# VARIATIONAL BAYES FOR NON-GAUSSIAN AUTOREGRESSIVE MODELS

W.D. Penny and S.J. Roberts

Department of Engineering Science,
Oxford University.
E-mail: wpenny,sjrob@robots.ox.ac.uk

**Abstract. We describe a Variational Bayesian (VB) learning algorithm for Non-Gaussian Autoregressive (AR) models. The noise is modelled as a Mixture of Gaussians rather than the usual single Gaussian. This allows different data points to be associated with different noise levels and effectively provides a robust estimation of AR coefficients. The VB framework is used to prevent overfitting and provides model order selection criteria both for AR order and noise model order. The algorithm is applied to synthetic data and to EEG.**

## INTRODUCTION

The standard Autoregressive (AR) model assumes that the noise is Gaussian and therefore that the AR coefficients can be set by minimising a least squares cost function. Least squares, however, is known to be sensitive to outliers. Therefore, if our time series is even marginally contaminated by artifacts the resulting AR coefficient estimates will be seriously degraded. See Bishop ([2], page 209) and Press *et al.* ([6], page 700) for a general discussion of this issue and a number of proposed solutions.

In this paper, we tackle this problem by modeling the noise with a Mixture of Gaussians in which different data points can be associated with different noise levels. This provides a robust estimation of AR coefficients via a weighted least squares approach; data points associated with high noise levels are downweighted in the AR estimation step. Moreover, the Variational Bayes (VB) framework [1] is used to prevent model overfitting and it provides model order selection criteria both for AR order and noise model order. Learning takes places in an Expectation-Maximisation (EM)-like strategy where, in the E-step, data points are soft assigned to different noise levels, and in the M-step, the noise levels, mixing coefficients and AR parameters are re-estimated. Readers unfamiliar with the VB framework are advised to first read our companion paper in these proceedings [5].

## NON-GAUSSIAN AUTOREGRESSIVE MODELS

We define an autoregressive model as

$$y_n = \boldsymbol{x}_n \boldsymbol{w} + e_n \tag{1}$$

where $y_n$ is the $n$th value of a time series, $\boldsymbol{w}$ is a column vector of AR coefficients (weights) and $\boldsymbol{x}_n = [y_{n-1}, y_{n-2}, ..., y_{n-p}]$ are the $p$ previous time series values. The AR model has additive noise $e_n$ which is usually modelled as a Gaussian. In this paper, however, we model the noise as a 1-dimensional Gaussian Mixture having $m$-components. Component $s$ has mixing coefficient $\pi_s$, mean $\mu_s$ and precision (inverse variance) $\beta_s$. We can write the parameters collectively as the vectors $\boldsymbol{w}$, $\boldsymbol{\pi} = [\pi_1, \pi_2, ..., \pi_m]$, $\boldsymbol{\mu} = [\mu_1, \mu_2, ..., \mu_m]$ and $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_m]$. The weights are drawn from a Gaussian prior (see later) with precision $\alpha$. We concatenate all the parameters into the overall parameter vector $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{w}, \alpha\}$. We are given a set of data points $D = \{y_n, \boldsymbol{x}_n\}$ with $n = 1..N$. The likelihood of a data point is given by the mixture model

$$p(y_n|\boldsymbol{\theta}) = \sum_{s=1}^{m} p(s_n = s|\boldsymbol{\pi}) p(y_n|, \boldsymbol{x}_n, s_n, \beta_s, \mu_s, \boldsymbol{w}) \tag{2}$$

where $s_n$ is an indicator variable indicating which component is selected for which data point. These are chosen probabilistically according to

$$p(s_n = s|\boldsymbol{\pi}) = \pi_s \tag{3}$$

Each component is a Gaussian with

$$p(y_n|\boldsymbol{x}_n, s_n, \beta_s, \mu_s, \boldsymbol{w}) = (2\pi)^{-1/2} \beta_s^{1/2} \exp\left(\frac{-\beta_s}{2}(y_n - [\mu_s + \boldsymbol{x}_n \boldsymbol{w}])^2\right) \tag{4}$$

The prior on the model parameters is

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\pi}) \prod_s p(\beta_s) \prod_s p(\mu_s) p(\boldsymbol{w}|\alpha) p(\alpha) \tag{5}$$

where the mixing prior is a symmetric Dirichlet

$$p(\boldsymbol{\pi}) = \frac{\Gamma(m\lambda_0)}{\Gamma(\lambda_0)^m} \prod_{s=1}^{m} \pi_s^{\lambda_0 - 1} \tag{6}$$

and $\Gamma(x)$ is the Gamma function [6]. The prior over the precisions is a Gamma

$$p(\beta_s) = \Gamma(\beta_s; b_0, c_0) \tag{7}$$

where the Gamma density is defined as

$$\Gamma(x; b, c) = \frac{1}{\Gamma(c)} \frac{x^{c-1}}{b^c} \exp\left(\frac{-x}{b}\right) \tag{8}$$

The prior over the means is a Normal

$$p(\mu_s) = N(\mu_s; m_0, v_0) \tag{9}$$

and the prior over the weights is a zero-mean Gaussian with an isotropic covariance having precision $\alpha$

$$p(\boldsymbol{w} \mid \alpha) = \left(\frac{\alpha}{2\pi}\right)^{p/2} \exp(-\alpha E_W) \tag{10}$$

where

$$E_W = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \tag{11}$$

The weight precision is drawn from a Gamma prior

$$p(\alpha) = \Gamma(\alpha; b_\alpha, c_\alpha) \tag{12}$$

The joint likelihood of a data point and indicator variable is

$$p(y_n, s_n|\theta) = p(s_n = s|\boldsymbol{\pi})p(y_n|\boldsymbol{x}_n, s_n, \beta_s, \mu_s, \boldsymbol{w}) \tag{13}$$

which, given that the data are IID, gives

$$p(\boldsymbol{Y}, \boldsymbol{S}|\theta) = \prod_{n=1}^{N} p(s_n = s|\boldsymbol{\pi})p(y_n|\boldsymbol{x}_n, s_n, \beta_s, \mu_s, \boldsymbol{w}) \tag{14}$$

over the whole date set, where $\boldsymbol{Y} = [y_1, y_2, ..., y_N]^T$ and $\boldsymbol{S} = [s_1, s_2, ..., s_N]^T$.

## VARIATIONAL BAYES

The aim of Variational Bayesian (VB) learning or 'Ensemble Learning' for models with hidden variables is to maximise the negative free energy [1]

$$F(p, m) = \left\langle \log \frac{p(\boldsymbol{Y}, \boldsymbol{S}|\boldsymbol{\theta})}{q(\boldsymbol{S})} \right\rangle_{\boldsymbol{S},\boldsymbol{\theta}} - KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) \tag{15}$$

where the first term corresponds to an average likelihood $L_{av}$ (where the expectation is taken wrt. $q(\boldsymbol{S})$ and $q(\boldsymbol{\theta})$) and the second term is the Kullback-Liebler (KL)-divergence between the approximate posteriors $q(\boldsymbol{\theta})$ and the priors $p(\boldsymbol{\theta})$. This objective function can be maximised via an EM-like strategy. In the E-step, $q(\boldsymbol{S})$ is updated according to

$$q(\boldsymbol{S}) \propto \exp[I(\boldsymbol{S})] \tag{16}$$

where

$$I(\boldsymbol{S}) = \langle \log p(\boldsymbol{Y}, \boldsymbol{S}|\theta) \rangle_{\boldsymbol{\theta}} \tag{17}$$

and the expectation is taken wrt. $q(\boldsymbol{\theta})$. In the M-step $q(\boldsymbol{\theta})$ is updated according to

$$q(\boldsymbol{\theta}) \propto \exp[I(\boldsymbol{\theta})]p(\boldsymbol{\theta}) \tag{18}$$

where

$$I(\boldsymbol{\theta}) = \langle \log p(\boldsymbol{Y}, \boldsymbol{S}|\theta) \rangle_{\boldsymbol{S}} \tag{19}$$

and the expectation is taken wrt. $q(\boldsymbol{S})$. See [1] for a derivation and discussion of this EM-like strategy.

To apply VB to Non-Gaussian Autoregressive models we approximate the posterior distribution over parameters with the factorised density

$$q(\boldsymbol{\theta}) = q(\boldsymbol{\pi})q(\boldsymbol{\beta})q(\boldsymbol{\mu})q(\boldsymbol{w})q(\alpha) \tag{20}$$

and the posterior distribution over hidden variables by $q(\boldsymbol{S})$. We then set each distribution so as to maximise $F(p, m)$. We omit the derivations due to lack of space but the procedure is similar to that used in [1]. The optimal posteriors factorize into the following parametric forms. For the precisions we have $q(\boldsymbol{\beta}) = \prod_s q(\beta_s)$ and Gamma densities

$$q(\beta_s) = \Gamma(\beta_s; b_s, c_s) \tag{21}$$

For the means, we have $q(\boldsymbol{\mu}) = \prod_s q(\mu_s)$ and Normal densities

$$q(\mu_s) = N(\mu_s; m_s, v_s) \tag{22}$$

For the mixing coefficients we have a Dirichlet

$$q(\pi) = \Gamma(\sum_{s'} \lambda_{s'}) \prod_{s=1}^{m} \frac{\pi_s^{\lambda_s - 1}}{\Gamma(\lambda_s)} \tag{23}$$

For the weights we have a Normal

$$q(\boldsymbol{w}) = N(\boldsymbol{w}; \hat{\boldsymbol{w}}, \hat{\boldsymbol{C}}) \tag{24}$$

and for the weight precision a Gamma

$$q(\alpha) = \Gamma(\alpha; b'_\alpha, c'_\alpha) \tag{25}$$

For the indicator posteriors we have $q(\boldsymbol{S}) = \prod_n q(s_n)$ and we write $\gamma_s^n \equiv q(s_n)$. The distributions are then optimised via an EM-like strategy where the E-step and M-step are as follows.

**E-step**

We first define some intermediate variables

$$\begin{aligned} \log \tilde{\pi}_s &= \Psi(\lambda_s) - \Psi(\sum_{s'} \lambda_{s'}) \\ \log \tilde{\beta}_s &= \Psi(c_s) + \log b_s \\ \bar{\beta}_s &= b_s c_s \end{aligned} \tag{26}$$

where $\Psi()$ is the digamma function [6] and

$$\begin{aligned}\tilde{\sigma}_s^2(n) &= \int q(\boldsymbol{w})q(\mu_s)(y_n - [\mu_s + \boldsymbol{x}_n \boldsymbol{w}])^2 d\boldsymbol{w} d\mu_s \\ &= y_n^2 - 2m_s y_n - 2\hat{y}_n y_n + m_s^2 + v_s \\ &+ 2m_s \hat{y}_n + \boldsymbol{x}_n \hat{\boldsymbol{C}} \boldsymbol{x}_n^T + \hat{y}_n^2\end{aligned} \tag{27}$$

where $\hat{y}_n = \boldsymbol{x}_n \hat{\boldsymbol{w}}$ is the mean AR prediction. The E-step then consists of updating the indicator posterior according to

$$\tilde{\gamma}_s^n = \tilde{\pi}_s \tilde{\beta}_s^{1/2} \exp\left[-\frac{1}{2}\bar{\beta}_s \tilde{\sigma}_s^2(n)\right] \tag{28}$$

We then normalise to give

$$\gamma_s^n = \frac{\tilde{\gamma}_s^n}{\sum_{s'} \tilde{\gamma}_{s'}^n} \tag{29}$$

which is the probability that component $s$ is responsible for data point $y_n$.

**M-step**

Again, we define some intermediate variables

$$\begin{aligned}\bar{\pi}_s &= \frac{1}{N}\sum_{n=1}^{N} \gamma_s^n \\ \bar{N}_s &= N\bar{\pi}_s \\ \bar{y}_s &= \frac{1}{N}\sum_{n=1}^{N} \gamma_s^n y_n \\ \bar{\boldsymbol{x}}_s &= \frac{1}{N}\sum_{n=1}^{N} \gamma_s^n \boldsymbol{x}_n\end{aligned} \tag{30}$$

where $\bar{\pi}_s$ is the proportion of data in component $s$ and $\bar{N}_s$ is the number of data points in component $s$ and the quantities $\bar{\boldsymbol{x}}_s$ and $\bar{y}_s$ are weighted data values. We then update the hyperparameters as follows. The mixing hyperparameters are updated by adding the data counts to the prior counts

$$\lambda_s = \bar{N}_s + \lambda_0 \tag{31}$$

If we define the average variance of component $s$ as

$$\tilde{\sigma}_s^2 = \frac{1}{N}\sum_{n=1}^{N} \gamma_s^n \tilde{\sigma}_s^2(n) \tag{32}$$

then the hyperparameters for the precisions are updated as

$$\begin{aligned}\frac{1}{b_s} &= \frac{N}{2}\tilde{\sigma}_s^2 + \frac{1}{b_0} \\ c_s &= \frac{\bar{N}_s}{2} + c_0\end{aligned} \tag{33}$$

We can understand these equations by looking at the corresponding mean variance (the inverse of the mean precision) which is given by $1/(b_s c_s)$. If we ignore terms involving the prior this comes out to be $\tilde{\sigma}_s^2/\bar{\pi}_s$ which is the average variance of that component re-weighted according to the number of examples that the component is responsible for.

If we first define the means and precisions of the component means *as estimated from the data* as

$$\begin{aligned} m_{data}(s) &= (\bar{\boldsymbol{x}}_s \hat{\boldsymbol{w}} - \bar{y}_s)/\bar{\pi}_s \\ \tau_{data}(s) &= \bar{N}_s \bar{\beta}_s \end{aligned} \tag{34}$$

then the means and precisions of the posteriors are given by

$$\begin{aligned} \tau_s &= \tau_0 + \tau_{data}(s) \\ m_s &= \frac{\tau_0}{\tau_s} m_0 + \frac{\tau_{data}(s)}{\tau_s} m_{data}(s) \end{aligned} \tag{35}$$

where $\tau_0 = 1/v_0$ is the prior precision and $\tau_s = 1/v_s$ is the posterior precision. The posterior for the AR coefficients is then updated using

$$\begin{aligned} \hat{\boldsymbol{C}} &= \left( \sum_s \bar{\beta}_s \boldsymbol{X}^T \boldsymbol{\Gamma}_s \boldsymbol{X} + \bar{\alpha} \boldsymbol{I} \right)^{-1} \\ \hat{\boldsymbol{w}} &= \hat{\boldsymbol{C}} \sum_s \bar{\beta}_s \left( \boldsymbol{X}^T \boldsymbol{\Gamma}_s \boldsymbol{Y} + m_s \boldsymbol{X}^T \boldsymbol{\Gamma}_s \boldsymbol{1} \right) \end{aligned} \tag{36}$$

where $\boldsymbol{\Gamma}_s = diag([\gamma_s^1, \gamma_s^2, ..., \gamma_s^N])$, the $n$th row of $\boldsymbol{X}$ contains $\boldsymbol{x}_n$, the $n$th entry in the column vector $\boldsymbol{Y}$ is $y_n$ and $\boldsymbol{1}$ is a column vector of 1's of length $N$. The updates for the weight precision are

$$\begin{aligned} 1/b'_\alpha &= \frac{1}{2} \hat{\boldsymbol{w}}^T \hat{\boldsymbol{w}} + \frac{1}{2} Tr(\hat{\boldsymbol{C}}) + \frac{1}{b_\alpha} \\ c'_\alpha &= \frac{p}{2} + c_\alpha \\ \bar{\alpha} &= b'_\alpha c'_\alpha \end{aligned} \tag{37}$$

### Co-Mean Noise Model

Whilst it is possible to use a full Gaussian Mixture model for the noise (as described above), the numerical examples in this paper use a simplified model in which all the means are zero. The resulting model then resembles weighted least squares where the weights are given by the precisions and responsibilities of the noise components. This simplifies a number of update equations. For the E-Step, we now have

$$\tilde{\sigma}_s^2(n) = y_n^2 - 2\hat{y}_n y_n + \boldsymbol{x}_n \hat{\boldsymbol{C}} \boldsymbol{x}_n^T + \hat{y}_n^2 \tag{38}$$

For the M-step equations 34 and 35 are dispensed with and the second line in equation 36 simplifies to

$$\hat{\boldsymbol{w}} = \hat{\boldsymbol{C}} \sum_s \bar{\beta}_s \boldsymbol{X}^T \boldsymbol{\Gamma}_s \boldsymbol{Y} \tag{39}$$

**Practicalities**

The prior distribution $p(\boldsymbol{\pi})$ is chosen to be unifrom. We set $\lambda_0 = 5$. For the Gamma distributions we use uninformative priors; for $p(\beta_s)$ we set $b_0 = 10^3$, $c_0 = 10^{-3}$, and for $p(\alpha)$ we set $b_\alpha = 10^3$ and $c_\alpha = 10^{-3}$.

The posterior distributions $q(\boldsymbol{\pi})$, $q(\beta_s)$, $q(\alpha)$ and $q(\boldsymbol{w})$ have the parameters $\lambda_s$, $b_s$, $c_s$, $b'_\alpha$, $c'_\alpha$, $\hat{\boldsymbol{w}}$ and $\hat{\boldsymbol{C}}$ which are initialised as follows. The posterior for the AR coefficients is initialised using the Maximum Likelihood (ML) solution [5]. The posterior for the weight precision is then initialised using equation 37. The remaining posteriors are set as follows. We first calculate the errors $e(n)$ from the ML model and then define a new variable $z(n) = |e(n) - \bar{e}|$ which is the absolute deviation of each error from the mean error $\bar{e}$. We then apply k-means clustering [2] to $z(n)$ which results in mixing coefficients $\lambda_z(s)$ and means $m_z(s)$. We then set $\lambda_s = 100\lambda_z(s)$. The parameters $b_s$ and $c_s$ are then set so as to achieve means of $1/m_z(s)^2$ and variances of $Var(1/m_z)$ (the mean and variance of a Gamma density are $bc$ and $b^2c$ respectively).

The VB equations are then applied iteratively until a consistent solution is reached. Convergence is measured by evaluating the negative free energy

$$F(p, m) = L_{av} - KL(\boldsymbol{w}) - KL(\boldsymbol{\alpha}) - KL(\boldsymbol{\pi}) - KL(\boldsymbol{\beta}) \tag{40}$$

where

$$L_{av} = H(q(\boldsymbol{S})) + \sum_{s=1}^{m} \bar{N}_s (\log \tilde{\pi}_s + 0.5 \log \tilde{\beta}_s) - 0.5N \sum_{s=1}^{m} \bar{\beta}_s \tilde{\sigma}_s^2 - 0.5N \log 2\pi \tag{41}$$

The entropy over the hidden variables is

$$H(q(\boldsymbol{S})) = - \sum_{n=1}^{N} \sum_{s=1}^{m} \gamma_s^n \log \gamma_s^n \tag{42}$$

The KL terms for Normal and Gamma densities can be computed from the equations given in [5]. For the Dirichlet density with $q(\boldsymbol{\pi}) = D(\boldsymbol{\pi}; \boldsymbol{\lambda}_q)$ and $p(\boldsymbol{\pi}) = D(\boldsymbol{\pi}; \boldsymbol{\lambda}_p)$ where the number of states is $m$ and $\lambda_{qt} = \sum_{s=1}^{m} \lambda_q(s)$, $\lambda_{pt} = \sum_{s=1}^{m} \lambda_p(s)$ the KL-divergence is

$$\begin{aligned} KL(q||p) &= \log \frac{\Gamma(\lambda_{qt})}{\Gamma(\lambda_{pt})} + \sum_{s=1}^{m} (\lambda_q(s) - \lambda_p(s))(\Psi(\lambda_q(s)) - \Psi(\lambda_{qt}) \\ &\quad + \sum_{s=1}^{m} \log \frac{\Gamma(\lambda_p(s))}{\Gamma(\lambda_q(s))} \end{aligned} \tag{43}$$

As the update for the AR coefficients is computationally more intensive than the updates for the other parameters (as it involves a matrix inversion) we perform this step only once every $W_t$ iterations. In our experiments we used $W_t = 5$. We evaluate $F(p, m)$ every $W_t$ iterations (after the AR updates) and terminate optimization if the proportionate increase from one evaluation to the next is less than 0.01 per cent.

## RESULTS

### Synthetic Data

We generated three seconds of synthetic data from an AR(5) model with coefficients

$$\boldsymbol{w}_{true} = [-1.8517, 1.3741, 0.1421, -0.6852, 0.3506]^T \tag{44}$$

and noise drawn from a 2-component Gaussian mixture model with mixing coefficients $\pi_1 = 0.9, \pi_2 = 0.1$ and variances $\sigma_1^2 = 1, \sigma_2^2 = 100$. We repeated the data generation process 10 times and used the negative free energy, $F(p, m)$ as a model order selection criterion. The results in Fig-

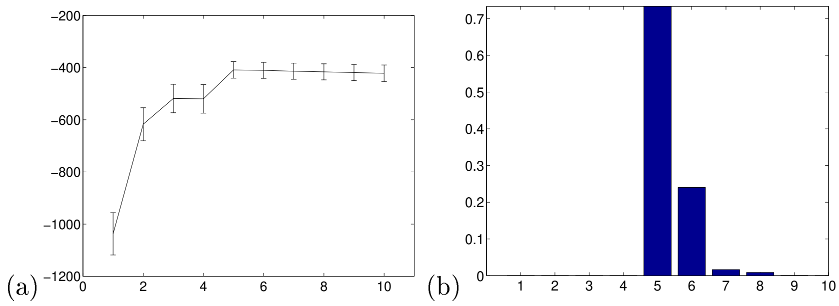


Figure 1: AR model order selection (a) $F(p, 2)$ versus $p$ and (b) the corresponding probabilities, $P(p)$ versus $p$.

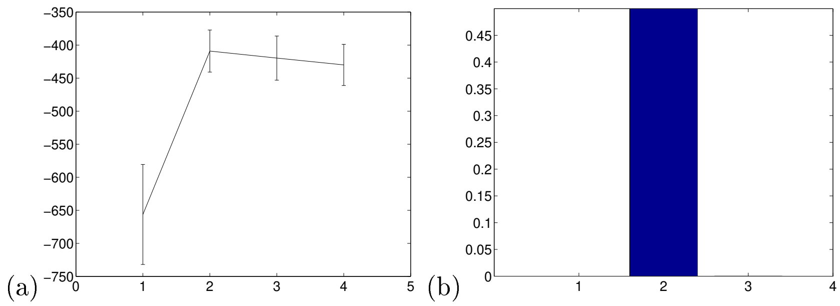


Figure 2: Noise model order selection (a) $F(5, m)$ versus $m$ and (b) the corresponding probabilities, $P(m)$ versus $m$.

ures 1 and 2 show that the VB model order selection criterion selects both the correct AR order and the correct noise model order. For more on VB model order selection, including a comparison with the Minimum Description Length (MDL) criterion, see our companion paper [5].

We also calculated the error with which the AR coefficients were estimated using the measure $||\hat{\boldsymbol{w}} - \boldsymbol{w}_{true}||$. Over the ten runs, the errors from the Non-Gaussian AR model with two components were less than those from the Gaussian AR model by an average factor of six.

**EEG Data**

We applied the model to a short sample of EEG data shown in Figure 3, the middle section of which is contaminated by an eye-movement artifact. For a Gaussian noise model ($m = 1$) the optimum AR model order was at $p = 27$ having $F_{27,1} = -2681$. For $m = 2$ and $m = 3$ the best models were also at $p = 27$ (this is not always the case) with $F_{27,2} = -2661$ and $F_{27,3} = -2671$. This shows that the non-Gaussian AR process with two noise components is the preferred model. This model took 25 EM-iterations to converge which included 5 re-estimates of the AR coefficients. The final mixing coefficients in the noise model were $\pi_1 = 0.78$ and $\pi_2 = 0.22$ and the variances (inverse precisions) were $\sigma_1^2 = 36.3$ and $\sigma_2^2 = 172$. Figure 3 shows

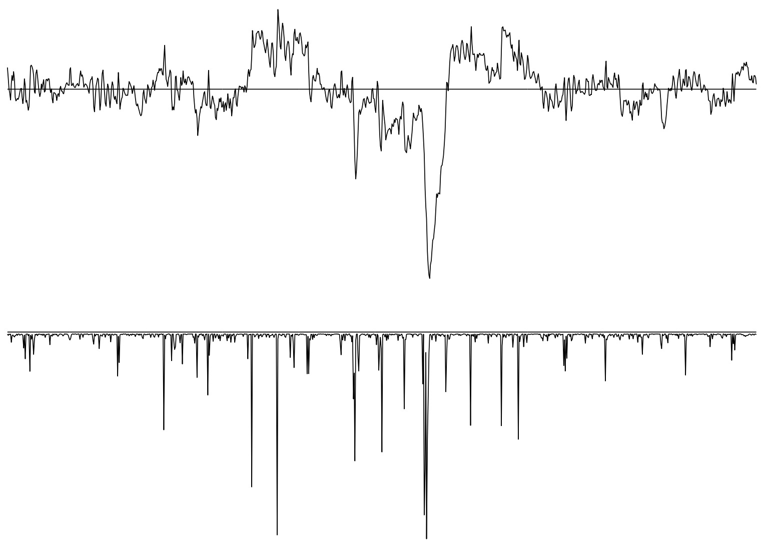

Figure 3: Top plot: Original EEG trace, Bottom plot: The log-probabilitity that the data belongs to the first (low-variance) noise component, $\log \gamma_1$ (calculated from equation 29). A high proportion of data in the middle section therefore clearly belongs to the second (high-variance) noise component. These points are down-weighted in the estimation of the AR coefficients.

which data points 'belong' to which noise component. The first component corresponds to the majority of the data (78% of it) and the second component picks out the outliers which are mostly in the middle section. These points are then effectively downweighted in the corresponding weighted least squares regression. The outliers therefore have a minimal influence on the estimation of the AR coefficients; this can be seen from equation 39 where $\bar{\beta}_s$ and $\mathbf{\Gamma}_s$ weight the samples accordingly.

## DISCUSSION

We have proposed a Non-Gaussian AR model for the modeling of stationary stochastic processes where the noise is modeled with a Mixture of Gaussians. The VB framework is used to prevent overfitting and it provides a model order selection criterion both for the AR order and the noise model order. In the experiments presented in the paper, we restricted the means in the noise mixture to be identically zero. The resulting model resembles weighted least squares and so provides robust estimation of AR coeffcients.

We envisage that a main application of the model is to EEG data; in an EEG recording up to 30% [4] of data blocks are corrupted by muscle, eye-movement or other artifact and, traditionally, these data blocks are discarded. As we have shown in the paper, however, the Non-Gaussian AR model is able to downweight the contaminated section of each block, thus allowing AR (and consequently spectral) estimation based on the uncontaminated section (this is viable as long as the uncontaminated section constitutes the majority of the block; in our eye-movement artifact example this held true as 78% of samples were uncontaminated). We also envisage using the Non-Gaussian AR algorithm to model the sources in an Independent Component Analysis model; this would extend the work in [3] by allowing for dynamic sources at the same time as using VB for model order estimation.

## REFERENCES

[1] H. Attias, “A Variational Bayesian Framework for Graphical Models,” in T. L. et al (ed.), **NIPS 12**, Cambridge, MA: MIT Press, 2000.
[2] C. M. Bishop, **Neural Networks for Pattern Recognition**, Oxford University Press, Oxford, 1995.
[3] R. Choudrey, W. Penny and S. Roberts, “An Ensemble Learning approach to Independent Component Analysis,” in **IEEE International Workshop on Neural Networks for Signal Processing (this volume)**, 2000.
[4] A. Gevins, M. Smith and D. Yu, “High-resolution EEG mapping of cortical activation related to working memory: Effects of task difficulty, type of processing, and practice,” **Cerebral Cortex**, vol. 7, no. 4, pp. 374–385, 1997.
[5] W. Penny and S. Roberts, “Bayesian Methods for Autoregressive Models,” in **IEEE International Workshop on Neural Networks for Signal Processing (this volume)**, 2000.
[6] W. H. Press, S. Teukolsky, W. Vetterling and B. Flannery, **Numerical Recipes in C**, Cambridge, 1992.