HIDDEN MARKOV INDEPENDENT COMPONENTS FOR BIOSIGNAL ANALYSIS

William D. Penny[1], Stephen J. Roberts[1] and Richard M. Everson[2]

1. Department of Engineering Science, University of Oxford, UK.*
2. Department of Computer Science, University of Exeter, UK.

INTRODUCTION

Much recent research in unsupervised learning builds on the idea of using generative models for modelling the probability distribution over a set of observations. These approaches suggest that powerful new data analysis tools may be derived by combining existing models using a probabilistic 'generative' framework. In this paper, we follow this approach and combine hidden Markov models (HMMs), Independent Component Analysis (ICA) and generalised autoregressive models (GAR) into a single generative model for the analysis of nonstationary multivariate time series.

Our motivation for this work derives from our desire to analyse biomedical signals which are known to be highly non-stationary. Moreover, in signals such as the electroencephalogram (EEG), for example, we have a number of sensors (electrodes) which detect signals emanating from a number of cortical sources via an unknown mixing process. This naturally fits an ICA approach which is further enhanced by noting that the sources themselves are characterised by their dynamic content. This leads us to the use of Generalised Autoregressive (GAR) processes to model the sources. The overall generative model is shown in Figure 1.

*Email: wpenny@robots.ox.ac.uk, Fax: +44 (0)1865 273908.

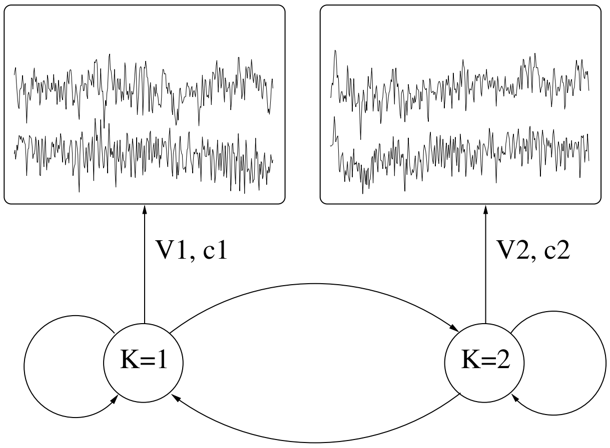

Figure 1: *Generative model. At time t the state of the HMM is $q_t = k$. Associated with each state is a mixing matrix $\boldsymbol{V}_k$ and a set of GAR coefficients, $\boldsymbol{c}_k^i$, one for each source i. The observations are generated as $\boldsymbol{x}_t = \boldsymbol{V}_k \boldsymbol{a}_t$ and the ith source is generated as $a_t^i = \boldsymbol{c}_k^i \tilde{\boldsymbol{a}}_t + e_t$ where $\tilde{\boldsymbol{a}}_t$ is a vector of previous source values and e_t is non-Gaussian noise.*

THEORY

Hidden Markov Models

The HMM model has K discrete hidden states, an initial state probability vector $\boldsymbol{\pi} = \pi_1, \pi_2, ..., \pi_K$, a state transition matrix $\boldsymbol{A}$ with entries a_{ij} and an observation density, $b_k(\boldsymbol{x}_t)$, for data point $\boldsymbol{x}_t$ and hidden state k. The parameters of the observation model (which in this paper, we consider to be an ICA model) are concatenated into the parameter set $\boldsymbol{B}$. The parameters of the HMM, as a whole, are concatenated into the parameter set $\boldsymbol{\theta} = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$. The probability of being in state k at time t given a particular observation sequence $\boldsymbol{X}_1^T$ is $\gamma_k[t]$. Given $\boldsymbol{\theta}$, $\gamma_k[t]$ can be calculated from the forward-backward algorithm [1].

The HMM can be trained using an Expectation-Maximisation (EM) algorithm [2] as follows. In an HMM parameterised by some vector $\hat{\boldsymbol{\theta}}$ the observed variable $\boldsymbol{x}_t$ 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 can be seen from the generative model in Figure 1. This allows the joint likelihood of an observation sequence $\boldsymbol{x}_1^T = [\boldsymbol{x}_1, \boldsymbol{x}_t, ..., \boldsymbol{x}_T]$ and hidden variable (state) sequence $\boldsymbol{q}_1^T = [q_1, q_2, ..., q_T]$ to be written as

$$p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{q}_1^T, \boldsymbol{x}_1^T) = p_{\hat{\boldsymbol{\theta}}}(q_1) \prod_{t=2}^{T} p_{\hat{\boldsymbol{\theta}}}(q_t \mid q_{t-1}) \prod_{t=1}^{T} p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t \mid q_t) \quad (1)$$

The joint log-likelihood may be written

$$\log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{q}_1^T, \boldsymbol{x}_1^T) = \log p_{\hat{\boldsymbol{\theta}}}(q_1) + \sum_{t=2}^{T} \log p_{\hat{\boldsymbol{\theta}}}(q_t \mid q_{t-1}) + \sum_{t=1}^{T} \log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t \mid q_t) \quad (2)$$

The EM algorithm requires us to maximise an *auxiliary function*, $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$, the expectation of the joint log-likelihood where the expectation is taken relative to the old distribution of hidden (state) variables, $p_{\boldsymbol{\theta}}(\boldsymbol{q}_1^T|\boldsymbol{x}_1^T)$ [1]. Hence,

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) + Q(\boldsymbol{A}, \hat{\boldsymbol{A}}) + Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) \quad (3)$$

where

$$Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}) = \sum_{\boldsymbol{q}_1^T} p_{\boldsymbol{\theta}}(\boldsymbol{q}_1^T|\boldsymbol{x}_1^T) \log p_{\hat{\boldsymbol{\theta}}}(q_1) \quad (4)$$

$$Q(\boldsymbol{A}, \hat{\boldsymbol{A}}) = \sum_{\boldsymbol{q}_1^T} p_{\boldsymbol{\theta}}(\boldsymbol{q}_1^T|\boldsymbol{x}_1^T) \sum_{t=2}^{T} \log p_{\hat{\boldsymbol{\theta}}}(q_t \mid q_{t-1}) \quad (5)$$

$$Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) = \sum_{\boldsymbol{q}_1^T} p_{\boldsymbol{\theta}}(\boldsymbol{q}_1^T|\boldsymbol{x}_1^T) \sum_{t=1}^{T} \log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t \mid q_t) \quad (6)$$

and $\sum_{\boldsymbol{q}_1^T}$ denotes a sum over all possible hidden state sequences. The three terms can be maximised separately giving rise to parameter update equations for each different part of the model. Maximising $Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}})$ and $Q(\boldsymbol{A}, \hat{\boldsymbol{A}})$ leads to the usual update equations for the initial state probabilities and the state transition matrix [1]. The term for the observation model parameters can be re-arranged as

$$Q(\boldsymbol{B}, \hat{\boldsymbol{B}}) = \sum_{k} \sum_{t} \gamma_k[t] \log p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t \mid q_t) \quad (7)$$

In principle, the observation model can be any probabilistic model. In speech recognition, for example, a Gaussian observation model or Gaussian Mixture model is often used and in previous work we have investigated the use of autoregressive models [3]. In this paper, we consider ICA models.

After the HMM has been trained, it can be applied to novel data sets and the most likely sequence of hidden states identified using the Viterbi algorithm [1].

Independent Component Analysis

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

$$\boldsymbol{x}_t = \boldsymbol{V}\boldsymbol{a}_t \quad (8)$$

where the underlying factors or sources, $\boldsymbol{a}_t$, are statistically independent and $\boldsymbol{V}$ is known as the *mixing matrix*. There are N observations and M sources. In this paper we consider square mixing only, i.e. $M = N$. ICA may also be viewed as a linear transformation to a new set of variables

$$\boldsymbol{a}_t = \boldsymbol{W}\boldsymbol{x}_t \quad (9)$$

[1] Maximising the auxiliary function can be shown, via Baye's rule and Jensen's inequality, to maximise the likelihood of the data, $p_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_t)$ [2].

whose components, $a_i[t]$, are statistically independent. The matrix $\boldsymbol{W}$ is known as the *unmixing matrix* and is the inverse of $\boldsymbol{V}$.

As noted by Mackay [4], it is possible to place ICA in a maximum likelihood framework. The likelihood of data points can be calculated from the likelihood of the source densities by noting that multivariate probability distributions are transformed as follows

$$p(\boldsymbol{x}_t) = \frac{p(\boldsymbol{a}_t)}{|\boldsymbol{J}|} \tag{10}$$

where $||$ denotes the absolute value and $\boldsymbol{J}$ is the Jacobian of the transformation (the Jacobian is the determinant of the matrix of partial derivatives) and $\boldsymbol{a}_t = \boldsymbol{W}\boldsymbol{x}_t$ [5]. Given that, in ICA, the transformation is linear and we are assuming independent sources, we have $|\boldsymbol{J}| = 1/|det(\boldsymbol{W})|$ and $p(\boldsymbol{a}_t) = \prod_i a_i[t]$. This allows us to write the log-likelihood as

$$\log p(\boldsymbol{x}_t) = \log |det(\boldsymbol{W})| + \sum_{i=1}^{M} \log p(a_i[t]) \tag{11}$$

where $p(a_i[t])$ is the probability density of the ith source component and $a_i[t] = \sum_j \boldsymbol{W}_{ij} x_j[t]$.

Generalised Autoregressive Sources

Pearlmutter and Parra [6] have proposed a 'contextual-ICA' algorithm where the sources are conditioned on previous source values. The observations are then generated from an instantaneous mixing of the sources. Their work focuses on using generalised autoregressive (GAR) models for modelling each source. The term 'generalised' is used because the AR models incorporate additive noise which is non-Gaussian. Specifically, Pearlmutter and Parra use an inverse-cosh noise distribution. In this paper we use GAR source models but with the more flexible Generalised Exponential (GE) noise distribution.

The model has p filter taps such that the GAR prediction for source i at time t is

$$\hat{a}_i[t] = -\sum_{d=1}^{p} c_i[d] a_i[t-d] \tag{12}$$

where $c_i[d]$ are the GAR coefficients for the ith source which can collectively be written as a vector $\boldsymbol{c}_i$. The prediction error is then $e_i[t] = a_i[t] - \hat{a}_i[t]$ which is distributed according to the GE distribution

$$G(e_i[t]; R_i, \beta_i) = \frac{R_i \beta_i^{1/R_i}}{2\Gamma(1/R_i)} \exp(-\beta_i |e_i[t]|^{R_i}) \tag{13}$$

where $\Gamma()$ is the usual gamma function [7], the mean of the distribution is zero, the width of the distribution is determined by $1/\beta_i$ and the weight of its tails is set by R_i. This gives rise to a Gaussian distribution for $R_i = 2$, a Laplacian for $R_i = 1$ and a uniform distribution in the limit $R_i \to \infty$. This parametric form is much more general than the inverse-cosh and has previously been used to good effect in stationary ICA models [8].

The probability density for the ith source is given by

$$p(a_i[t]) = G(e_i[t]; R_i, \beta_i) \tag{14}$$

We note that for the special case of $p = 0$ the GAR source model reduces to a GE source model. Experiments later in this paper contrast the use of GAR versus GE sources.

Hidden Markov ICA

A Hidden Markov ICA model has an unmixing matrix, $\boldsymbol{W}_k$, and source density parameter vectors, $\boldsymbol{R}_k, \boldsymbol{\beta}_k$ for each state k. These parameters can be optimised by minimising a cost function defined by substituting the ICA log-likelihood (equation 11) into the HMM auxiliary function (equation 7) giving

$$Q = \sum_k Q_k \tag{15}$$

where

$$Q_k = \log |det(\boldsymbol{W}_k)| + \frac{1}{\gamma_k} \sum_{t=1}^{T} \gamma_k[t] \sum_{i=1}^{M} \log p(a_i[t]) \tag{16}$$

where we have divided through by

$$\gamma_k = \sum_{t=1}^{T} \gamma_k[t] \tag{17}$$

Gradient-based learning rules for updating the parameters of the HMICA model can be derived by evaluating the appropriate derivatives of Q_k eg. $\frac{\partial Q_k}{\partial \boldsymbol{W}_k}$, $\frac{\partial Q_k}{\partial R_{k,i}}$, $\frac{\partial Q_k}{\partial \beta_{k,i}}$ and $\frac{\partial Q_k}{\partial \boldsymbol{c}_{k,i}}$. The remainder of this section shows how these derivatives can be computed. The unmixing matrix can be learnt by following the gradient

$$\frac{\partial Q_k}{\partial \boldsymbol{W}_k} = \boldsymbol{W}_k^{-T} + \frac{1}{\gamma_k} \sum_t \gamma_k[t] \boldsymbol{z}_t \boldsymbol{x}_t^T \tag{18}$$

where $z_i(t) = \partial \log p(a_i[t]) / \partial a_i[t]$. It is, however, more efficient to use a 'covariant' [4] algorithm which follows the 'natural gradient' [9] which is obtained by post-multiplying the gradient by $\boldsymbol{W}_k^T \boldsymbol{W}_k$. This gives

$$\left[\frac{\partial Q_k}{\partial \boldsymbol{W}_k} \right]_{NAT} = \boldsymbol{W}_k + \frac{1}{\gamma_k} \sum_t \gamma_k[t] \boldsymbol{z}_t \boldsymbol{y}_t^T \tag{19}$$

where

$$\boldsymbol{y}_t = \boldsymbol{W}_k^T \boldsymbol{a}_t \tag{20}$$

The above gradients are identical to those used in standard ICA except that, in the HMICA update for state k, each data point is weighted by an amount proportional to the 'responsibility' state k has for that data point.

For GAR sources the gradient can be calculated by substituting the GAR probability $p(a_i[t])$ from equation 14 into the above expression for $z_i(t)$. Full details of the resulting HMICA natural gradient algorithm and optimisation methods for the GAR coefficients are given in [10].

RESULTS

Synthetic Data

We generated a two-dimensional time series containing two epochs each with $N = 200$ samples. We first generated two GE sources, the first being Gaussian ($R = 2$) with variance $1/\beta = 2$, and the second being Laplacian ($R = 1$) with width parameter $1/\beta = 1$. These source parameters were the same for both epochs. Observations for the first epoch of the time series were then generated by mixing the sources with the matrix

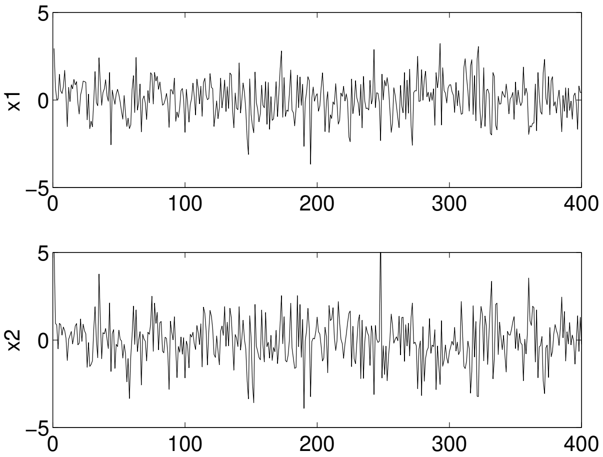

Figure 2: *Synthetic Data. The mixing matrices underlying the two time series are switched at t=200*

$$\boldsymbol{V}_1 = \begin{pmatrix} 0.8321 & 0.5547 \\ 0.4472 & 0.8944 \end{pmatrix} \tag{21}$$

In the second epoch, the matrix was changed to

$$\boldsymbol{V}_2 = \begin{pmatrix} 0.8944 & -0.4472 \\ -0.5547 & 0.8231 \end{pmatrix} \tag{22}$$

The resulting data set is plotted in Figure 2.

We then trained two ICA models on each separate data partition to serve as a benchmark against which the HMICA solution could be compared[2].

A two-state HMICA model was then trained on the data. The model was initialised by setting $R = 2$ and $\beta = 1$ for both sources and both states. Each unmixing matrix was initially set to the identity matrix with uniform noise in the range $[-0.05, 0.05]$ added on to each matrix element. The diagonal elements of the state transition matrix were set

[2] If we were to compare the HMICA solution against the known true models then any difference could be due to (i) incorrect partitioning in the HMICA model or (ii) having a small, finite data sample or (iii) some artifact of the covariant ICA algorithm. By comparing against the covariant ICA solutions we are therefore sure to be focusing only on (i).

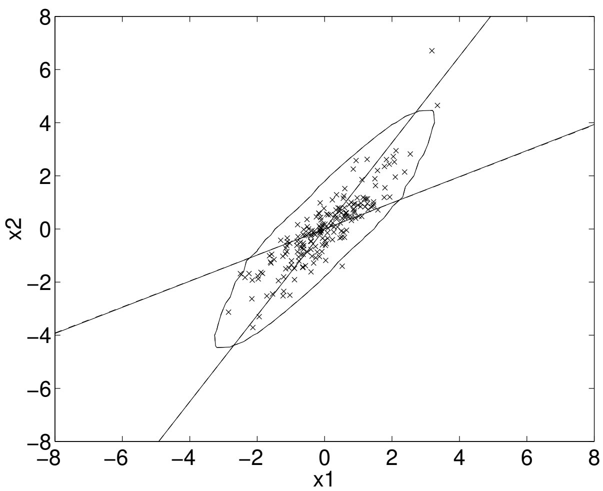

Figure 3: *Synthetic Data. ICA model for state 1. The straight lines show the directions of the independent components and the curvy lines show a single probability contour plotted at 0.01 of the probability density at the origin. These probability densities were calculated using equation 10.*

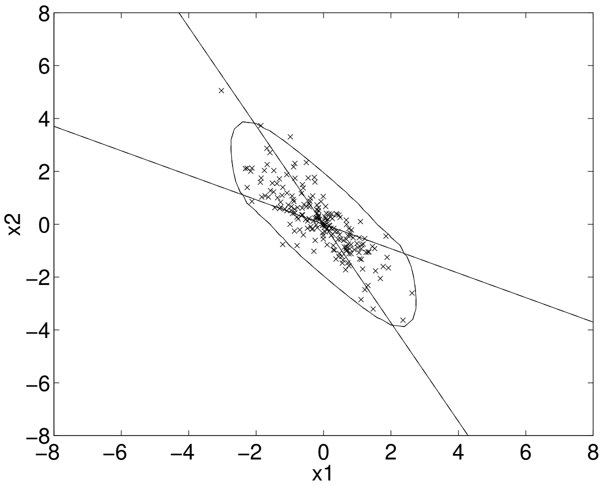

Figure 4: *Synthetic Data. ICA model for state 2.*

so that the average state duration density was D_s samples (see below). Five iterations of the EM algorithm were then sufficient for the HMICA solution to converge. The estimates for R and β were so accurate that no difference could be seen in the plotted probability contours (see Figures 3 and 4). This accuracy is due to the correct partitioning of the data into the two separate epochs; application of the Viterbi algorithm led to 100% of samples being correctly classified. Experimentation with initialisation of the state transition matrix showed that convergence was typically achieved in five to ten EM iterations with D_s (see above) ranging from 10 to 400 samples. Convergence was even possible with D_s as low as three, after ten to twenty EM iterations.

EEG Data

We applied the HMICA model to EEG data recorded whilst a subject performed a motor imagery task; the subject was instructed to imagine opening and closing his hand in response to an external cue. Figure 5 shows two channels of EEG and the timing of the task cue. This subject possessed a particularly strong mu-rhythm (an 8-12Hz rhythm in sensorimotor cortex) which disappeared during imagery (after the cue).

We trained two HMICA models on this data set; one with GAR sources and one with GE. Each converged within ten EM iterations. As can be seen from Figure 5 the one with GAR sources provides a good partitioning of the data, whereas use of the GE source model provides a less clear partitioning. This difference can be explained by the high source Signal-to-Noise (SNR) ratio before the cue; during high SNR periods the sources are likely to be multimodal (a pure sinewave, for example, has a bimodal density). In these periods a GAR source model is more appropriate because, by fitting the dynamics of the signal, the residual will become unimodal.

We note that it is essential to have a non-stationary model in order for these periods of strong rhythmic source activity to be identified. Applying stationary-ICA to the time series without the partitioning provided by the HMM misses this activity completely. GAR source models and HMM switching are therefore complementary.

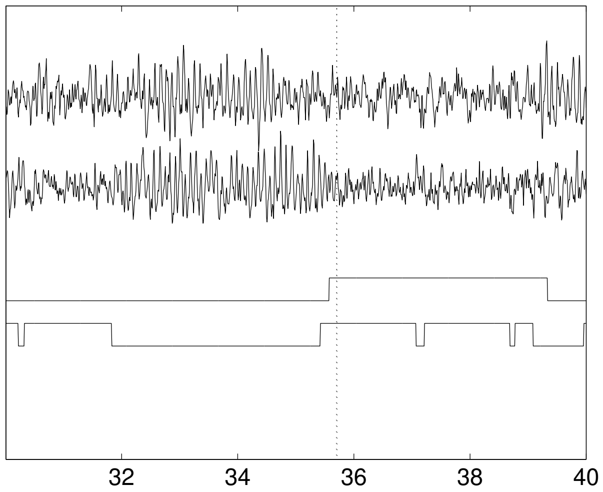

Figure 5: *EEG data recorded before and after motor imagery task. The top two traces show EEG recordings from electrodes over left and right sensorimotor cortex. The next two traces show partitionings from an HMICA model; the top one using GAR sources and the bottom one using GE sources. The dashed vertical line indicates the cue indicating when the task should be performed.*

CONCLUSIONS & FUTURE WORK

We have proposed a generative model for the analysis of nonstationary multivariate time series. The model combines three existing statistical models, HMMs, ICA and GARs, into a single overall model where the components learn together. The model can (i) learn the dynamics of the underlying sources (ii) learn the parameters of the mixing process and (iii) detect discrete changes in either the source dynamics or the mixing or both. This last property enables the model to partition a multivariate time series into different regimes and so may be used for classification purposes. Moreover, each regime is characterised by a known mixing of stationary sources. The model is therefore readily interpretable.

As compared to previous work in ICA, the main novelty of our approach is in the use of an HMM to switch between ICA models. Like other modellers [6], we have also used GAR source models, although ours are different in the choice of noise model; instead of an inverse-cosh we use the more flexible generalised exponential [8].

Whilst our model goes some way to bridging the gap between statistical models and the known properties of biomedical data, such as EEG, there remain many areas for future refinement. The most unrealistic assumption at present is that of square mixing; for data such as EEG it is unlikely that the number of sources is the same as the number of electrodes. A non-square ICA algorithm is therefore required, such as those developed by Attias [11] . These algorithms also have the advantage of being able to cope with observation noise (actually, the observation noise is essential in developing a probabilistic model; if you have fewer sources than observations the 'extra' observations can be explained as noise).

The second main area for improvement is the use of automatic model order selection methods; both for the order of the source model and to choose the optimal number of sources. Finally, following Knuth [12], we could use our knowledge of the physics of the mixing process to choose suitable priors for the mixing matrices.

With regard to the analysis of biomedical data, the ICA model we have described has two very attractive properties. Firstly, because the E-step of the hidden Markov learning automatically partitions the data into similar regimes, HMICA will be able to detect task-related changes that are not consistently time-locked to the experimental task block design. In event-related EEG or fMRI studies [13] these changes would not otherwise show up; they would disappear in the averaging process. Secondly, by tuning each GAR source to its optimal model order (using, for example, a Minimum Description Length criterion) pure noise sources, having zero model order, can be differentiated from dynamic sources, having non-zero model order. By focussing only on the dynamic sources HMICA can deliver a small number of interesting sources rather than the usual very large number of sources which may be difficult to interpret.

References

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

[2] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[3] W.D. Penny and S.J. Roberts. Dynamic models for nonstationary signal segmentation. *Computers and Biomedical Research*, 32(6):483–502, 1999.

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

[5] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.

[6] B. A. Pearlmutter and L.C. Parra. Maximum Likelihood Blind Source Separation: A Context-Sensitive Generalization of ICA. In *Advances in Neural Information Processing Systems 9*, pages 613–619. MIT Press, 1997.

[7] W. H. Press, S.A. Teukolsky, W.T. Vetterling, and B.V.P. Flannery. *Numerical Recipes in C*. Cambridge, 1992.

[8] R. Everson and S.J. Roberts. Independent Component Analysis: A flexible nonlinearity and decorrelating manifold approach. *Neural Computation*, 11(8), 1999.

[9] S. Amari, A. Cichocki, and H.H. Yang. A New Learning Algorithm for Blind Signal Separation. In D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, Cambridge, MA, 1996. MIT Press.

[10] W.D. Penny, R.M. Everson, and S.J. Roberts. Hidden Markov Independent Components Analysis. In M. Girolami, editor, *Advances in Independent Component Analysis*. Springer, 2000.

[11] H. Attias. Independent Factor Analysis. *Neural Computation*, 11(4):803–851, 1999.

[12] K.H. Knuth. Difficulties Applying Recent Blind Source Separation Techniques to EEG and MEG. In J.T. Rychert and C.R. Smith, editors, *Maximum Entropy and Bayesian Methods*, pages 209–222. Dordrecht, 1998.

[13] M.J. McKeown, S. Makeig, G.G. Brown, T.P. Jung, S.S. Kindermann, A.J. Bell, and T.J. Sejnowski. Analysis of fMRI data by blind separation into independent spatial components. *Human Brain Mapping*, 6:160–188, 1998.

Acknowledgements

William Penny is supported by the UK Engineering and Physical Sciences Research Council grant number GR/M 41339. The empirical work in this paper was based around hidden Markov model software routines made available by Zoubin Ghahramani ($http://gatsby.ucl.ac.uk/zoubin$).