \log p_i((\boldsymbol{W}_{q_t}^{-1})_{ij} \boldsymbol{y}_{t,j}) \right) \tag{45}$$

and

$$\gamma(s) = \sum_t \gamma_t(s) \tag{46}$$

If the sources are modelled as $\boldsymbol{x}_t = \boldsymbol{V}_s \boldsymbol{y}_t$ where $\boldsymbol{V}_s = \boldsymbol{W}_s^{-1}$ then differentiating Q_{ICA_s} w.r.t $\boldsymbol{V}_s$ gives the steepest ascent learning rule

$$(\boldsymbol{V}_s)_{ij} \propto \frac{\partial Q_{ICA_s}}{\partial (\boldsymbol{V}_s)_{ij}} = \gamma(s)(\boldsymbol{V}_s^{-1})_{ji} + \sum_t \gamma_t(s) \boldsymbol{y}_{t,j} z_{i,t} \tag{47}$$

where $a_{i,t} = (\boldsymbol{V}_s)_{ij} \boldsymbol{y}_{t,j}$ and $z_i = \partial \log p_i(a_{i,t}) / \partial a_{i,t}$. This is a batch version of equation 14 in Bell and Sejnowski [1] (they make the further assumption that the density of source components is an inverse cosh) with the addition that each example is weighted by $\gamma_t(s)$, the probability that the ICA model for state s is 'active' at time t.

4 Discussion

In [5] an autoregressive model was used but the observation density referred to the density of a sequence of lagged observations rather than the observation which we are trying to predict. The AR observation model embedded in an HMM as described in this paper is due to [2]. We have extended this model to the case of multivariate autoregressive densities, principal component densities and independent component densities. We owe our approach to ideas from the paper by Roweis and Ghahramani [6].

References

[1] A.J. Bell and T.J. Sejnowski. An information maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1004–1034, 1995.

[2] A.M. Fraser and A. Dimitriadis. Forecasting probability densities by using hidden Markov models with mixed states. In A.S. Weigend and N.A. Gershenfeld, editors, *Time series prediction: forecasting the future and understanding the past*, pages 265–282. Addison-Wesley, 1994.

[3] D.J.C. Mackay. Maximum likelihood and covariant algorithms for independent component analysis. Technical report, Cavendish Laboratory, University of Cambridge, 1996.

[4] W. H. Press, S.A. Teukolsky, W.T. Vetterling, and B.VP. Flannery. *Numerical Recipes in C, second edition.* Cambridge, 1992.

[5] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[6] S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. Technical report, Computation and Neural Systems, California Institute of Technology, 1997.

[7] M. E. Tipping and C. M. Bishop. Probabilistic Principal Component Analysis. Technical report, Neural Computing Research Group, Aston University, 1997.

A The EM algorithm

Given a parameter vector $\boldsymbol{\theta}$, a vector of observations $\boldsymbol{y}$ and a vector of hidden (missing or unknown) variables $\boldsymbol{q}$, the EM algorithm will set the parameters to their maximum likelihood values. The hidden variables may be, for example, the label of the Gaussian which generated each data point in a Gaussian mixture model or the label of the state which generated each data point in a Hidden Markov Model. EM operates as follows:

1. Set parameter vector $\boldsymbol{\theta}$ to initial value.
2. Choose $\hat{\boldsymbol{\theta}}$ to maximise $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \varepsilon_{\boldsymbol{q}|\boldsymbol{y},\boldsymbol{\theta}} \left[\log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{q}, \boldsymbol{y}) \right]$ where ε denotes the expectation operator.
3. Set $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$. If not converged go to 2.

The algorithm works because an increase in the function $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ ensures that the likelihood of the data is increased, ie $p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{y}) > p_{\boldsymbol{\theta}}(\boldsymbol{y})$. This statement can be proved by decomposing the log likelihood using Baye's rule

$$\log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{y}) = \log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{q}, \boldsymbol{y}) - \log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{q} \mid \boldsymbol{y}) \tag{48}$$

and taking expectatations w.r.t the (old) distribution of hidden variables

$$\log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{y}) = \varepsilon_{\boldsymbol{q}|\boldsymbol{y},\boldsymbol{\theta}} \left[\log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{q}, \boldsymbol{y}) \right] - \varepsilon_{\boldsymbol{q}|\boldsymbol{y},\boldsymbol{\theta}} \left[\log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{q} \mid \boldsymbol{y}) \right] \tag{49}$$

(the term of the left-hand side is unaffected as it does not depend on $\boldsymbol{q}$). Now, Jensen's inequality states that for a convex function $f()$ and coefficients λ_j which lie between 0 and 1 and sum to 1

$$f\left(\sum_j \lambda_j x_j\right) \geq \sum_j \lambda_j f(x_j) \tag{50}$$

This can be used to prove the Gibbs/Kullback-Liebler inequality

$$\sum_j p_{\theta_1}(x_j) log \left(\frac{p_{\theta_2}(x_j)}{p_{\theta_1}(x_j)}\right) \leq 0 \tag{51}$$

which can be written as

$$\varepsilon_{\theta_1}[\log p_{\theta_2}(x)] \leq \varepsilon_{\theta_1}[\log p_{\theta_1}(x)] \tag{52}$$

This implies that the second term in equation 49 (including the minus sign) is a minimum when when $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$; thus changing $\boldsymbol{\theta}$ can only increase the second term. Hence, the log-likelihood (and therefore the likelihood) is increased when the first term is increased, that is when $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) \geq Q(\boldsymbol{\theta}, \boldsymbol{\theta})$.

B The forward-backward algorithm

This is an efficient method for calculating the probability of an observation sequence $\boldsymbol{y}_1^T = \boldsymbol{y}_1\boldsymbol{y}_2...\boldsymbol{y}_T$ [5]. The probabilities in this section are all dependent on the model parameters, $\boldsymbol{\theta}$, although this dependence is left out of the notation. The algorithm uses a 'forward variable'

$$\alpha_t(j) = P(\boldsymbol{y}_1^t, q_t = S_j) \tag{53}$$

This variable is calculated via initialisation

$$\alpha_1(j) = \pi_j b_j(\boldsymbol{y}_1) \tag{54}$$

and iteration

$$\alpha_{t+1}(j) = \left[\sum_i^N \alpha_t(i) a_{ij}\right] b_j(\boldsymbol{y}_{t+1}) \tag{55}$$

The probability of the sequence is obtained by the marginalisation

$$P(\boldsymbol{y}_1^T) = \sum_j^N P(\boldsymbol{y}_1^T, q_T = S_j) = \sum_j^N \alpha_T(j) \tag{56}$$

We also introduce the 'backward variable'. This can also be used to calculate the probability of the observation sequence. Its main use, however, is in setting the parameters of the HMM (see section 2.3).

$$\beta_t(j) = P(\boldsymbol{y}_{t+1}^T | q_t = S_j) \tag{57}$$

This variable is calculated via initialisation

$$\beta_T(j) = 1 \tag{58}$$

and iteration

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(\boldsymbol{y}_{t+1}) \beta_{t+1}(j) \tag{59}$$

We now also define some intermediate variables which are used in the paper. The probabilities are all dependent on the model parameters, $\boldsymbol{\theta}$, although this dependence is left out of the notation. We define the variable $\xi_t(i,j)$ as

$$\begin{aligned} \xi_t(i,j) &= P(q_t = S_i, q_{t+1} = S_j | \boldsymbol{y}_1^T) \\ &= \frac{P(q_t = S_i, q_{t+1} = S_j, \boldsymbol{y}_1^T)}{P(\boldsymbol{y}_1^T)} \end{aligned} \tag{60}$$

This is calculated in terms of the forward and backward variables as

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{y}_{t+1}) \beta_{t+1}(j)}{\sum_i \sum_j \alpha_t(i) a_{ij} b_j(\boldsymbol{y}_{t+1}) \beta_{t+1}(j)} \tag{61}$$

We now define the variable $\gamma_t(i)$

$$\gamma_t(i) = P(q_t = S_i | \boldsymbol{y}_1^T) \tag{62}$$

This can be calculated as

$$\gamma_t(i) = \sum_j \xi_t(i,j) \tag{63}$$

In a practical application the forward, backward and intermediate variables must be scaled so as to prevent underflow.

C The Viterbi algorithm

The simplest procedure for finding the hidden state associated with a given observation one is, at each time point, to select the hidden state with the highest probability. That is

$$q_t^* = \max_i P(q_t = S_i) = \max_i \gamma_t(i) \tag{64}$$

We call this single-state decoding. A more widely used method is to select the most likely *sequence* of hidden states, $\boldsymbol{q}_1^T = q_1 q_2 ... q_T$. This is calculated according to the Viterbi algorithm [5] which uses the intermediate variable

$$\delta_t(j) = \max_{\boldsymbol{q}_1^{t-1}} P(\boldsymbol{q}_1^{t-1}, q_t = S_j, \boldsymbol{y}_1^t) \tag{65}$$

We also need a variable to keep a record of the most likely state sequence. This is $\Psi_t(j)$ which stores the pre-cursor state of q_t^* in the most likely sequence q_1^{*T}. The variables are calculated via initialisation

$$\begin{aligned} \delta_1(j) &= \pi_j b_j(\boldsymbol{y}_1) \\ \Psi_t(j) &= 0 \end{aligned} \tag{66}$$

and iteration

$$\begin{aligned} \delta_t(j) &= \max_i \left[\delta_{t-1}(i) a_{ij}\right] b_j(\boldsymbol{y}_t) \\ \Psi_t(j) &= \arg\max_i \left[\delta_{t-1}(i) a_{ij}\right] \end{aligned} \tag{67}$$