CHAPTER 11

Hierarchical Models

W. Penny and R. Henson

INTRODUCTION

Hierarchical models are central to many current analyses of functional imaging data including random effects analysis (Chapter 12), electroencephalographic (EEG) source localization (Chapter 28 to 30) and spatiotemporal models of imaging data (Chapters 25 and 26 and Friston *et al*., 2002b). These hierarchical models posit linear relations among variables with error terms that are Gaussian. The general linear model (GLM), which to date has been so central to the analysis of functional imaging data, is a special case of these hierarchical models consisting of just a single layer.

Model fitting and statistical inference for hierarchical models can be implemented using a parametric empirical Bayes (PEB) algorithm described in Chapter 24 and in Friston *et al*. (2002a). The algorithm is sufficiently general to accomodate multiple hierarchical levels and allows for the error covariances to take on arbitrary form. This generality is particularly appealing as it renders the method applicable to a wide variety of modelling scenarios. Because of this generality, however, and the complexity of scenarios in which the method is applied, readers wishing to learn about PEB for the first time are advised to read this chapter first. Chapter 24 then goes on to discuss the more general case. It also shows that the variance components that are estimated using PEB, can also be estimated using an algorithm from classical statistics called restricted maximum likelihood (ReML).

In this chapter, we provide an introduction to hierarchical models and focus on some relatively simple examples. This chapter covers the relevant mathematics and numerical examples are presented in the following chapter. Each model and PEB algorithm we present is a special case of that described in Friston *et al*. (2002a). While there are a number of tutorials on hierarchical modelling (Lee, 1997; Carlin and Louis, 2000) what we describe here has been tailored for functional imaging applications. We also note that a tutorial on hierarchical models is, to our minds, also a tutorial on Bayesian inference, as higher levels act as priors for parameters in lower levels. Readers are therefore encouraged to also consult background texts on Bayesian inference, such as Gelman (1995).

This chapter focuses on two-level models and shows how one computes the posterior distributions over the first- and second-level parameters. These are derived, initially, for completely general designs and error covariance matrices. We then consider two special cases: (i) models with equal error variances; and (ii) separable models. We assume initially that the covariance components are known, and then in the section on PEB, we show how they can be estimated. A numerical example is then given showing PEB in action. The chapter then describes how Bayesian inference can be implemented for hierarchical models with arbitrary probability distributions (e.g. non-Gaussian), using the belief propagation algorithm. We close with a discussion.

In what follows, the notation $\mathrm{N}(m, \Sigma)$ denotes a uni/multivariate normal distribution with mean m and variance/covariance Σ and lower-case ps denote probability densities. Upper case letters denote matrices, lower case denote column vectors and x^T denotes the transpose of x. We will also make extensive use of the normal density, i.e. if $p(x) = \mathrm{N}(m, \Sigma)$ then:

$$p(x) \propto \exp\left(-\frac{1}{2}(x-m)^T \Sigma^{-1}(x-m)\right) \qquad \mathbf{11.1}$$

We also use $\mathrm{Var}[]$ to denote variance, $\otimes$ to denote the Kronecker product and X^+ to denote the pseudo-inverse.

Statistical Parametric Mapping, by Karl Friston et al.
ISBN–13: 978-0-12-372560-8 ISBN–10: 0-12-372560-7

TWO-LEVEL MODELS

We consider two-level linear Gaussian models of the form:

$$y = Xw + e$$
$$w = M\mu + z \qquad \textbf{11.2}$$

where the errors are zero mean Gaussian with covariances Cov $[e] = C$ and $\text{Cov}[z] = P$. The model is shown graphically in Figure 11.1. The column vectors y and w have K and N entries respectively. The vectors w and μ are the first- and second-level parameters and X and M are the first- and second-level design matrices. Models of this form have been used in functional imaging. For example, in random effects analysis, the second level models describe the variation of subject effect sizes about a population effect size, μ. In Bayesian inference with shrinkage priors, the second-level models variation of effect-size over voxels around a whole-brain mean effect size of $\mu - 0$ (i.e. for a given cognitive challenge, the response of a voxel chosen at random is, on average, zero). See, for example, Friston *et al*. (2002b).

The aim of Bayesian inference is to make inferences about w and μ based on the posterior distributions $p(w|y)$ and $p(\mu|y)$. These can be derived as follows. We first note that the above equations specify the likelihood and prior probability distributions:

$$p(y|w) \propto \exp\left(-\frac{1}{2}(y - Xw)^T C^{-1}(y - Xw)\right) \qquad \textbf{11.3}$$
$$p(w) \propto \exp\left(-\frac{1}{2}(w - M\mu)^T P^{-1}(w - M\mu)\right)$$

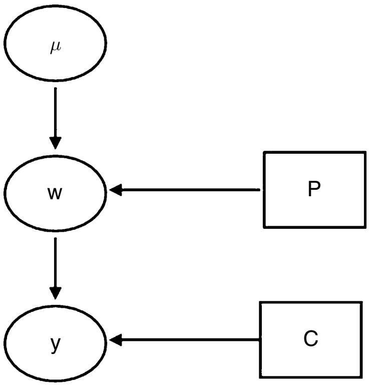

FIGURE 11.1 Two-level hierarchical model. The data y are explained as deriving from an effect w and a zero-mean Gaussian random variation with covariance C. The effects w in turn are random effects deriving from a superordinate effect μ and zero-mean Gaussian random variation with covariance P. The goal of Bayesian inference is to make inferences about μ and w from the posterior distributions $p(\mu|y)$ and $p(w|y)$.

The posterior distribution is then:

$$p(w|y) \propto p(y|w)p(w) \qquad \textbf{11.4}$$

Taking logs and keeping only those terms that depend on w gives:

$$\begin{aligned}\log p(w|y) &= -\frac{1}{2}(y - Xw)^T C^{-1}(y - Xw) \qquad \textbf{11.5}\\ &\quad -\frac{1}{2}(w - M\mu)^T P^{-1}(w - M\mu) + ..\\ &= -\frac{1}{2}w^T(X^T C^{-1}X + P^{-1})w\\ &\quad + w^T(X^T C^{-1}y + P^{-1}M\mu) + ..\end{aligned}$$

Taking logs of the Gaussian density $p(x)$ in Eqn. **11.1** and keeping only those terms that depend on x gives:

$$\log p(x) = -\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}m + .. \qquad \textbf{11.6}$$

Comparing Eqn. **11.5** with terms in the above equation shows that:

$$\begin{aligned}p(w|y) &= \text{N}(m, \Sigma)\\ \Sigma^{-1} &= X^T C^{-1}X + P^{-1}\\ m &= \Sigma(X^T C^{-1}y + P^{-1}M\mu) \qquad \textbf{11.7}\end{aligned}$$

The posterior distribution over the second-level coefficient is given by Bayes' rule as:

$$p(\mu|y) = \frac{p(y|\mu)p(\mu)}{p(y)} \qquad \textbf{11.8}$$

However, because we do not have a prior $p(\mu)$, this posterior distribution becomes identical to the likelihood term, $p(y|\mu)$, which can be found by eliminating the first-level parameters from our two equations, i.e. by substituting the second level equation into the first giving:

$$y = XM\mu + Xz + e \qquad \textbf{11.9}$$

which can be written as:

$$y = \tilde{X}\mu + \tilde{e} \qquad \textbf{11.10}$$

where $\tilde{X} = XM$ and $\tilde{e} = Xz + e$. The solution to Eqn. **11.10** then gives:

$$\begin{aligned}p(\mu|y) &= \text{N}(\hat{\mu}, \Sigma_\mu)\\ \hat{\mu} &= (\tilde{X}^T\tilde{C}^{-1}\tilde{X})^{-1}\tilde{X}^T\tilde{C}^{-1}y \qquad \textbf{11.11}\\ \Sigma_\mu &= (\tilde{X}^T\tilde{C}^{-1}\tilde{X})^{-1}\end{aligned}$$

where the covariance term:

$$\begin{aligned}\tilde{C} &= \mathrm{Cov}[\tilde{e}] \\ &= XPX^T + C\end{aligned} \tag{11.12}$$

We have now achieved our first goal, the posterior distributions of first- and second-level parameters being expressed in terms of the data, design and error-covariance matrices. We now consider the special cases of sensor fusion, equal variance models and separable models.

Sensor fusion

The first special case is the univariate model:

$$\begin{aligned}y &= w + e \\ w &= \mu + z\end{aligned} \tag{11.13}$$

with a single scalar data point, y, and variances $C = 1/\beta,\ P = 1/\alpha$ specified in terms of the data precision β and the prior precision α (the 'precision' is the inverse variance). Plugging these values into Eqn. **11.7** gives

$$\begin{aligned}p(w|y) &= \mathrm{N}(m, \lambda^{-1}) \\ \lambda &= \beta + \alpha \\ m &= \frac{\beta}{\lambda}y + \frac{\alpha}{\lambda}\mu\end{aligned} \tag{11.14}$$

Despite its simplicity, this model possesses two important features of Bayesian learning in linear-Gaussian models. The first is that 'precisions add' – the posterior precision is the sum of the data precision and the prior precision. The second is that the posterior mean is the sum of the data mean and the prior mean, each weighted by their relative precisions. A numerical example is shown in Figure 11.2.

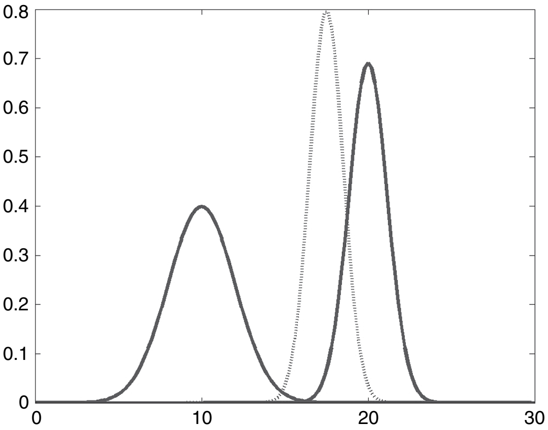

FIGURE 11.2 Bayes rule for univariate Gaussians. The two solid curves show the probability densities for the prior $p(w) = \mathrm{N}(\mu, \alpha^{-1})$ with $\mu = 20$ and $\alpha = 1$ and the likelihood $p(y|w) = \mathrm{N}(w, \beta^{-1})$ with $w = 25$ and $\beta = 3$. The dotted curve shows the posterior distribution, $p(w|y) = \mathrm{N}(m, \lambda^{-1})$ with $m = 23.75$ and $\lambda = 4$, as computed from Eqn. **11.14**. The posterior distribution is closer to the likelihood because the likelihood has higher precision.

Equal variance

This special case is a two-level multivariate model as in Eqn. **11.2**, but with isotropic covariances at both the first and second levels. We have $C = \beta^{-1}I_K$ and $P = \alpha^{-1}I_N$. This means that observations are independent and have the same error variance. This is an example of the errors being independent and identically distributed (IID), where, in this case, the distribution is a zero-mean Gaussian having a particular variance. In this chapter, we will also use the term 'sphericity' for any model with IID errors. Models without IID errors will have 'non-sphericity' (as an aside we note that IID is not actually a requirement of 'sphericity' and readers looking for a precise definition are referred to Winer *et al*. (1991) and to Chapter 10).

On a further point of terminology, the unknown vectors w and μ will be referred to as 'parameters', whereas variables related to error covariances will be called 'hyperparameters'. The variables α and β are therefore hyperparameters. The posterior distribution over first level parameters is given by:

$$\begin{aligned}p(w|y) &= \mathrm{N}(\hat{w}, \hat{\Sigma}) \\ \hat{\Sigma} &= (\beta X^T X + \alpha I_N)^{-1} \\ \hat{w} &= \hat{\Sigma}\left(\beta X^T y + \alpha M\mu\right)\end{aligned} \tag{11.15}$$

Note that if $\alpha = 0$, we recover the maximum likelihood estimate:

$$\hat{w}_{ML} = (X^T X)^{-1} X^T y \tag{11.16}$$

This is the familiar ordinary least squares (OLS) estimate used in the GLM (Holmes *et al*., 1997). The posterior distribution over the second level parameters is given by Eqn. **11.12** with:

$$\tilde{C} = \beta^{-1} I_K + \alpha^{-1} X X^T \tag{11.17}$$

Separable model

We now consider 'separable models' which can be used, for example, for random effects analysis. Figure 11.3

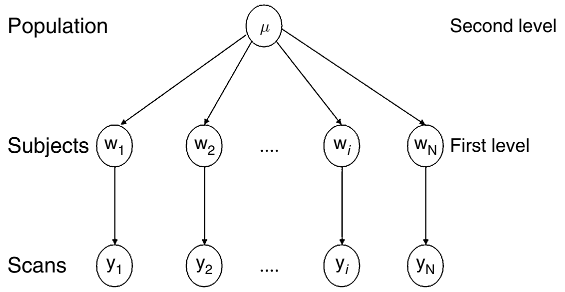

FIGURE 11.3 Generative model for random effects analysis.

shows the corresponding generative model. In these models, the first-level splits into N separate submodels. For each submodel, i, there are n_i observations. These form the n_i-element vector y_i giving information about the parameter w_i via the design vector x_i. For functional magnetic resonance imaging (fMRI) analysis, these design vectors comprise stimulus functions, e.g. boxcars or delta functions, convolved with an assumed haemodynamic response. The overall first-level design matrix X then has a block-diagonal form $X = \mathrm{blkdiag}(x_1, \ldots, x_i, \ldots, x_N)$ and the covariance is given by $C = \mathrm{diag}[\beta_1 1_{n_1}^T, \ldots, \beta_i 1_{n_i}^T, \ldots, \beta_N 1_{n_N}^T]$, where 1_n is a column vector of 1s with n entries. For example, for $N = 3$ groups with $n_1 = 2$, $n_2 = 3$ and $n_3 = 2$ observations in each group:

$$X = \begin{bmatrix} x_1(1) & 0 & 0 \\ x_1(2) & 0 & 0 \\ 0 & x_2(1) & 0 \\ 0 & x_2(2) & 0 \\ 0 & x_2(3) & 0 \\ 0 & 0 & x_3(1) \\ 0 & 0 & x_3(2) \end{bmatrix} \qquad \mathbf{11.18}$$

and $C^{-1} = \mathrm{diag}[\beta_1, \beta_1, \beta_2, \beta_2, \beta_2, \beta_3, \beta_3]$. The covariance at the second level is $P = \alpha^{-1} I_N$, as before, and we also assume that the second level design matrix is a column of 1s, $M = 1_N$. The posterior distribution over first level parameters is found by substituting X and C into Eqn. **11.7**. This gives a distribution which factorizes over the different first level coefficients such that:

$$p(w|y) = \prod_{i=1}^{N} p(w_i|y) \qquad \mathbf{11.19}$$

$$p(w_i|y) = N(\hat{w}_i, \hat{\Sigma}_{ii})$$

$$\hat{\Sigma}_{ii}^{-1} = \beta_i x_i^T x_i + \alpha$$

$$\hat{w}_i = \hat{\Sigma}_{ii} \beta_i x_i^T y_i + \hat{\Sigma}_{ii} \alpha \mu$$

The posterior distribution over second level parameters is, from Eqn. **11.12**, given by:

$$p(\mu|y) = \mathrm{N}(\hat{\mu}, \sigma_\mu^2) \qquad \mathbf{11.20}$$

$$\sigma_\mu^2 = \frac{1}{\sum_{i=1}^{N} x_i^T (\alpha^{-1} x_i x_i^T + \beta_i^{-1})^{-1} x_i}$$

$$\hat{\mu} = \sigma_\mu^2 \sum_{i=1}^{N} x_i^T (\alpha^{-1} x_i x_i^T + \beta_i^{-1})^{-1} y_i$$

We note that, in the absence of any second level variability, i.e. $\alpha \to \infty$, the estimate $\hat{\mu}$ reduces to the mean of the first level coefficients weighted by their precision:

$$\hat{\mu} = \frac{\sum_i \beta_i x_i^T y_i}{\sum_i \beta_i x_i^T x_i} \qquad \mathbf{11.21}$$

PARAMETRIC EMPIRICAL BAYES

In the previous section, we have shown how to compute the posterior distributions $p(w|y)$ and $p(\mu|y)$. As can be seen from Eqns **11.7** and **11.11**, however, these equations depend on covariances P and C. In this section, we show how covariance components can be estimated for the special cases of equal variance models and separable models.

In Friston *et al.* (2002a), the covariances are decomposed using:

$$C = \sum_j \lambda_j^1 Q_j^1 \qquad \mathbf{11.22}$$

$$P = \sum_j \lambda_j^2 Q_j^2$$

where Q_j^1 and Q_j^2 are basis functions that are specified by the modeller, depending on the application in mind. For example, for analysis of fMRI data from a single subject, two basis functions are used, the first relating to error variance and the second relating to temporal autocorrelation (Friston *et al.*, 2002b). The hyperparameters $\lambda = [\{\lambda_j^1\}, \{\lambda_j^2\}]$ are unknown, but can be estimated using the PEB algorithm described in Friston *et al.* (2002a). Variants of this algorithm are known as the *evidence framework* (Mackay, 1992) or *maximum likelihood II (ML-II)* (Berger, 1985). The PEB algorithm is also referred to as simply *empirical Bayes*, but we use the term PEB to differentiate it from the non-parametric empirical Bayes methods described in Carlin and Louis (2000). The hyperparameters are set so as to maximize the evidence (also known as the marginal likelihood):

$$p(y|\lambda) = \int p(y|w, \lambda) p(w|\lambda) dw \qquad \mathbf{11.23}$$

This is the likelihood of the data after we have integrated out the first-level parameters. For the two multivariate special cases described above, by substituting in our expressions for the prior and likelihood, integrating, taking logs and then setting the derivatives to zero, we can derive a set of update rules for the hyperparameters. These derivations are provided in the following two sections.

Equal variance

For the equal variance model, the objective function is:

$$p(y|\alpha,\beta) = \int p(y|w,\beta)p(w|\alpha)dw \tag{11.24}$$

Substituting in expressions for the likelihood and prior gives:

$$\begin{aligned} p(y|\alpha,\beta) &= \left(\frac{\beta}{2\pi}\right)^{K/2}\left(\frac{\alpha}{2\pi}\right)^{N/2} \\ &\quad\times\int \exp\left(-\frac{\beta}{2}e(w)^Te(w) - \frac{\alpha}{2}z(w)^Tz(w)\right)dw \end{aligned}$$

where $e(w) = y - Xw$ and $z(w) = w - M\mu$. By rearranging the terms in the exponent (and keeping all of them, unlike before) where we were only interested in w-dependent terms) the integral can be written as:

$$\begin{aligned} I &= \left[\int \exp\left(-\frac{1}{2}(w-\hat{w})^T\hat{\Sigma}^{-1}(w-\hat{w})\right)dw\right] \\ &\quad\times\left[\exp\left(-\frac{\beta}{2}e(\hat{w})^Te(\hat{w}) - \frac{\alpha}{2}z(\hat{w})^Tz(\hat{w})\right)\right] \end{aligned} \tag{11.25}$$

where the second term is not dependent on w. The first factor is then simply given by the normalizing constant of the multivariate Gaussian density:

$$(2\pi)^{N/2}|\hat{\Sigma}|^{1/2} \tag{11.26}$$

Hence,

$$\begin{aligned} p(y|\alpha,\beta) &= \left(\frac{\beta}{2\pi}\right)^{K/2}\alpha^{N/2}|\hat{\Sigma}|^{1/2} \\ &\quad\times\exp\left(-\frac{\beta}{2}e(\hat{w})^Te(\hat{w}) - \frac{\alpha}{2}z(\hat{w})^Tz(\hat{w})\right) \end{aligned}$$

where $|\hat{\Sigma}|$ denotes the determinant of $\hat{\Sigma}$. Taking logs gives the 'log-evidence':

$$\begin{aligned} F &= \frac{K}{2}\log\frac{\beta}{2\pi} + \frac{N}{2}\log\alpha + \frac{1}{2}\log|\hat{\Sigma}| \\ &\quad-\frac{\beta}{2}e(\hat{w})^Te(\hat{w}) - \frac{\alpha}{2}z(\hat{w})^Tz(\hat{w}) \end{aligned} \tag{11.27}$$

To find equations for updating the hyperparameters, we must differentiate F with respect to α and β and set the derivative to zero. The only possibly problematic term is the log-determinant, but this can be differentiated by first noting that the inverse covariance is given by:

$$\hat{\Sigma}^{-1} = \beta X^TX + \alpha I_N \tag{11.28}$$

If λ_j are the eigenvalues of the first term, then the eigenvalues of $\hat{\Sigma}^{-1}$ are $\lambda_j + \alpha$. Hence,

$$|\hat{\Sigma}^{-1}| = \prod_j(\lambda_j + \alpha) \tag{11.29}$$

$$|\hat{\Sigma}| = \frac{1}{\prod_j(\lambda_j+\alpha)}$$

$$\log|\hat{\Sigma}| = -\sum_j\log(\lambda_j+\alpha)$$

$$\frac{\partial}{\partial\alpha}\log|\hat{\Sigma}| = -\sum_j\frac{1}{\lambda_j+\alpha}$$

Setting the derivative $\partial F/\partial\alpha$ to zero then gives:

$$\begin{aligned} \alpha z(\hat{w})^Tz(\hat{w}) &= N - \sum_j\frac{\alpha}{\lambda_j+\alpha} \\ &= \sum_j\frac{\lambda_j+\alpha}{\lambda_j+\alpha} - \sum_j\frac{\alpha}{\lambda_j+\alpha} \\ &= \sum_j\frac{\lambda_j}{\lambda_j+\alpha} \end{aligned} \tag{11.30}$$

This is an implicit equation in α which leads to the following update rule. We first define the quantity γ which is computed from the 'old' value of α:

$$\gamma = \sum_{j=1}^N\frac{\lambda_j}{\lambda_j+\alpha} \tag{11.31}$$

and then let:

$$\frac{1}{\alpha} = \frac{z(\hat{w})^Tz(\hat{w})}{\gamma} \tag{11.32}$$

The update for β is derived by first noting that the eigenvalues λ_j are linearly dependent on β. Hence,

$$\frac{\partial\lambda_j}{\partial\beta} = \frac{\lambda_j}{\beta} \tag{11.33}$$

The derivative of the log-determinant is then given by:

$$\frac{\partial}{\partial\beta}\log|\hat{\Sigma}^{-1}| = \frac{1}{\beta}\sum_j\frac{\lambda_j}{\lambda_j+\alpha} \tag{11.34}$$

which leads to the update:

$$\frac{1}{\beta} = \frac{e(\hat{w})^T e(\hat{w})}{K - \gamma} \qquad \textbf{11.35}$$

The PEB algorithm consists of iterating the update rules in Eqn. **11.31**, Eqn. **11.32**, Eqn. **11.35** and the posterior estimates in Eqn. **11.15**, until convergence.

The update rules in Eqn. **11.31**, Eqn. **11.32** and Eqn. **11.35** can be interpreted as follows. For every j for which $\lambda_j >> \alpha$, the quantity γ increases by 1. As α is the prior precision and λ_j is the data precision (of the jth 'eigencoefficient'), γ therefore measures the number of parameters that are determined by the data. Given K data points, the quantity $K - \gamma$ therefore corresponds to the number of degrees of freedom in the data set. The variances α^{-1} and β^{-1} are then updated based on the sum of squares divided by the appropriate degrees of freedom.

Separable models

For separable models, the objective function is:

$$p(y|\alpha, \{\beta_i\}) = \int p(y|w, \{\beta_i\}) p(w|\alpha) dw \qquad \textbf{11.36}$$

Because the second-level here is the same as for the equal variance case, so is the update for alpha. The updates for β_i are derived in a similar manner as before, but we also make use of the fact that the first-level posterior distribution factorizes (see Eqn. **11.20**). This decouples the updates for each β_i and results in the following PEB algorithm:

$$\begin{aligned}
\hat{e}_i &= y_i - \hat{w}_i x_i \qquad \textbf{11.37}\\
\hat{z}_i &= \hat{w}_i - \hat{\mu}\\
\lambda_i &= \beta_i x_i^T x_i\\
\gamma_i &= \frac{\lambda_i}{\lambda_i + \alpha}\\
\gamma &= \sum_i \gamma_i\\
\beta_i &= (n_i - \gamma_i)/\hat{e}_i^T \hat{e}_i\\
\alpha &= \gamma / \hat{z}^T \hat{z}\\
\hat{w}_i &= (\beta_i x_i^T y_i + \alpha\mu)/(\lambda_i + \alpha)\\
d_i &= (\alpha^{-1} x_i x_i^T + \beta_i^{-1} I_{n_i})^{-1}\\
\sigma_\mu^2 &= 1/(\sum_i x_i^T d_i x_i)\\
\hat{\mu} &= \sigma_\mu^2 \sum_i x_i^T d_i y_i
\end{aligned}$$

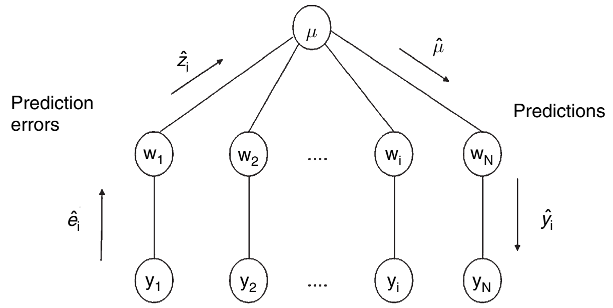

FIGURE 11.4 Part of the PEB algorithm for separable models requires the upwards propagation of prediction errors and downwards propagation of predictions. This passing of messages between nodes in the hierarchy is a special case of the more general belief propagation algorithm referred to in Figure 11.5.

Initial values for $\hat{w}_i$ and β_i are set using OLS, $\hat{\mu}$ is initially set to the mean of $\hat{w}_i$ and α is initially set to 0. The equations are then iterated until convergence (in our examples in Chapter 12, we never required more than ten iterations). While the above updates may seem somewhat complex, they can perhaps be better understood in terms of messages passing among nodes in a hierarchical network. This is shown in Figure 11.4 for the 'prediction' and 'prediction error' variables.

The PEB algorithms we have described show how Bayesian inference can take place when the variance components are unknown (in the previous section, we assumed the variance components were known). An application of this PEB algorithm to random effects analysis is provided in the next chapter. We now provide a brief numerical example demonstrating the iterations with PEB updates.

NUMERICAL EXAMPLE

This numerical example caricatures the use of PEB for estimating effect sizes from functional imaging data described in Chapter 23. The approach uses a 'global shrinkage prior' which embodies a prior belief that, across the brain: (i) the average effect is zero, $\mu = 0$; and (ii) the variability of responses follows a Gaussian distribution with precision α. Mathematically, we can write $p(w_i) = \mathrm{N}(0, \alpha^{-1})$. Plate 5(a) (see colour plate sections) shows effect sizes generated from this prior for a $N = 20$-voxel brain and $\alpha = 1$.

Chapter 23 allows for multiple effects to be expressed at each voxel and for position emission tomography (PET)/fMRI data to be related to effect sizes using the full flexibility of general linear models. Here, we just assume that data at each voxel are normally distributed about the effect size at that voxel. That is, $p(y_i|w_i) = \mathsf{N}(w_i, \beta_i^{-1})$. Plate 5(b) shows $n_i = 10$ data points at each voxel generated from this likelihood. We have allowed the observation noise precision β_i to be different at each voxel. Voxels 2, 15 and 18, for example, have noisier data than others.

Effect sizes were then estimated from these data using maximum likelihood (ML) and PEB. ML estimates are shown in Plate 5(c) and (d). These are simply computed as the mean value observed at each voxel. PEB was implemented using the updates in Eqn. **11.37** with $\mu = 0$ and $x_i = 1_{n_i}$ and initialized with $\alpha = 0$ and β_i and $\hat{w}_i$ set to ML-estimated values.

Eqn. **11.37** was then iterated, resulting in effect size estimates shown in Plate 6 before iterations one, three, five and seven. These estimates seem rather stable after only two or three iterations. Only the effects at voxels 5 and 15 seem markedly changed between iterations three and seven. The corresponding estimates of α were 0, 0.82, 0.91 and 0.95, showing convergence to the true prior response precision value of 1.

It is well known that PEB provides estimates that are, on average, more accurate than ML. Here, we quantify this using, σ_s, the standard deviation across voxels of the difference between the true and estimated effects. For ML, $\sigma_s = 0.71$ and for PEB, $\sigma_s = 0.34$. That PEB estimates are twice as accurate on average can be seen by comparing Plate 6(a) and (d). Of course, PEB is only better 'on average'. It does better at most voxels at the expense of being worse at a minority, for example, voxel 2. This trade-off is discussed further in Chapter 22.

PEB can do better than ML because it uses more information: here, the information that effects have a mean of zero across the brain and follow a Gaussian variability profile. This shows the power of Bayesian estimation, which combines prior information with data in an optimal way. In this example, a key parameter in this trade-off is the parameter γ_i which is computed as in Eqn. **11.37**. This quantity is the ratio of the data precision to the posterior precision. A value of 1 indicates that the estimated effect is determined solely by the data, as in ML. A value of 0 indicates the estimate is determined solely by the prior. For most voxels in our data set, we have $\gamma_i \approx 0.9$, but for the noisy voxels 2, 15 and 18, we have $\gamma_i \approx 0.5$. PEB thus relies more on prior information where data are unreliable.

PEB will only do better than ML if the prior is chosen appropriately. For functional imaging data, we will never know what the 'true prior' is, just as we will never know what the 'true model' is. But some priors and models are better than others, and there is a formal method for deciding between them. This is 'Bayesian model selection' and is described in Chapter 35.

Finally, we note that the prior used here does not use spatial information i.e. there is no notion that voxel 5 is 'next to' voxel 6. It turns out that for functional imaging data, spatial information is important. In Chapter 25, we describe Bayesian fMRI inference with spatial priors. Bayesian model selection shows that models with spatial priors are preferred to those without (Penny *et al.*, 2006).

BELIEF PROPAGATION

This chapter has focused on the special case of two-level models and Gaussian distributions. It is worthwhile noting that the general solution to inference in tree-structured hierarchical models, which holds for all distributions, is provided by the 'sum-product' or 'belief propagation' algorithm (Pearl, 1988; Jordan and Weiss, 2002). This is a message passing algorithm which aims to deliver the marginal distributions[1] at each point in the hierarchy. It does this by propagating evidence up the hierarchy and marginal distributions down. If the downward messages are passed after the upward messages have reached the top, then this is equivalent to propagating the posterior beliefs down the hierarchy. This is shown schematically in Figure 11.

This general solution is important as impacts on non-Gaussian and/or nonlinear hierarchical models. Of particular relevance are the models of inference in cortical hierarchies (Friston, 2003) referred to in later chapters of the book. In these models, evidence flows up the hierarchy, in the form of prediction errors, and marginal distributions flow down, in the form of predictions. Completion of the downward pass explains late components of event-related potentials which are correlated with, e.g. extra-classical receptive field effects (Friston, 2003). This general solution also motivates a data analysis approach known as Bayesian model averaging (BMA), described further in Chapter 35, where, e.g. x_3 in Figure 11.5 embodies assumptions about model structure. The downward pass of belief propagation then renders our final inferences independent of these assumptions. See Chapter 16 of Mackay, (2003) and Ghahramani, 1998) for further discussion of these issues.

[1] The probability distribution over a set of variables is known as the joint distribution. The distribution over a subset is known as the marginal distribution.

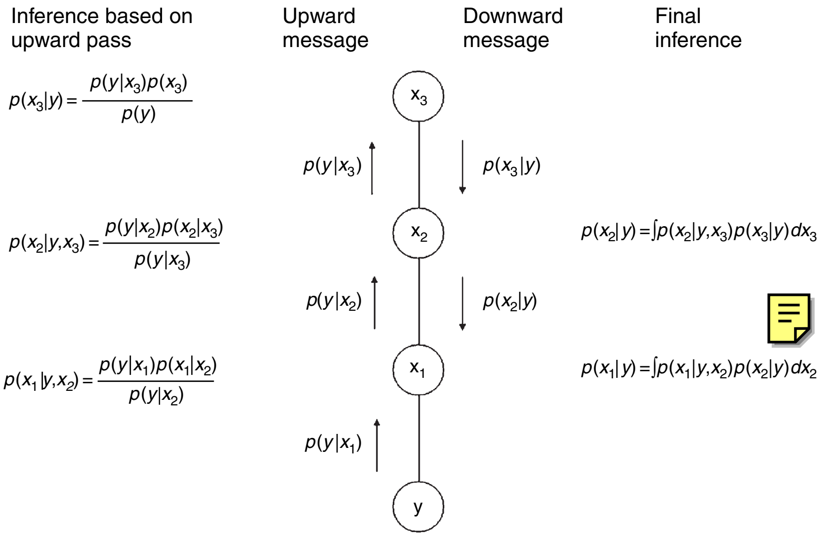
Inference based on upward pass
Upward message
Downward message
Final inference
$p(x_3|y) = \frac{p(y|x_3)p(x_3)}{p(y)}$
$p(x_2|y,x_3) = \frac{p(y|x_2)p(x_2|x_3)}{p(y|x_3)}$
$p(x_1|y,x_2) = \frac{p(y|x_1)p(x_1|x_2)}{p(y|x_2)}$
x_3
x_2
x_1
y
$p(y|x_3)$
$p(y|x_2)$
$p(y|x_1)$
$p(x_3|y)$
$p(x_2|y)$
$p(x_2|y) = \int p(x_2|y,x_3)p(x_3|y)dx_3$
$p(x_1|y) = \int p(x_1|y,x_2)p(x_2|y)dx_2$

FIGURE 11.5 Belief propagation for inference in hierarchical models. This algorithm is used to update the marginal densities, i.e. to update $p(x_i)$ to $p(x_i|y)$. Inferences based on purely the upward pass are contingent on variables in the layer above, whereas inferences based on upwards and downwards passes are not. Completion of the downward pass delivers the marginal density. Application of this algorithm to the two-level Gaussian model will produce the update Eqn. **11.7** and Eqn. **11.11**. More generally, this algorithm can be used for Bayesian model averaging, where e.g. x_3 embodies assumptions about model structure, and as a model of inference in cortical hierarchies, where e.g. completion of the downward pass explains extra-classical receptive field effects (Friston, 2003).

DISCUSSION

We have described Bayesian inference for some particular two-level linear-Gaussian hierarchical models. A key feature of Bayesian inference in this context is that the posterior distributions are Gaussian with precisions that are the sum of the data and prior precisions. The posterior means are the sum of the data and prior means, but each weighted according to their relative precision. With zero prior precision, two-level models reduce to a single-level model (i.e. a GLM) and Bayesian inference reduces to the familiar maximum-likelihood estimation scheme. With non-zero and, in general unknown, prior means and precisions, these parameters can be estimated using PEB. These covariance components can also be estimated using the ReML algorithm from classical statistics. The relation between PEB and ReML is discussed further in Chapter 22.

We have described two special cases of the PEB algorithm, one for equal variances and one for separable models. Both algorithms are special cases of a general approach described in Friston *et al.*, (2002a) and in Chapter 24. In these contexts, we have shown that PEB automatically partitions the total degrees of freedom (i.e. number of data points) into those to be used to estimate the hyperparameters of the prior distribution and those to be used to estimate hyperparameters of the likelihood distribution. The next chapter describes how PEB can be used in the context of random effects analysis.

REFERENCES

Berger JO, (1985) *Statistical decision theory and Bayesian analysis*. Springer-Verlag, [**11.1]

Carlin BP, Louis TA (2000) *Bayes and empirical Bayes methods for data analysis*. ~~Chapman and Hall,~~ [**11.2]

Friston KJ, Penny WD, Phillips C *et al*. (2002a) Classical and Bayesian inference in neuroimaging: theory. *NeuroImage* **16**: 465–83

Friston KJ, Glaser DE, Henson RNA *et al*. (2002b) Classical and Bayesian inference in neuroimaging: applications. *NeuroImage* **16**: 484–512

Friston K (2003) Learning and inference in the brain. *Neural Netw* **16**: 1325–52

Gelman A, Carlin JB, Stern HS *et al*. (1995) *Bayesian data analysis*. Chapman and Hall, Boca Raton

Ghahramani Z (1998) Learning dynamic bayesian networks. In *Adaptive processing of temporal information*, Giles CL, Gori M (ed). Springer-Verlag, [**11.3]

Holmes AP, Poline JB, Friston KJ (1997) Characterizing brain images with the general linear model. In *Human Brain Function*, Frackowiak RSJ, Friston KJ, Frith C *et al*. (eds). Academic Press, pp 59–84. [**11.4]

Jordan M, Weiss Y (2002) Graphical models: probabilistic inference. In *The Handbook of Brain Theory and Neural Networks*, Arbib M (ed.) MIT Press, [**11.5]

Lee PM (1997) *Bayesian statistics: an introduction*, 2nd edn. Arnold, [**11.6]

Mackay DJC (1992) Bayesian interpolation. *Neural Comput* **4**: 415–47

Mackay DJC (2003) *Information theory, inference and learning algorithms*. Cambridge University Press, Cambridge

Pearl J (1988) *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kauffman, [**11.7]

Penny WD, Flandin G, Trujillo-Barreto N (2006) Bayesian comparison of spatially regularised general linear models. *Hum Brain Mapp* [**11.8]

Winer BJ, Brown DR, Michels KM (1991) *Statistical principles in experimental design*. McGraw-Hill, [**11.9]