Bayesian analysis of single-subject fMRI data:

SPM5 implementation

William D. Penny, Nelson Trujillo-Bareto, Guillaume Flandin

Wellcome Department of Imaging Neuroscience, UCL, London UK.

Key Words: Variational Bayes, fMRI, spatial priors, effect size, general linear model, autoregressive model, Laplacian, smoothing, spatial regularisation

Introduction

This document describes aspects of how Bayesian analysis of single subject fMRI is implemented in SPM5. The algorithm is described in detail in (Penny et al., 2004). Since that paper, however, the algorithm has been extended to allow for spatial regularisation of AR coefficients. The implementation has also been speeded up by using cross-covariance formulae for computing some necessary time domain quantities, and the algorithm can be applied to data from multiple sessions. Further, PPMs based on contrasts of parameter estimates are computed differently depending on whether they are specified beforehand or post-hoc. If they are specified beforehand they use the posterior covariance from VB, whereas if they are specified post-hoc the posterior covariance is approximated using a Taylor-series approach. We now described each of these developments in some detail.

Spatial regularisation of AR coefficients

The generative model described in Figure 1 of (Penny et al., 2004) has been updated to allow for spatial regularisation of AR coefficients, as shown in Figure 1 of this document. In SPM 5, Bayesian analysis allows you to specify voxel-wise AR(p) models with arbitrary p (same p for all voxels).

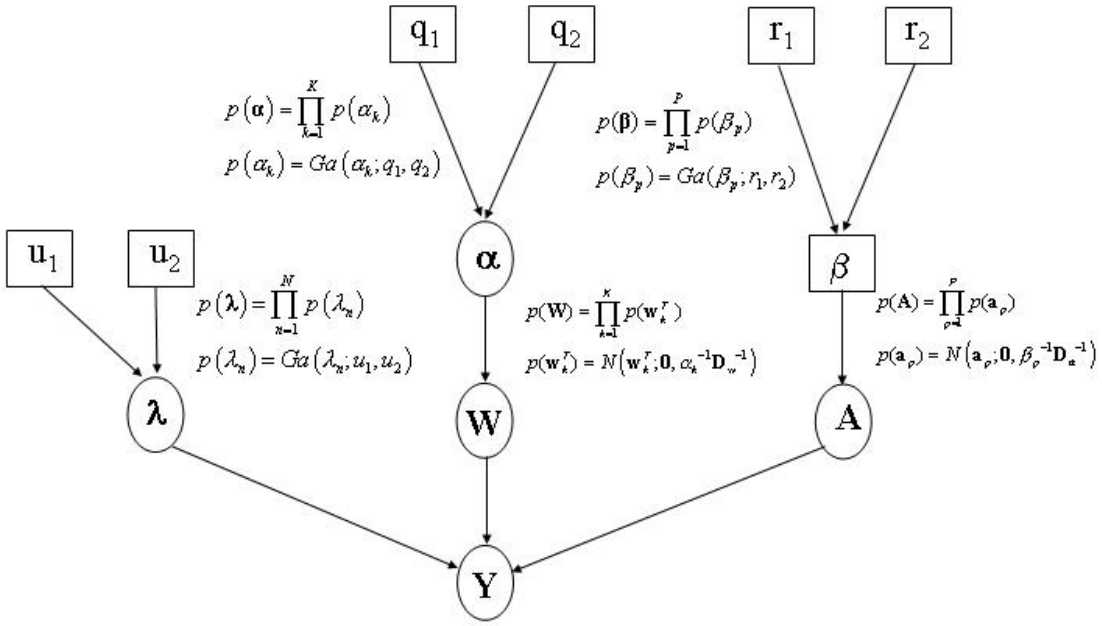


Figure 1. *The figure shows the probabilistic dependencies underlying our generative model for fMRI data. The quantities in square brackets are constants and those in circles are random variables. The spatial regularization coefficients* $\boldsymbol{\alpha}$ *constrain the regression coefficients* $\boldsymbol{W}$*, and the spatial regularization coefficients* $\boldsymbol{\beta}$ *constrain the AR coefficients* $\boldsymbol{A}$*. The parameters* $\boldsymbol{\lambda}$ *control the observation noise precision at each voxel. The graph*

shows that the joint probability of parameters and data can be written

$$p(\mathbf{Y},\mathbf{W},\mathbf{A},\boldsymbol{\lambda},\boldsymbol{\alpha},\boldsymbol{\beta}) = p(\mathbf{Y}\,|\,\mathbf{W},\mathbf{A},\boldsymbol{\lambda})\,p(\mathbf{W}\,|\,\boldsymbol{\alpha})\,p(\mathbf{A}\,|\,\boldsymbol{\beta})\,p(\boldsymbol{\lambda}\,|\,u_1,u_2)\,p(\boldsymbol{\alpha}\,|\,q_1,q_2)\,p(\boldsymbol{\beta}\,|\,r_1,r_2)$$

where the first term is the likelihood and the other terms are priors.

The corresponding update equations for the approximate posterior are shown in Figure 2.

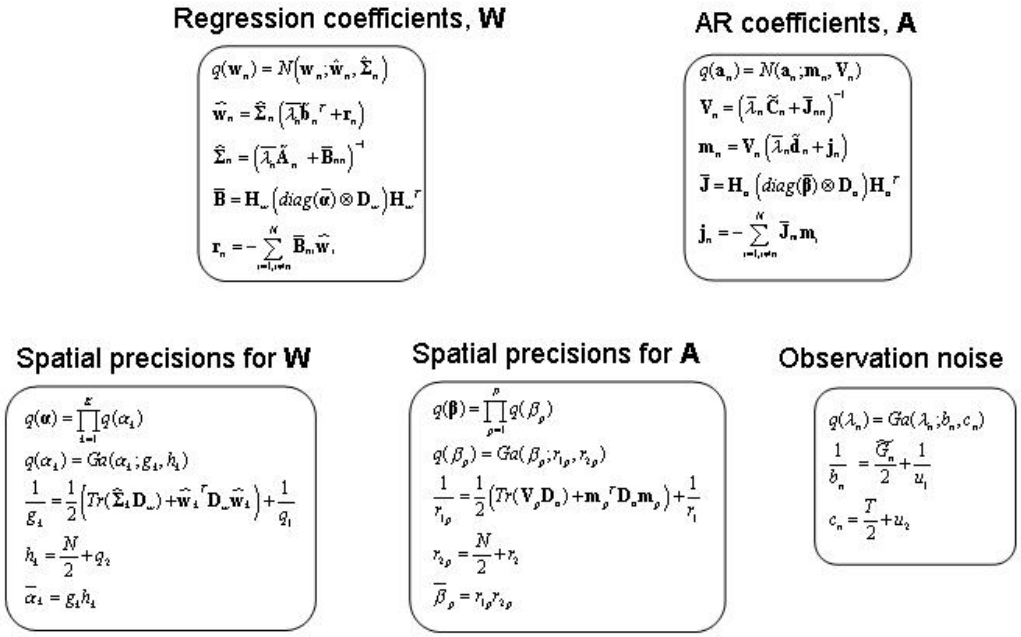


Figure 2. *Approximate posterior and summary of VB-update equations*

Cross covariance formulae

The formulae in the appendix of (Penny et al., 2003) for computing the quantities $\{\tilde{\mathbf{A}},\tilde{\mathbf{b}},\tilde{\mathbf{C}},\tilde{\mathbf{d}},\tilde{g}\}$ (equations 63, 64, 50 and 77 respectively) though correct (apart from a few missing transpose operators here and there !) are computationally inefficient. This is

because of the sums over time. For the lengths of time series typical in fMRI (eg. 300-400 scans) this creates a bottleneck when implementing the algorithm in MATLAB. The equations, which contain terms only up to second order (ie. quadratic), can however be re-arranged to isolate the sums over time so that they can be pre-computed. This effectively amounts to computing the following cross-covariances. This first set of terms depends on the design matrix only and therefore can be pre-computed for the whole volume

$$\mathbf{G}_x = \mathbf{X}^T\mathbf{X}$$
$$[k \times k]$$
$$\mathbf{S} = Cov(\tilde{\mathbf{X}}_t, \tilde{\mathbf{X}}_t)$$
$$[k^2 \times p^2]$$
$$\mathbf{R}_1 = Cov(\tilde{\mathbf{X}}_t, \mathbf{x}_t)$$
$$[k^2 \times p]$$

where the *Cov()* operator only involves terms from *p+1* to *T* where *p* is the order of the AR model and *T* is the number of time points, $\mathbf{X}$ is the design matrix, and $\tilde{\mathbf{X}}$ is the embedded design matrix. Underneath each equation is the dimension into which the result is re-shaped into (where necessary). The following terms depend on the design matrix and the data and therefore can be pre-computed for each slice

$$\mathbf{R}_{xy} = Cov(\mathbf{d}_t, \mathbf{x}_t)$$
$$[p \times k]$$
$$\mathbf{D} = Cov(\mathbf{d}_t, \tilde{\mathbf{X}}_t)$$
$$[k \times p^2]$$
$$\mathbf{G}_{xy} = Cov(\mathbf{y}_t, \tilde{\mathbf{X}}_t)$$
$$[p \times k]$$
$$\mathbf{G}_y = Cov(\mathbf{d}_t, \mathbf{d}_t^T)$$
$$[p \times p]$$
$$\mathbf{g}_y = Cov(\mathbf{y}_t, \mathbf{d}_t)$$
$$[p \times 1]$$
$$\mathbf{g}_{xy} = Cov(\mathbf{X}_t^T, \mathbf{y}_t)$$
$$[k \times 1]$$

where **y** is the fMRI time series and **d** is the embedded time series. The new update formulae can then be related to these quantities. For T=400, the new formulae are about 200 times quicker.

Approximating Posterior Covariance Matrices

In the SPM implementation of this algorithm we do not store full covariance matrices for each voxel as this would require too much disk space. Instead we store the posterior standard deviations of parameter estimates

$$d_n = \sqrt{diag(\hat{\Sigma}_n)} \tag{1}$$

(effectively images of error bars), AR coefficents, a_n, and noise standard deviation, $\sqrt{1/\lambda_n}$. These are stored in the files `SDbeta_000k.img`, SessM_`AR_000p.img` and SessM_`SDerror.img` where M is the session number. The approximate posterior covariances are then formed using a Taylor series expansion as follows. For each slice, *s*, we first compute the averages

$$\begin{aligned} \lambda_s &= \langle \lambda_n \rangle \\ a_s &= \langle a_n \rangle \\ V_s &= \langle V_n \rangle \\ b_s &= \langle B_{nn} \rangle \end{aligned} \tag{2}$$

where V_n is the posterior covariance of AR coefficients, and B_{nn} is the spatial precision, and from these compute a slice-specific error covariance matrix, $\hat{\Sigma}_s$. This is then normalised to produce a slice-specific error correlation matrix, R_s. The error correlation matrix at voxel *n* is then approximated using

$$R_n = R_s + \frac{dR_s}{d\lambda_s}(\lambda_n - \lambda_s) + \sum_{p=1}^{P} \frac{dR_s}{da_s(i)} \left(a_n(i) - a_s(i)\right) \tag{3}$$

where Jacobian matrices are stored for each slice. In SPM the slice-specific information is held in `SPM.Sess(M).PPM.slice(z).mean`. Finally the approximate covariance at voxel *n* is formed using

$$C_n = (d_n d_n^T) .* R_n \quad (4)$$

This approximate covariance is used when inferences are made about contrasts of parameter estimates. Figure 3 plots the resulting approximate variance versus the true variance for two contrasts from the face fMRI data sets (see Penny et al. 2005).

Contrasts

If contrasts are specified beforehand (when the batch mode is used) then PPM inference is based on the posterior covariances as estimated using the VB algorithm. This is implemented by creating contrast images, `con*.img`, and contrast standard deviation (sd) images, `con_sd*.img`, and updating them as the parameters in each slice are estimated. These are then used by the contrast manager later on. This 'pre-computation' of the contrast and contrast SD makes the specification of PPMs with different effect-size thresholds computationally efficient.

If contrasts are specified post-hoc (ie. after parameter estimation by `spm_spm_vb.m`) then PPMs are based on posterior covariances that are approximated using the Taylor series approximation described in the previous section.

Multiple Sessions

Multiple sessions are handled by modelling the data in each session separately and then concatenating the parameter estimates into a single larger model. This then, as

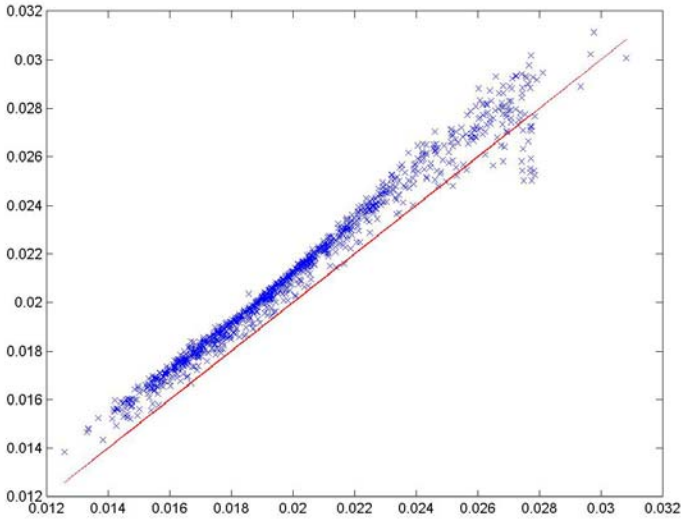
usual, allows for contrasts that span different sessions, so that conditions that span multiple sessions can be assessed. For contrasts that are specified prior to estimation, inference is based on the posterior covariances as estimated using the VB algorithm. This is implemented by creating contrast images (and contrast standard deviation images) as described above. For contrasts specified post hoc inference is based on the Taylor series approximation to the posterior covariance described earlier. This uses session-specific AR and noise precision parameters which are stored in `SessM_AR_000p.img` and `SessM_SDerror.img` where M is the session number.

High Pass Filtering

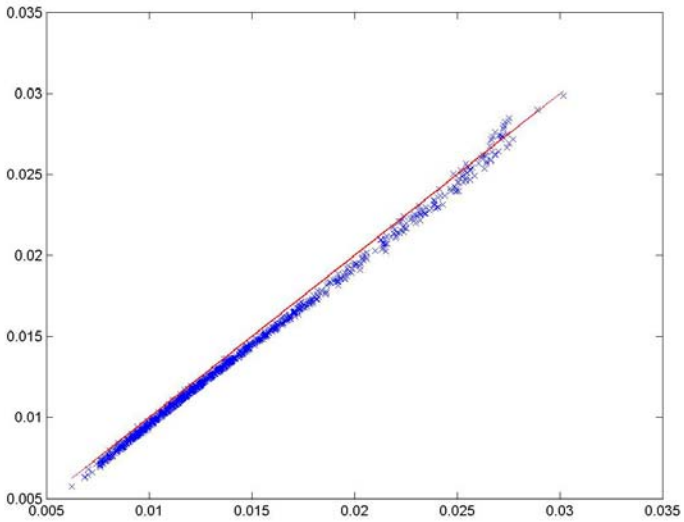
High pass filtering is implemented by filtering the data, y, within the function `spm_spm_vb.m` before passing it to the model fitting routine, `spm_vb_glmar.m`. This is done by forming the matrix $R_0=I-X_0X_0^+$, where X_0 is the matrix of DCT basis functions, and filtering using $y_{\_filtered}=R_0y$.

Acknowledgements

The authors are supported by the Wellcome Trust. The authors would also like to thank Rik Henson, Felix Blankenburg and Chloe Hutton.



(a)



(b)

Figure 3. **Approximating the Posterior Covariance**. Plots of approximate variance versus true variance for the face fMRI data and contrasts (a) main effect of faces and (b) main effect of fame. Crosses mark values at each of the voxels in slice *z=10* and

the straight line shows $y=x$. The second contrast, being a differential contrast, shows smaller error.

Reference List

1. Penny,W., Kiebel,S., and Friston,K. (2003). Variational Bayesian inference for fMRI time series. Neuroimage. *19*, 727-741.

2. Penny,W., Trujillo-Barreto,N.J., and Friston,K. (2005) Bayesian fMRI time series analysis with spatial priors. Neuroimage. 24(2), 350-362.