Mathematics for Brain Imaging

Will Penny

Functional Imaging Laboratory
Wellcome Trust Centre for Neuroimaging
Institute of Neurology, UCL
12 Queen Square, London WC1N 3BG, UK

April 8, 2008

http://www.fil.ion.ucl.ac.uk/~wpenny/mbi/

Contents

Chapter 1

General Linear Models I

1.1 Maximum Likelihood Estimation

We can learn the mean and variance of a Gaussian distribution using the Maximum Likelihood (ML) framework as follows. A Gaussian variable x_n has the PDF

$$p(x_n) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} \right) \tag{1.1}$$

which is also called the likelihood of the data point. Given N Independent and Identically Distributed (IID) (it is often assumed that the data points, or errors, are independent and come from the same distribution) samples $y = [y_1, y_2, .., y_N]$ we have

$$p(y) = \prod_{n=1}^{N} p(y_n) \tag{1.2}$$

which is the likelihood of the data set. We now wish to set μ and σ^2 so as to maximise this likelihood. For numerical reasons (taking logs gives us bigger numbers) this is more conveniently achieved by maximising the log-likelihood (note: the maximum is given by the same values of μ and σ)

$$L \equiv \log p(y) = -\frac{N}{2}\log 2\pi - \frac{N}{2}\log\sigma^2 - \sum_{n=}^{N}\frac{(y_n-\mu)^2}{2\sigma^2} \qquad (1.3)$$

The optimal values of μ and σ are found by setting the derivatives $\frac{dL}{d\mu}$ and $\frac{dL}{d\sigma}$ to zero. This gives

$$\mu = \frac{1}{N}\sum_{n=1}^{N} y_n \qquad (1.4)$$

and

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(y_n-\mu)^2 \qquad (1.5)$$

We note that the last formula is different to the usual formula for estimating variance

$$\sigma^2 = \frac{1}{N-1}\sum_{n=1}^{N}(x_n-\mu)^2 \qquad (1.6)$$

because of the difference in normalisation. The last estimator of variance is preferred as it is an *unbiased* estimator (see later section on bias and variance).

If we had an input-dependent mean, $\mu_n = wx_n$, then the optimal value for w can be found by maximising L. As only the last term in equation 1.3 depends on w this therefore corresponds to minimisation of the squared errors between μ_n and y_n. This provides the connection between ML estimation and Least Squares (LS) estimation; ML reduces to LS for the case of Gaussian noise.

1.2 Correlation and Regression

1.2.1 Correlation

The *covariance* between two variables x and y is measured as

$$\sigma_{xy} = \frac{1}{N-1}\sum_{n=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \tag{1.7}$$

where μ_x and μ_y are the means of each variable. Note that $\sigma_{yx} = \sigma_{xy}$. Sometimes we will use the notation

$$Var(x, y) = \sigma_{xy} \tag{1.8}$$

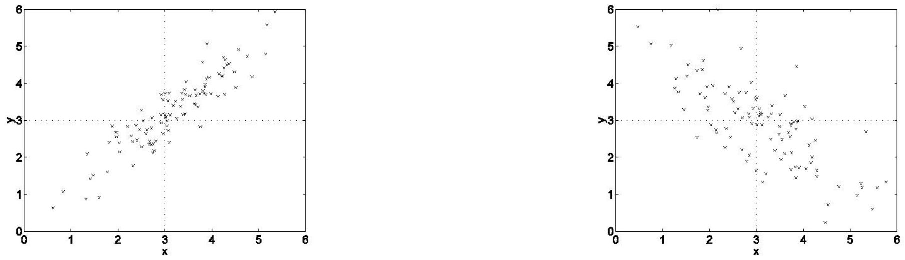
If x tends to be above its mean when y is above its mean then σ_{xy} will be positive. If they tend to be on opposite sides of their means σ_{xy} will be negative. The *correlation* or *Pearson's correlation coefficient* is a normalised covariance

$$r = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{1.9}$$

such that $-1 \leq r \leq 1$, a value of -1 indicating perfect negative correlation and a value of $+1$ indicating perfect positive correlation; see Figure 1.1. A value of 0 indicates no correlation. The strength of a correlation is best measured by r^2 which takes on values between 0 and 1, a value near to 1 indicating strong correlation (regardless of the sign) and a value near to zero indicating a very weak correlation.

1.2.2 Linear regression

We now look at modelling the relationship between two variables x and y as a linear function; given a collection of N data



(a) (b)

Figure 1.1: *(a) Positive correlation, $r = 0.9$ and (b) Negative correlation, $r = -0.7$. The dotted horizontal and vertical lines mark μ_x and μ_y.*

points $\{x_i, y_i\}$, we aim to estimate y_i from x_i using a linear model

$$\hat{y}_i = ax_i + b \tag{1.10}$$

where we have written $\hat{y}$ to denote our estimated value. Regression with one input variable is often called *univariate* linear regression to distinguish it from *multivariate* linear regression where we have lots of inputs. The goodness of fit of the model to the data may be measured by the least squares cost function

$$E = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{1.11}$$

The values of a and b that minimize the above cost function can be calculated by setting the first derivatives of the cost function to zero and solving the resulting simultaneous equations (derivatives are used to find maxima and minima of functions).

The result is derived as follows. We can find the slope a and offset b by minising the cost function

$$E = \sum_{i=1}^{N} (y_i - ax_i - b)^2 \tag{1.12}$$

Differentiating with respect to a gives

$$\frac{\partial E}{\partial a} = -2 \sum_{i=1}^{N} x_i (y_i - a x_i - b) \tag{1.13}$$

Differentiating with respect to b gives

$$\frac{\partial E}{\partial b} = -2 \sum_{i=1}^{N} (y_i - a x_i - b) \tag{1.14}$$

By setting the above derivatives to zero we obtain the *normal equations* of the regression. Re-arranging the normal equations gives

$$a \sum_{i=1}^{N} x_i^2 + b \sum_{i=1}^{N} x_i = \sum_{i=1}^{N} x_i y_i \tag{1.15}$$

and

$$a \sum_{i=1}^{N} x_i + bN = \sum_{i=1}^{N} y_i \tag{1.16}$$

By substituting the mean observed values μ_x and μ_y into the last equation we get

$$b = \mu_y - a \mu_x \tag{1.17}$$

Now let

$$\begin{aligned} S_{xx} &= \sum_{i=1}^{N} (x_i - \mu_x)^2 && (1.18) \\ &= \sum_{i=1}^{N} x_i^2 - N \mu_x^2 \end{aligned}$$

$$(1.19)$$

and

$$\begin{aligned} S_{xy} &= \sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) && (1.20) \\ &= \sum_{i=1}^{N} x_i y_i - N\mu_x\mu_y \\ & && (1.21) \end{aligned}$$

Substiting for b into the first normal equation gives

$$a\sum_{i=1}^{N} x_i^2 + (\mu_y - a\mu_x)\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} x_i y_i \qquad (1.22)$$

Re-arranging gives

$$\begin{aligned} a &= \frac{\sum_{i=1}^{N} x_i y_i - \mu_y \sum_{i=1}^{N} x_i}{\sum_{i=1}^{N} x_i^2 + \mu_x \sum_{i=1}^{N} x_i} && (1.23) \\ &= \frac{\sum_{i=1}^{N} x_i y_i - N\mu_x\mu_y}{\sum_{i=1}^{N} x_i^2 + N\mu_x^2} \\ &= \frac{\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)}{\sum_{i=1}^{N}(x_i - \mu_x)^2} \\ &= \frac{\sigma_{xy}}{\sigma_x^2} \end{aligned}$$

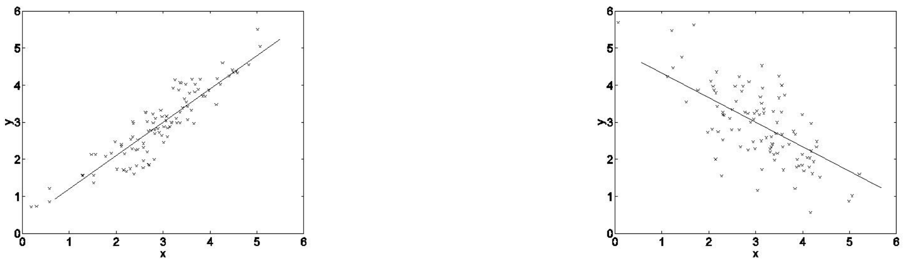
To summarise, the solutions are

$$a = \frac{\sigma_{xy}}{\sigma_x^2} \qquad (1.24)$$

and

$$b = \mu_y - a\mu_x \qquad (1.25)$$

where μ_x and μ_y are the mean observed values of the data and σ_x^2 and σ_{xy} are the input variance and input-output covariance.



(a) (b)

Figure 1.2: *The linear regression line is fitted by minimising the vertical distance between itself and each data point. The estimated lines are (a)* $\hat{y} = 0.9003x + 0.2901$ *and (b)* $\hat{y} = -0.6629x + 4.9804$.

This enables least squares fitting of a regression line to a data set as shown in Figure 1.2.

The model will fit some data points better than others; those that it fits well constitute the *signal* and those that it does'nt fit well constitute the *noise*. The strength of the noise is measured by the noise variance

$$\sigma_e^2 = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{1.26}$$

and the strenth of the signal is given by $\sigma_y^2 - \sigma_e^2$. The *signal-to-noise ratio* is therefore $(\sigma_y^2 - \sigma_e^2)/\sigma_e^2$.

Splitting data up into signal and noise components in this manner (ie. breaking down the variance into what the model *explains* and what it does not) is at the heart of statistical procedures such as analysis of variance (ANOVA) [24].

1.2.3 Relation to correlation

The correlation measure r is intimately related to the linear regression model. Indeed (by substituting σ_{xy} from equation 4.57

into equation 1.24) r may be expressed as

$$r = \frac{\sigma_x}{\sigma_y} a \tag{1.27}$$

where a is the slope of the linear regression model. Thus, for example, the sign of the slope of the regression line defines the sign of the correlation. The correlation is, however, also a function of the standard deviation of the x and y variables; for example, if σ_x is very large, it is possible to have a strong correlation even though the slope may be very small.

The relation between r and linear regression emphasises the fact that r is only a measure of *linear* correlation. It is quite possible that two variables have a strong nonlinear relationship (ie. are nonlinearly correlated) but that $r = 0$. Measures of nonlinear correlation will be discussed in a later lecture.

The strenth of correlation can also be expressed in terms of quantites from the linear regresssion model

$$r^2 = \frac{\sigma_y^2 - \sigma_e^2}{\sigma_y^2} \tag{1.28}$$

where σ_e^2 is the noise variance and σ_y^2 is the variance of the variable we are trying to predict. Thus r^2 is seen to measure the proportion of variance explained by a linear model, a value of 1 indicating that a linear model perfectly describes the relationship between x and y.

1.2.4 Finding the uncertainty in estimating the slope

The data points may be written as

$$\begin{aligned} y_i &= \hat{y}_i + e_i \qquad (1.29) \\ &= a x_i + b + e_i \end{aligned}$$

where the noise, e_i has mean zero and variance σ_e^2. The mean and variance of each data point are

$$E(y_i) = ax_i + b \tag{1.30}$$

and

$$Var(y_i) = Var(e_i) = \sigma_e^2 \tag{1.31}$$

We now calculate the variance of the estimate a. From earlier we see that

$$a = \frac{\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)}{\sum_{i=1}^{N}(x_i - \mu_x)^2} \tag{1.32}$$

Let

$$c_i = \frac{(x_i - \mu_x)}{\sum_{i=1}^{N}(x_i - \mu_x)^2} \tag{1.33}$$

We also note that $\sum_{i=1}^{N} c_i = 0$ and $\sum_{i=1}^{N} c_i x_i = 1$. Hence,

$$\begin{aligned} a &= \sum_{i=1}^{N} c_i(y_i - \mu_y) && (1.34) \\ &= \sum_{i=1}^{N} c_i y_i - \mu_y \sum_{i=1}^{N} c_i \end{aligned}$$

$$(1.35)$$

The mean estimate is therefore

$$\begin{aligned} E(a) &= \sum_{i=1}^{N} c_i E(y_i) - \mu_y \sum_{i=1}^{N} c_i && (1.36) \\ &= a\sum_{i=1}^{N} c_i x_i + b\sum_{i=1}^{N} c_i - \mu_y \sum_{i=1}^{N} c_i \\ &= a \end{aligned}$$

$$(1.37)$$

The variance is

$$Var(a) = Var(\sum_{i=1}^{N} c_i y_i - \mu_y \sum_{i=1}^{N} c_i) \tag{1.38}$$

The second term contains two fixed quantities so acts like a constant. Hence,

$$\begin{aligned} Var(a) &= Var(\sum_{i=1}^{N} c_i y_i) \qquad (1.39)\\ &= \sum_{i=1}^{N} c_i^2 Var(y_i) \\ &= \sigma_e^2 \sum_{i=1}^{N} c_i^2 \\ &= \frac{\sigma_e^2}{\sum_{i=1}^{N}(x_i - \mu_x)^2} \\ &= \frac{\sigma_e^2}{(N-1)\sigma_x^2} \end{aligned}$$

1.3 Inference

When we estimate the mean and variance from small samples of data our estimates may not be very accurate. But as the number of samples increases our estimates get more and more accurate and as this number approaches infinity the sample mean approaches the true mean or *population* mean. In what follows we refer to the sample means and variances as m and s and the population means and standard deviations as μ and σ.

Hypothesis Testing: Say we have a hypothesis ***H*** which is *The mean value of my signal is 32*. This is often referred to as the *null hypothesis* or H_0. We then get some data and test $\boldsymbol{H}$ which is then either *accepted* or *rejected* with a certain probability or *significance level*, p. Very often we choose $p = 0.05$ (a value used throughout science).

We can do a *one-sided* or a *two-sided* statistical test depending on exactly what the null hypothesis is. In a one-sided test our hypothesis may be (i) our parameter is less than x or (ii) our parameter is greater than x. For two-sided tests our hypothesis is of the form (iii) our parameter is x. This last hypothesis can be rejected if the sample statistic is either much smaller or much greater than it should be if the parameter truly equals x.

1.3.1 Regression

In a linear regression model we are often interested in whether or not the gradient is significantly different from zero or other value of interest.

To answer the question we first estimate the variance of the slope and then perform a t-test. In the appendix we show that the variance of the slope is given by [1]

$$\sigma_a^2 = \frac{\sigma_e^2}{(N-1)\sigma_x^2} \tag{1.40}$$

We then calculate the t-statistic

$$t = \frac{a - a_h}{\sigma_a} \tag{1.41}$$

where a_h is our hypothesized slope value (eg. a_h may be zero) and look up $p(t)$ with $N - 2$ DF (we have used up 1DF to

[1] When estimating σ_x^2 we should divide by $N - 1$ and when estimating σ_e^2 we should divide by $N - 2$.

estimate the input variance and 1DF to estimate the noise variance). In the data plotted in Figure 1.2(b) the estimated slope is $a = -0.6629$. From the data we also calculate that $\sigma_a = 0.077$. Hence, to find out if the slope is significantly non-zero we compute $CDF_t(t)$ where $t = -0.6629/0.077 = -8.6$. This has a p-value of 10^{-13} ie. a very significant value. To find out if the slope is significantly different from -0.7 we calculate $CDF_t(t)$ for $t = (-0.6629 + 0.7)/0.077 = 0.4747$ which gives a p-value of 0.3553 ie. not significantly different (again, we must bear in mind that we need to do a two-sided test; see earlier).

1.3.2 Correlation

Because of the relationship between correlation and linear regression we can find out if correlations are significantly non-zero by using exactly the same method as in the previous section; if the slope is significantly non-zero then the corresponding correlation is also significantly non-zero.

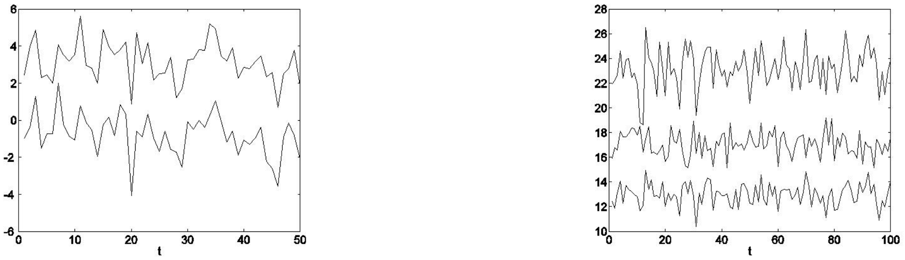
By substituting $a = (\sigma_y/\sigma_x)r$ (this follows from equation 1.24 and equation 1.9) and $\sigma_e^2 = (1 - r^2)\sigma_y^2$ (from equation 1.28) into equation 1.40 and then σ_a into equation 1.41 we get the test statistic [2]

$$t = \frac{r\sqrt{N-2}}{\sqrt{1-r^2}} \tag{1.42}$$

which has $N - 2$ DF.

For example, the two signals in Figure 1.3(a) have, over the $N = 50$ given samples, a correlation of $r = 0.8031$ which gives $t = 9.3383$ and a p-value of 10^{-12}. We therefore reject the hypothesis that the signals are not correlated; they clearly are.

[2] Strictly, we should use $\sigma_e^2 = \frac{N-1}{N-2}(1 - r^2)\sigma_y^2$ to allow for using $N - 2$ in the denominator of σ_e^2.



(a) (b)

Figure 1.3: *Two signals (a) sample correlation* $r = 0.8031$ *and (b) sample correlation,* r=0.1418*. Strong correlation; by shifting and scaling one of the time series (ie. taking a linear function) we can make it look like the other time series.*

The signals in Figure 1.3(b) have a correlation of $r = 0.1418$ over the $N = 50$ given samples which gives $t = 0.9921$ and a p-value of $p = 0.1631$. We therefore accept the null hypothesis that the signals are not correlated.

1.4 Linear algebra

1.4.1 Transposes and Inner Products

A collection of variables may be treated as a single entity by writing them as a *vector*. For example, the three variables x_1, x_2 and x_3 may be written as the vector

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{1.43}$$

Bold face type is often used to denote vectors (scalars - single variables - are written with normal type). Vectors can be written as *column vectors* where the variables go down the page or as *row vectors* where the variables go across the page (it needs to be made clear when using vectors whether $\boldsymbol{x}$ means a row vector or a column vector - most often it will mean a column vector and in our text it will *always* mean a column vector, unless we say otherwise). To turn a column vector into a row vector we use the *transpose* operator

$$\boldsymbol{x}^T = [x_1, x_2, x_3] \tag{1.44}$$

The transpose operator also turns row vectors into column vectors. We now define the *inner product* of two vectors

$$\begin{aligned} \boldsymbol{x}^T\boldsymbol{y} &= [x_1, x_2, x_3]\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \\ &= x_1y_1 + x_2y_2 + x_3y_3 \\ &= \sum_{i=1}^{3} x_iy_i \end{aligned} \tag{1.45}$$

which is seen to be a scalar. The *outer product* of two vectors produces a matrix

$$\begin{aligned} \boldsymbol{x}\boldsymbol{y}^T &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}[y_1, y_2, y_3] \\ &= \begin{bmatrix} x_1y_1 & x_1y_2 & x_1y_3 \\ x_2y_1 & x_2y_2 & x_2y_3 \\ x_3y_1 & x_3y_2 & x_3y_3 \end{bmatrix} \end{aligned} \tag{1.46}$$

An $N \times M$ matrix has N rows and M columns. The ijth entry of a matrix is the entry on the jth column of the ith row. Given a matrix $\boldsymbol{A}$ (matrices are also often written in bold type) the ijth entry is written as $\boldsymbol{A}_{ij}$. When applying the transpose operator to a matrix the ith row becomes the ith column. That is, if

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{1.47}$$

then

$$\boldsymbol{A}^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \tag{1.48}$$

A matrix is *symmetric* if $\boldsymbol{A}_{ij} = \boldsymbol{A}_{ji}$. Another way to say this is that, for symmetric matrices, $\boldsymbol{A} = \boldsymbol{A}^T$.

Two matrices can be multiplied if the number of columns in the first matrix equals the number of rows in the second. Multiplying $\boldsymbol{A}$, an $N \times M$ matrix, by $\boldsymbol{B}$, an $M \times K$ matrix, results in $\boldsymbol{C}$, an $N \times K$ matrix. The ijth entry in $\boldsymbol{C}$ is the inner product between the ith row in $\boldsymbol{A}$ and the jth column in $\boldsymbol{B}$. As an example

$$\begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \end{bmatrix} \begin{bmatrix} 1 & 3 & 7 & 2 \\ 4 & 3 & 4 & 1 \\ 5 & 6 & 4 & 2 \end{bmatrix} = \begin{bmatrix} 34 & 39 & 42 & 15 \\ 64 & 75 & 87 & 30 \end{bmatrix} \tag{1.49}$$

Given two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ we note that

$$(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T \tag{1.50}$$

1.4.2 Properties of matrix multiplication

Matrix multiplication is associative

$$(\boldsymbol{AB})\boldsymbol{C} = \boldsymbol{A}(\boldsymbol{BC}) \tag{1.51}$$

distributive

$$\boldsymbol{A}(\boldsymbol{B} + \boldsymbol{C}) = \boldsymbol{AB} + \boldsymbol{AC} \tag{1.52}$$

but not commutative

$$\boldsymbol{AB} \neq \boldsymbol{BA} \tag{1.53}$$

1.4.3 Covariance matrices

In the previous chapter the covariance, σ_{xy}, between two variables x and y was defined. Given p variables there are $p \times p$ covariances to take account of. If we write the covariances between variables x_i and x_j as σ_{ij} then all the covariances can be summarised in a *covariance matrix* which we write below for $p = 3$

$$\boldsymbol{C} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{bmatrix} \tag{1.54}$$

The ith diagonal element is the covariance between the ith variable and itself which is simply the variance of that variable; we therefore write σ_i^2 instead of σ_{ii}. Also, note that because $\sigma_{ij} = \sigma_{ji}$ covariance matrices are symmetric.

We now look at computing a covariance matrix from a given data set. Suppose we have p variables and that a single observation $\boldsymbol{x}_i$ (a row vector) consists of measuring these variables and suppose there are N such observations. We now make a matrix $\boldsymbol{X}$ by putting each $\boldsymbol{x}_i$ into the ith row. The matrix $\boldsymbol{X}$ is therefore an $N \times p$ matrix whose rows are made up of different observation vectors. If all the variables have zero mean then the covariance matrix can then be evaluated as

$$\boldsymbol{C} = \frac{1}{N-1}\boldsymbol{X}^T\boldsymbol{X} \tag{1.55}$$

This is a multiplication of a $p \times N$ matrix, $\boldsymbol{X}^T$, by a $N \times p$ matrix, $\boldsymbol{X}$, which results in a $p \times p$ matrix. To illustrate the use of covariance matrices for time series, figure 1.4 shows 3 time

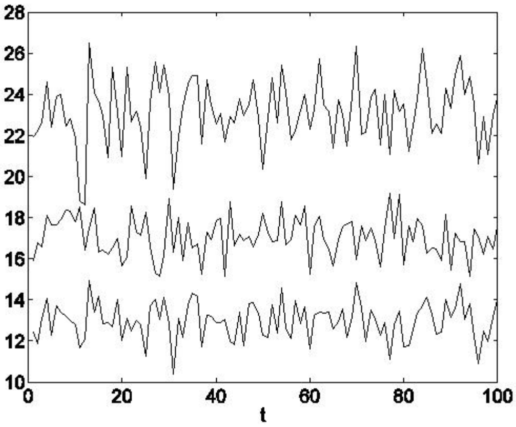


Figure 1.4: *Three time series having the covariance matrix $\boldsymbol{C}_1$ and mean vector $\boldsymbol{m}_1$ shown in the text. The top and bottom series have high covariance but none of the other pairings do.*

series which have the following covariance relation

$$\boldsymbol{C}_1 = \begin{bmatrix} 1 & 0.1 & 1.6 \\ 0.1 & 1 & 0.2 \\ 1.6 & 0.2 & 2.0 \end{bmatrix} \tag{1.56}$$

and mean vector

$$\boldsymbol{m}_1 = [13, 17, 23]^T \tag{1.57}$$

1.4.4 Diagonal matrices

A *diagonal matrix* is a square matrix ($M = N$) where all the entries are zero except along the diagonal. For example

$$\boldsymbol{D} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 6 \end{bmatrix} \tag{1.58}$$

There is also a more compact notation for the same matrix

$$\boldsymbol{D} = diag([4, 1, 6]) \tag{1.59}$$

If a covariance matrix is diagonal it means that the covariances between variables are zero, that is, the variables are all uncorrelated. Non-diagonal covariance matrices are known as *full* covariance matrices. If $\boldsymbol{V}$ is a vector of variances $\boldsymbol{V} = [\sigma_1^2, \sigma_2^2, \sigma_3^2]^T$ then the corresponding diagonal covariance matrix is $\boldsymbol{V}_d = diag(\boldsymbol{V})$.

1.4.5 The correlation matrix

The correlation matrix, $\boldsymbol{R}$, can be derived from the covariance matrix by the equation

$$\boldsymbol{R} = \boldsymbol{BCB} \tag{1.60}$$

where $\boldsymbol{B}$ is a diagonal matrix of inverse standard deviations

$$\boldsymbol{B} = diag([1/\sigma_1, 1/\sigma_2, 1/\sigma_3]) \tag{1.61}$$

1.4.6 The identity matrix

The identity matrix is a diagonal matrix with ones along the diagonal. Multiplication of any matrix, $\boldsymbol{X}$ by the identity matrix results in $\boldsymbol{X}$. That is

$$\boldsymbol{IX} = \boldsymbol{X} \tag{1.62}$$

The identity matrix is the matrix equivalent of multiplying by 1 for scalars.

1.4.7 Matrix inverse

Given a matrix $\boldsymbol{X}$ its inverse $\boldsymbol{X}^{-1}$ is defined by the properties

$$\begin{aligned} \boldsymbol{X}^{-1}\boldsymbol{X} &= \boldsymbol{I} \qquad (1.63)\\ \boldsymbol{X}\boldsymbol{X}^{-1} &= \boldsymbol{I} \end{aligned}$$

where $\boldsymbol{I}$ is the identity matrix. The inverse of a diagonal matrix with entries d_{ii} is another diagonal matrix with entries $1/d_{ii}$. This satisfies the definition of an inverse, eg.

$$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.64}$$

More generally, the calculation of inverses involves a lot more computation. Before looking at the general case we first consider the problem of solving simultaneous equations. These constitute relations between a set of *input or independent* variables $\boldsymbol{x}_i$ and a set of *output or dependent* variables y_i. Each input-output pair constitutes an observation. In the following example we consider just $N = 3$ observations and $p = 3$ dimensions per observation

$$\begin{array}{lll} 2w_1 & +w_2 + w_3 & = 5 \\ 4w_1 & -6w_2 & = -2 \\ -2w_1 & +7w_2 + 2w_3 & = 9 \end{array}$$

which can be written in matrix form

$$\begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -2 \\ 9 \end{bmatrix} \tag{1.65}$$

or in matrix form

$$\boldsymbol{X}\boldsymbol{w} = \boldsymbol{y} \tag{1.66}$$

This system of equations can be solved in a systematic way by subtracting multiples of the first equation from the second and third equations and then subtracting multiples of the second equation from the third. For example, subtracting twice the first equation from the second and -1 times the first from the third gives

$$\begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 8 & 3 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -12 \\ 14 \end{bmatrix} \tag{1.67}$$

Then, subtracting -1 times the second from the third gives

$$\begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -12 \\ 2 \end{bmatrix} \tag{1.68}$$

This process is known as *forward elimination*. We can then substitute the value for w_3 from the third equation into the second etc. This process is *back-substitution*. The two processes are together known as *Gaussian elimination*. Following this through for our example we get $\boldsymbol{w} = [1, 1, 2]^T$.

When we come to invert a matrix (as opposed to solve a system of equations as in the previous example) we start with the equation

$$\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I} \tag{1.69}$$

and just write down all the entries in the $\boldsymbol{A}$ and $\boldsymbol{I}$ matrices in one big matrix

$$\begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 \\ 4 & -6 & 0 & 0 & 1 & 0 \\ -2 & 7 & 2 & 0 & 0 & 1 \end{bmatrix} \tag{1.70}$$

We then perform forward elimination [3] until the part of the matrix corresponding to $\boldsymbol{A}$ equals the identity matrix; the matrix on the right is then $\boldsymbol{A}^{-1}$ (this is because in equation 1.69 if $\boldsymbol{A}$ becomes $\boldsymbol{I}$ then the left hand side is $\boldsymbol{A}^{-1}$ and the right side

[3] We do not perform back-substitution but instead continue with forward elimination until we get a diagonal matrix.

must equal the left side). We get

$$\begin{bmatrix} 1 & 0 & 0 & \frac{12}{16} & \frac{-5}{16} & \frac{-6}{16} \\ 0 & 1 & 0 & \frac{4}{8} & \frac{-3}{8} & \frac{-2}{8} \\ 0 & 0 & 1 & -1 & 1 & 1 \end{bmatrix} \tag{1.71}$$

This process is known as the *Gauss-Jordan* method. For more details see Strang's excellent book on Linear Algebra [44] where this example was taken from.

Inverses can be used to solve equations of the form $\boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}$. This is achieved by multiplying both sides by $\boldsymbol{X}^{-1}$ giving

$$\boldsymbol{w} = \boldsymbol{X}^{-1}\boldsymbol{y} \tag{1.72}$$

Hence,

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} \frac{12}{16} & \frac{-5}{16} & \frac{-6}{16} \\ \frac{4}{8} & \frac{-3}{8} & \frac{-2}{8} \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ -2 \\ 9 \end{bmatrix} \tag{1.73}$$

which also gives $\boldsymbol{w} = [1, 1, 2]^T$.

The inverse of a product of matrices is given by

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1} \tag{1.74}$$

Only square matrices are invertible because, for $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, if $\boldsymbol{y}$ and $\boldsymbol{x}$ are of different dimension then we will not necessarily have a one-to-one mapping between them.

1.4.8 Orthogonality

The length of a d-element vector $\boldsymbol{x}$ is written as $||\boldsymbol{x}||$ where

$$\begin{aligned} ||\boldsymbol{x}||^2 &= \sum_{i=1}^{d} x_i^2 \\ &= \boldsymbol{x}^T\boldsymbol{x} \end{aligned} \tag{1.75}$$

Two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are *orthogonal* if

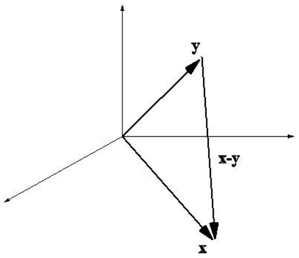


Figure 1.5: *Two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. These vectors will be orthogonal if they obey Pythagoras' relation ie. that the sum of the squares of the sides equals the square of the hypoteneuse.*

$$||\boldsymbol{x}||^2 + ||\boldsymbol{y}||^2 = ||\boldsymbol{x} - \boldsymbol{y}||^2 \tag{1.76}$$

That is, if

$$x_1^2 + ... + x_d^2 + y_1^2 + ... + y_d^2 = (x_1 - y_1)^2 + ... + (x_d - y_d)^2 \tag{1.77}$$

Expanding the terms on the right and re-arranging leaves only the cross-terms

$$\begin{aligned} x_1 y_1 + ... + x_d y_d &= 0 \qquad (1.78) \\ \boldsymbol{x}^T \boldsymbol{y} &= 0 \end{aligned}$$

That is, two vectors are orthogonal if their inner product is zero.

1.4.9 Angles between vectors

Given a vector $\boldsymbol{b} = [b_1, b_2]^T$ and a vector $\boldsymbol{a} = [a_1, a_2]^T$ we can

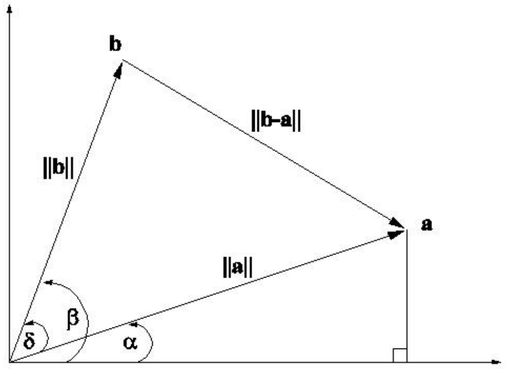


Figure 1.6: *Working out the angle between two vectors.*

work out that

$$\begin{aligned} \cos\alpha &= \frac{a_1}{||\boldsymbol{a}||} \\ \sin\alpha &= \frac{a_2}{||\boldsymbol{a}||} \\ \cos\beta &= \frac{b_1}{||\boldsymbol{b}||} \\ \sin\beta &= \frac{b_2}{||\boldsymbol{b}||} \end{aligned} \quad (1.79)$$

$$(1.80)$$

Now, $cos\delta = cos(\beta - \alpha)$ which we can expand using the trig identity

$$\cos(\beta - \alpha) = \cos\beta\cos\alpha + \sin\beta\sin\alpha \quad (1.81)$$

Hence

$$\cos(\delta) = \frac{a_1 b_1 + a_2 b_2}{||\boldsymbol{a}||||\boldsymbol{b}||} \quad (1.82)$$

More generally, we have

$$\cos(\delta) = \frac{\boldsymbol{a}^T\boldsymbol{b}}{||\boldsymbol{a}||||\boldsymbol{b}||} \quad (1.83)$$

Because, $\cos \pi/2 = 0$, this again shows that vectors are orthogonal for $\boldsymbol{a}^T\boldsymbol{b} = 0$. Also, because $|\cos \delta| \leq 1$ where $|x|$ denotes the absolute value of x we have

$$|\boldsymbol{a}^T\boldsymbol{b}| \leq ||\boldsymbol{a}||||\boldsymbol{b}|| \tag{1.84}$$

which is known as the *Schwarz Inequality.*

1.4.10 Projections

The projection of a vector $\boldsymbol{b}$ onto a vector $\boldsymbol{a}$ results in a projection vector $\boldsymbol{p}$ which is the point on the line $\boldsymbol{a}$ which is closest to the point $\boldsymbol{b}$. Because $\boldsymbol{p}$ is a point on $\boldsymbol{a}$ it must be some scalar multiple of it. That is

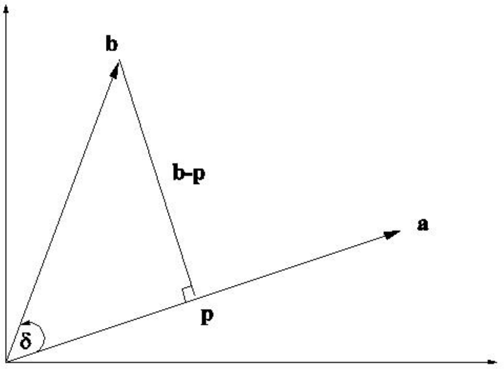


Figure 1.7: *The projection of* $\boldsymbol{b}$ *onto* $\boldsymbol{a}$ *is the point on* $\boldsymbol{a}$ *which is closest to* $\boldsymbol{b}$*.*

$$\boldsymbol{p} = w\boldsymbol{a} \tag{1.85}$$

where w is some coefficient. Because $\boldsymbol{p}$ is the point on $\boldsymbol{a}$ *closest* to $\boldsymbol{b}$ this means that the vector $\boldsymbol{b} - \boldsymbol{p}$ is orthogonal to $\boldsymbol{a}$.

Therefore

$$\begin{aligned} \boldsymbol{a}^T(\boldsymbol{b}-\boldsymbol{p}) &= 0 \\ \boldsymbol{a}^T(\boldsymbol{b}-w\boldsymbol{a}) &= 0 \end{aligned} \tag{1.86}$$

Re-arranging gives

$$w = \frac{\boldsymbol{a}^T\boldsymbol{b}}{\boldsymbol{a}^T\boldsymbol{a}} \tag{1.87}$$

and

$$\boldsymbol{p} = \frac{\boldsymbol{a}^T\boldsymbol{b}}{\boldsymbol{a}^T\boldsymbol{a}}\boldsymbol{a} \tag{1.88}$$

We refer to $\boldsymbol{p}$ as the *projection vector* and to w as the *projection.*

1.5 Multiple Regression

A good practical introduction to the material on regression is presented by Kleinbaum et al. [24]. More details of matrix manipulations are available in Weisberg [47] and Strang has a great in-depth intro to linear algebra [44]. See also relevant material in *Numerical Recipes* [42]. See Chatfield's book on multivariate analysis for more details [9].

For a multivariate linear data set, the dependent variable y_i is modelled as a linear combination of the input variables $\boldsymbol{x}_i$ and an error term [4]

$$y_i = \boldsymbol{x}_i\boldsymbol{w} + e_i \tag{1.89}$$

where $\boldsymbol{x}_i$ is a row vector, $\boldsymbol{w}$ is a column vector and e_i is an error. The overall goodness of fit can be assessed by the least

[4] The error term is introduced because, very often, given a particular data set it will not be possible to find an exact linear relationship between $\boldsymbol{x}_i$ and y_i for every i. We therefore cannot directly estimate the weights as $\boldsymbol{X}^{-1}\boldsymbol{y}$.

squares cost function

$$E = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{1.90}$$

where $\hat{\boldsymbol{y}}_i = \boldsymbol{x}_i \boldsymbol{w}$.

1.5.1 Estimating the weights

The least squares cost function can be written in matrix notation as

$$E = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) \tag{1.91}$$

where $\boldsymbol{X}$ is an N-by-p matrix whose rows are made up of different input vectors and $\boldsymbol{y}$ is a vector of targets. The weight vector that minimises this cost function can be calculated by setting the first derivative of the cost function to zero and solving the resulting equation.

By expanding the brackets and collecting terms (using the matrix identity $(\boldsymbol{AB})^T = \boldsymbol{B}^T \boldsymbol{A}^T$ we get

$$E = \boldsymbol{y}^T \boldsymbol{y} - 2\boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{y} + \boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{w} \tag{1.92}$$

The derivative with respect to $\boldsymbol{w}$ is [5]

$$\frac{\partial E}{\partial \boldsymbol{w}} = -2\boldsymbol{X}^T \boldsymbol{y} + 2\boldsymbol{X}^T \boldsymbol{X} \boldsymbol{w} \tag{1.93}$$

Equating this derivative to zero gives

$$(\boldsymbol{X}^T \boldsymbol{X})\boldsymbol{w} = \boldsymbol{X}^T \boldsymbol{y} \tag{1.94}$$

which, in regression analysis, is known as the 'normal equation'. Hence,

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y} \tag{1.95}$$

[5] From matrix calculus [27] we know that the derivative of $\boldsymbol{c}^T \boldsymbol{B} \boldsymbol{c}$ with respect to $\boldsymbol{c}$ is $(\boldsymbol{B}^T + \boldsymbol{B})\boldsymbol{c}$. Also we note that $\boldsymbol{X}^T \boldsymbol{X}$ is symmetric.

This is the general solution for multivariate linear regression [6]. It is a unique minimum of the least squares error function (ie. this is the only solution).

Once the weights have been estimated we can then estimate the error or noise variance from

$$\sigma_e^2 = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{1.96}$$

1.5.2 Understanding the solution

If the inputs are zero mean then the input covariance matrix multiplied by N-1 is

$$\boldsymbol{C}_x = \boldsymbol{X}^T \boldsymbol{X} \tag{1.97}$$

The weights can therefore be written as

$$\hat{\boldsymbol{w}} = \boldsymbol{C}_x^{-1} \boldsymbol{X}^T \boldsymbol{y} \tag{1.98}$$

ie. the inverse covariance matrix times the inner products of the inputs with the output (the ith weight will involve the inner product of the ith input with the output).

Single input

For a single input $\boldsymbol{C}_x^{-1} = 1/(N-1)\sigma_{x_1}^2$ and $\boldsymbol{X}^T \boldsymbol{y} = (N-1)\sigma_{x_1 y}$. Hence

$$\hat{w}_1 = \frac{\sigma_{x_1 y}}{\sigma_{x_1}^2} \tag{1.99}$$

This is *exactly* the same as the estimate for the slope in linear regression (first lecture). This is re-assuring.

[6] In practice we can use the equivalent expression $\hat{\boldsymbol{w}} = \boldsymbol{X}^{+1} \boldsymbol{y}$ where $\boldsymbol{X}^{+1}$ is the pseudo-inverse [44]. This method is related to Singular Value Decomposition and is discussed later.

Uncorrelated inputs

For two uncorrelated inputs

$$\boldsymbol{C}_x^{-1} = \begin{bmatrix} \frac{1}{(N-1)\sigma_{x_1}^2} & 0 \\ 0 & \frac{1}{(N-1)\sigma_{x_2}^2} \end{bmatrix} \tag{1.100}$$

We also have

$$\boldsymbol{X}^T \boldsymbol{y} = \begin{bmatrix} (N-1)\sigma_{x_1,y} \\ (N-1)\sigma_{x_2,y} \end{bmatrix} \tag{1.101}$$

The two weights are therefore

$$\begin{aligned} \hat{w}_1 &= \frac{\sigma_{x_1 y}}{\sigma_{x_1}^2} \\ \hat{w}_2 &= \frac{\sigma_{x_2 y}}{\sigma_{x_2}^2} \end{aligned} \tag{1.102}$$

Again, these solutions are the same as for the univariate linear regression case.

General case

If the inputs are correlated then a coupling is introduced in the estimates of the weights; weight 1 becomes a function of $\sigma_{x_2 y}$ as well as $\sigma_{x_1 y}$

$$\hat{\boldsymbol{w}} = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} \\ \sigma_{x_1 x_2} & \sigma_{x_2}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sigma_{x_1,y} \\ \sigma_{x_2,y} \end{bmatrix} \tag{1.103}$$

1.5.3 Inference

Some of the inputs in a linear regression model may be very useful in predicting the output. Others, not so. So how do we find which inputs or *features* are useful ? This problem is known as feature selection.

The problem is tackled by looking at the coefficients of each input (ie. the weights) and seeing if they are significantly non-zero. The procedure is identical to that described for univariate linear regression.

The only added difficulty is that we have more inputs and more weights, but the procedure is basically the same. Firstly, we have to estimate the variance on each weight. This is done in the next section. We then compare each weight to zero using a t-test.

Functions of random vectors

For a vector of random variables, $\boldsymbol{z}$, and a matrix of constants, $\boldsymbol{C}$, and a vector of constants, $\boldsymbol{d}$, we have

$$Var(\boldsymbol{Cz}+\boldsymbol{d}) = \boldsymbol{C}[Var(\boldsymbol{z})]\boldsymbol{C}^T \tag{1.104}$$

where, here, Var() denotes a covariance matrix. This is a generalisation of the result for scalar random variables $Var(cz) = c^2 Var(z)$.

The covariance between a pair of random vectors is given by

$$Var(\boldsymbol{C}_1\boldsymbol{z}, \boldsymbol{C}_2\boldsymbol{z}) = \boldsymbol{C}_1[Var(\boldsymbol{z})]\boldsymbol{C}_2^T \tag{1.105}$$

The weight covariance matrix

Different instantiations of target noise will generate different estimated weight vectors according to equation 1.95. For the case of Gaussian noise we do not actually have to compute the weights on many instantiations of the target noise and then compute the sample covariance [7]; the corresponding weight covariance matrix is given by the equation

$$\boldsymbol{\Sigma} = Var((\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}) \tag{1.106}$$

Substituting $\boldsymbol{y} = \boldsymbol{X}\hat{\boldsymbol{w}} + \boldsymbol{e}$ gives

$$\boldsymbol{\Sigma} = Var((\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} + (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{e}) \tag{1.107}$$

This is in the form of $Var(\boldsymbol{C}\boldsymbol{z}+\boldsymbol{d})$ (see earlier) with $\boldsymbol{d}$ being given by the first term which is constant, $\boldsymbol{C}$ being given by $(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ and $\boldsymbol{z}$ being given by $\boldsymbol{e}$. Hence,

$$\begin{aligned}
\boldsymbol{\Sigma} &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T[Var(\boldsymbol{e})][(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T]^T \qquad (1.108)\\
&= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\sigma_e^2\boldsymbol{I})[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T]^T\\
&= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\sigma_e^2\boldsymbol{I})\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}
\end{aligned}$$

Re-arranging further gives

$$\boldsymbol{\Sigma} = \sigma_e^2(\boldsymbol{X}^T\boldsymbol{X})^{-1} \tag{1.109}$$

In the appendix we show that this can be evaluated as

$$\boldsymbol{\Sigma} = \sigma_e^2(\boldsymbol{X}^T\boldsymbol{X})^{-1} \tag{1.110}$$

The correlation in the inputs introduces a correlation in the weights; for uncorrelated inputs the weights will be uncorrelated. The variance of the jth weight, w_j, is then given by the

[7] But this type of procedure is the basis of bootstrap estimates of parameter variances. See [13].

jth diagonal entry in the covariance matrix

$$\sigma_{w_j}^2 = \mathbf{\Sigma}_{jj} \tag{1.111}$$

To see if a weight is significantly non-zero we then compute $CDF_t(t)$ (the cumulative density function; see earlier lecture) where $t = w_j / \sigma_{w_j}$ and if it is above some threshold, say $p = 0.05$, the corresponding feature is removed.

Note that this procedure, which is based on a t-test, is exactly equivalent to a similar procedure based on a partial F-test (see, for example, [24] page 128).

If we do remove a weight then we must recompute all the other weights (and variances) *before* deciding whether or not the other weights are significantly non-zero. This usually proceeds in a stepwise manner where we start with a large number of features and reduce them as necessary (*stepwise backward selection*) or gradually build up the number of features (*stepwise forward selection*) [24].

Note that, if the weights were uncorrelated we could do feature selection in a single step; we would not have to recompute weight values after each weight removal. This provides one motivation for the use of orthogonal transforms in which the weights *are* uncorrelated. Such transforms include Fourier and Wavelet transforms as we shall see in later lectures.

1.5.4 Equivalence of t-test and F-test for feature selection

When adding a new variable x_p to a regression model we can test to see if the increase in the proportion of variance explained is *significant* by computing

$$F = \frac{(N-1)\sigma_y^2 \left[r^2(y, \hat{y}_p) - r^2(y, \hat{y}_{p-1}) \right]}{\sigma_e^2(p)} \tag{1.112}$$

where $r^2(y, \hat{y}_p)$ is the square of the correlation between y and the regression model with all p variables (ie. including x_p) and $r^2(y, \hat{y}_{p-1})$ is the square of the correlation between y and the regression model without x_p. The denominator is the noise variance from the model including x_p. This statistic is distributed according to the F-distribution with $v_1 = 1$ and $v_2 = N - p - 2$ degrees of freedom.

This test is identical to the double sided t-test on the t-statistic computed from the regression coefficient a_p, described in this lecture (see also page 128 of [24]). This test is also equivalent to seeing if the partial correlation between x_p and y is significantly non-zero (see page 149 of [24]).

1.5.5 Example

Suppose we wish to predict a time series x_3 from two other time series x_1 and x_2. We can do this with the following regression model [8]

$$x_3 = w_0 + w_1 x_1 + w_2 x_2 \tag{1.113}$$

and the weights can be found using the previous formulae. To cope with the constant, w_0, we augment the $\boldsymbol{X}$ vector with an additional column of 1's.

We analyse data having covariance matrix $\boldsymbol{C}_1$ and mean vector $\boldsymbol{m}_1$ (see equations 1.57 and 1.56 in an earlier lecture). $N = 50$

[8] Strictly, we can only apply this model if the samples *within* each time series are independent (see later). To make them independent we can randomize the time index thus removing any correlation between lagged samples. We therefore end up with a random variables rather than time series.

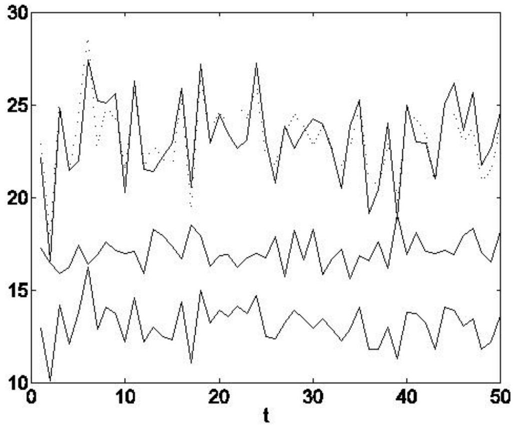


Figure 1.8: *Three time series having the correlation matrix* $\boldsymbol{C}_1$ *and mean vector* $\boldsymbol{m}_1$ *shown in the text. The dotted line shows the value of the third time series as predicted from the other two using a regression model.*

data points were generated and are shown in Figure 1.8. The weights were then estimated from equation 1.95 as

$$\begin{aligned} \hat{\boldsymbol{w}} &= [w_1, w_2, w_0]^T \\ &= [1.7906, -0.0554, 0.6293]^T \end{aligned} \tag{1.114}$$

Note that w_1 is much bigger than w_2. The weight covariance matrix was estimated from equation 1.110 as

$$\boldsymbol{\Sigma} = \begin{bmatrix} 0.0267 & 0.0041 & -0.4197 \\ 0.0041 & 0.0506 & -0.9174 \\ -0.4197 & -0.9174 & 21.2066 \end{bmatrix} \tag{1.115}$$

giving $\sigma_{w_1} = 0.1634$ and $\sigma_{w_2} = 0.2249$. The corresponding t-statistics are $t_1 = 10.96$ and $t_2 = -0.2464$ giving p-values of 10^{-15} and 0.4032. This indicates that the first weight is significantly different from zero but the second weight is not ie. x_1 is a good predictor of x_3 but x_2 is not. We can therefore remove x_2 from our regression model.

Question: But what does linear regression tell us about the data that the correlation/covariance matrix does'nt ? *Answer:* Partial correlations.

1.5.6 Partial Correlation

Remember (see eg. equation 1.28 from lecture 1), the square of the correlation coefficient between two variables x_1 and y is given by

$$r^2_{x_1 y} = \frac{\sigma^2_y - \sigma^2_e(x_1)}{\sigma^2_y} \tag{1.116}$$

where $\sigma^2_e(x_1)$ is the variance of the errors from using a linear regression model based on x_1 to predict y. Writing $\sigma^2_y = \sigma^2_e(0)$, ie. the error with no predictive variables

$$r^2_{x_1 y} = \frac{\sigma^2_e(0) - \sigma^2_e(x_1)}{\sigma^2_e(0)} \tag{1.117}$$

When we have a second predictive variable x_2, the square of the *partial correlation* between x_2 and y is defined as

$$r^2_{x_2 y|x_1} = \frac{\sigma^2_e(x_1) - \sigma^2_e(x_1, x_2)}{\sigma^2_e(x_1)} \tag{1.118}$$

where $\sigma^2_e(x_1, x_2)$ is the variance of the errors from the regression model based on x_1 and x_2. It's the extra proportion of variance in y explained by x_2. It's different to $r^2_{x_2 y}$ because x_2 may be correlated to x_1 which itself explains some of the variance in y. After *controlling* for this, the resulting proportionate reduction in variance is given by $r^2_{x_2 y|x_1}$. More generally, we can define pth order partial correlations which are the correlations between two variables after controlling for p variables.

The sign of the partial correlation is given by the sign of the corresponding regression coefficient.

Relation to regression coefficients

Partial correlations are to regression coefficients what the correlation is to the slope in univariate linear regression. If the partial correlation is significantly non-zero then the corresponding regression coefficient will also be. And vice-versa.

Chapter 2

General Linear Models II

2.1 Generalised Inverse

For GLM

$$y = X\beta + e$$

where X is a $N \times k$ design matrix and $p(e) = \mathsf{N}(0, \sigma^2 I_N)$, we can estimate the coefficients from the normal equations

$$(X^T X)\beta = X^T y$$

If rank of X, denoted $r(X)$, is k (ie. full rank) then $X^T X$ has an inverse (it is 'nonsingular') and

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

But if $r(x) < k$ we can have $X\beta_1 = X\beta_2$ (ie. same predictions) with $\beta_1 \neq \beta_2$ (different parameters). The parameters are then not therefore 'unique', 'identifiable' or 'estimable'.

For example, a design matrix sometimes used in the Analysis of Variance (ANOVA)

$$X = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{2.1}$$

has $k = 3$ columns but rank $r(X) = 2$ ie. only two linearly independent columns (any column can be expressed as a linear combination of the other two).

For models such as these $X^T X$ is not invertible, so we must resort to the *generalised inverse*, X^-. This is defined as any matrix X^- such that $XX^-X = X$. It can be shown that in the general case

$$\begin{aligned} \hat{\beta} &= (X^T X)^- X^T y \qquad (2.2) \\ &= X^- y \end{aligned}$$

If X is full-rank, $X^T X$ is invertible and $X^- = (X^T X)^{-1} X^T$.

There are many generalise inverses. We would often choose the pseudo-inverse (pinv in MATLAB)

$$\hat{\beta} = X^+ y \tag{2.3}$$

Take home message: avoid rank-deficient designs. If X is full rank, then $X^+ = X^- = (X^T X)^{-1} X^T$.

2.2 Estimating error variance

An *unbiased* estimate for the error variance σ^2 can be derived as follows. Let

$$X\hat{\beta} = Py$$

where P is the *projection matrix*

$$\begin{aligned} P &= X(X^TX)^-X^T \qquad (2.4) \\ &= XX^- \end{aligned}$$

Py projects the data y into the space of X. P has two important properties (i) it is symmetric $P^T = P$, (ii) $PP{=}P$. This second property follows from it being a projection. If what is being projected is already in X space (ie. Py) then looking for that component of it that is in X space will give the same thing ie. $PPy = Py$.

Then residuals are

$$\begin{aligned} \hat{e} &= y - X\hat{\beta} \qquad (2.5) \\ &= (I - P)y \\ &= Ry \end{aligned}$$

where $R = I_N{-}XX^-$ is the *residual-forming* matrix. Remember, $\hat{e}$ is that component of the data, orthogonal to the 'space' X. Ry is another projection matrix, but one that projects the data y into the orthogonal complement of X. Similarly, R has the two properties (i) $R^T = R$ and (ii) $RR = R$.

We now look seek an unbiased estimator of the variance by first looking at the expected sum of squares

$$\begin{aligned} E[\hat{e}^T\hat{e}] &= E[y^TR^TRy] \qquad (2.6) \\ &= E[y^TRy] \end{aligned}$$

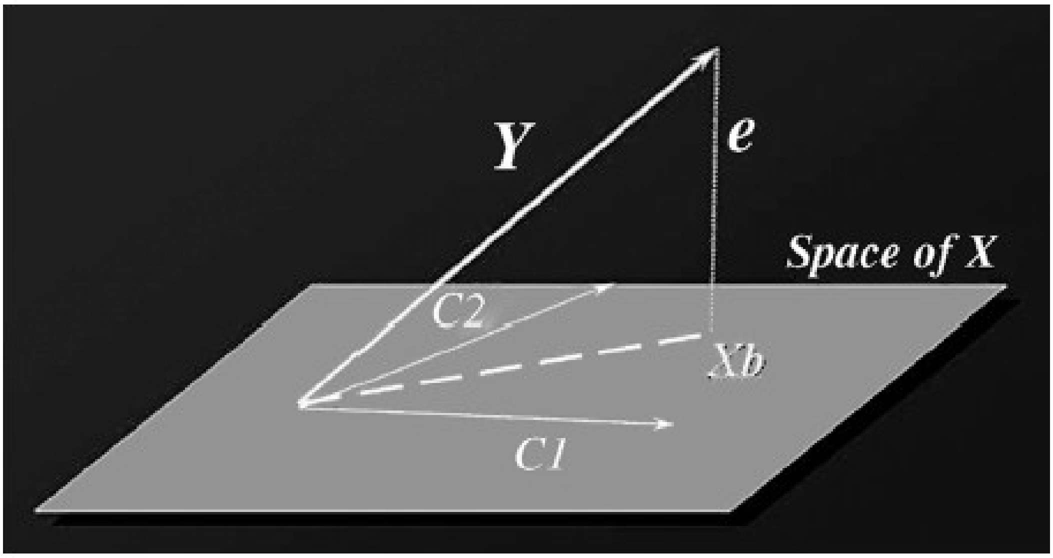


FIGURE 9.15 Geometrical perspective: estimation. The data Y are projected orthogonally onto the space of the design matrix (X) defined by two regressors $C1$ and $C2$. The error e is the distance between the data and the smallest possible within the model space.

We now use the standard result: If $p(a) = N(\mu, V)$ then

$$E[a^T Ba] = \mu^T B\mu + Tr(BV)$$

So, if $p(y) = N(X\hat{\beta}, \sigma^2 I_N)$ then

$$\begin{aligned} E[y^T Ry] &= \hat{\beta}^T X^T RX\hat{\beta} + Tr(\sigma^2 R) \quad (2.7) \\ &= \hat{\beta}^T (X^T X - X^T XX^- X)\hat{\beta} + Tr(\sigma^2 R) \\ &= Tr(\sigma^2 (I - P)) \\ &= \sigma^2 (N - r(P)) \\ &= \sigma^2 (N - k) \end{aligned}$$

So, an *unbiased* estimate of the variance is

$$\begin{aligned} \hat{\sigma^2} &= (y^T Ry)/(N - k) \quad (2.8) \\ &= RSS/(N - k) \end{aligned}$$

where the RSS is 'Residual Sum of Squares'. Remember, the ML variance estimate is

$$\hat{\sigma^2}_{ML} = (y^T R y)/N$$

2.3 Comparing nested GLMs

Full model:

$$y = X_0\beta_0 + X_1\beta_1 + e$$

Reduced model:

$$y = X_0\beta_0 + e_0$$

Consider the test-statistic

$$f = \frac{(RSS_{red} - RSS_{full})/(k-p)}{RSS_{full}/(N-k)}$$

where 'Residual Sum of Squares (RSS)' are

$$RSS_{full} = \hat{e}^T \hat{e}$$

$$RSS_{red} = \hat{e}_0^T \hat{e}_0$$

We can re-write in terms of 'Extra Sum of Squares'

$$f = \frac{ESS/(k-p)}{RSS_{full}/(N-k)}$$

where

$$ESS = RSS_{red} - RSS_{full}$$

We can compute these quantities using

$$\begin{aligned} RSS_{full} &= y^T R y \qquad (2.9) \\ RSS_{red} &= y^T R_0 y \end{aligned}$$

We expect the denominator to be

$$E[RSS_{full}/(N-k)] = \sigma^2 \tag{2.10}$$

and, under the null ($\beta_1 = 0$), we have $\sigma_0^2 = \sigma^2$ and therefore expect the numerator to be

$$E[(RSS_{red} - RSS_{full})/(k-p)] = \sigma^2 \tag{2.11}$$

where $r(R_0 - R) = k - p$ (mirroring the earlier expectation calculation). Under the null, we therefore expect a test statistic of unity

$$< f >= \frac{\sigma^2}{\sigma^2}$$

as both numerator and denominator are unbiased estimates of error variance. We might naively expect to get a numerator of zero, under the null. But this is not the case because, in any finite sample, ESS will be non zero. When we then divide by $(k-p)$ we get $E[ESS/(k-p)] = \sigma^2$.

When the full model is better we get a larger f value.

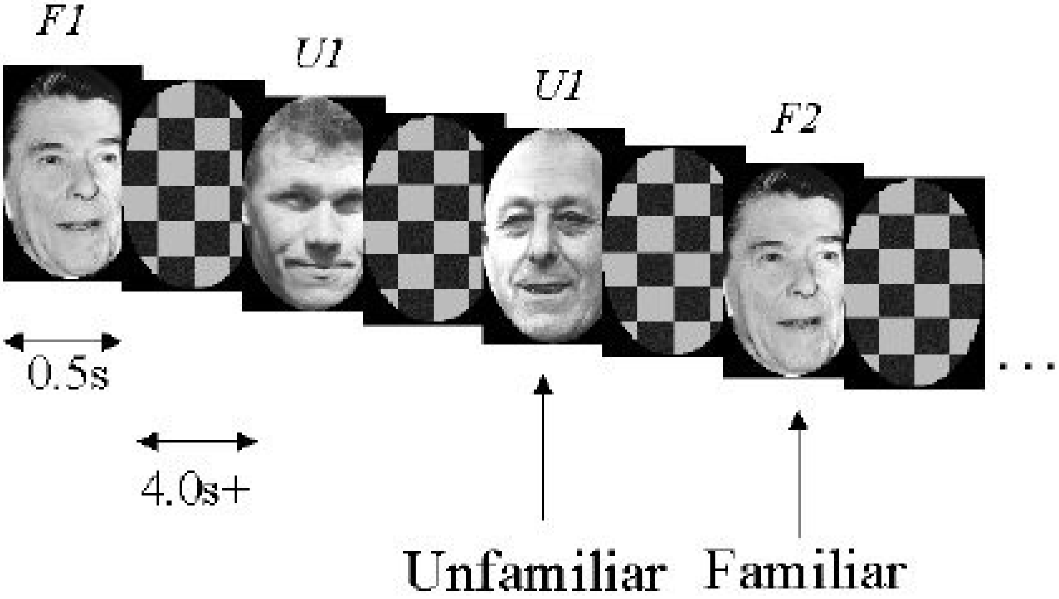
2.4 Partial correlation and R^2

The square of the partial correlaton coefficient

$$R^2_{y,X_1|X_0} = \frac{RSS_{red} - RSS_{full}}{RSS_{red}} \tag{2.12}$$

is the (square) of the correlation between y and $X_1\beta_1$ after controlling for the effect of $X_0\beta_0$. Abbreviating the above to R^2, the F-statistic can be re-written as

$$f = \frac{R^2/(k-p)}{(1-R^2)/(N-k)}$$



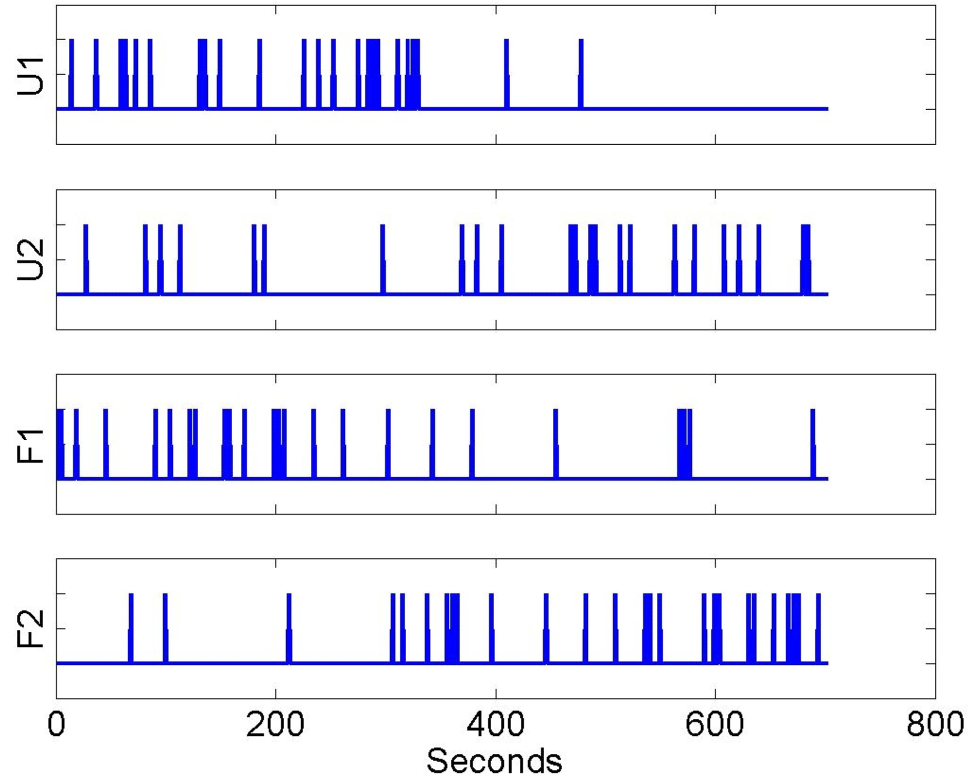
Model comparison tests are identical to tests of partial correlation.

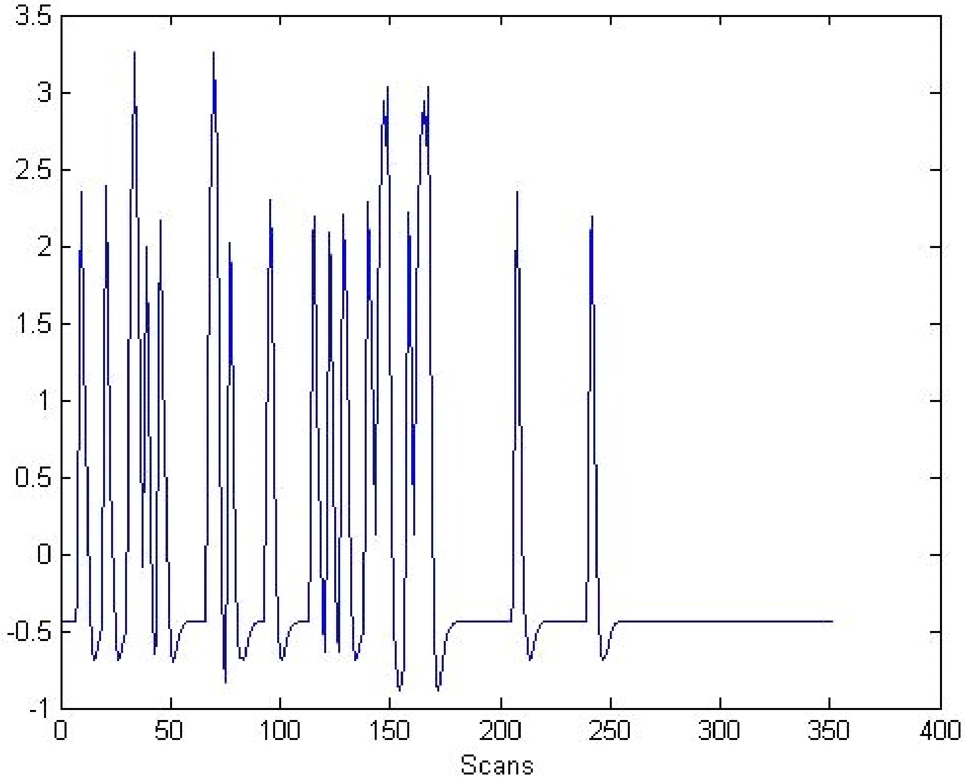
In X_0 explains no variance eg. it is a constant or empty matrix then

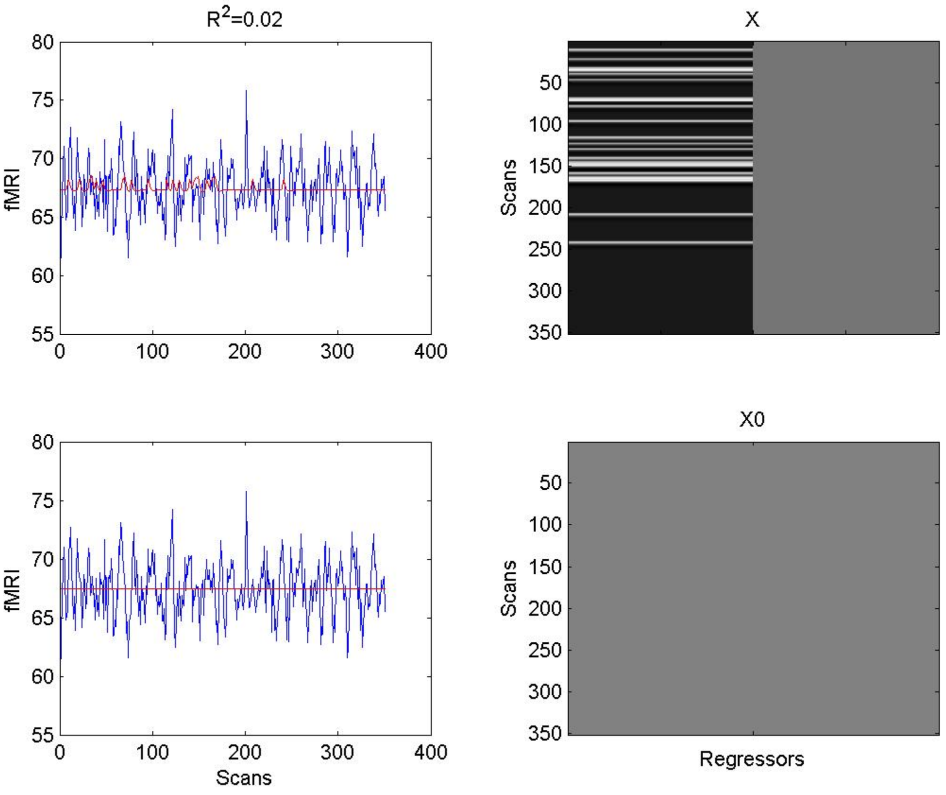
$$R^2 = \frac{Y^T Y - Y^T R Y}{Y^T Y} \tag{2.13}$$

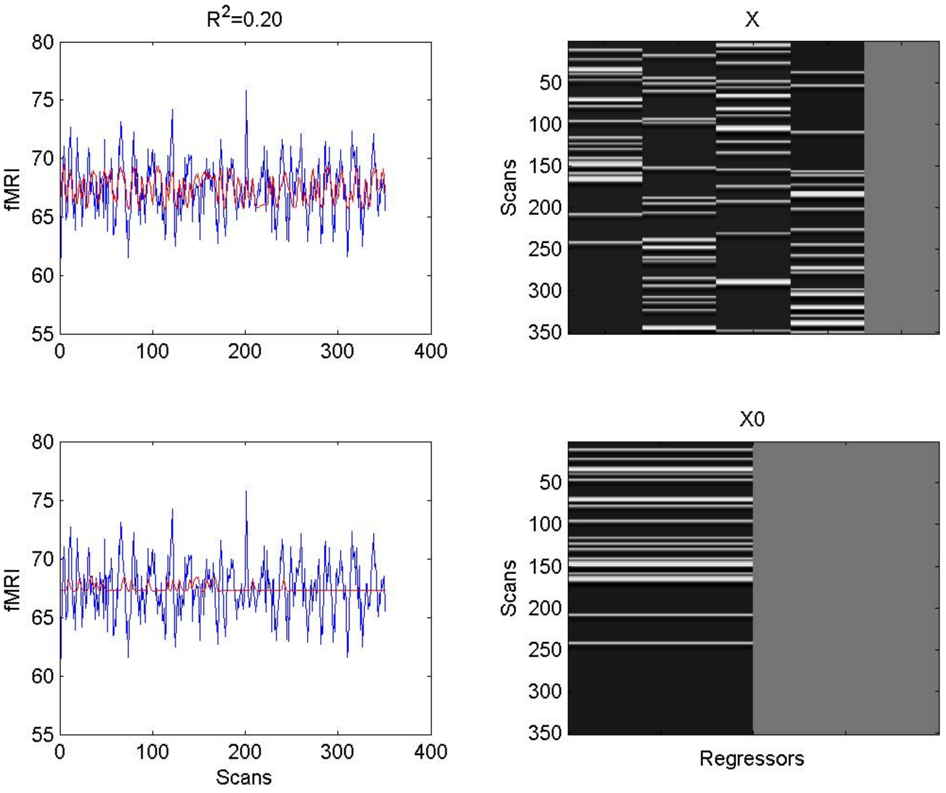
which is the proportion of variance explained by the model with design matrix X. More generally, if X_0 is not the empty matrix then R^2 is that proportion of the variability unexplained by the reduced model X_0 that is explained by the full model X.

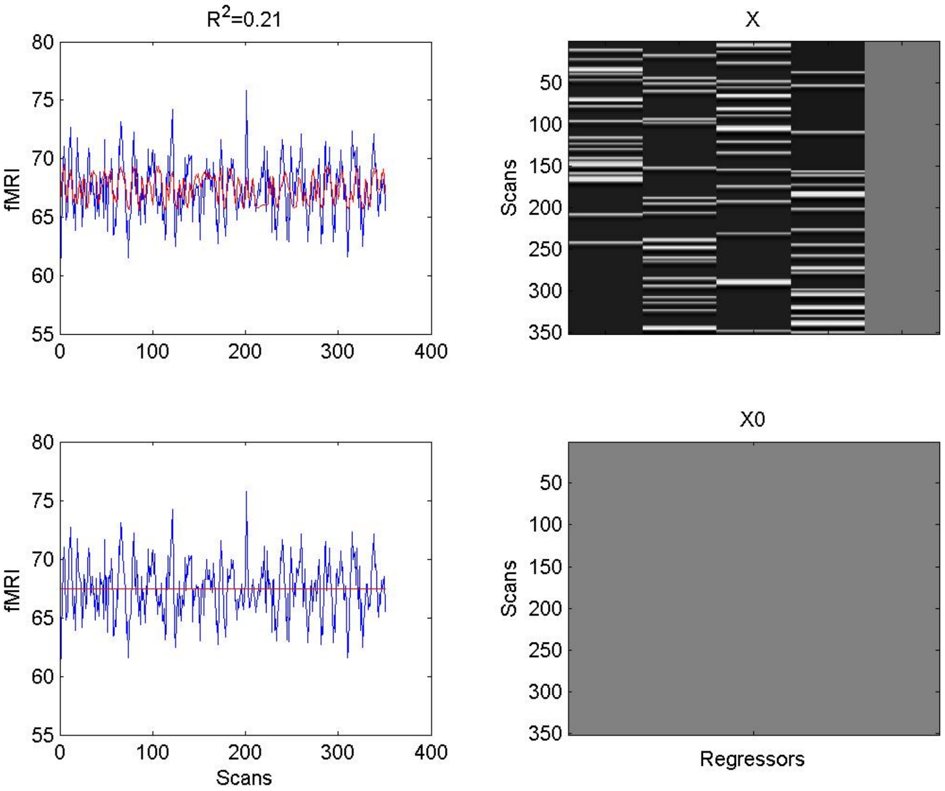
2.5 Examples


U1
U2
F1
F2
0
200
400
600
800
Seconds


3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
50
100
150
200
250
300
350
400
Scans


R^2=0.02
fMRI
80
75
70
65
60
55
0
100
200
300
400
Scans
X
X0
Scans
50
100
150
200
250
300
350
Regressors


R^2=0.20
fMRI
80
75
70
65
60
55
0
100
200
300
400
Scans
X
X0
Scans
50
100
150
200
250
300
350
Regressors



2.6 How large must f be for a ‘significant’ improvement ?

Under the null ($\beta_1 = 0$), f follows an F-distribution with $k-p$ numerator degrees of freedom (DF) and $N-k$ denominator DF.

Info on PDFs and transforming them.

2.7 Contrasts

We can also compare nested models using *contrasts*. This is more efficient, as we only need to estimate parameters of the *full* model.

For a contrast matrix C we wish to test the hypothesis $C^T\beta = 0$. This can correspond to a model comparison, as before, if C is chosen appropriately. But it is also more general, as we can test any effect which can be expressed as

$$C^T\beta = H^T X\beta$$

for some H. This defines a space of estimable contrasts.

The contrast C defines a subspace $X_c = XC$. As before, we can think of the hypothesis $C^T\beta = 0$ as comparing a full model, X, versus a reduced model which is now given by $X_0 = XC_0$ where C_0 is a contrast orthogonal to C ie.

$$C_0 = I_k - CC^-$$

A test statistic can then be generated as before where $R_0 = I_N - X_0X_0^-$, $M = R_0 - R$ and

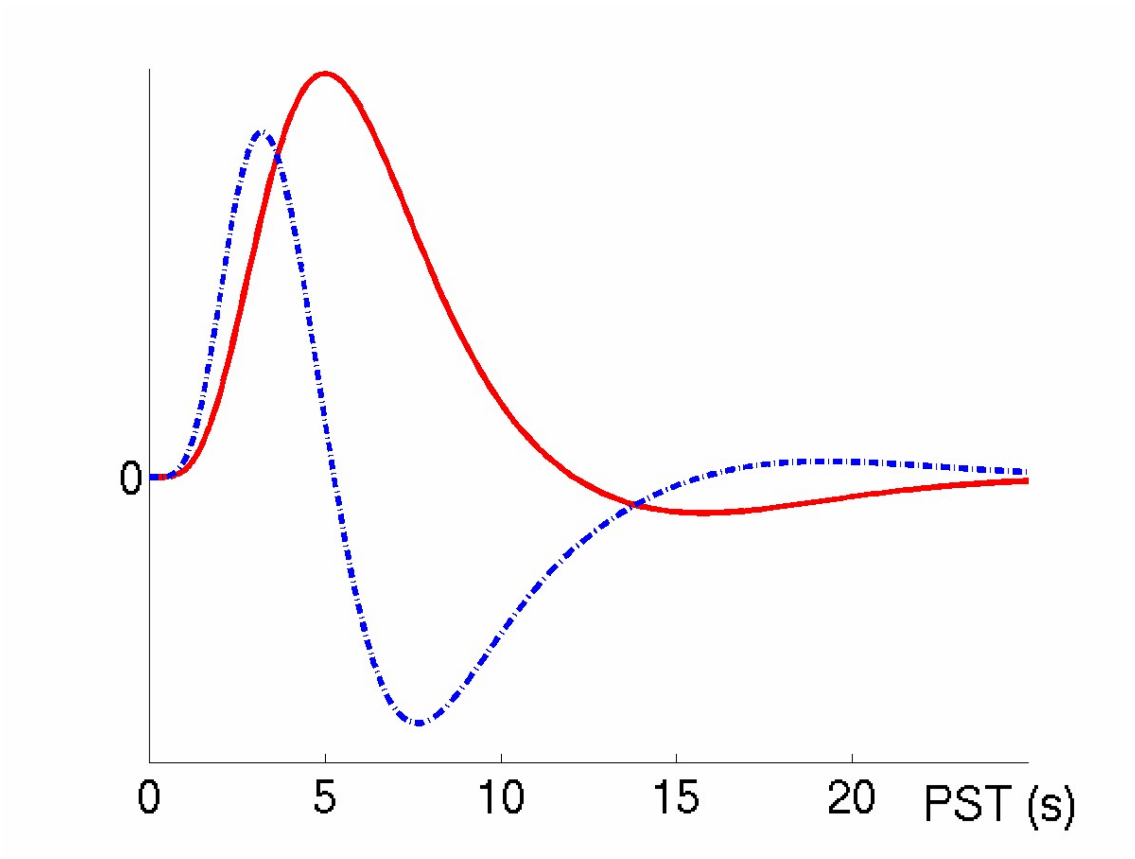
$$f = \frac{y^T My/r(M)}{y^T Ry/r(R)}$$

In fMRI, the use of contrasts allows us to test for (i) main effects and interactions in factorial designs, (ii) choice of hemodynamic basis sets. Importantly, we do not need to refit models.

The numerator can be calculated efficiently as

$$y^T My = \hat{c}^T \left[C^T(X^T X)^- C\right]^- \hat{c} \tag{2.14}$$

where $\hat{c} = C^T\hat{\beta}$ is the estimated effect size. See Christensen [10] for details.

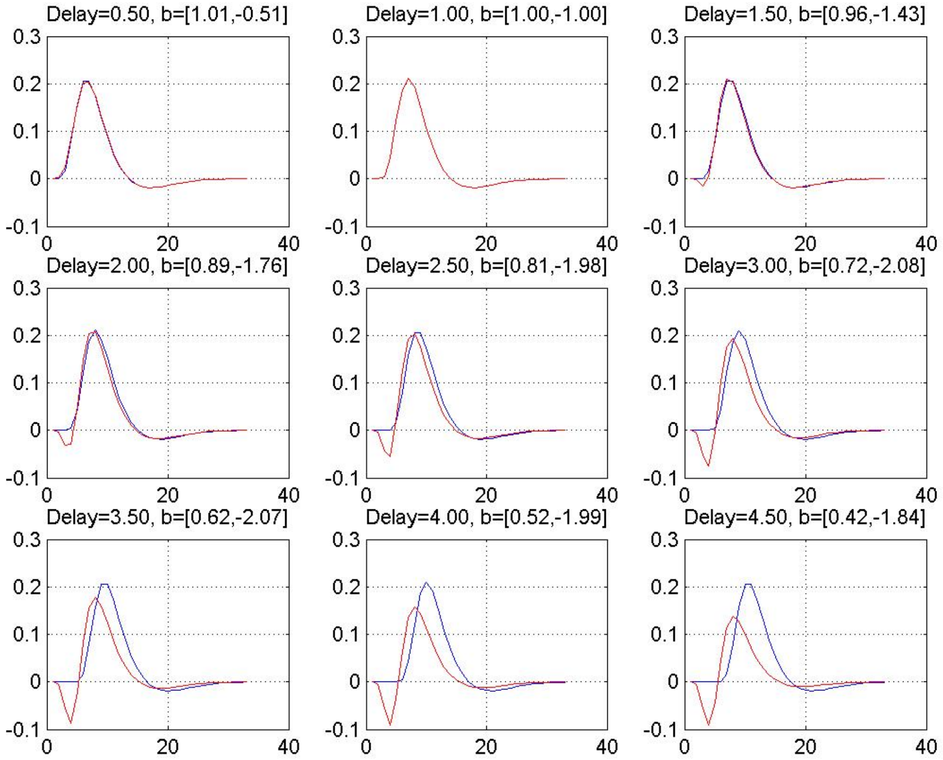


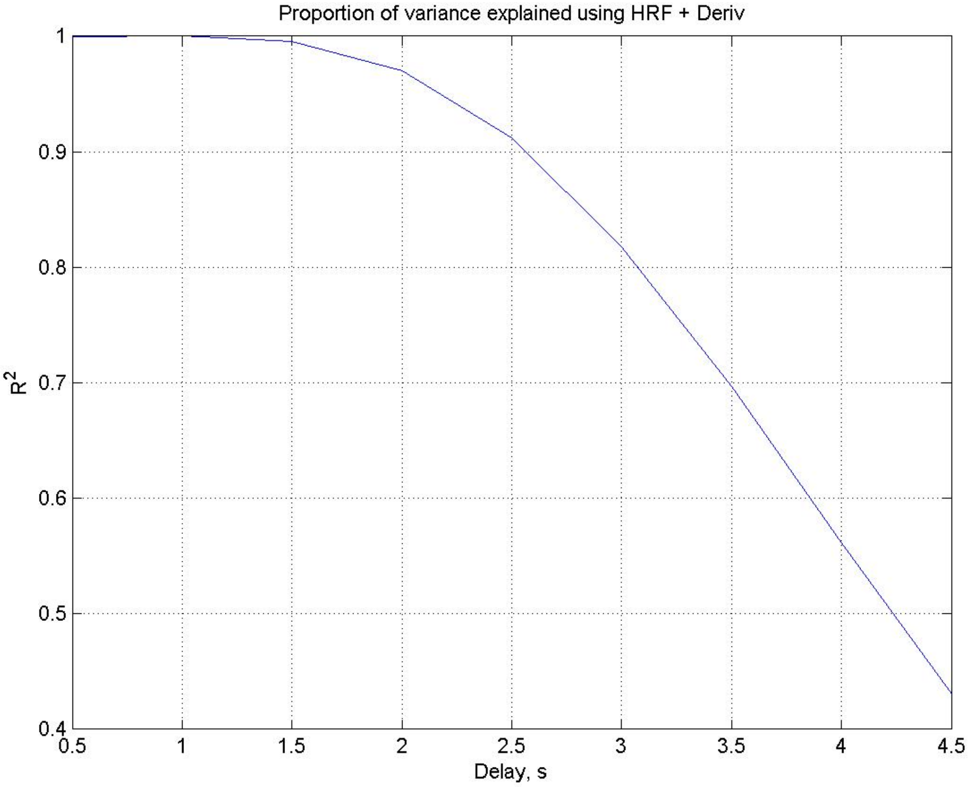
2.8 Hemodynamic basis functions

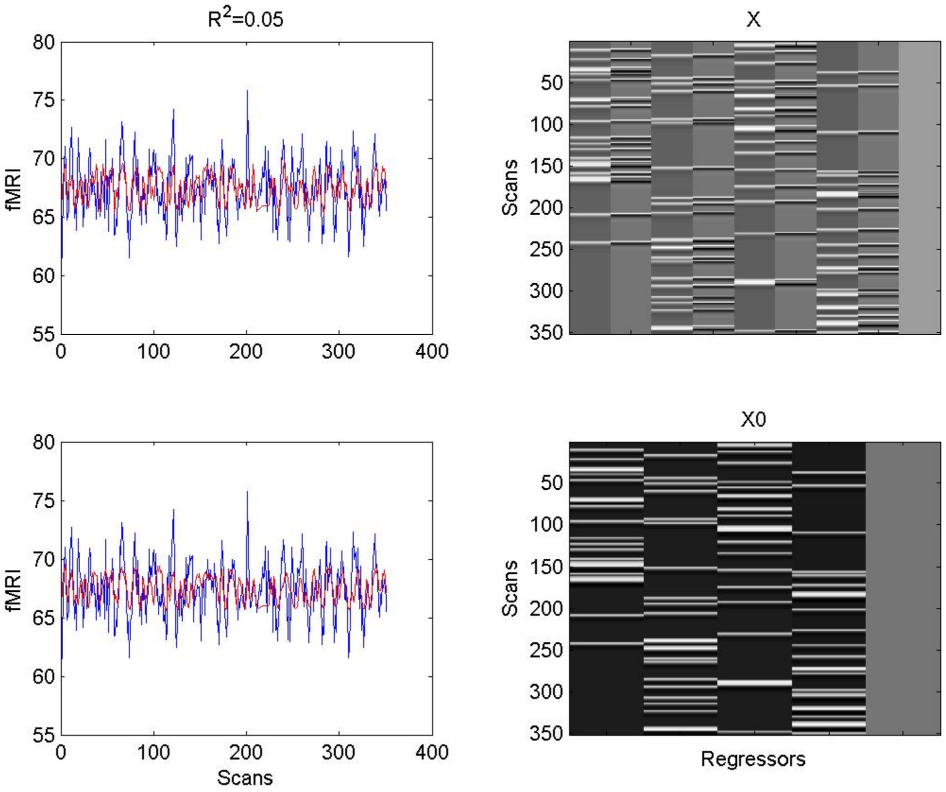
If $C(t,u)$ is the 'Canonical' basis function for event offset u then, using a first-order Taylor series approximation

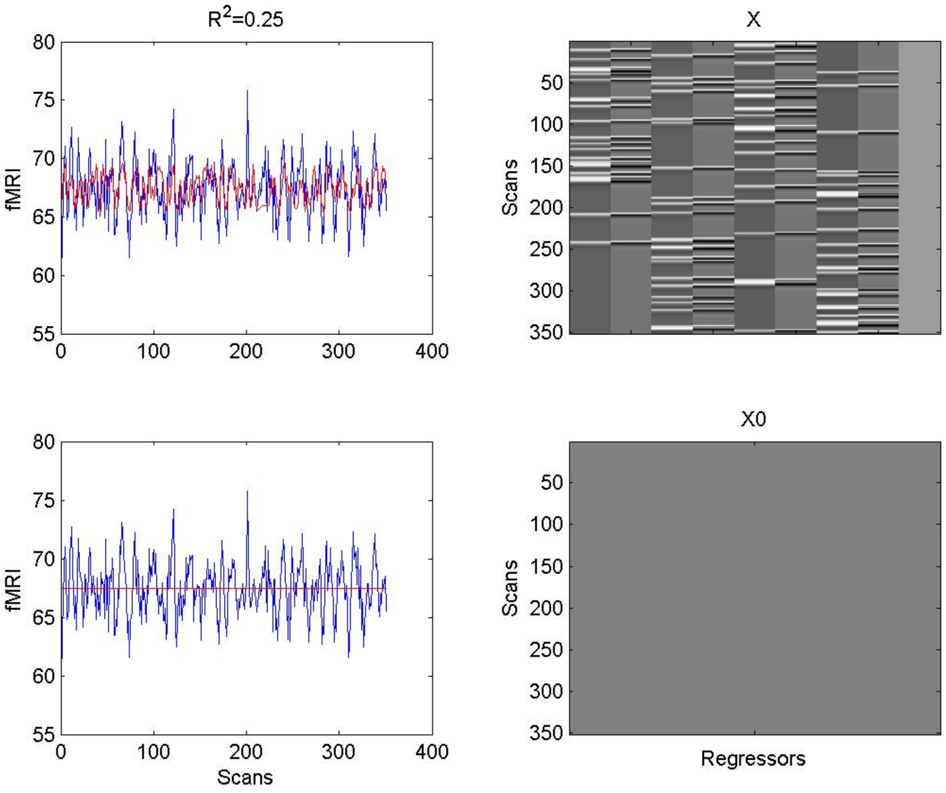
$$\begin{aligned} C(t,u_0+h) &\approx C(t,u_0) + h\frac{dC(t,u)}{du} \\ &\approx C(t,u_0) + hD(t,u_0) \end{aligned} \tag{2.15}$$

where the derivative is evaluated at $u = u_0$. This will allow us to accomodate small errors in event timings, or earlier/later rises in the hemodynamic response.


Delay=0.50, b=[1.01,-0.51]
Delay=1.00, b=[1.00,-1.00]
Delay=1.50, b=[0.96,-1.43]
Delay=2.00, b=[0.89,-1.76]
Delay=2.50, b=[0.81,-1.98]
Delay=3.00, b=[0.72,-2.08]
Delay=3.50, b=[0.62,-2.07]
Delay=4.00, b=[0.52,-1.99]
Delay=4.50, b=[0.42,-1.84]
0.3
0.2
0.1
0
-0.1
0
20
40


Proportion of variance explained using HRF + Deriv
R²
1
0.9
0.8
0.7
0.6
0.5
0.4
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Delay, s


R^2=0.05
X
X0
fMRI
Scans
Regressors
80
75
70
65
60
55
0
100
200
300
400
50
150
250
350


R^2=0.25
X
X0
fMRI
Scans
Regressors
80
75
70
65
60
55
0
100
200
300
400
50
150
250
350

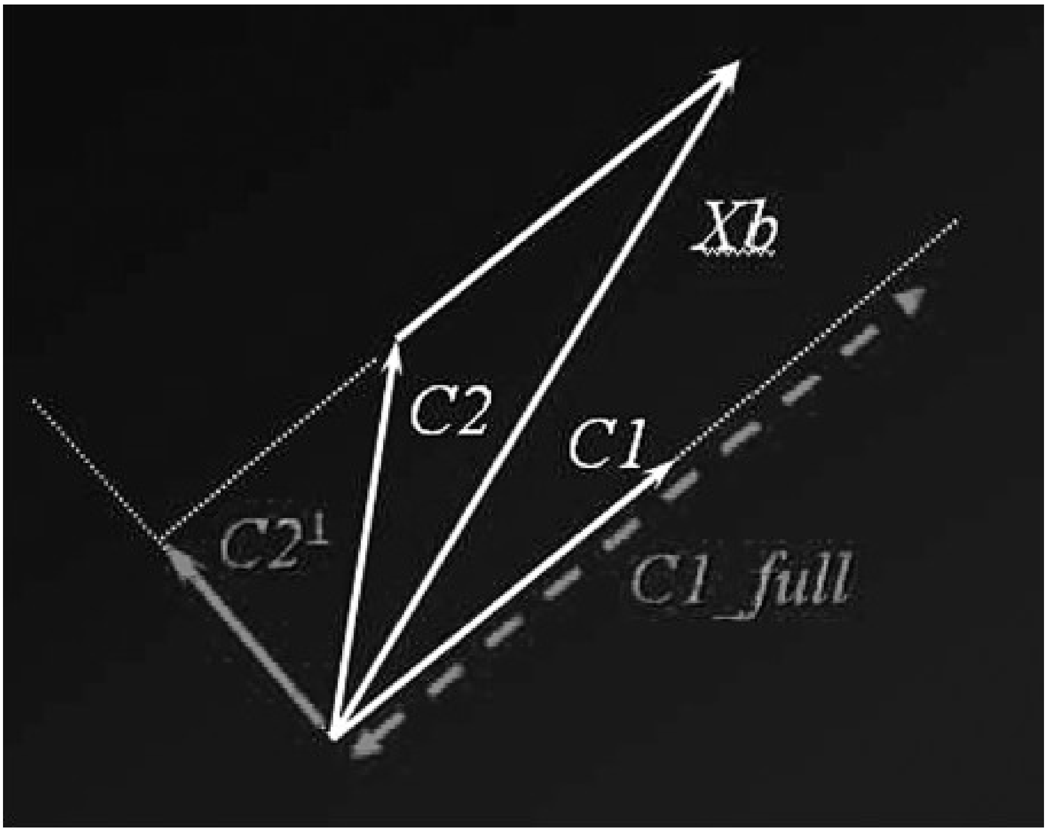


FIGURE 9.16 Hypothesis testing: the geometrical perspective. With a model defined by the two regressors $C1$ and $C2$, testing for $C2$ in effect measures its part orthogonal to C1. If the model is explicitly orthogonalized, (i.e. $C2$ is replaced by $C2^{orth}$), the test of $C2$ is unchanged, but the test of $C1$ is, and will capture more variablity, as indicated by $C1_{full}$.

Chapter 3

Random Field Theory

3.1 Inference for random fields

A random field is a set of random variables defined at every point in space. To find out if our z-score is ‘significant’ we need to find out the probability of getting a score that size (or greater) in the abscence of signal. In the absence of signal, we have just error fields. In brain imaging the error fields are spatially correlated and can be described by stochastic processes over space.

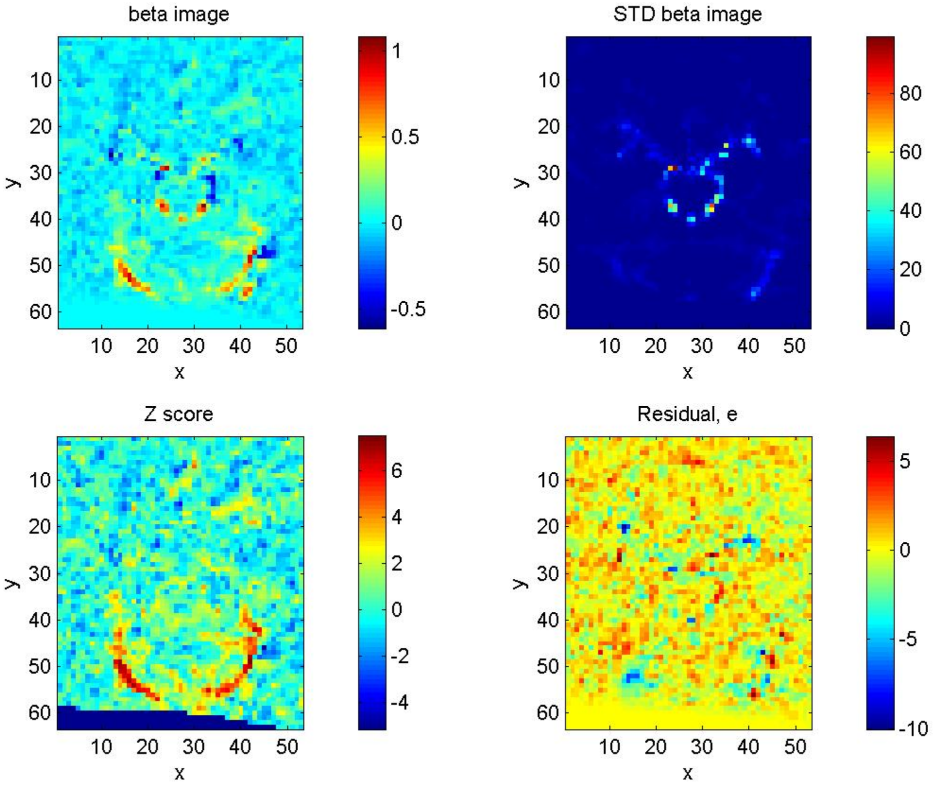


Figure 3.1: Face data: U1 effect.

3.1.1 Family Wise Error

We wish to find the probability, under the null hypothesis, that the maximum statistic over the field is larger than some threshold u. That is

$$p(U_{max} > u|H_0) \tag{3.1}$$

This is the probability of a Family Wise Error (FWE). An FWE is a false positive anywhere in the image.

3.2 Gaussian processes

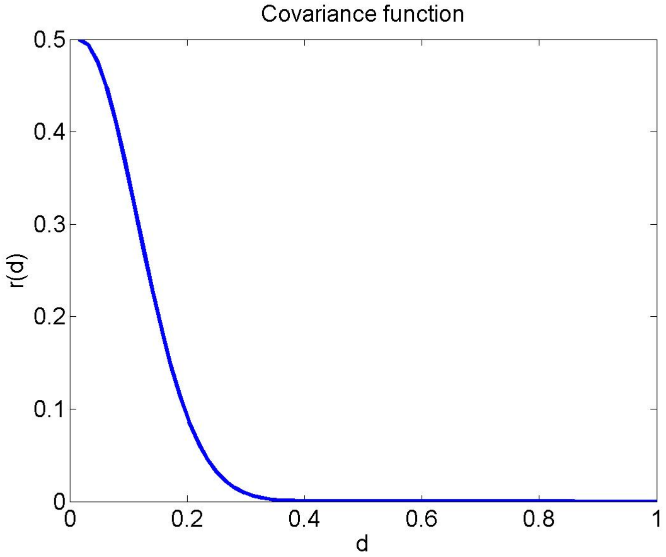
A stochastic process $x(v)$ is a Gaussian process if for any N samples the joint distribution $p(x(v_1), x(v_2), ..., x(v_N))$ is a multivariate Gaussian.

A Gaussian random field/process has a Gaussian distribution at every point and at every collection of points.

Gaussian processes (GPs) are therefore defined by a mean function $m(v)$ and a covariance function

$$r(u, v) = E\left([x(u) - m(u)]^T [x(v) - m(v)]\right) \tag{3.2}$$

A Gaussian process is stationary if $r(u, v) = r(u - v)$. We can then write the covariance function as $r(d)$ where $d = r - v$. In what follows we will assume the mean function to be zero at all points. GPs and their properties are then defined solely by their covariance function.



3.2.1 Example 1

A Gaussian covariance function is given by

$$r(d) = \sigma^2 \exp\left(-\frac{d^2}{2s^2}\right) \tag{3.3}$$

with power $\sigma^2 = 0.5$ and smoothness $s^2 = 0.1^2$

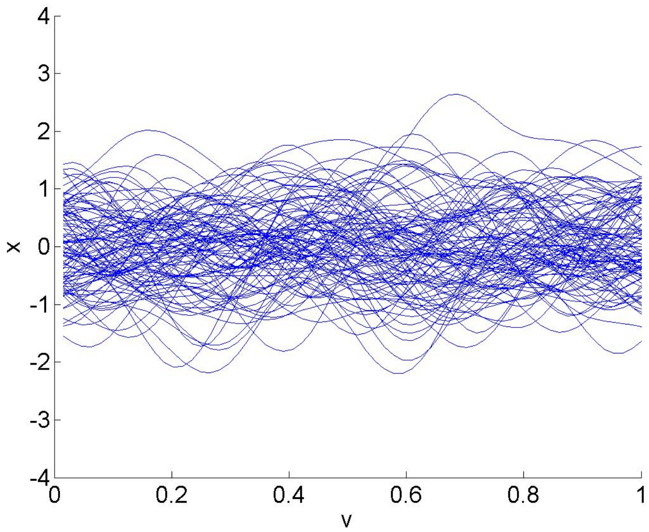


Figure 3.2: 100 realisations of a Gaussian process with previous Gaussian covariance function. See also NETLAB demo.

3.2.2 Power and roughness

For stationary processes, the distribution of the max statistic is determined solely by the power and roughness.

The power or variance of a stationary zero-mean Gaussian process is given by the covariance function at lag 0

$$E(|x(v)|^2) = r_x(0) \tag{3.4}$$

Given any Gaussian process we can create a new one by taking derivatives eg. $y = x'(v) = dx(v)/dv$. Using Fourier methods (see eg. page 325 in [34]) or making use of symmetry properties of the covariance function (see eg. page 314 in [34]) it can be shown that the covariance function of y is given by

$$r_y(v) = -r_x^{''}(v) \tag{3.5}$$

The power of the stochastic process y is

$$E(|y(v)|^2) = r_y(0) \tag{3.6}$$

Combining this with the result above shows that variance of the slope is given by

$$E(|\frac{dx(v)}{dv}|^2) = -r_x^{''}(0) \tag{3.7}$$

The 'roughness', λ is then given by the following ratio

$$\lambda^{1/2} = \frac{-r_x^{''}(0)}{r(0)} \tag{3.8}$$

For unit power fields we have $\lambda^{1/2} = -r_x^{''}(0)$. The 'smoothness' is defined as the inverse of the roughness.

For the results that follow, the covariance function can be chosen arbitrarily. However, some results are simplified if the covariance function has a particular form. For example, the covariance function could itself be Gaussian.

3.2.3 Gaussian covariance function

A Gaussian covariance function is given by

$$r(d) = \sigma^2 \exp\left(-\frac{d^2}{2s^2}\right) \tag{3.9}$$

At distance 0, $r(0) = \sigma^2$. Spatial derivatives are then given by

$$r'(d) = -\frac{d}{s^2} r(d) \tag{3.10}$$

Hence, $r'(0) = 0$. The second derivative is given by

$$\begin{aligned} r^{''}(d) &= -\frac{d}{s^2} \times -\frac{d}{s^2} r(d) - \frac{1}{s^2} r(d) \\ &= \left(\frac{d^2 - s^2}{s^4}\right) r(d) \end{aligned} \tag{3.11}$$

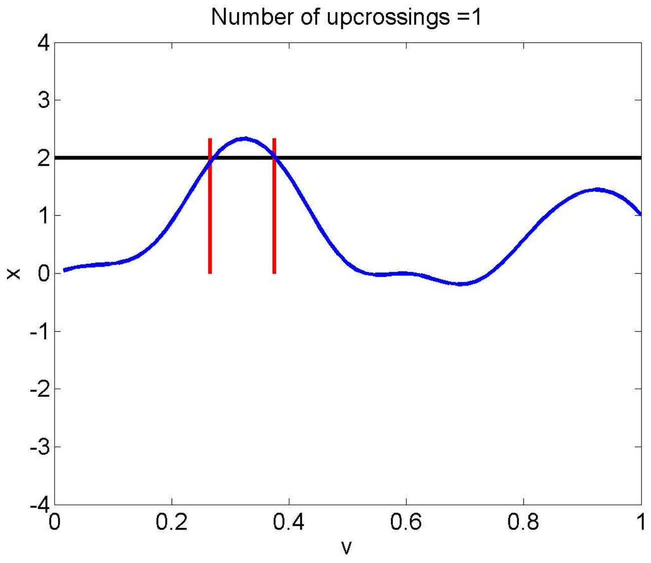


Figure 3.3: Crossings of a 1D field

Re-arranging shows that the roughness is given by,

$$\begin{aligned} \lambda^{1/2} &= -\frac{r''(0)}{r(0)} \\ &= \frac{1}{s^2} \end{aligned} \tag{3.12}$$

The roughness of a GP with a Gaussian CF is therefore $1/s^2$. The smoothness is then the square of the length scale s.

3.3 Crossings of one-dimensional processes

In a stationary 1-dimensional zero-mean Gaussian field the expected number of crossings, N_c, in the interval $[0, 1]$ of the level u is (page 606, [34])

$$E(N_c) = p_x(u)E(|x'(t)|) \tag{3.13}$$

That is the density at u multiplied by the expected slope. The density is the usual Gaussian

$$p_x(u) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left(-\frac{u^2}{2\sigma^2}\right) \tag{3.14}$$

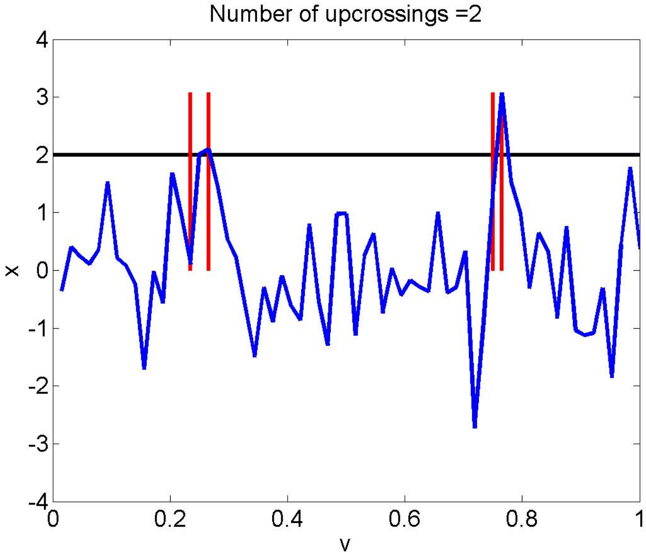


Figure 3.4: Crossings of a rougher higher power 1D field

and it can be shown that

$$E(|x'(t)|)^2 = \frac{-2r''(0)}{\pi} \tag{3.15}$$

The crossing density is therefore

$$E(N_c) = \frac{\lambda^{1/2}}{\pi\sigma} \exp\left(-\frac{u^2}{2\sigma^2}\right) \tag{3.16}$$

The expected number of upcrossings, N_u, is therefore half that (see also page 67, [1])

$$E(N_u) = \frac{\lambda^{1/2}}{2\pi\sigma} \exp\left(-\frac{u^2}{2\sigma^2}\right) \tag{3.17}$$

where $\sigma^2 = E(|x(v)|^2) = r(0)$ is the power and $\lambda^{1/2} = -r''(0)$ is the roughness. So, the greater the roughness the more upcrossings we expect. At high thresholds, u, $E(c)$ is the probability that the maximum of the process is larger than u.

3.4 Multi-dimensional processes

We assume standard Gaussian variates at each location (ie. $\sigma = r(0) = 1$). We first define an *excursion set* as the set of voxels where the statistical field exceeds a fixed threshold u.

3.4.1 Euler characteristic

See eg. [48]. The Euler characteristic, c, counts the number of disconnected components minus the number of 'holes' plus the number of 'hollows'. For high thresholds u the holes and hollows disappear and c counts the number of local maxima.

For large x the Euler characteristic, c, approaches the number of local maxima. Raising the threshold further either the global maxima is above threshold or it is not. So the *expected value* of c is then the probability that the global maximum exceeds the threshold u.

3.4.2 Expected Euler characteristic

The expected value of c for an N-dimensional stationary Gaussian process is given by (page 111 [1])

$$E[c] = V|\Lambda|^{1/2}(2\pi)^{-(N+1)/2}b(N,u)\exp\left(-\frac{u^2}{2}\right) \tag{3.18}$$

where

$$b(N,u) = \sum_{j=0}^{(N-1)/2}(-1)^j\frac{(2j)!}{j!2^j}u^{N-1-2j} \tag{3.19}$$

This general result rests on a theorem from differential topology known as Morse's theorem. Results for dimensions $N < 3$ can

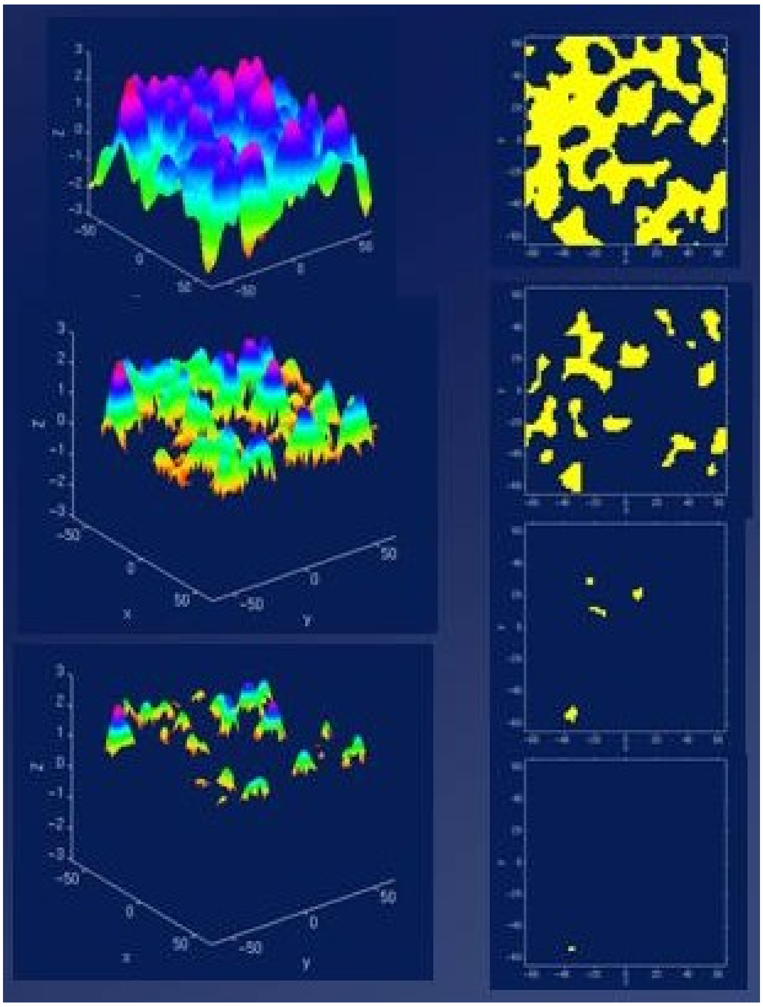


Figure 3.5: Thresholding a 2D field

be derived without this.

For $N = 1$ we have $b(N, u) = 1$

$$E[c] = V\frac{\lambda^{1/2}}{2\pi}\exp\left(-\frac{u^2}{2}\right) \tag{3.20}$$

which is the same result as earlier for the expected number of upcrossings (assuming $V = 1$, $\sigma = 1$).

For $N = 2$ (eg. brain slice) we have $b(N, u) = u$ and

$$E[c] = V|\Lambda|^{1/2}(2\pi)^{-3/2}u\exp\left(-\frac{u^2}{2}\right) \tag{3.21}$$

For $N = 3$ (eg. brain volume) we have $b(N, u) = u^2 - 1$ and

$$E[c] = V|\Lambda|^{1/2}(2\pi)^{-2}(u^2 - 1)\exp\left(-\frac{u^2}{2}\right) \tag{3.22}$$

3.4.3 Gaussian smoothing

One can create a Gaussian process by convolving IID Gaussian noise with a Gaussian kernel (ie. a Gaussian with covariance matrix Λ^{-1}. For a 3D field, if the principal axes of Λ coincide with the x, y and z directions then the off-diagonal elements of Λ are zero. If f_x, f_y and f_z are the Full Width at Half Maximums (FWHMs) in the x, y and z directions then the roughness is given by [51]

$$|\Lambda|^{1/2} = (f_x f_y f_z)^{-1}(4\ln 2)^{3/2} \tag{3.23}$$

If we then define the number of resels as

$$R = \frac{V}{f_x f_y f_z} \tag{3.24}$$

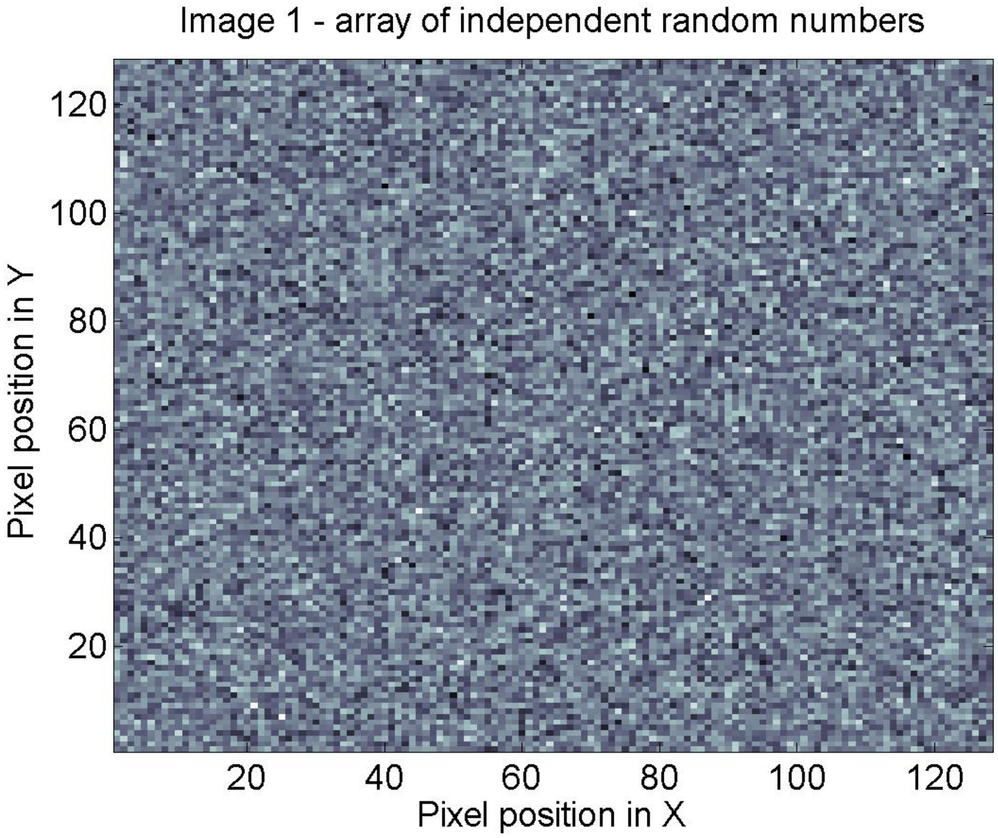


Figure 3.6:

then for volumetric data we can write (see eg. [])

$$E[c] = R(4\ln 2)^{3/2}(2\pi)^{-2}(u^2-1)\exp\left(-\frac{u^2}{2}\right) \tag{3.25}$$

The above formula only applies to stationary Gaussian fields with Gaussian CFs (these can be created by smoothing IID data with a Gaussian kernel). But because roughness is a property at zero lag, in practice the above formula works well if the covariance function at zero lag is similar to that of a Gaussian CF. It does'nt matter what the tails of the CF look like. So the result can be used for non-Gaussian covariance functions as long as the above holds [50].

3.4.4 Slice data

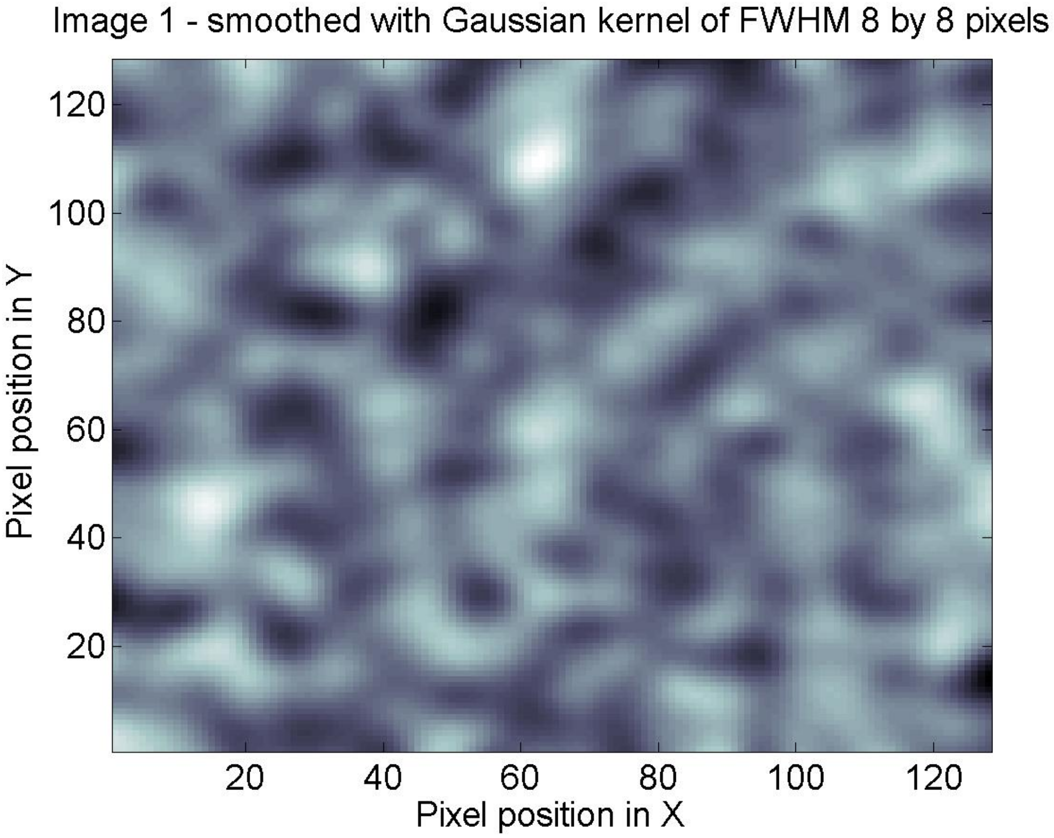


Image 1 - smoothed with Gaussian kernel of FWHM 8 by 8 pixels
120
100
80
60
40
20
Pixel position in Y
20
40
60
80
100
120
Pixel position in X

Figure 3.7:

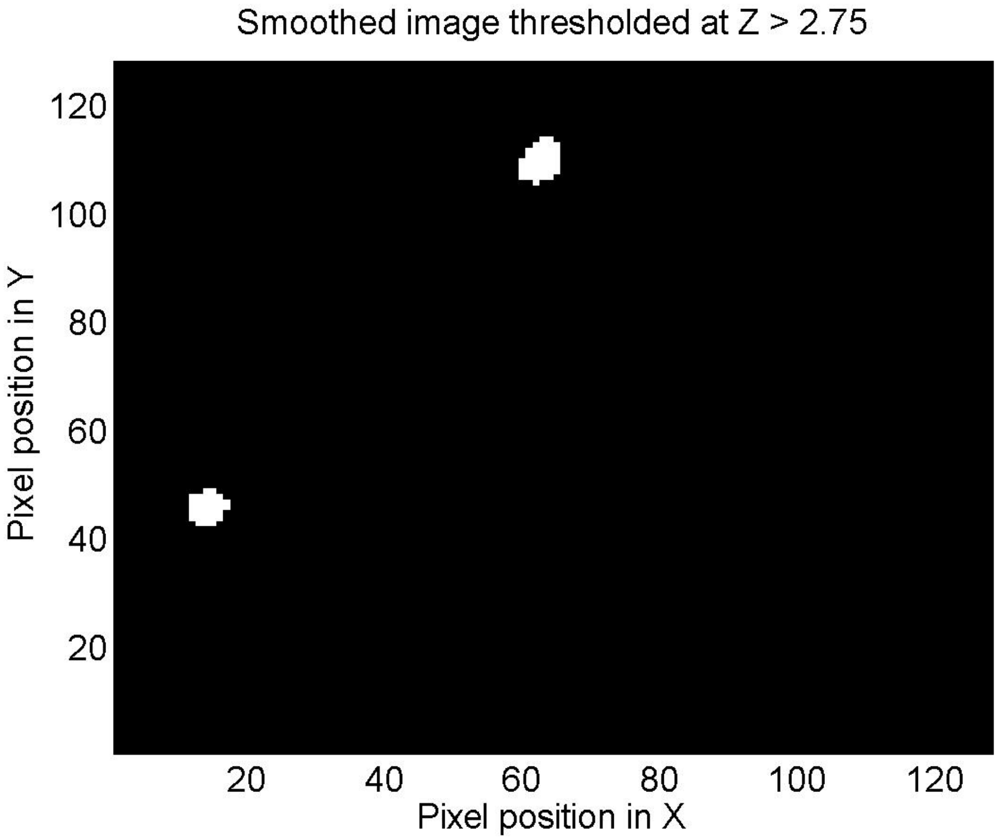

Smoothed image thresholded at Z > 2.75
120
100
80
60
40
20
Pixel position in Y
20
40
60
80
100
120
Pixel position in X

Figure 3.8:

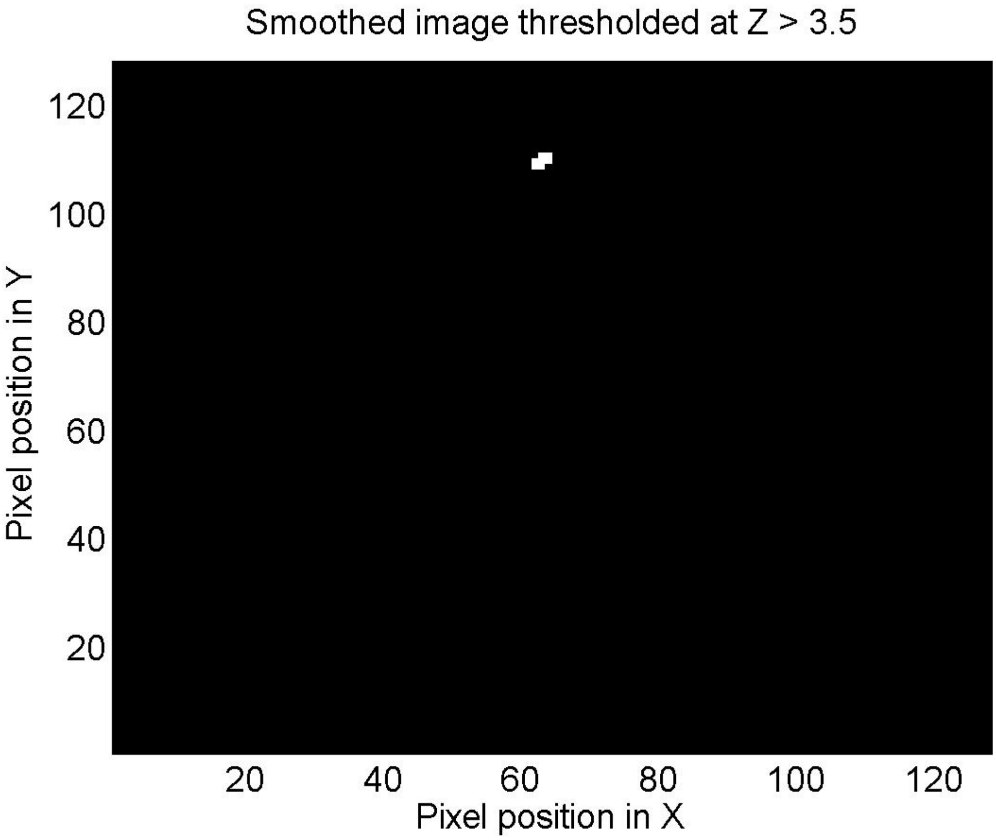

Smoothed image thresholded at Z > 3.5
120
100
80
60
40
20
Pixel position in Y
20
40
60
80
100
120
Pixel position in X

Figure 3.9:

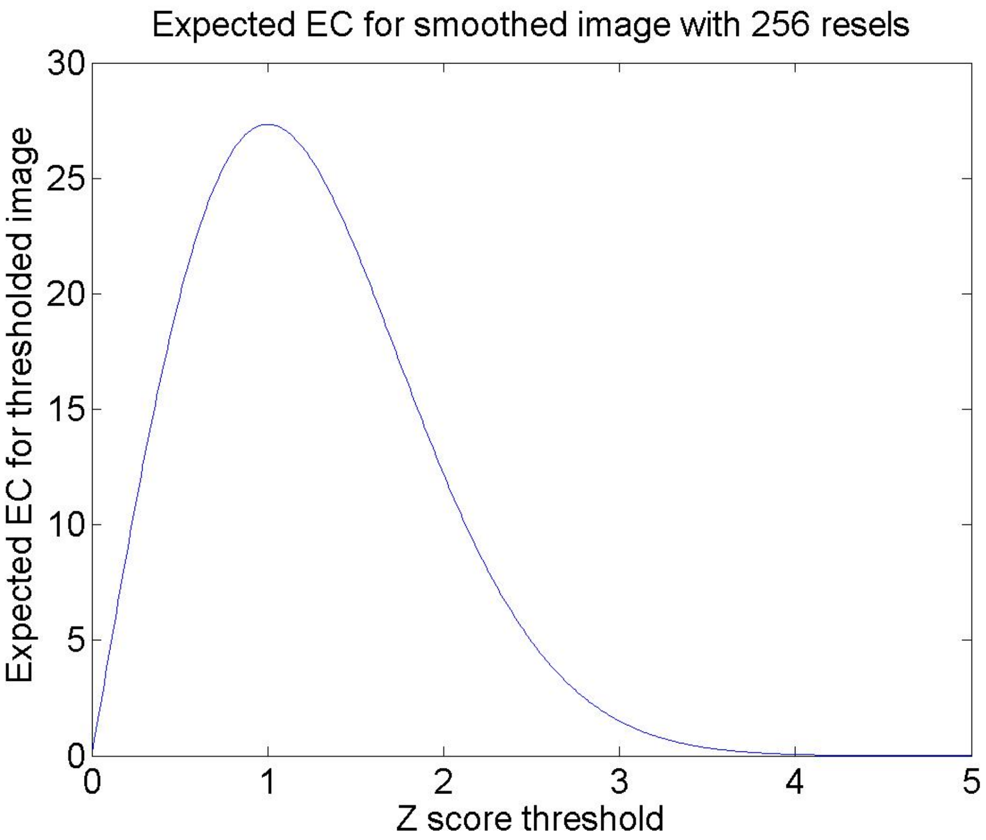

Expected EC for smoothed image with 256 resels
Expected EC for thresholded image
Z score threshold
0
5
10
15
20
25
30
0
1
2
3
4
5

Figure 3.10:

3.5 Further issues

3.5.1 Estimating roughness

Roughness can be estimated using numerical derivatives of the residuals in each of the x, y and z directions. These are then averaged over different residual images (SPM uses 64). These are stored as a Resels Per Voxel (RPV) image, then averaged over voxels [51]. See eg. face data.

3.5.2 Discretisation

The application of continuous theory to data sampled at discrete points requires that voxel size be eg. 3 times as small as the smoothness of the field. The theory in [51] also requires the search region to be considerably larger than the smoothness (see below).

3.5.3 Non-Gaussian processes

The results have been extended to t, χ^2 and F random fields [50]. This extension also provides accurate approximations for small search volumes (see Small Volume Correction (SVC) button in SPM). In this work Worsley derives the 'unified formula'

$$E(c) = \sum_{N=1}^{3} R_N(V) p_N(u) \tag{3.26}$$

where N is the dimension of the field, V is the search volume, $R_N(V)$ is the number of resels in dimension V, and $p_N(V)$ is the EC density for threshold u. The above equation can be solved for u to find the appropriate threshold.

3.5.4 Inferences about extent

For a given level u, one can work out the probability that the extent of an activation is greater than k. This is known as a cluster-level inference [14].

3.5.5 Nonstationary fields

The assumption of stationarity is reasonable for PET or smoothed fMRI data. But functional data projected onto unfolded or flattened cortical surfaces or anatomical data such as deformation vectors are highly non-isotropic. Such data can be dealt with by warping voxel coordinates so the effective FWHM is constant [49]. The method has a minor impact on height inferences but a major impact on extent inferences. It is therefore most useful for eg. cluster-level inference for VBM.

Chapter 4

Multivariate Models

4.1 Linear algebra

4.1.1 Orthogonal Matrices

The set of vectors $\boldsymbol{q}_1..\boldsymbol{q}_k$ are *orthogonal* if

$$\boldsymbol{q}_j^T\boldsymbol{q}_k \;=\; \begin{matrix} 0 & j \neq k \\ d_{jk} & j = k \end{matrix} \tag{4.1}$$

If these vectors are placed in columns of the matrix $\boldsymbol{Q}$ then

$$\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{D} \tag{4.2}$$

4.1.2 Orthonormal Matrices

The set of vectors $\boldsymbol{q}_1..\boldsymbol{q}_k$ are *orthonormal* if

$$\boldsymbol{q}_j^T\boldsymbol{q}_k \;=\; \begin{matrix} 0 & j \neq k \\ 1 & j = k \end{matrix} \tag{4.3}$$

If these vectors are placed in columns of the matrix $\boldsymbol{Q}$ then

$$\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I} \tag{4.4}$$

Hence, the transpose equals the inverse

$$\boldsymbol{Q}^T = \boldsymbol{Q}^{-1} \tag{4.5}$$

The vectors $\boldsymbol{q}_1..\boldsymbol{q}_k$ are said to provide an *orthonormal basis*. This means that *any* vector can be written as a linear combination of the basis vectors. A trivial example is the two-dimensional cartesian coordinate system where $\boldsymbol{q}_1 = [1, 0]^T$ (the x-axis) and $\boldsymbol{q}_2 = [0, 1]^T$ (the y-axis). More generally, to represent the vector $\boldsymbol{x}$ we can write

$$\boldsymbol{x} = \tilde{x}_1\boldsymbol{q}_1 + \tilde{x}_2\boldsymbol{q}_2 + ... + \tilde{x}_d\boldsymbol{q}_d \tag{4.6}$$

To find the appropriate coefficients $\tilde{x}_k$(the co-ordinates in the new basis), multiply both sides by $\boldsymbol{q}_k^T$. Due to the orthonormality property all terms on the right disappear except one leaving

$$\tilde{x}_k = \boldsymbol{q}_k^T\boldsymbol{x} \tag{4.7}$$

The new coordinates are the projections of the data onto the basis functions (re. definition of projections in earlier lecture, there is no denominator since $\boldsymbol{q}_k^T\boldsymbol{q}_k = 1$). In matrix form, equation 4.6 can be written as $\boldsymbol{x} = \boldsymbol{Q}\tilde{\boldsymbol{x}}$ which therefore has the solution $\tilde{\boldsymbol{x}} = \boldsymbol{Q}^{-1}\boldsymbol{x}$. But given that $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^T$ we have

$$\tilde{\boldsymbol{x}} = \boldsymbol{Q}^T\boldsymbol{x} \tag{4.8}$$

So for orthonormal bases, eg. Fourier or wavelets, data can be transformed from data to parameter space and vice-versa without inverse operators (not so for GLM with arbitrary design matrix).

4.1.3 Determinants

The determinant of a two-by-two matrix

$$\boldsymbol{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{4.9}$$

is given by

$$\det(\boldsymbol{A}) = ad - bc \tag{4.10}$$

The determinant of a three-by-three matrix

$$\boldsymbol{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \tag{4.11}$$

is given by

$$\det(\boldsymbol{A}) = a \det\left(\begin{bmatrix} e & f \\ h & i \end{bmatrix}\right) - b \det\left(\begin{bmatrix} d & f \\ g & i \end{bmatrix}\right) + c \det\left(\begin{bmatrix} d & e \\ g & h \end{bmatrix}\right)$$

Determinants are important because of their properties. In particular, if two rows of a matrix are equal then the determinant is zero eg. if

$$\boldsymbol{A} = \begin{bmatrix} a & b \\ a & b \end{bmatrix} \tag{4.12}$$

then

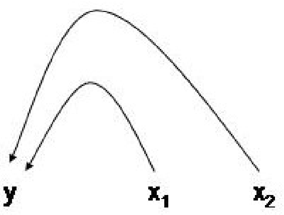
$$\det(\boldsymbol{A}) = ab - ba = 0 \tag{4.13}$$

In this case the transformation from $\boldsymbol{x} = [x_1, x_2]^T$ to $\boldsymbol{y} = [y_1, y_2]^T$ given by

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \tag{4.14}$$

reduces two pieces of information (x_1 and x_2) to one piece of information

$$y = y_1 = y_2 = ax_1 + bx_2 \tag{4.15}$$



In this case it is not possible to reconstruct $\boldsymbol{x}$ from $\boldsymbol{y}$; the transformation is not invertible - the matrix $\boldsymbol{A}$ does not have an inverse and the value of the determinant provides a test for this: If $\det(\boldsymbol{A}) = 0$ the matrix $\boldsymbol{A}$ is not invertible; it is *singular*. Conversely, if $\det(\boldsymbol{A}) \neq 0$ then $\boldsymbol{A}$ *is* invertible.

Another important property of determinants is that they measure the 'volume' of a matrix. For a 3-by-3 matrix the three rows of the matrix form the edges of a cube. The determinant is the volume of this cube. For a d-by-d matrix the rows form the edges of a 'parallepiped'. Again, the determinant is the volume.

We also write

$$\det(\boldsymbol{A}) = |\boldsymbol{A}| \tag{4.16}$$

4.1.4 Eigenanalysis

The square matrix $\boldsymbol{A}$ has eigenvalues λ and eigenvectors $\boldsymbol{q}$ if

$$\boldsymbol{A}\boldsymbol{q} = \lambda\boldsymbol{q} \tag{4.17}$$

Therefore

$$(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{q} = 0 \tag{4.18}$$

To satisfy this equation either $\boldsymbol{q} = 0$, which is uninteresting, or the matrix $\boldsymbol{A} - \lambda\boldsymbol{I}$ must reduce $\boldsymbol{q}$ to the null vector (a single point). For this to happen $\boldsymbol{A} - \lambda \boldsymbol{I}$ must be singular. Hence

$$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0 \tag{4.19}$$

Eigenanalysis therefore proceeds by (i) solving the above equation to find the eigenvalues λ_i and then (ii) substituting them into equation 4.17 to find the eigenvectors. For example, if

$$\boldsymbol{A} \;=\; \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix} \tag{4.20}$$

then

$$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = (4-\lambda)(-3-\lambda) - (-5)(2) = 0 \tag{4.21}$$

which can be rearranged as

$$\begin{aligned} \lambda^2 - \lambda - 2 &= 0 \qquad (4.22) \\ (\lambda+1)(\lambda-2) &= 0 \end{aligned}$$

Hence the eigenvalues are $\lambda = -1$ and $\lambda = 2$. Substituting back into equation 4.17 gives an eigenvector $\boldsymbol{q}_1$ which is any multiple of $[1, 1]^T$. Similarly, eigenvector $\boldsymbol{q}_2$ is any multiple of $[5, 2]^T$.

We now note that the determinant of a matrix is also equal to the product of its eigenvalues

$$\det(\boldsymbol{A}) = \prod_k \lambda_k \tag{4.23}$$

We also define the *Trace* of a matrix as the sum of its diagonal elements

$$Tr(\boldsymbol{A}) = \sum_k a_{kk} \tag{4.24}$$

and note that it is also equal to the sum of the eigenvalues

$$Tr(\boldsymbol{A}) = \sum_k \lambda_k \tag{4.25}$$

Eigenanalysis applies only to square matrices.

4.1.5 Diagonalization

If we put the eigenvectors into the columns of a matrix

$$\boldsymbol{Q} = \begin{bmatrix} | & | & . & | \\ | & | & . & | \\ \boldsymbol{q}_1 & \boldsymbol{q}_2 & . & \boldsymbol{q}_d \\ | & | & . & | \\ | & | & . & | \end{bmatrix} \tag{4.26}$$

then, because, $\boldsymbol{A}\boldsymbol{q}_k = \lambda_k \boldsymbol{q}_k$, we have

$$\boldsymbol{A}\boldsymbol{Q} = \begin{bmatrix} | & | & . & | \\ | & | & . & | \\ \lambda_1\boldsymbol{q}_1 & \lambda_2\boldsymbol{q}_2 & . & \lambda_d\boldsymbol{q}_d \\ | & | & . & | \\ | & | & . & | \end{bmatrix} \tag{4.27}$$

If we put the eigenvalues into the matrix $\boldsymbol{\Lambda}$ then the above matrix can also be written as $\boldsymbol{Q}\boldsymbol{\Lambda}$. Therefore,

$$\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{\Lambda} \tag{4.28}$$

Pre-multiplying both sides by $\boldsymbol{Q}^{-1}$ gives

$$\boldsymbol{Q}^{-1}\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{\Lambda} \tag{4.29}$$

This shows that any square matrix can be converted into a diagonal form (provided it has distinct eigenvalues; see eg. [44] p. 255).

4.1.6 Spectral Theorem

For any real *symmetric* matrix all the eigenvalues will be real and there will be d distinct eigenvalues and orthogonal eigenvectors. They can be normalised and placed into the matrix $\boldsymbol{Q}$.

Since $\boldsymbol{Q}$ is now orthonormal we have $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^T$. Hence

$$\boldsymbol{Q}^T \boldsymbol{A} \boldsymbol{Q} = \boldsymbol{\Lambda} \tag{4.30}$$

Pre-multiplying by $\boldsymbol{Q}$ and post-multiplying by $\boldsymbol{Q}^T$ gives

$$\boldsymbol{A} = \boldsymbol{Q} \boldsymbol{\Lambda} \boldsymbol{Q}^T \tag{4.31}$$

which is known as the *spectral theorem*. It says that any real, symmetric matrix can be represented as above where the columns of $\boldsymbol{Q}$ contain the eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix containing the eigenvalues, λ_i. Equivalently,

$$\boldsymbol{A} \;=\; \begin{bmatrix} | & | & . & | \\ | & | & . & | \\ \boldsymbol{q}_1 & \boldsymbol{q}_2 & . & \boldsymbol{q}_d \\ | & | & . & | \\ | & | & . & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & & \\ & & & \\ & & & \lambda_d \end{bmatrix} \begin{bmatrix} - & - & \boldsymbol{q}_1 & - & - \\ - & - & \boldsymbol{q}_2 & - & - \\ . & . & . & & . \\ - & - & \boldsymbol{q}_d & - & - \end{bmatrix} \tag{4.32}$$

This can also be written as a summation

$$\boldsymbol{A} = \sum_{k=1}^{d} \lambda_k \boldsymbol{q}_k \boldsymbol{q}_k^T \tag{4.33}$$

This provides a particularly efficient way to compute powers of matrices

$$\boldsymbol{A}^k = \boldsymbol{Q} \boldsymbol{\Lambda}^k \boldsymbol{Q}^T \tag{4.34}$$

This is particularly useful for solving multivariate difference and differential equations (see later lecture). Using the above with $k = -1$ shows $\det(A^{-1}) = 1/\det(A)$.

4.1.7 Quadratic Forms

The quadratic function

$$f(\boldsymbol{x}) = a_{11}x_1^2 + a_{12}x_1x_2 + a_{21}x_2x_1 + ... + a_{dd}x_d^2 \tag{4.35}$$

can be written in matrix form as

$$f(\boldsymbol{x}) \;=\; [x_1, x_2, ..., x_d] \begin{bmatrix} a_{11} & a_{12} & a_{1d} \\ a_{21} & a_{22} & a_{2d} \\ & & \\ & & \\ a_{d1} & a_{d2} & a_{dd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_d \end{bmatrix} \tag{4.36}$$

which is written compactly as

$$f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \tag{4.37}$$

If $f(\boldsymbol{x}) > 0$ for any non-zero $\boldsymbol{x}$ then $\boldsymbol{A}$ is said to be positive-definite. Similarly, if $f(\boldsymbol{x}) \geq 0$ then $\boldsymbol{A}$ is positive-semi-definite.

If we substitute $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$ and $\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{y}$ where $\boldsymbol{y}$ are the projections onto the eigenvectors, then we can write

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{y}^T \boldsymbol{\Lambda} \boldsymbol{y} \qquad (4.38) \\ &= \sum_i y_i^2 \lambda_i \end{aligned}$$

Hence, for positive-definiteness we must therefore have $\lambda_i > 0$ for all i (ie. positive eigenvalues).

4.2 Principal Component Analysis

Given a set of data vectors $\{\boldsymbol{x}_n\}$ we can construct a covariance matrix

$$\boldsymbol{C} = \frac{1}{N} \sum_n (\boldsymbol{x}_n - \bar{\boldsymbol{x}})(\boldsymbol{x}_n - \bar{\boldsymbol{x}})^T \tag{4.39}$$

or, if we construct a matrix $\boldsymbol{X}$ with rows equal to $\boldsymbol{x}_n - \bar{\boldsymbol{x}}$ then

$$\boldsymbol{C} = \frac{1}{N}\boldsymbol{X}^T\boldsymbol{X} \tag{4.40}$$

Because covariance matrices are real and symmetric we can apply the spectral theorem

$$\boldsymbol{C} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T \tag{4.41}$$

If the eigenvectors (columns of $\boldsymbol{Q}$) are normalised to unit length, they constitute an orthonormal basis. If the eigenvalues are then ordered in magnitude such that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d$ then the decomposition is known as Principal Component Analysis (PCA). The projection of a data point $\boldsymbol{x}_n$ onto the principal components is

$$\boldsymbol{y}_n = \boldsymbol{Q}^T\boldsymbol{x}_n \tag{4.42}$$

The mean projection is

$$\bar{\boldsymbol{y}} = \boldsymbol{Q}^T\bar{\boldsymbol{x}} \tag{4.43}$$

The covariance of the projections is given by the matrix

$$\boldsymbol{C}_y = \frac{1}{N}\sum_n (\boldsymbol{y}_n - \bar{\boldsymbol{y}})(\boldsymbol{y}_n - \bar{\boldsymbol{y}})^T \tag{4.44}$$

Substituting in the previous two expressions gives

$$\begin{aligned} \boldsymbol{C}_y &= \frac{1}{N}\sum_n \boldsymbol{Q}^T(\boldsymbol{x}_n - \bar{\boldsymbol{x}})(\boldsymbol{x}_n - \bar{\boldsymbol{x}})^T\boldsymbol{Q} \\ &= \boldsymbol{Q}^T\boldsymbol{C}\boldsymbol{Q} \\ &= \boldsymbol{\Lambda} \end{aligned} \tag{4.45}$$

where $\boldsymbol{\Lambda}$ is the diagonal eigenvalue matrix with entries λ_k ($\sigma_k^2 = \lambda_k$). This shows that the variance of the kth projection is given by the kth eigenvalue. Moreover, it says that the projections are uncorrelated. PCA may therefore be viewed as a linear transform

$$\boldsymbol{y} = \boldsymbol{Q}^T \boldsymbol{x} \tag{4.46}$$

which produces uncorrelated data.

4.2.1 The Multivariate Gaussian Density

In d dimensions the general multivariate normal probability density can be written

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{C}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\bar{\boldsymbol{x}})^T C^{-1}(\boldsymbol{x}-\bar{\boldsymbol{x}})\right) \tag{4.47}$$

where the mean $\bar{\boldsymbol{x}}$ is a d-dimensional vector, $\boldsymbol{C}$ is a $d \times d$ covariance matrix, and $|\boldsymbol{C}|$ denotes the determinant of $\boldsymbol{C}$. Because the determinant of a matrix is the product of its eigenvalues then for covariance matrices, where the eigenvalues correspond to variances, the determinant is a single number which represents the total volume of variance. The quantity

$$M(\boldsymbol{x}) = (\boldsymbol{x}-\bar{\boldsymbol{x}})^T C^{-1}(\boldsymbol{x}-\bar{\boldsymbol{x}}) \tag{4.48}$$

which appears in the exponent is called the *Mahalanobis distance* from $\boldsymbol{x}$ to $\bar{\boldsymbol{x}}$.

4.2.2 Singular Value Decomposition

The eigenvalue-eigenvector factorisation (see equation 4.31)

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T \tag{4.49}$$

applies to real symmetric matrices only. There is an equivalent factorisation for rectangular matrices, having N rows and d

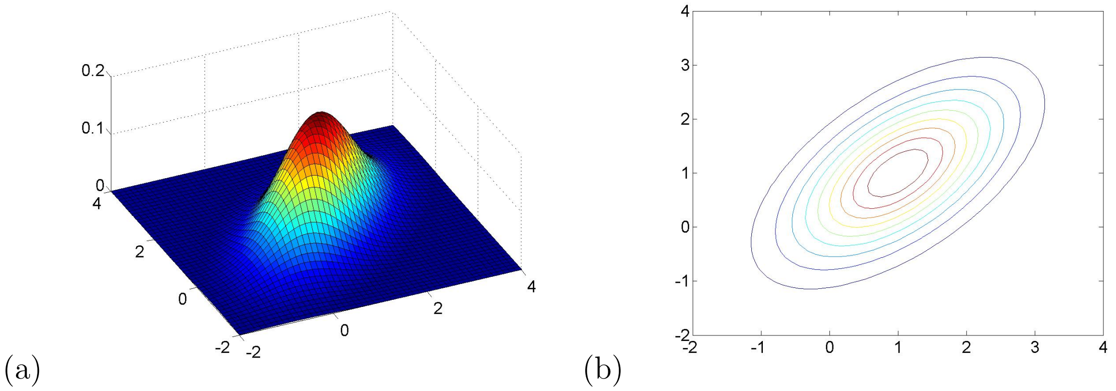


Figure 4.1: *(a) 3D-plot and (b) contour plot of Multivariate Gaussian PDF with* $\boldsymbol{\mu} = [1,1]^T$ *and* $\boldsymbol{C}_{11} = \boldsymbol{C}_{22} = 1$ *and* $\boldsymbol{C}_{12} = \boldsymbol{C}_{21} = 0.6$ *ie a positive correlation of* $r = 0.6$.

columns, called Singular Value Decomposition (SVD)

$$\boldsymbol{A} = \boldsymbol{Q}_1 \boldsymbol{D} \boldsymbol{Q}_2^T \tag{4.50}$$

where $\boldsymbol{Q_1}$ is an orthonormal N-by-N matrix, $\boldsymbol{Q_2}$ is an orthonormal d-by-d matrix, $\boldsymbol{D}$ is a diagonal matrix of dimension N-by-d and the kth diagonal entry in $\boldsymbol{D}$ is known as the kth singular value, σ_k.

If we substitute the SVD of $\boldsymbol{A}$ into $\boldsymbol{A}^T\boldsymbol{A}$, after some rearranging, we get

$$\boldsymbol{A}^T\boldsymbol{A} \;=\; \boldsymbol{Q}_2 \boldsymbol{D}^T \boldsymbol{D} \boldsymbol{Q}_2^T \tag{4.51}$$

which is of the form $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$ where $\boldsymbol{Q} = \boldsymbol{Q}_2$ and $\boldsymbol{\Lambda} = \boldsymbol{D}^T\boldsymbol{D}$. This shows that the columns of $\boldsymbol{Q}_2$ contain the eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$ and that $\boldsymbol{D}$ contains the square roots of the corresponding eigenvalues. Similarly, by substituting the SVD of $\boldsymbol{A}$ into $\boldsymbol{A}\boldsymbol{A}^T$ we can show that the columns of $\boldsymbol{Q}_1$ are the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^T$.

Relation to PCA

Given a data matrix $\boldsymbol{X}$ constructed as before (see PCA section), except that the matrix is scaled by a normalisation factor $\sqrt{1/N}$, then $\boldsymbol{X}^T\boldsymbol{X}$ is equivalent to the covariance matrix $\boldsymbol{C}$. If we therefore decompose $\boldsymbol{X}$ using SVD, the principal components will apear in $\boldsymbol{Q}_2$ and the square roots of the corresponding eigenvalues will appear in $\boldsymbol{D}$.

Therefore we can implement PCA in one of two ways (i) compute the covariance matrix and perform an eigendecomposition or (ii) use SVD directly on the (normalised) data matrix.

See eg. `alan_svd.m`.

4.2.3 PET verbal fluency data

Subject scanned under two alternating conditions (i) word generation and (ii) word shadowing. Six repetitions of each. GLM analysis to select voxels showing significant variation over the 12 scans. Zero mean voxel activities over scans.

Create matrix M of dimension $N_{scans} \times N_{voxels}$. Application of SVD

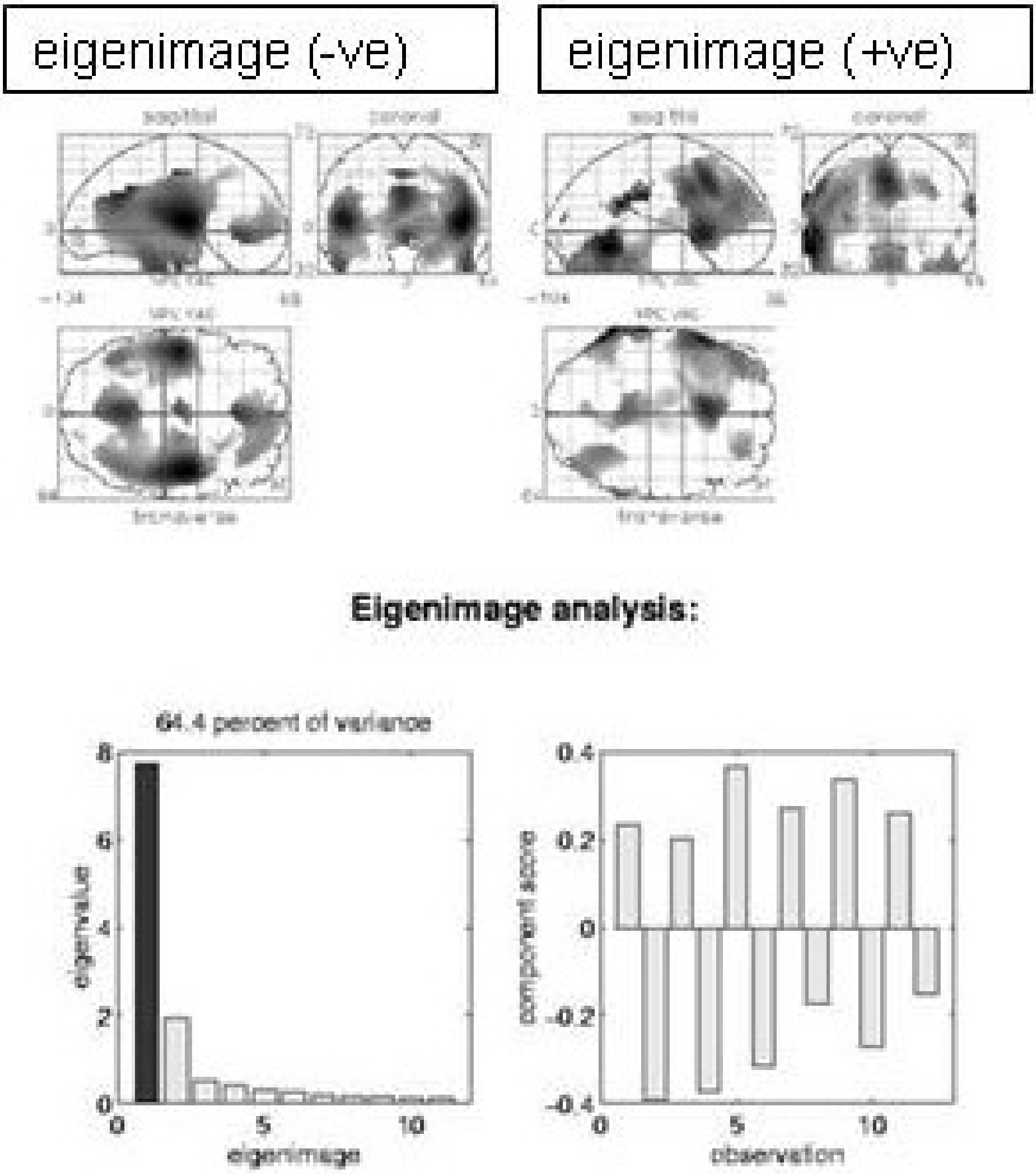
$$USV^T = M \tag{4.52}$$

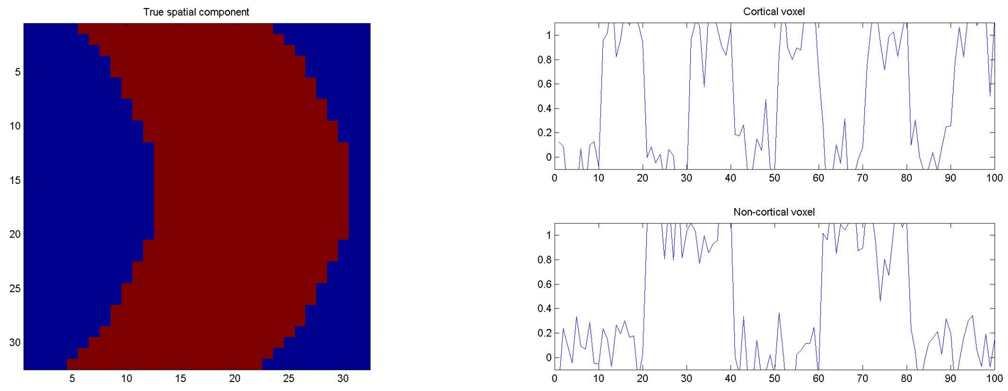
places temporal components (eigenvariates) in columns of U and spatial components (eigenimages) in columns of V. Diagonal elements in S show that first mode accounts for 64% variance, second 16%.

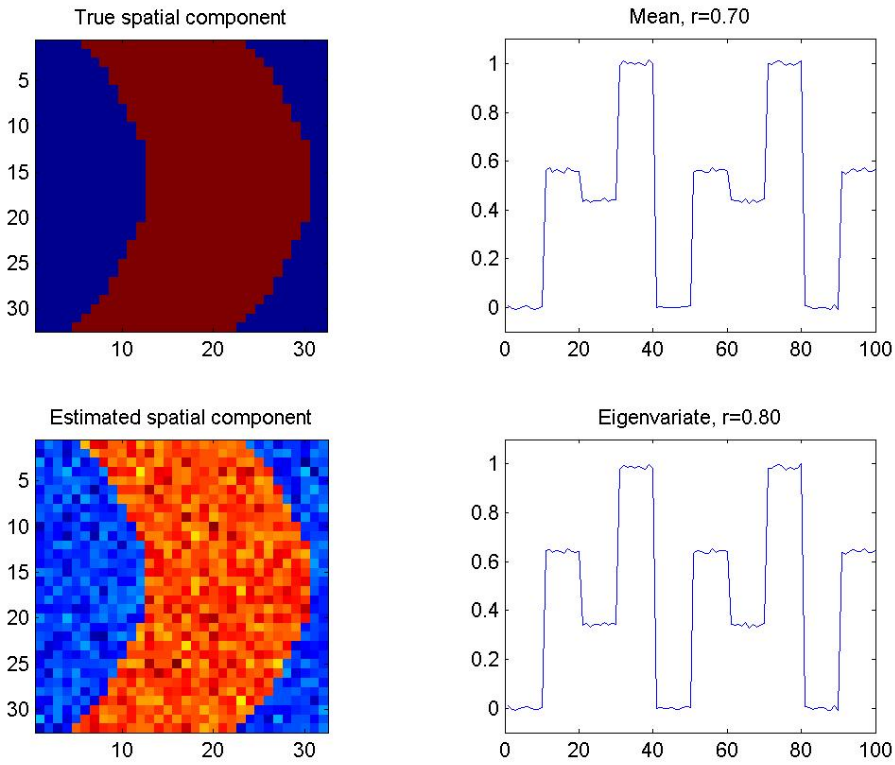
First eigenimage has positive loadings in anterior cingulate, left DLPFC, Broca's area, thalamic nuclei and cerebellum (regions showing higher activity in generation than shadowing). Negative loadings bitemporally and in posterior cingulate.

4.2.4 Summarising regional activity

See `region_svd.m`


eigenimage (-ve)
eigenimage (+ve)
sagittal
coronal
transverse
Eigenimage analysis:
64.4 percent of variance
eigenvalue
eigenimage
0
2
4
6
8
5
10
component score
observation
0.4
0.2
-0.2
-0.4


True spatial component
5
10
15
20
25
30
Cortical voxel
Non-cortical voxel
0
0.2
0.4
0.6
0.8
1
10
20
30
40
50
60
70
80
90
100



4.3 Structural Equation Modelling

Structural Equation Models (SEMs) comprise a set of regions and a set of directed connections. Importantly, a causal semantics is ascribed to these connections where an arrow from A to B means that A causes B. Causal relationships are thus not inferred from the data but are assumed a-priori [35].

We consider networks comprising N regions in which the activity at time t is given by the $N \times 1$ vector y_t. If there are T time points and Y is an $N \times T$ data matrix comprising $t = 1..T$ such vectors then the likelihood of the data is given by

$$p(Y|\theta) = \prod_{t=1}^{T} p(y_t|\theta) \tag{4.53}$$

where θ are the parameters of an SEM.

The second SEM equation specifies the generative model at time t

$$p(y_t|\theta) = \mathsf{N}\left(y_t; 0, \Sigma(\theta)\right) \tag{4.54}$$

which denotes that the activities are zero mean Gaussian variates with a covariance, $\Sigma(\theta)$, that is a function of the connectivity matrix θ. The form of this function is specified implicitly by the regression equation that describes how activity in one area is related to activity in other areas via a set of path coefficients, M, as

$$y_t = My_t + e_t \tag{4.55}$$

where e_t are zero mean Gaussian innovations or errors of covariance R. Typically R will be a diagonal matrix and we write the error variance in region i as σ_i^2. Regions are connected together via the $N \times N$ path coefficient matrix M where the M_{ij} denotes a connection from region j to region i. The parameters of an SEM, θ, are the unknown elements of M and R. Re-write as

$$y_t = (I_N - M)^{-1} e_t \tag{4.56}$$

This form is particularly useful as it shows us how to generate data from the model. Firstly, we generate the Gaussian variates e_t and then pre-multiply by $(I_N - M)^{-1}$. This is repeated for each t. This form also allows us to express the covariance of y_t as a function of θ

$$\Sigma(\theta) = (I_N - M)^{-1} R (I_N - M)^{-T} \tag{4.57}$$

4.3.1 Estimation

Given a set of parameters θ we can compute the likelihood of a data set from equations 4.53, 4.54 and 4.57. Given a data set one can therefore find the connectivity matrix that maximises the (log) likelihood using standard optimisation methods [42].

$$L(\theta) = -\frac{T}{2}\log|\Sigma(\theta)| - \frac{NT}{2}\log 2\pi - \frac{1}{2}\sum_{t=1}^{T} y_t^T \Sigma(\theta)^{-1} y_t$$

If we define the sample covariance as

$$S = \frac{1}{T}\sum_{t=1}^{T} y_t y_t^T \tag{4.58}$$

then, by noting that the last term is a scalar and that the trace of a scalar is that same scalar value, and using the circularity property of the trace operator (that is, $\mathsf{Tr}(AB) = \mathsf{Tr}(BA)$), we can write

$$L(\theta) = -\frac{T}{2}\log|\Sigma(\theta)| - \frac{NT}{2}\log 2\pi - \frac{T}{2}\mathsf{Tr}(S\Sigma(\theta)^{-1})$$

If we use unbiased estimates of the sample covariance matrix then we replace T's in the above equation by $T-1$'s. If we now also drop those terms that are not dependent on the model parameters we get

$$L(\theta) \;=\; -\frac{T-1}{2}\left(\log|\Sigma(\theta)| + \mathsf{Tr}(S\Sigma(\theta)^{-1})\right) \tag{4.59}$$

Maximum likelihood estimates can therefore be obtained by maximising the above function.

4.3.2 Inference

Statistical inference is based on the likelihood ratio for comparing models i and j is

$$R_{ij} = \frac{p(Y|\theta, m=i)}{p(Y|\theta, m=j)} \tag{4.60}$$

If $L(\theta_i)$ and $L(\theta_j)$ are the corresponding log-likelihoods then the log of the likelihood ratio is

$$\log R_{ij} = L(\theta_i) - L(\theta_j) \tag{4.61}$$

Under the null hypothesis that the models are identical, and for large T, $-2 \log R_{ij}$ is distributed as a chi-squared variable having degrees of freedom equal to the difference in number of parameters between the models (see p.265 in [7]). This only applies to nested models.

A special case of the above test arises when one wishes to evaluate the goodnees of fit of a single model. We will denote this as 'model 1'. This can be achieved by comparing the likelihood of model 1 to the likelihood of the least restrictive (most complex) model one could possibly adopt ('model 0') with covariance equal to the sample covariance ie. $\Sigma(\theta) = S$. The has likelihood

$$\begin{aligned} L_0 &= -\frac{T-1}{2}\left(\log|S| + \mathsf{Tr}(SS^{-1})\right) & (4.62)\\ &= -\frac{T-1}{2}\left(\log|S| + N\right) & (4.63) \end{aligned}$$

The corresponding (log) likelihood ratio is

$$\log R_{10} = -\frac{T-1}{2}\left(\log|\Sigma(\theta)| + \mathsf{Tr}(S\Sigma(\theta)^{-1}) - \log|S| - N\right) \tag{4.64}$$

which in turn has a corresponding chi-squared value

$$\chi^2 = (T-1)F(\theta) \tag{4.65}$$

where

$$F(\theta) = \log|\Sigma(\theta)| + \mathsf{Tr}(S\Sigma(\theta)^{-1}) - \log|S| - N \tag{4.66}$$

The corresponding degrees of freedom are equal to the degrees of freedom in model 0, k, minus the degrees of freedom in model 1, q. For an N-dimensional covariance matrix there are k=N(N+1)/2 degrees of freedom. For model 1, q equals the total number of connectivity and variance parameters to be estimated. The associate χ^2 test provides a way of assessing if an SEM fits the data sufficiently.

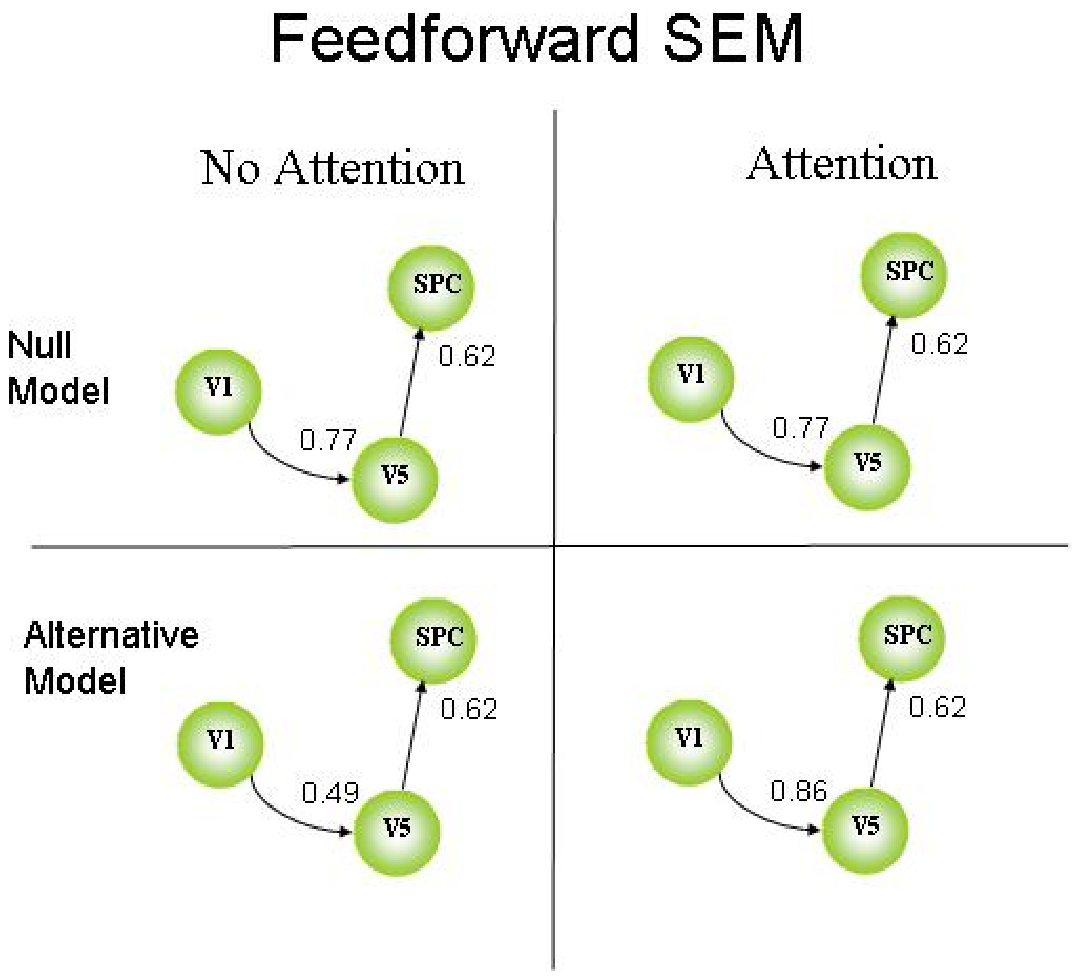
For more general model comparisons the χ^2 statistic associated with the LR test can be written as

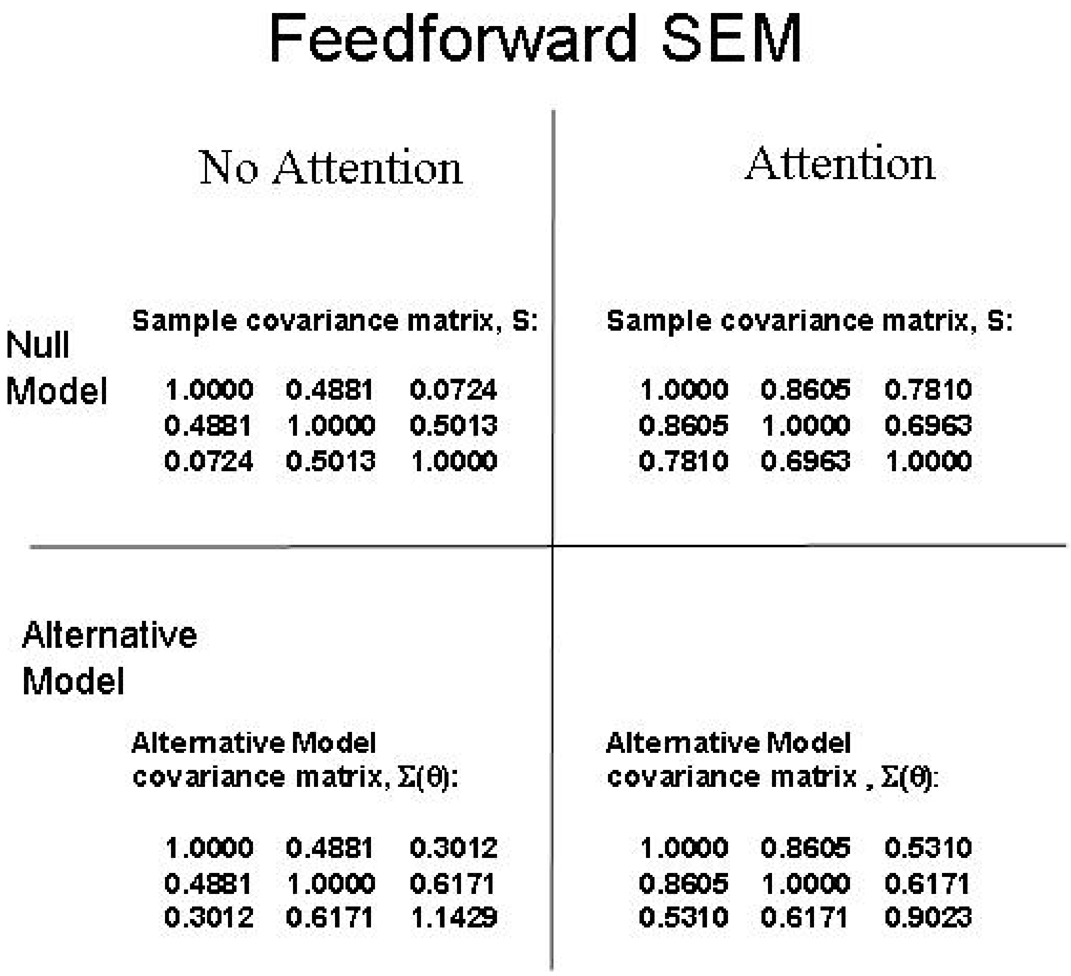
$$\chi^2 = (T-1)(F(\theta_1) - F(\theta_2)) \tag{4.67}$$

4.3.3 Attention to visual motion fMRI data

We first use a feedforward architecture. The null model has all parameters fixed between conditions giving $k = 8$ (two path coefficients and six error variance parameters). The alternative model allows V1-V5 to change giving $q = 9$. The alternative model fits better ($p = 0.003$).

But in comparison to the sample covariance, where $k = N(N+1)/2 = 6$ degrees of freedom per data set and two data sets (two conditions) ie. $k = 12$, the alternative model is sig. different ($p < 1e-5$). It is therefore not a good model of the data.


Feedforward SEM
No Attention
Attention
Null Model
SPC
0.62
V1
0.77
V5
SPC
0.62
V1
0.77
V5
Alternative Model
SPC
0.62
V1
0.49
V5
SPC
0.62
V1
0.86
V5



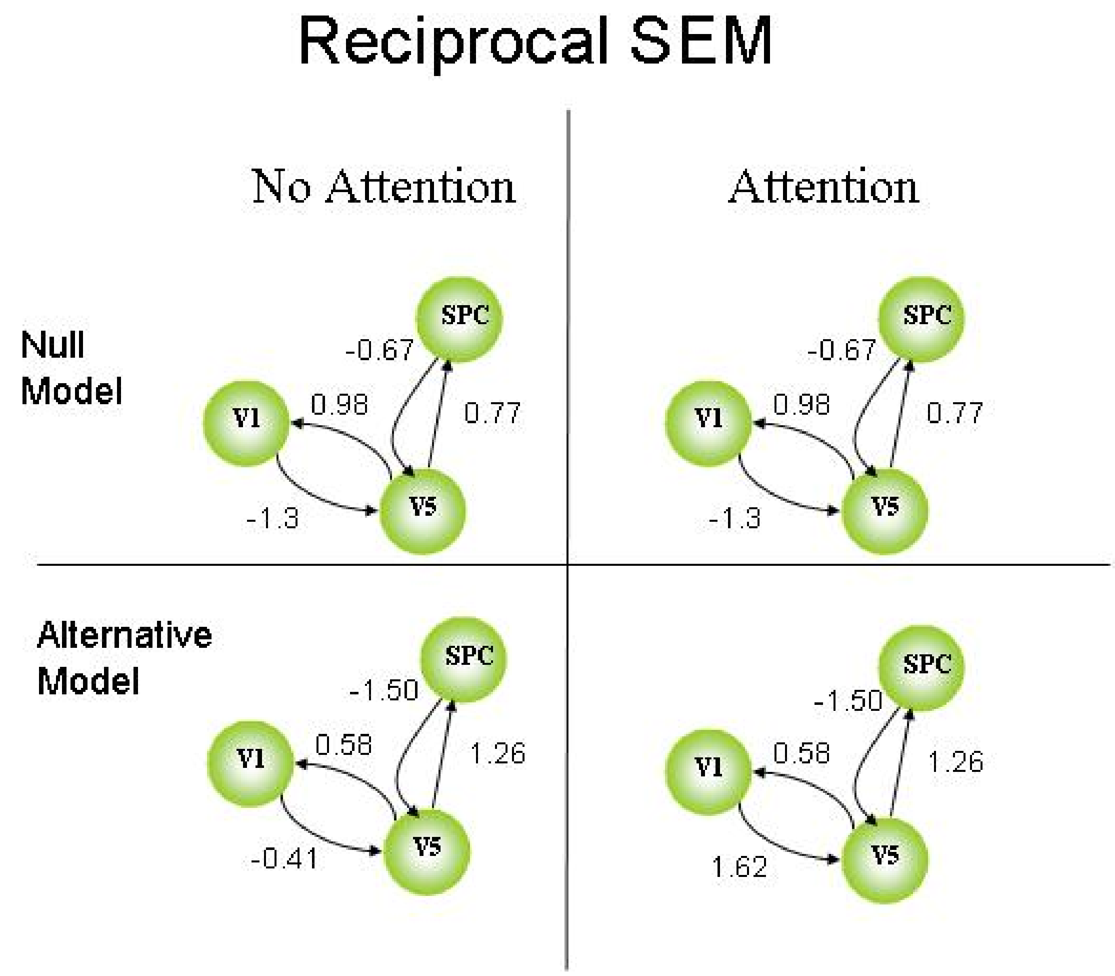
Feedforward SEM

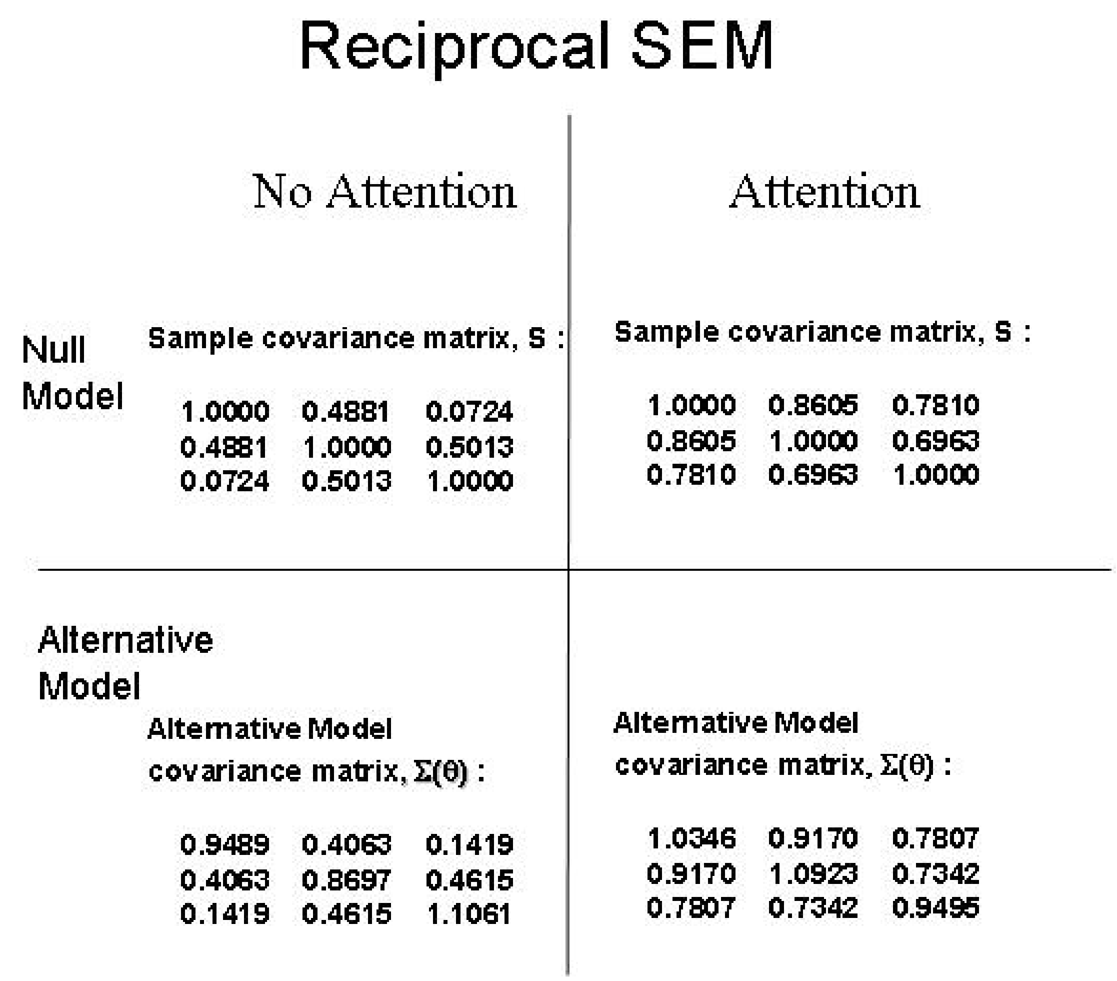
	No Attention	Attention
Null Model	Sample covariance matrix, S: 1.0000 0.4881 0.0724 0.4881 1.0000 0.5013 0.0724 0.5013 1.0000	Sample covariance matrix, S: 1.0000 0.8605 0.7810 0.8605 1.0000 0.6963 0.7810 0.6963 1.0000
Alternative Model	Alternative Model covariance matrix, $\Sigma(\theta)$: 1.0000 0.4881 0.3012 0.4881 1.0000 0.6171 0.3012 0.6171 1.1429	Alternative Model covariance matrix , $\Sigma(\theta)$: 1.0000 0.8605 0.5310 0.8605 1.0000 0.6171 0.5310 0.6171 0.9023

We now use a recpirocal architecture. The null model has all parameters fixed between conditions giving $k = 10$. The alternative model allows V1-V5 to change giving $q = 11$. The alternative model fits better ($p = 9e - 6$).

In comparison to the sample covariance, where $k = 12$, the alternative model is not sig. different ($p = 0.05$) (well, its borderline !). It is therefore an acceptable model of the data.

As compared to the feedforward model, the correlations between V1 and SPC are modelled more accurately (at a minor cost of not modelling V1-V5 and V5-SPC correlations quite so accurately).


Reciprocal SEM
No Attention
Attention
Null Model
Alternative Model
SPC
V1
V5
-0.67
0.98
0.77
-1.3
SPC
V1
V5
-0.67
0.98
0.77
-1.3
SPC
V1
V5
-1.50
0.58
1.26
-0.41
SPC
V1
V5
-1.50
0.58
1.26
1.62



Reciprocal SEM

No Attention

Attention

Null Model

Sample covariance matrix, S :

$$\begin{matrix} 1.0000 & 0.4881 & 0.0724 \\ 0.4881 & 1.0000 & 0.5013 \\ 0.0724 & 0.5013 & 1.0000 \end{matrix}$$

Sample covariance matrix, S :

$$\begin{matrix} 1.0000 & 0.8605 & 0.7810 \\ 0.8605 & 1.0000 & 0.6963 \\ 0.7810 & 0.6963 & 1.0000 \end{matrix}$$

Alternative Model

Alternative Model covariance matrix, $\Sigma(\theta)$:

$$\begin{matrix} 0.9489 & 0.4063 & 0.1419 \\ 0.4063 & 0.8697 & 0.4615 \\ 0.1419 & 0.4615 & 1.1061 \end{matrix}$$

Alternative Model covariance matrix, $\Sigma(\theta)$:

$$\begin{matrix} 1.0346 & 0.9170 & 0.7807 \\ 0.9170 & 1.0923 & 0.7342 \\ 0.7807 & 0.7342 & 0.9495 \end{matrix}$$

Chapter 5

Variance Components

5.1 GLMs with multiple covariance components

Given the usual GLM

$$y = X\beta + e \tag{5.1}$$

where β are the true but unknown parameters, and $Cov(e) = V(\lambda)$ is the error covariance parameterised by unknown parameters λ ie. 'hyperparameters'. These more general models are useful for eg. (a) fMRI analysis allowing for correlated errors and (b) analysis of data from a group of subjects.

We now address two questions

- If we know V how do we estimate β ?
- How do we estimate V ?

The two answers are (i) WLS and (ii) ReML.

5.2 Weighted Least Squares

If we know V then we can estimate β by maximising the likelihood

$$L = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|V| - \frac{1}{2}(y - X\beta)^T V^{-1}(y - X\beta)$$

We can derive the normal equations in the usual way, by setting the appropriate derivatives to zero.

$$\frac{dL}{d\beta} = X^T V^{-1} y - X^T V^{-1} X\beta$$

This leads to the solution

$$\hat{\beta}_{ML} = (X^T V^{-1} X)^{-1} X^T V^{-1} y \tag{5.2}$$

This is often referred to as Weighted Least Squares (WLS), $\hat{\beta}_{ML} = \hat{\beta}_{WLS}$.

For isotropic error covariance $V = \lambda I$, $\hat{\beta}_{ML} = \hat{\beta}_{OLS}$, the Ordinary Least Squares (OLS) solution

$$\hat{\beta}_{OLS} = (X^T X)^{-1} X^T y \tag{5.3}$$

5.3 Restricted Maximum Likelihood (ReML)

If we don't know V we can estimate it using ReML. In Maximum Likelihood (ML), λ are estimated by maximising the likelihood

$$p(y|\beta, \lambda) \tag{5.4}$$

The idea behind ReML is to find λ that maximise the restricted likelihood

$$p(y|\lambda) \tag{5.5}$$

This does not depend on the parameters β. We can write it as

$$p(y|\lambda) = \int p(y|\beta,\lambda) d\beta \tag{5.6}$$

We will now use a quadratic identity, derived in the following section, to solve the integral.

5.3.1 Quadratic Identity

Let $y_t = X\beta$ be the true, but unknown, mean data values. And $\hat{y} = X\hat{\beta}$ be the predictions of the fitted model

Then

$$\begin{aligned}
(y - X\beta)^T V^{-1}(y - X\beta) &= (y - y_t)^T V^{-1}(y - y_t) \qquad (5.7)\\
&= (y - y_t + \hat{y} - \hat{y})^T V^{-1}(y - y_t + \hat{y} - \hat{y})\\
&= (y - \hat{y})^T V^{-1}(y - \hat{y})\\
&+ (-y_t + \hat{y})^T V^{-1}(-y_t + \hat{y})\\
&= (y - \hat{y})^T V^{-1}(y - \hat{y})\\
&+ (y_t - \hat{y})^T V^{-1}(y_t - \hat{y})\\
&= (y - X\hat{\beta})^T V^{-1}(y - X\hat{\beta})\\
&+ (X\beta - X\hat{\beta})^T V^{-1}(X\beta - X\hat{\beta})\\
&= (y - X\hat{\beta})^T V^{-1}(y - X\hat{\beta})\\
&+ (\beta - \hat{\beta})^T (X^T V^{-1} X)(\beta - \hat{\beta})
\end{aligned}$$

So, we have shown that

$$\begin{aligned}
(y - X\beta)^T V^{-1}(y - X\beta) &= (y - X\hat{\beta})^T V^{-1}(y - X\hat{\beta}) \qquad (5.8)\\
&+ (\beta - \hat{\beta})^T (X^T V^{-1} X)(\beta - \hat{\beta})
\end{aligned}$$

The second term depends on the parameters β but the first does not.

5.3.2 ReML integral

For an $N \times p$ full-rank design matrix, the likelihood is

$$p(y|\beta, \lambda) = (2\pi)^{-N/2}|V|^{-1/2} \exp\left(\frac{1}{2}(y - X\beta)^T V^{-1}(y - X\beta)\right) \tag{5.9}$$

Using our quadratic identity we can write

$$\begin{aligned} p(y|\beta, \lambda) &= (2\pi)^{-N/2}|V|^{-1/2} \qquad (5.10) \\ &\times \exp\left(\frac{1}{2}(y - X\hat{\beta})^T V^{-1}(y - X\hat{\beta})\right) \\ &\times \exp\left(\frac{1}{2}(\beta - \hat{\beta})^T (X^T V^{-1} X)(\beta - \hat{\beta})\right) \end{aligned}$$

The restricted likelihood is then given by

$$\begin{aligned} p(y|\lambda) &= \int p(y|\beta, \lambda) d\beta \qquad (5.11) \\ &= (2\pi)^{-N/2}|V|^{-1/2} \\ &\times \exp\left(\frac{1}{2}(y - X\hat{\beta})^T V^{-1}(y - X\hat{\beta})\right) \\ &\times \int \exp\left(\frac{1}{2}(\beta - \hat{\beta})^T (X^T V^{-1} X)(\beta - \hat{\beta})\right) d\beta \\ &= (2\pi)^{-N/2}|V|^{-1/2} \\ &\times \exp\left(\frac{1}{2}(y - X\hat{\beta})^T V^{-1}(y - X\hat{\beta})\right) \\ &\times (2\pi)^{p/2}|X^T V^{-1} X|^{-1/2} \end{aligned}$$

where we've noted that the integral is just the normalising constant for a multivariate Gaussian. Taking logs gives the ReML objective function

$$\begin{aligned} L_R(\lambda) &= \log p(y|\lambda) \qquad (5.12) \\ &= -\frac{N-p}{2}\log 2\pi - \frac{1}{2}\log|V| - \frac{1}{2}\log|X^T V^{-1} X| \\ &- \frac{1}{2}(y - X\hat{\beta})^T V^{-1}(y - X\hat{\beta}) \end{aligned}$$

This does not depend on the parameters β. It does depend on $\hat{\beta}$, but we can substitute in expressions for this from earlier, so that L_R is just a function of X, y and λ.

5.3.3 Single variance component

For a single variance component

$$V(\lambda) = \lambda Q \qquad (5.13)$$

an analytic expression for λ can be found

Maximum Likelihood

The likelihood function is

$$L(\lambda) = -\frac{1}{2}\log|V| - \frac{1}{2} r^T V^{-1} r + ... \qquad (5.14)$$

where the residuals are $r = y - X\beta$. The gradient with respect to λ is

$$g = -\frac{1}{2}Tr(V^{-1}Q) + \frac{1}{2} r^T V^{-1} Q V^{-1} r \qquad (5.15)$$

For a single variance component we get the estimate

$$\lambda = \frac{r^T Q^{-1} r}{N} \qquad (5.16)$$

For isotropic errors $Q = I$ we have

$$\lambda = \frac{r^T r}{N} \qquad (5.17)$$

This is biased.

ReML

We write $g_R = \frac{dL_R(\lambda)}{d\lambda}$ as the gradient of the ReML function. This can be shown to be

$$g_R = -\frac{1}{2}Tr(PQ) + \frac{1}{2}y^T P^T Q P y \tag{5.18}$$

where the projection matrix $P = V^{-1}R_{WLS}$ and

$$R_{WLS} = I - X(X^T V^{-1} X)^{-1} X^T V^{-1} \tag{5.19}$$

It's the same as the ML gradient but with P instead of V^{-1}. We are working in a subspace of V^{-1} that is orthogonal to the WLS estimates.

Setting $g = 0$ gives

$$\lambda = \frac{r^T Q^{-1} r}{Tr(R)} \tag{5.20}$$

where

$$R = I - X(X^T Q^{-1} X)^{-1} X^T Q^{-1} \tag{5.21}$$

is the residual forming matirx and $r = Ry$ are the residuals.

If $Q = I$, ie. isotropic error, $R = I - X(X^T X)^{-1} X^T$ and

$$\lambda = \frac{r^T r}{N - k} \tag{5.22}$$

which is an *unbiased* estimate of the error variance (unlike the ML estimate).

5.3.4 Linear constraints

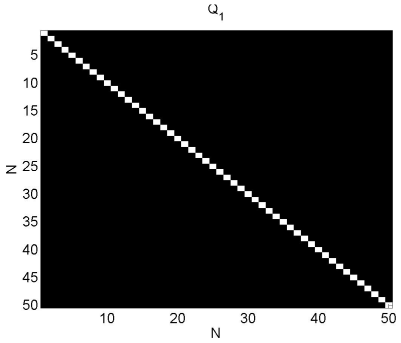
If the error covariance takes the following form

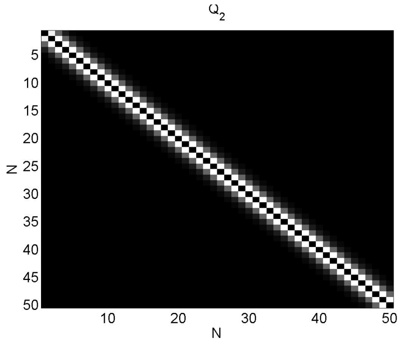
$$V(\lambda) = \sum_k \lambda_k Q_k \tag{5.23}$$

where Q_k is a known matrix and λ_k is the kth unknown hyperparameter, then λ can be found by maximising the ReML objective function. This could be implemented by any optimisation method eg. one could follow the gradient g where

$$g_k = -\frac{1}{2} Tr(PQ_k) + \frac{1}{2} y^T P^T Q_k P y \tag{5.24}$$

A better algorithm, based on Fisher scoring, has been derived by Harville [20]. See also Friston et al. [17][15] for applications to brain imaging. This algorithm is implemented in `spm_reml.m`.





5.3.5 fMRI time series

Correlated fMRI time series can be dealt with by having eg. two covariance components: one for the additive noise, Q_1, and a second for the temporal autocorrelation, Q_2. Q_2 is based on a first-order autoregressive model $e_t = ae_{t-1} + z_t$ with a fixed coefficient $a = 0.2$. Variations in a can be accomodated by specifying a third basis function Q_3 which is a Taylor expansion of Q_2 [15].

5.4 Hierarchical General Linear Models

Given the hierarchical model (eg. 3 levels)

$$\begin{aligned} y &= X_1\beta_1 + e_1 \\ \beta_1 &= X_2\beta_2 + e_2 \\ \beta_2 &= X_3\beta_3 + e_3 \end{aligned} \tag{5.25}$$

In the analysis of group data this can enable us, for example, to relate population effects, β_3, to subject effects β_2 to session effects β_1. The vector y contains the data from all trials in all sessions from all subjects.

The error covariances at each level C_3, C_2 and C_1 describe between-subject variance, between-session variance and between-trial variance.

We can substitute from β_2 into the second equation, then β_1 into the first to give a collapsed model

$$y = X\beta_3 + e \tag{5.26}$$

where

$$\begin{aligned} X &= X_1X_2X_3 \\ e &= e_1 + X_1e_2 + X_1X_2e_3 \end{aligned} \tag{5.27}$$

The error covariance, $Cov(e) = C$ is

$$C = C_1 + X_1C_2X_2^T + X_1X_2C_3X_2^TX_1^T \tag{5.28}$$

The hierarchical structure introduces this particular structure into the error covariance of the collapsed model.

We can then run ReML to estimate the variance components λ (parameters of C). The population effect is then estimated using WLS in the usual way.

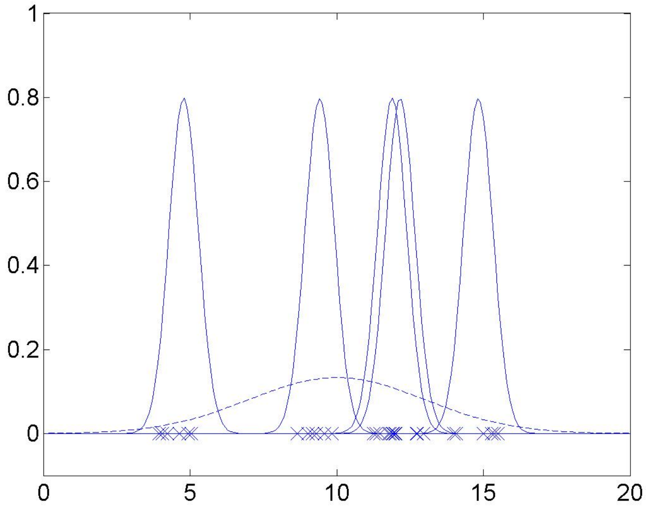


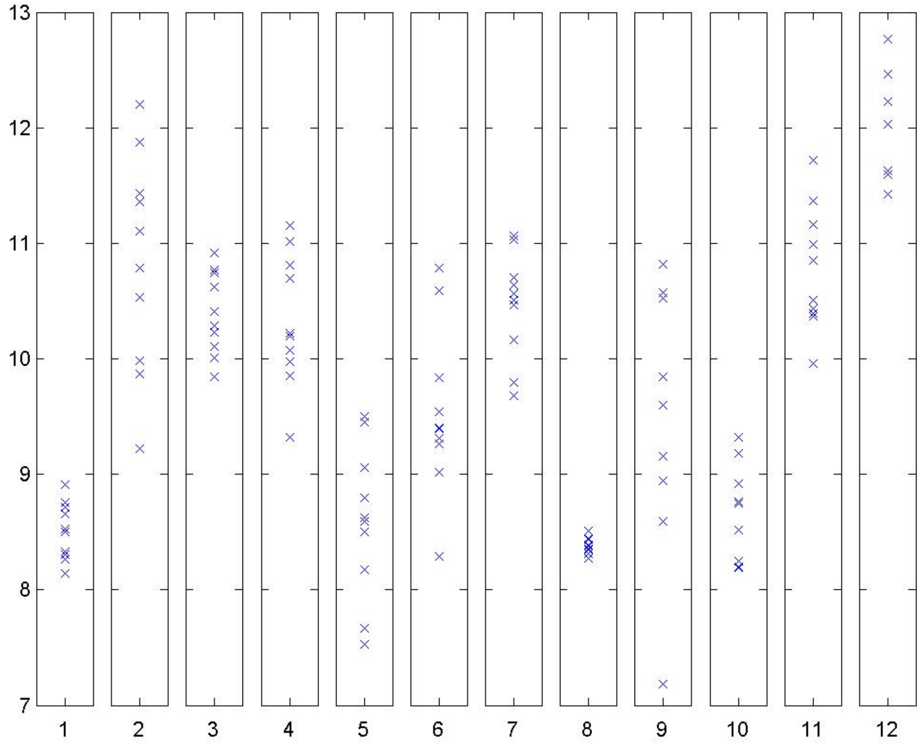
Figure 5.1: *Synthetic data illustrating the probability model underlying random effects analysis. The dotted line is the Gaussian distribution underlying the second level model with mean β_2, the population effect, and variance σ_b^2, the between-subject variance. The mean subject effects, $\beta_1(i)$, are drawn from this distribution. The solid lines are the Gaussians underlying the first level models with means $\beta_1(i)$ and variances σ_w^2. In this example the within-subject/between-trial variance is the same for all subjects. The crosses are the observed effects y_{ij} which are drawn from the solid Gaussians.*

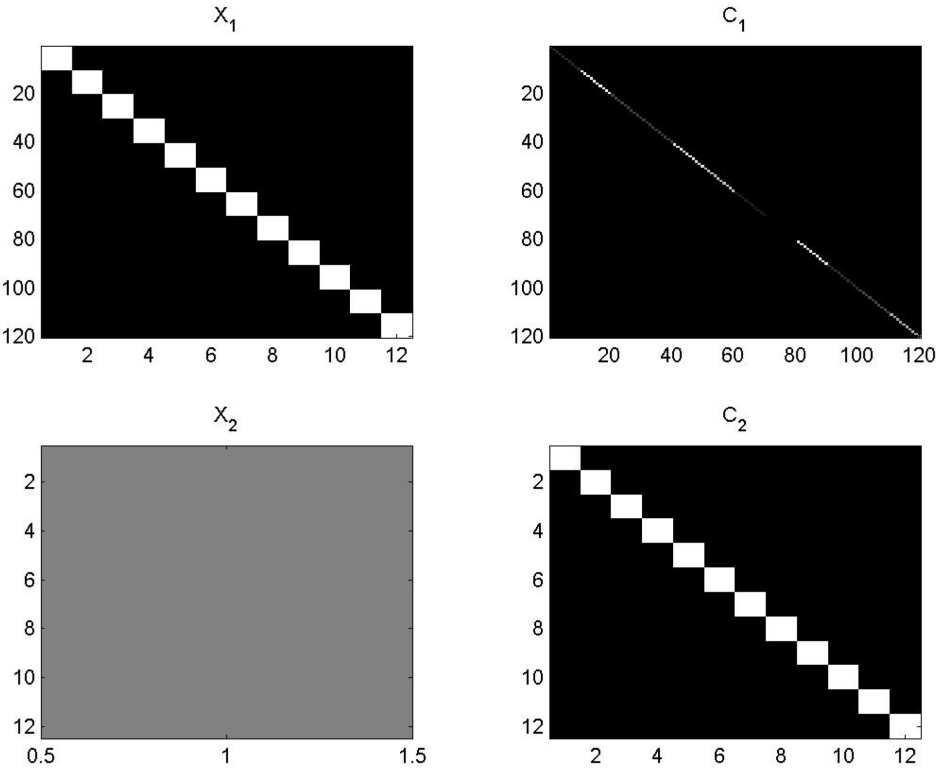
5.4.1 Example

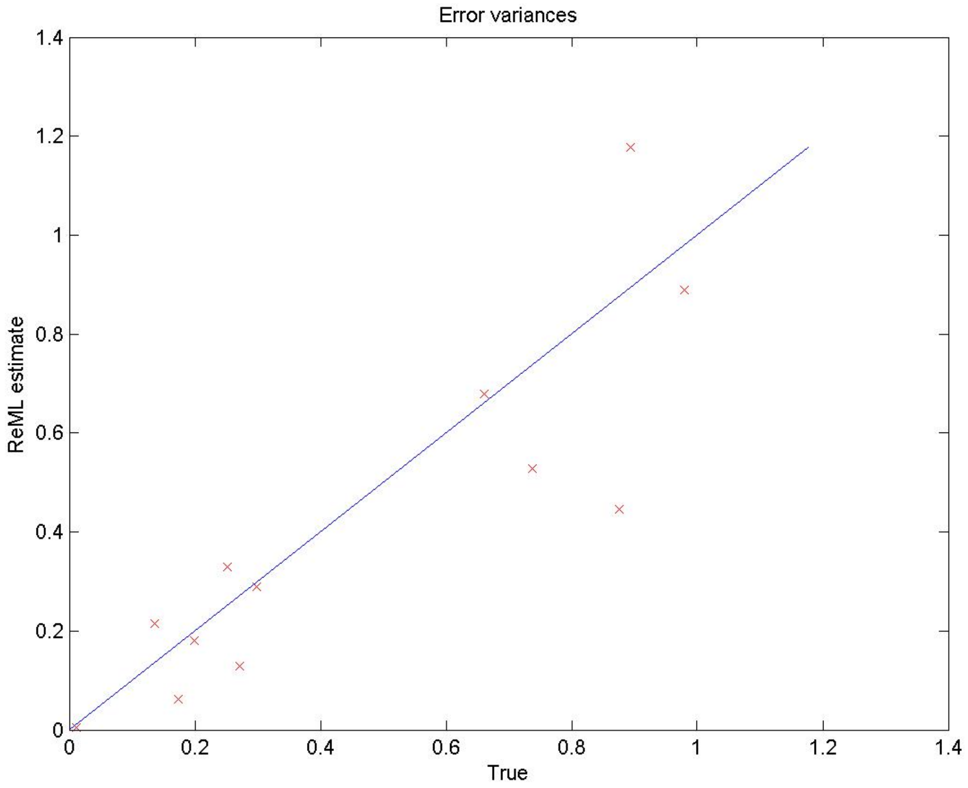
Two-level hierarchical model

$$\begin{aligned} y &= X_1\beta_1 + e_1 \\ \beta_1 &= X_2\beta_2 + e_2 \end{aligned} \tag{5.29}$$

where β_1 contains subject effects, β_2 the population effect. The errors are between-trial errors e_1 and between-subject errors e_2. In brain imaging such a model is known as a Random-Effects (RFX) analysis, as the subject effects are viewed as random variables (there is a between-subject error). The aim is to make an inference about the population effect.


13
12
11
10
9
8
7
1
2
3
4
5
6
7
8
9
10
11
12


X1
20
40
60
80
100
120
2
4
6
8
10
12
C1
20
40
60
80
100
120
20
40
60
80
100
120
X2
2
4
6
8
10
12
0.5
1
1.5
C2
2
4
6
8
10
12
2
4
6
8
10
12



5.4.2 Summary statistic approach

This involves simply taking a Summary Statistic (SS) eg. the mean, from one level and using it as data for the level above. For 'balanced designs', this gives us the correct results on average [37]. This requires the same number of trials per subject and the same between-trial error variance. The two-level hierarchical model is approximated as two separate single level models

$$\begin{aligned} y &= X_1\hat{\beta}_1 + e_1 \\ \hat{\beta}_1 &= X_2\beta_2 + e_2 \end{aligned} \tag{5.30}$$

The first level effects are estimated for each subject, saved as 'contrast images' and entered as data for a separate 2nd-level model.

5.5 fMRI data from multiple sessions

This section compares RFX analysis as implemented using SS versus ReML. The dataset comprises 1,200 images that were acquired in 10 sessions of 120 scans each. These data have been described elsewhere [16].

Each session contained a different number of events, so strictly, violates SS assumptoions. The experimental design involved 30-second epochs of single word streams and a passive listening task. The words were concrete, monosyllabic nouns presented at a number of different rates. The word rate was varied pseudo-randomly over epochs within each session. Further details of the paradigm and analysis details are given in [18]. The results of the SS and ReML analyses have been thresholded at $p < 0.05$, corrected for the entire search volume.

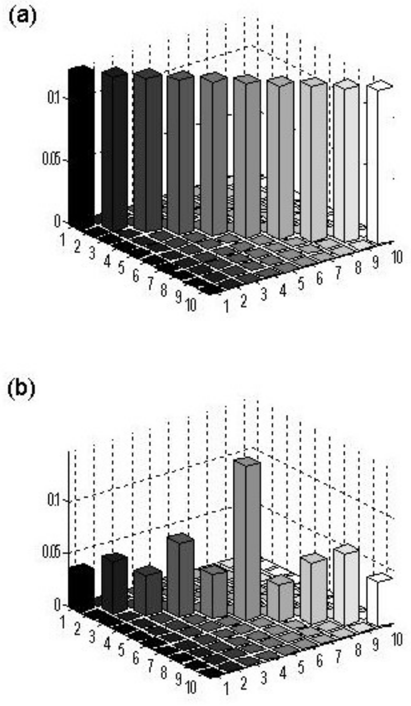


Figure 5.2: *Within-session variance as (a) assumed by SS and (b) estimated using ReML. This shows that within-session variance can vary by up to a factor of four, although this makes little difference to the final inference.*

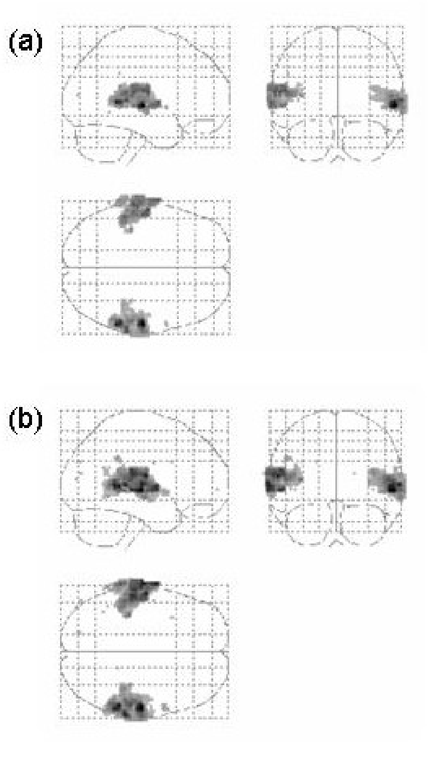


Figure 5.3: *SPMs showing the effect of words in the population using (a) SS and (b) ReML approaches.*

Chapter 6

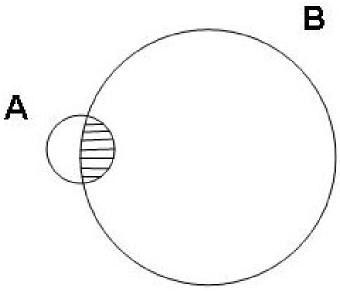
Bayesian Methods

6.1 Contents

Bayes rule for

- Gaussians
- General Linear Models

and Parametric Empirical Bayes (PEB). Application to M/EEG source localisation.



Given probabilities $p(A)$, $p(B)$, and the joint probability $p(A, B)$, we can write the conditional probabilities

$$\begin{aligned} p(B|A) &= \frac{p(A,B)}{p(A)} \\ p(A|B) &= \frac{p(A,B)}{p(B)} \end{aligned}$$

Eliminating $p(A, B)$ gives Bayes rule

$$p(B|A) = \frac{p(A|B)p(B)}{p(A)}$$

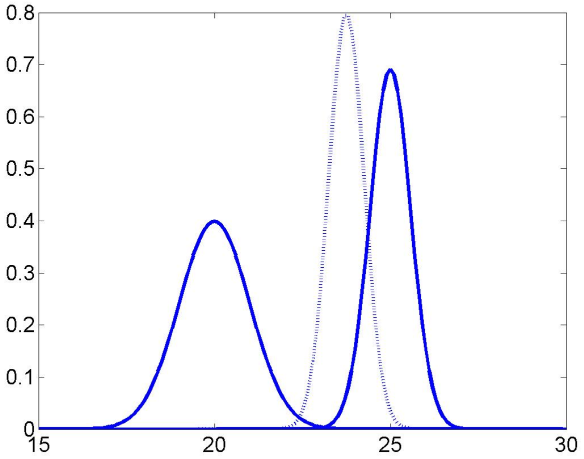


Figure 6.1: *Bayes rule for univariate Gaussians. The two solid curves show the probability densities for the prior* $m_0 = 20$*,* $p_0 = 1$ *and the likelihood* $m_D = 25$ *and* $p_D = 3$*. The dotted curve shows the posterior distribution with* $m = 23.75$ *and* $p = 4$*. The posterior is closer to the likelihood because the likelihood has higher precision.*

6.1.1 Gaussians

'Precision' is inverse variance eg. variance of 0.1 is precision of 10.

For a Gaussian prior with mean m_0 and precision p_0, and a Gaussian likelihood with mean m_D and precision p_D the posterior is Gaussian with

$$\begin{aligned} p &= p_0 + p_D \\ m &= \frac{p_0}{p} m_0 + \frac{p_D}{p} m_D \end{aligned}$$

So, (1) precisions add and (2) the posterior mean is the sum of the prior and data means, but each weighted by their relative precision.

6.1.2 Bayesian GLM

If $p(x) = \mathsf{N}(m, \Sigma)$ then

$$p(x) \propto \exp\left(-\frac{1}{2}(x-m)^T \Sigma^{-1}(x-m)\right) \tag{6.1}$$

A Bayesian GLM is defined as

$$\begin{aligned} y &= X\beta + e_1 \\ \beta &= \mu + e_2 \end{aligned} \tag{6.2}$$

where the errors are zero mean Gaussian with covariances $\mathsf{Cov}[e_1] = C_1$ and $\mathsf{Cov}[e_2] = C_2$.

$$\begin{array}{ll} p(y|\beta) & \propto \exp\left(-\frac{1}{2}(y-X\beta)^T C_1^{-1}(y-X\beta)\right) \\ p(\beta) & \propto \exp\left(-\frac{1}{2}(\beta-\mu)^T C_2^{-1}(\beta-\mu)\right) \end{array} \tag{6.3}$$

The posterior distribution is then

$$p(\beta|y) \propto p(y|\beta)p(\beta) \tag{6.4}$$

Taking logs and keeping only those terms that depend on β gives

$$\begin{aligned}\log p(\beta|y) &= -\frac{1}{2}(y - X\beta)^T C_1^{-1}(y - X\beta) \\ &- \frac{1}{2}(\beta - \mu)^T C_2^{-1}(\beta - \mu) + .. \\ &= -\frac{1}{2}\beta^T(X^T C_1^{-1} X + C_2^{-1})\beta \\ &+ \beta^T(X^T C_1^{-1} y + C_2^{-1}\mu) + ..\end{aligned} \tag{6.5}$$

Taking logs of the Gaussian density $p(x)$ in equation 6.2 and keeping only those terms that depend on x gives

$$\log p(x) = -\frac{1}{2}x^T \Sigma^{-1} x + x^T \Sigma^{-1} m + .. \tag{6.6}$$

Comparing equation 6.5 with terms in the above equation shows that

$$\begin{aligned}p(\beta|y) &= \mathsf{N}(m, \Sigma) \\ \Sigma^{-1} &= X^T C_1^{-1} X + C_2^{-1} \\ m &= \Sigma(X^T C_1^{-1} y + C_2^{-1}\mu)\end{aligned} \tag{6.7}$$

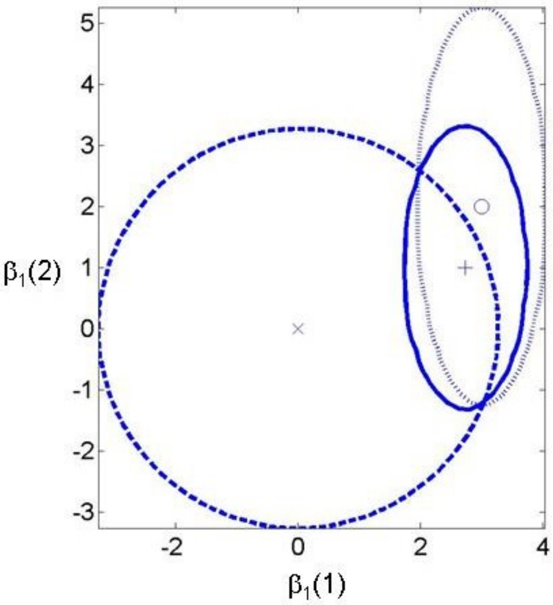


Figure 6.2: GLMs with two parameters. The prior (dashed line) has mean $\mu = [0,0]^T$ (cross) and precision $C_1^{-1} = \mathsf{diag}([1,1])$. The likelihood (dotted line) has mean $X^T y = [3,2]^T$ (circle) and precision $(X^T C_1^{-1} X)^{-1} = \mathsf{diag}([10,1])$. The posterior (solid line) has mean $m = [2.73,1]^T$ (cross) and precision $\Sigma^{-1} = \mathsf{diag}([11,2])$. In this example, the measurements are more informative about $\beta(1)$ than $\beta(2)$. This is reflected in the posterior distribution.

6.1.3 Augmented Form

From before

$$\begin{aligned} p(\beta|y) &= \mathsf{N}(m,\Sigma) \\ \Sigma^{-1} &= X^T C_1^{-1} X + C_2^{-1} \\ m &= \Sigma(X^T C_1^{-1} y + C_2^{-1}\mu) \end{aligned} \tag{6.8}$$

This can also be written as

$$\begin{aligned} \Sigma^{-1} &= \bar{X}^T V^{-1} \bar{X} \\ m &= \Sigma(\bar{X}^T V^{-1} \bar{y}) \end{aligned} \tag{6.9}$$

where

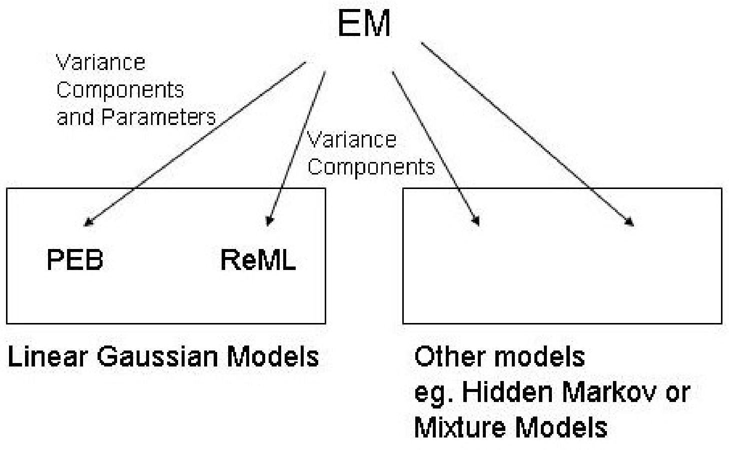
$$\begin{aligned} \bar{X} &= \begin{bmatrix} X \\ I \end{bmatrix} \\ V &= \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \\ \bar{y} &= \begin{bmatrix} y \\ \mu \end{bmatrix} \end{aligned} \tag{6.10}$$

where we've augmented the data matrix with prior expectations. Estimation in a Bayesian GLM is therefore equivalent to Maximum Likelihood estimation (ie. for IID covariances this is the same as Weighted Least Squares) with *augmented* data. Our prior beliefs can be thought of as extra data points.

6.2 Parametric Empirical Bayes

For a Bayesian GLM

$$\begin{aligned} y &= X\beta + e_1 \\ \beta &= \mu + e_2 \end{aligned} \tag{6.11}$$


EM
Variance Components and Parameters
Variance Components
PEB
ReML
Linear Gaussian Models
Other models eg. Hidden Markov or Mixture Models

with linear covariance constraints

$$\begin{aligned} C_1 &= \sum_i \lambda_i Q_i \\ C_2 &= \sum_j \lambda_j Q_j \end{aligned} \tag{6.12}$$

the covariance components can be estimated using ReML (last lecture). We can then make inferences about intermediate level parameters eg. β using Bayes rule (earlier in this lecture).

Also, the ReML algorithm can be reformulated into two steps (i) estimation the posterior distribution over β's and (ii) hyperparameter estimation (λ's). This reformulation is known as Parametric Empirical Bayes (PEB). The difference is that, in ReML, step (i) is embedded into step (ii). For ReML the goal is to estimate variance components, for PEB the goal is to estimate (intermediate level) parameters.

PEB is a special case of an Expectation-Maximisation (EM) algorithm where (i) E-Step: estimate posterior distribution over β's (ii) M-Step: update λ's. PEB/ReML are specific to linear Gaussian models but EM is generic, ie. there is an EM algorithm for mixture models, hidden Markov models etc.

For hierarchical linear models the PEB/EM algorithm is

- E-Step: Update distribution over parameters β

$$\begin{aligned} \Sigma^{-1} &= \bar{X}^T V^{-1} \bar{X} \\ m &= \Sigma(\bar{X}^T V^{-1} \bar{y}) \end{aligned} \tag{6.13}$$

- M-Step: Update hyperparameters λ_i (and therefore V) by following gradient g_i

$$\begin{aligned} r &= \bar{y} - \bar{X} m \\ g_i &= -\frac{1}{2} Tr(V^{-1} Q_i) + \frac{1}{2} Tr(\Sigma \bar{X}^T V^{-1} Q_i V^{-1} \bar{X}) \\ &+ \frac{1}{2} r^T V^{-1} Q_i V^{-1} r \end{aligned} \tag{6.14}$$

The M-Step is identical to ReML (last lecture) as the gradient can be expressed as

$$\begin{aligned} g_i &= -\frac{1}{2} Tr(PQ_i) + \frac{1}{2} y^T P^T Q_i P y \\ P &= V^{-1} - V^{-1} \bar{X} (\bar{X}^T V^{-1} \bar{X})^{-1} \bar{X}^T V^{-1} \end{aligned} \tag{6.15}$$

Whether or not EM or ReML is more computationally efficient for estimating variance components depends on the sparsity of the covariance constraints Q_i. For more details (and Fisher scoring implementation) see [17].

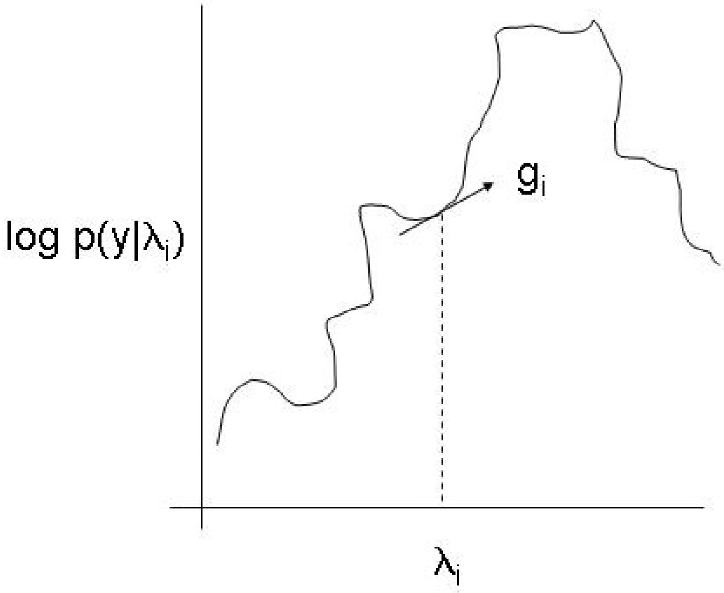


Figure 6.3: *EM and ReML estimate hyperparameters λ_i by following the gradient to the (local) maximum.*

6.2.1 Global Shrinkage Priors

Used in eg. fMRI analysis [15]. Special case of hierarchical model

$$\begin{aligned} y &= X\beta + e_1 \quad (6.16) \\ \beta &= \mu + e_2 \end{aligned}$$

with 20 voxels and 10 data points per voxel

$$\begin{aligned} X &= I_{20} \otimes 1_{10} \quad (6.17) \\ C_1 &= \sum_{i=1}^{20} \frac{1}{v_i} Q_i \\ C_2 &= \frac{1}{\alpha} I_{20} \end{aligned} \quad (6.18)$$

The parameter $\beta(i)$ encodes the effect size at voxel i. This model assumes that across the brain (i) average effect size is

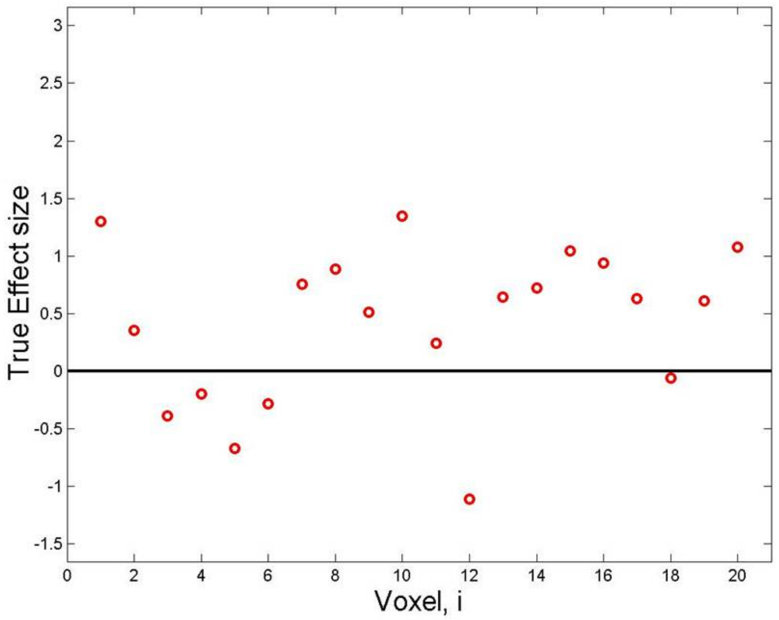


Figure 6.4: *Across the 20-voxel brain (i) average effect size is zero, $\mu = 0$, the variability of responses follows a Gaussian with precision α. True effect sizes (red circles).*

zero, $\mu = 0$, and (ii) the variability of responses follows a Gaussian with precision α. Hyperparameters are $\lambda = \{v_i, \alpha\}$.

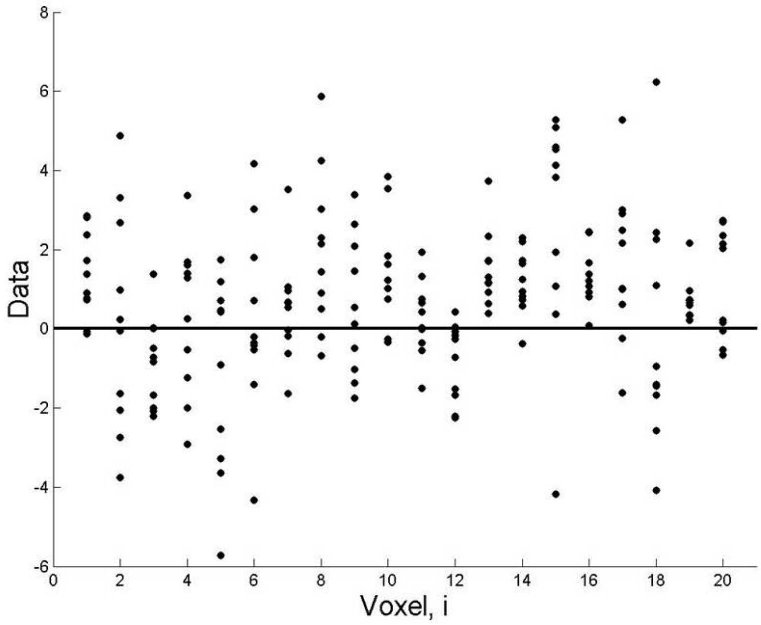


Figure 6.5: *Data at each voxel are normally distributed about the effect size at that voxel with precision λ_i eg. voxels 2, 5 and 15 have noisier data than others.*

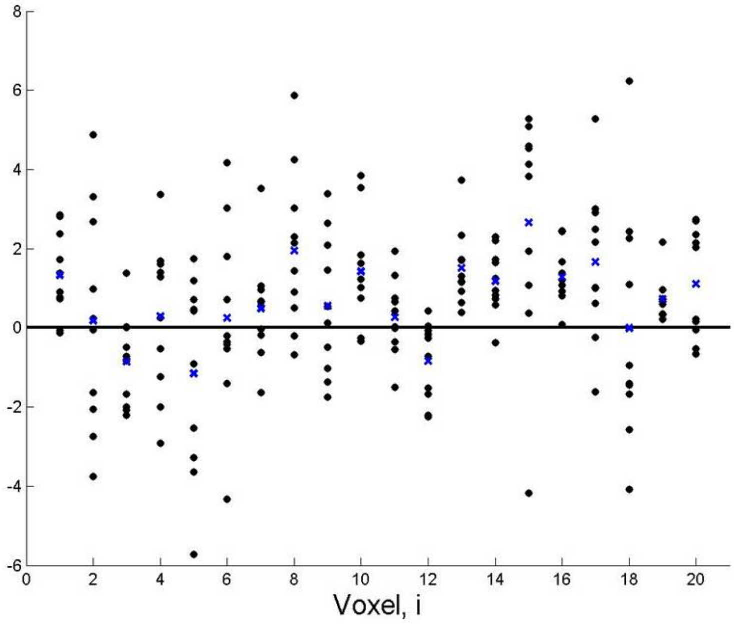


Figure 6.6: *Previous graph but with sample means (blue crosses) also at each voxel.*

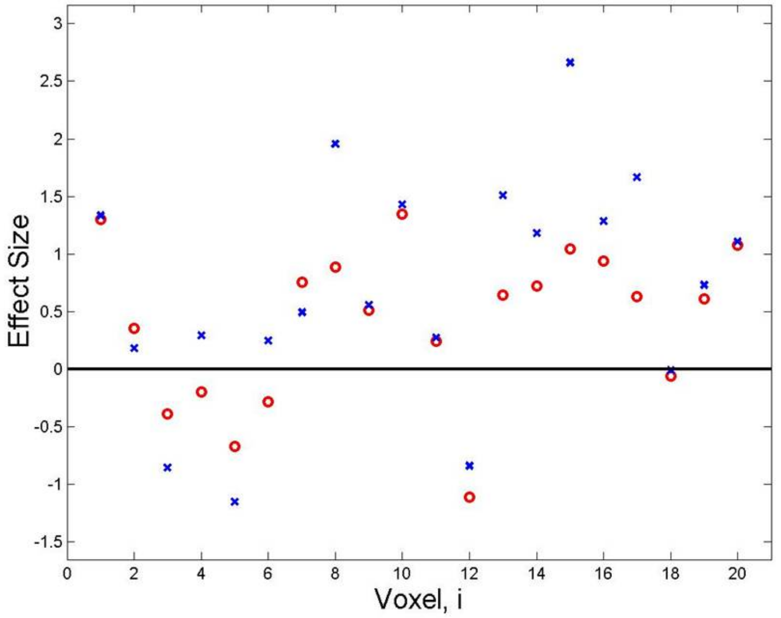


Figure 6.7: Sample means (also ML estimates - blue crosses) and true effect sizes (red circles). Estimation error =0.71.

For this model the PEB algorithm has a simple form. By setting the gradients g_i to zero we can get the following updates for the hyperparameters $\lambda = \{v_i, \alpha\}$.

$$\begin{aligned} \beta(i) &= \frac{\gamma_i}{N} \sum_{n=1}^{N} y_{in} \qquad (6.19) \\ \frac{1}{v_i} &= \frac{1}{N - \gamma_i} \sum_{n=1}^{N} (y_{in} - \beta(i))^2 \\ \gamma_i &= \frac{N v_i}{N v_i + \alpha} \\ \frac{1}{\alpha} &= \frac{1}{\sum_i \gamma_i} \sum_{i=1}^{V} \beta(i)^2 \end{aligned}$$

where y_{in} is the nth scan at the ith voxel, γ_i is the ratio of the data precision to the posterior precision.

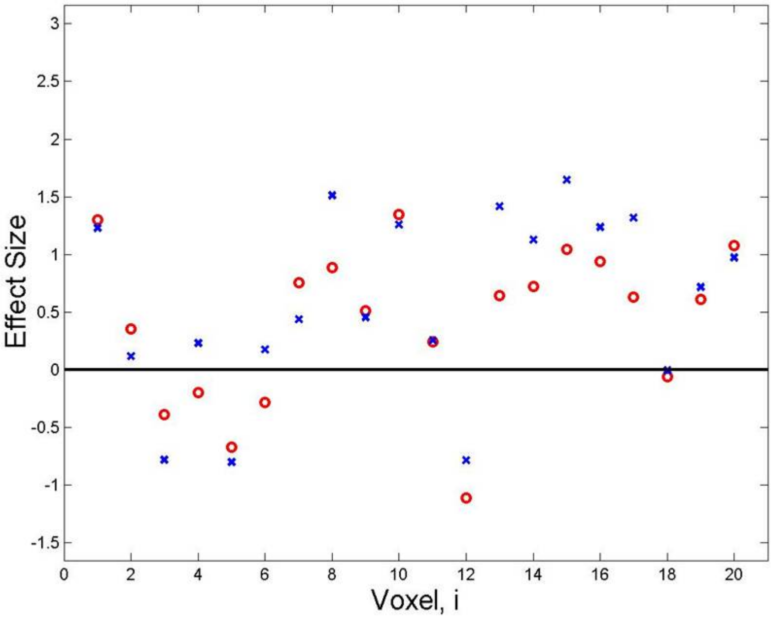


Figure 6.8: *After PEB iteration 3*

Without a prior, $\gamma_i = 1$ we get

$$\frac{1}{v_i} = \frac{1}{N-1} \sum_{n=1}^{N} (y_{in} - \beta(i))^2 \tag{6.20}$$

This is the familiar 'unbiased' estimate, if we only have to estimate variance components at a single level. The PEB updates partition the total degrees of freedom N into those used to estimate first or second level hyperparameters.

See code `em1.m`.

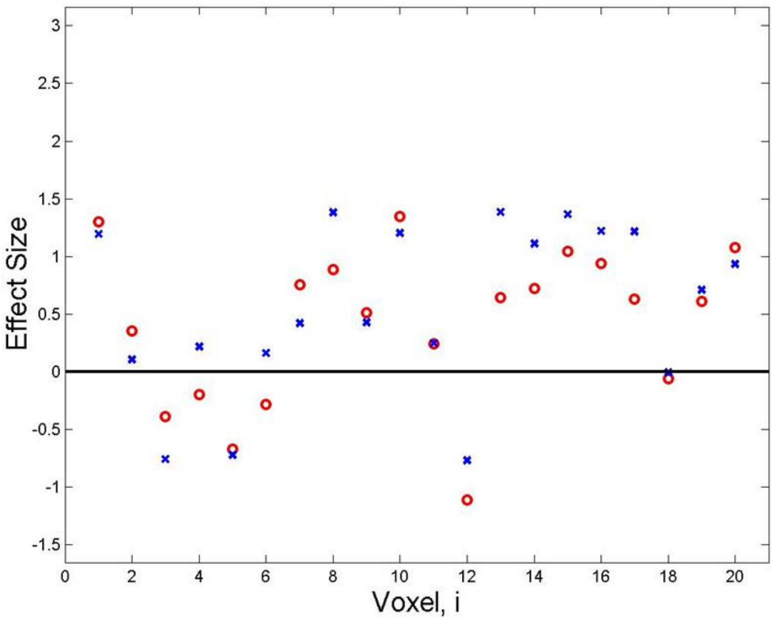


Figure 6.9: *After PEB iteration 7. Estimation error =0.34.*

On average, across the brain, PEB is more accurate than ML. It does better at most voxels at the expense of being worse at a minority eg. voxel 2.

For most voxels we have $\gamma_i = 0.9$, but for the noisy voxels 2, 15 and 18 we have $\gamma_i = 0.5$. PEB thus relies more on prior information where data are unreliable.

6.2.2 EEG Source Reconstruction

To 'reconstruct' EEG data at a *single time point* use the model

$$\begin{aligned} y &= X\beta + e_1 \qquad (6.21) \\ \beta &= \mu + e_2 \end{aligned}$$

where X is a lead-field matrix transforming Current Source Density (CSD) β at V voxels in brain space into EEG voltages y at S electrodes. For more on this see eg. [3].

$$\begin{aligned} C_1 &= \sum_i \lambda_i Q_i \qquad (6.22) \\ C_2 &= \sum_j \lambda_j Q_j \end{aligned}$$

$$(6.23)$$

where Q_i defines structure of sensor noise, and Q_j source noise ie. uncertainty in sources. In the application that follows we use $Q_i = I$ and $Q_j = L$, a 'Laplacian' matrix set up so that we expect the squared difference between neighboring voxels to be λ_j ie. this enforces a smoothness constraint.

The data in this analysis is from [22].

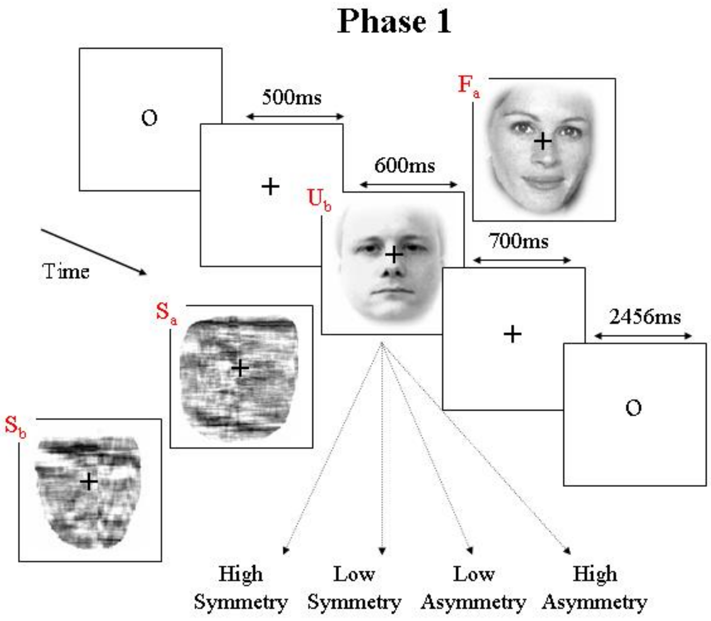


Figure 6.10: *Subjects are presented images of faces and scrambled faces and are asked to make symmetry judgements.*

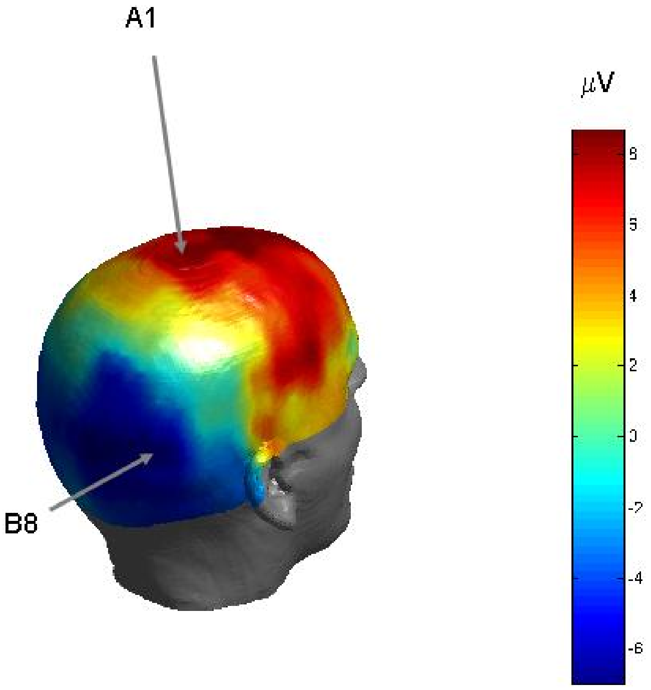


Figure 6.11: *Electrode voltages at 160ms post-stimulus, y. This is an Event-Related Potential (ERP), the result of averaging the responses to many (86) trials.*

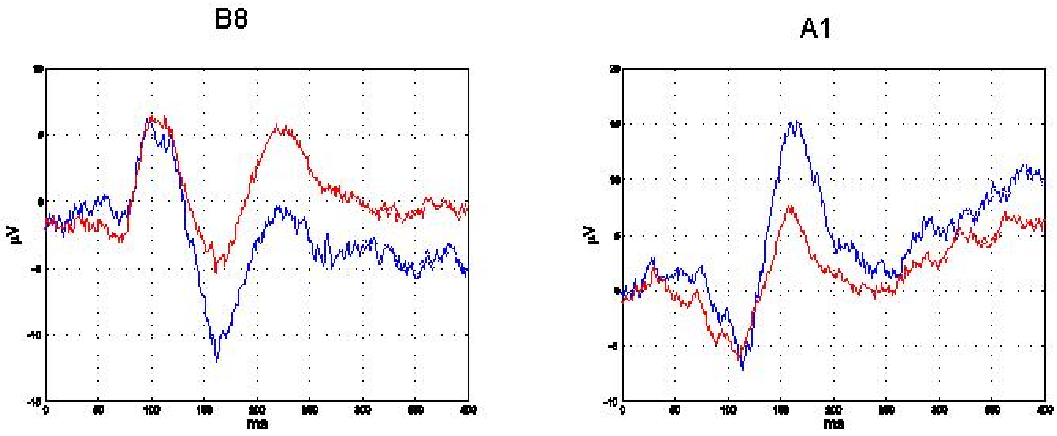


Figure 6.12: *Voltages at two different electrodes for faces (blue) and scrambled faces (red). These are Event-Related Potentials (ERPs), the result of averaging the responses to many (86) trials.*

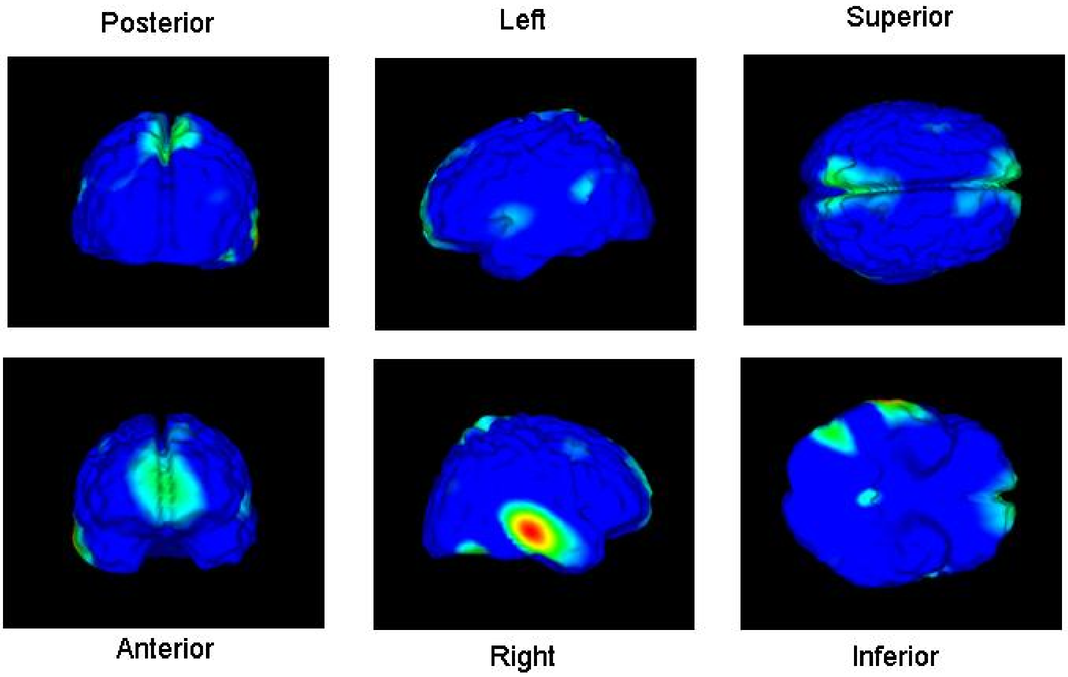


Figure 6.13: Estimate of CSD, β. Computed as the CSD difference for faces minus scrambled faces.

Chapter 7

Model Comparison

7.1 Contents

Making inferences about models

- Bayes factors
- Evidence for Bayesian GLMs
- Multimodal imaging
- Bayesian Model Averaging
- Nonlinear M/EEG source localisation.

7.2 Bayes Factors

Bayes rule for data y and 'model' or 'hypothesis' i

$$p(m = i|y) = \frac{p(y|m = i)p(m = i)}{p(y)}$$

In this context, $p(y|m = i)$, is known as the evidence for model i. Similarly for model j

$$p(m = j|y) = \frac{p(y|m = j)p(m = j)}{p(y)}$$

Dividing one by the other gives

$$\frac{p(m = i|y)}{p(m = j|y)} = \frac{p(y|m = i)}{p(y|m = j)} \times \frac{p(m = i)}{p(m = j)}$$

This is the fundamental relationship

$$PosteriorOdds = BayesFactor \times PriorOdds$$

The Bayes factor is a ratio of model evidences. It tells you how the odds have changed. It can be written BF_{ij}.

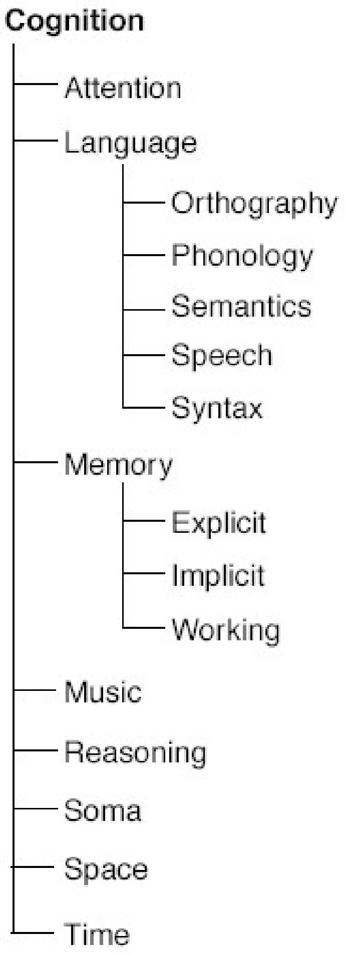


Figure 7.1: Cognitive processes, m, described in BrainMap database.

7.2.1 Inferring cognitive processes

Poldrack[41] considers the relationship between engagement of cognitive processes, m, and activation of brain regions y. For example, using the BrainMap database, the frequency of language studies, L, that give rise to Broca activations (20mm ROI at $x = -37, y = 18, z = 18$mm) can be used to estimate

$$\begin{aligned} p(y = B|m = L) &= \frac{p(y = B, m = L)}{p(m = L)} \qquad (7.1) \\ &= \frac{166}{869} = 0.191 \end{aligned}$$

Simiarly, given the number of non-language studies, $\bar{L}$, that also activate Broca's area

$$\begin{aligned} p(y = B|m = \bar{L}) &= \frac{p(y = B, m = \bar{L})}{p(m = \bar{L})} \\ &= \frac{199}{2353} = 0.085 \end{aligned} \tag{7.2}$$

This gives rise to a Bayes factor

$$\begin{aligned} BF_{L,\bar{L}} &= \frac{p(y = B|m = L)}{p(y = B|m = \bar{L})} \\ &= \frac{0.191}{0.085} = 2.3 \end{aligned} \tag{7.3}$$

That is, after seeing a Broca activation, the odds that a language process has been engaged are larger by a factor 2.3.

For equal prior odds $p(m = L) = p(m = \bar{L}) = 0.5$, the posterior probability of language processes given a Broca activation is

$$\begin{aligned} p(m = L|y = B) &= \frac{p(y = B|m = L)p(m = L)}{p(y = B|m = L)p(m = L) + p(y = B|m = \bar{L})p(m = \bar{L})} \\ &= \frac{0.191}{0.191 + 0.085} = 0.69 \end{aligned}$$

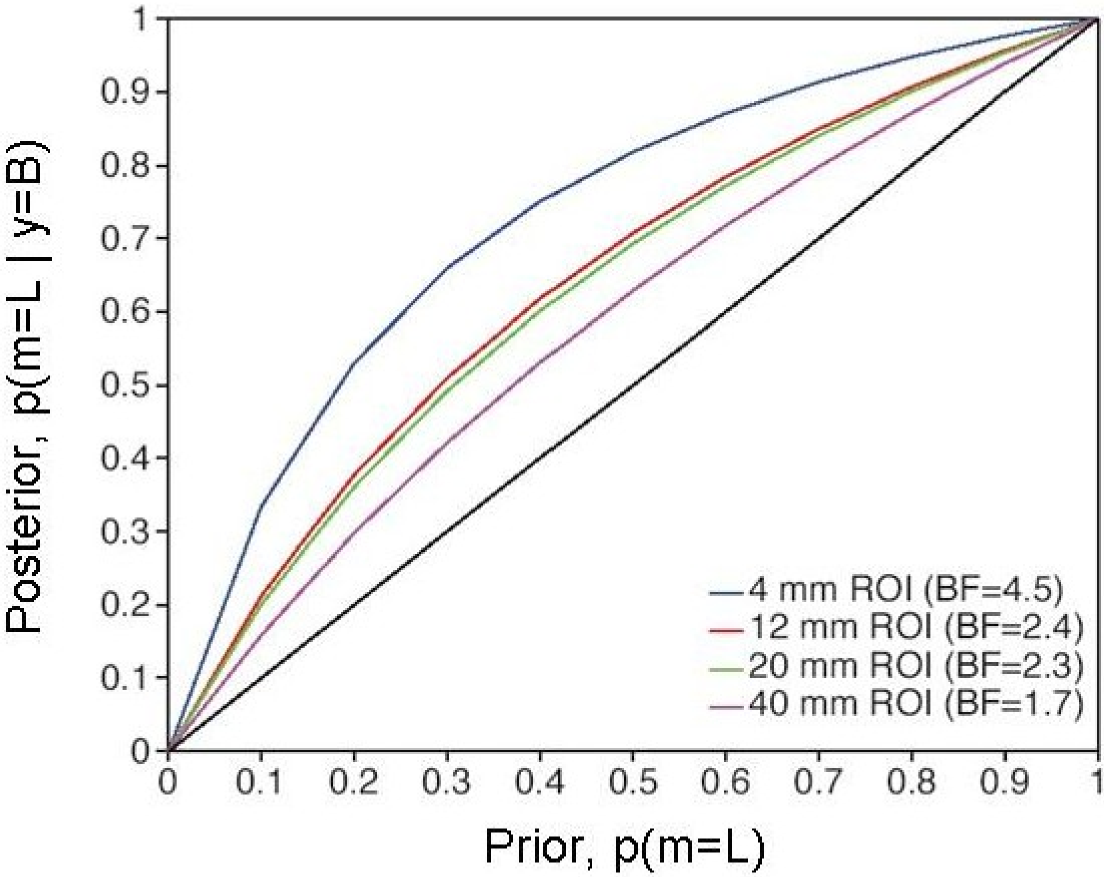


Figure 7.2: Effect of ROI size on posterior probability. Power of reverse inference is increased using smaller, more selective regions.

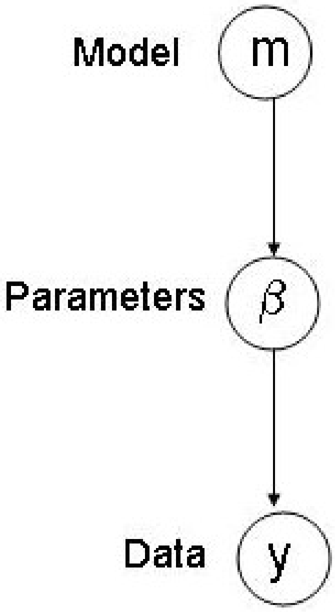


Figure 7.3: *Hierarchical generative model in which members of a model class, indexed by m, are considered as part of the hierarchy. Typically, m indexes the structure of the model. This might be the connectivity pattern in a dynamic causal model or set of anatomical or functional constraints in a source reconstruction model. Once a model has been chosen from the distribution $p(m)$, its parameters are generated from the parameter prior $p(\theta|m)$ and finally data is generated from the likelihood $p(y|\theta, m)$.*

7.3 Making inferences about models

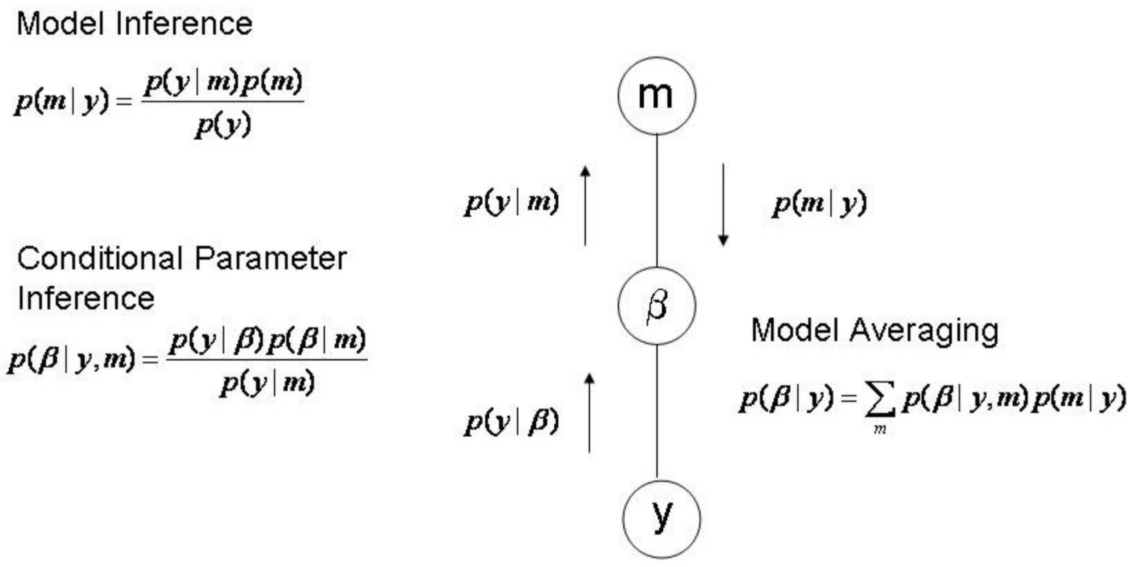

Model Inference
$p(m|y)=\frac{p(y|m)p(m)}{p(y)}$
m
$p(y|m)$
$p(m|y)$
Conditional Parameter Inference
$p(\beta|y,m)=\frac{p(y|\beta)p(\beta|m)}{p(y|m)}$
β
Model Averaging
$p(\beta|y)=\sum_m p(\beta|y,m)p(m|y)$
$p(y|\beta)$
y

Figure 7.4: *In Bayesian Model Selection (BMS), the posterior model probability $p(m|y)$, is used to select a single 'best' model. In Bayesian Model Averaging (BMA), inferences are based on all models and $p(m|y)$ is used as a weighting factor. Only in BMA, are parameter inferences based on the correct marginal density $p(\theta|y)$.*

7.4 Evidence for Bayesian GLMs

For a Bayesian GLM

$$\begin{aligned} y &= X\beta + e_1 \\ \beta &= \mu + e_2 \end{aligned} \tag{7.4}$$

with linear covariance constraints

$$\begin{aligned} C_1 &= \sum_i \lambda_i Q_i \\ C_2 &= \sum_j \lambda_j Q_j \end{aligned} \tag{7.5}$$

From lecture 6 we know that the posterior distribution over regression coefficients is

$$\begin{aligned} \Sigma^{-1} &= \bar{X}^T V^{-1} \bar{X} \\ \hat{\beta} &= \Sigma(\bar{X}^T V^{-1} \bar{y}) \end{aligned} \tag{7.6}$$

where

$$\begin{aligned}
\bar{X} &= \begin{bmatrix} X \\ I \end{bmatrix} \qquad (7.7) \\
V &= \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \\
\bar{y} &= \begin{bmatrix} y \\ \mu \end{bmatrix}
\end{aligned}$$

where we've augmented the data matrix with prior expectations.

We'll assume that we've run PEB and so have estimated parameters, $\hat{\beta}$, and hyperparameters, $\hat{\lambda}$. We now wish to compute the model evidence $p(y|m)$.

From lecture 5 we know that

$$\begin{aligned}
p(y|\lambda, m) &= (2\pi)^{-N/2}|V|^{-1/2} \qquad (7.8) \\
&\times \exp\left(\frac{1}{2}(\bar{y} - \bar{X}\hat{\beta})^T V^{-1}(\bar{y} - \bar{X}\hat{\beta})\right) \\
&\times |\bar{X}^T V^{-1} \bar{X}|^{-1/2}
\end{aligned}$$

By substuting in the expressions for V, $\bar{y}$ and $\bar{X}$ and taking logs we can write the log evidence as

$$\log p(y|\lambda, m) = Accuracy(m) - Complexity(m) \tag{7.9}$$

where

$$\begin{aligned} Accuracy(m) &= -\frac{1}{2}\log|C_1| - \frac{1}{2}(y - X\hat{\beta})^T C_1^{-1}(y - X\hat{\beta}) \\ Complexity(m) &= \frac{1}{2}\log|C_2| - \frac{1}{2}\log|\Sigma| + \frac{1}{2}(\mu - \hat{\beta})^T C_2^{-1}(\mu - \hat{\beta}) \end{aligned}$$

The second term is referred to as 'complexity' because eg. the quadratic term scales with the number of parameters in the model. A model with high evidence must therefore provide a good trade-off between accuracy and complexity.

This trade-off is also employed in other more ad-hoc model selection schemes eg. AIC and BIC have complexity terms embodying fixed costs for each parameter of 1 (AIC) and $\frac{1}{2}\log N$. See eg. [39] for more details.

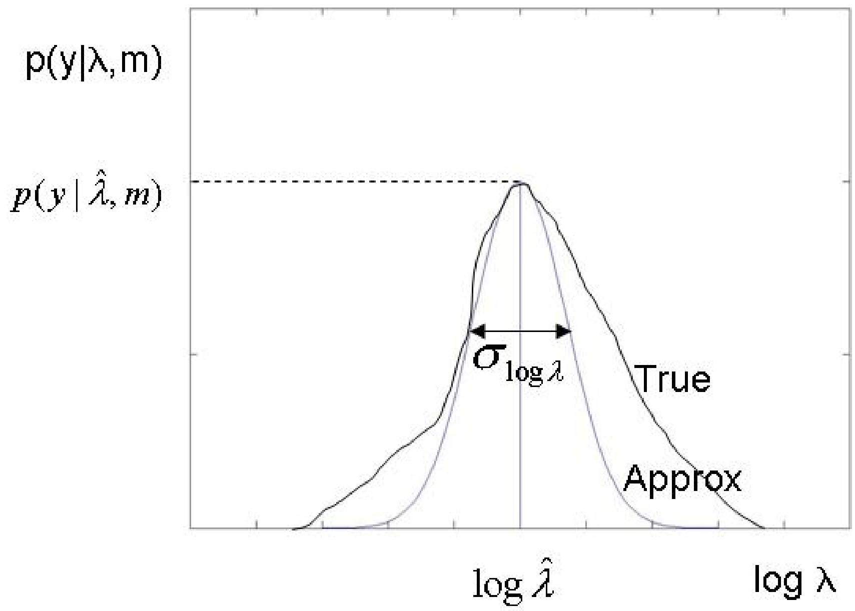


Figure 7.5: *Approximating the hyperparameter uncertainty with a Gaussian in log space.*

7.4.1 Integrating out hyperparameters

To get the evidence $p(y|m)$ we must integrate out the uncertainty in the hyperparameters λ.

$$p(y|m) = \int p(y|\lambda, m) d\lambda \tag{7.10}$$

To do this we'll assume that the hyperparameters have a Gaussian distribution about their estimated value, $\hat{\lambda}$. As the hyperparameters must be positive we'll assume that this distribution is in log space. If we have a single hyperparameter then

$$p(y|\lambda, m) = p(y|\hat{\lambda}, m) \exp\left(-\frac{(\log\lambda - \log\hat{\lambda})^2}{2\sigma^2_{\log\lambda}}\right) \tag{7.11}$$

where $\sigma^2_{\log\lambda}$ is our uncertainty (variance) in the (log) estimated hyperparamater. We can then evaluate the integral to give

$$p(y|m) = p(y|\hat{\lambda}, m)(2\pi)^{1/2}\sigma_{\log\lambda} \tag{7.12}$$

The last terms are just the normalising constant for the Gaussian density. This expression for the evidence takes into account uncertainty in the estimation of the hypeparameters. If we have H hyperparameters then we get

$$p(y|m) = p(y|\hat{\lambda}, m)(2\pi)^{H/2}\prod_{h=1}^{H}\sigma_{\log\lambda_h} \tag{7.13}$$

7.5 Multimodal Imaging

Source reconstruction of EEG using fMRI location priors [29]. To ‘reconstruct’ EEG data at a *single time point* use the model

$$\begin{aligned} y &= X\beta + e_1 \\ \beta &= \mu + e_2 \end{aligned} \tag{7.14}$$

where X is a lead-field matrix transforming Current Source Density (CSD) β at V voxels in brain space into EEG voltages y at S electrodes. We use $\mu = 0$.

$$\begin{aligned} C_1 &= \sum_i \lambda_i Q_i \\ C_2 &= \sum_j \lambda_j Q_j \end{aligned} \tag{7.15}$$

where Q_i defines structure of sensor noise, and Q_j source noise ie. uncertainty in sources. In the application that follows we use $Q_i = I$ and $Q_j = L$, a ‘Laplacian’ or ‘smoothness’ matrix set up

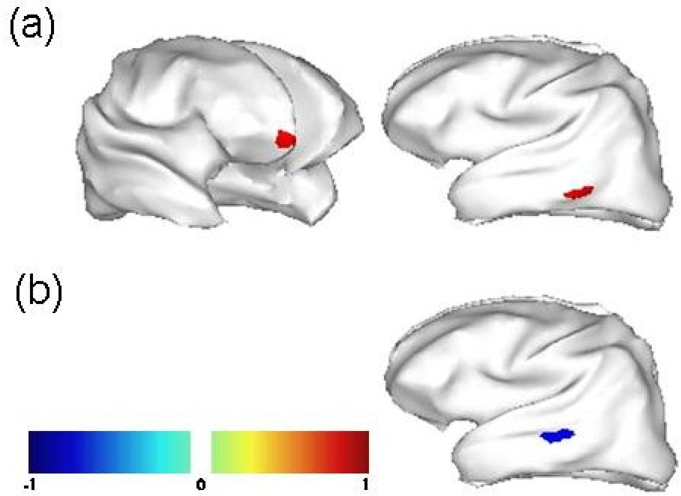


Figure 7.6: *Inflated cortical representation of (a) two simulated source locations ('valid' prior) and (b) 'invalid' prior location.*

so that we expect the squared difference between neighboring voxels to be λ_j. Also consider extra Q_j's to incorporate valid and invalid location priors from eg. fMRI [29].

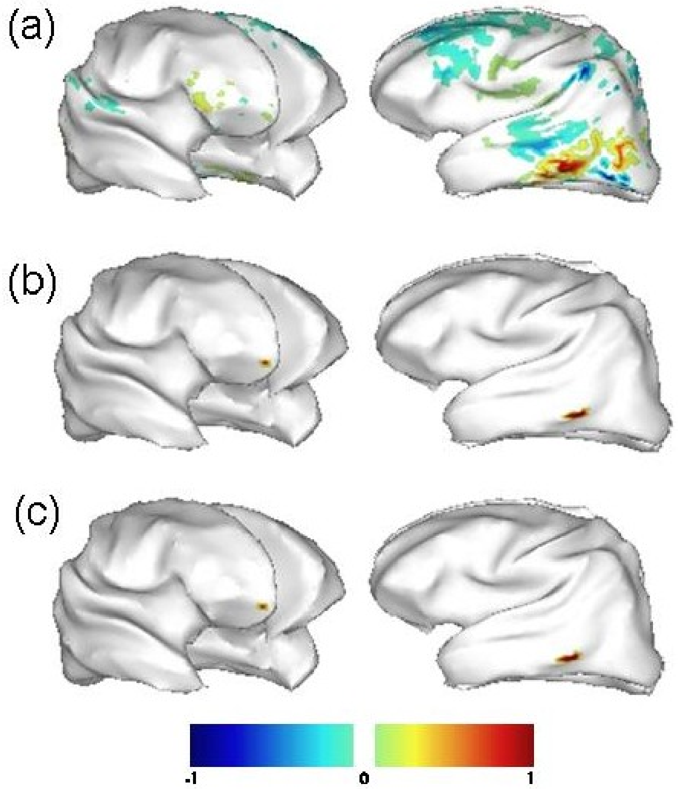


Figure 7.7: *Inflated cortical representation of representative source reconstructions using (a) smoothness prior, (b) smoothness and valid priors and (c) smoothness, valid and invalid priors. The reconstructed values have been normalised between -1 and 1.*

7.6 Nonlinear source reconstruction

Trujillo-Barreto et al. [45] describe a nonlinear source reconstruction algorithm based on combining reconstructions from a very large number of different models $m = 1..M$, using Bayesian Model Averaging (BMA)

$$p(\beta|y) = \sum_m p(\beta|y,m)p(m|y) \tag{7.16}$$

where $p(\beta|y,m)$ is the estimated CSD from model m and $p(m|y)$ is the posterior probability of model m. If all models are equi-likely apriori then $p(m|y) = p(y|m)$. We therefore need to

- Fit model m to get CSD estimates
- Estimate model evidence $p(y|m)$
- Search model space M

Model space contains $M = 2^{71}$ models. There's no point fitting models that will have a low evidence. Use a greedy search strategy where eg. at search iteration i our model contains regions 13, 40-45 and 62. Add/delete a region chosen uniformly at random and select it for iteration $i+1$ if evidence is higher. Keep all models with evidence greater than 1/20th of max so far - this is Occam's window of models.

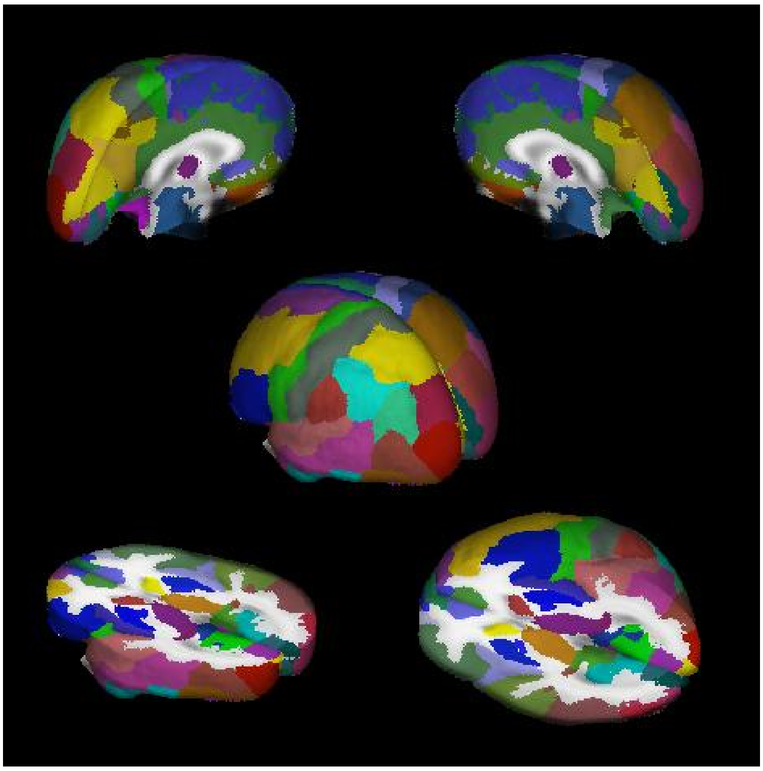


Figure 7.8: 3D segmentation of 71 structures of the Probabilistic MRI Atlas developed at the Montreal Neurological Institute. As shown in the color scale, brain areas belonging to different hemispheres were segmented separately.

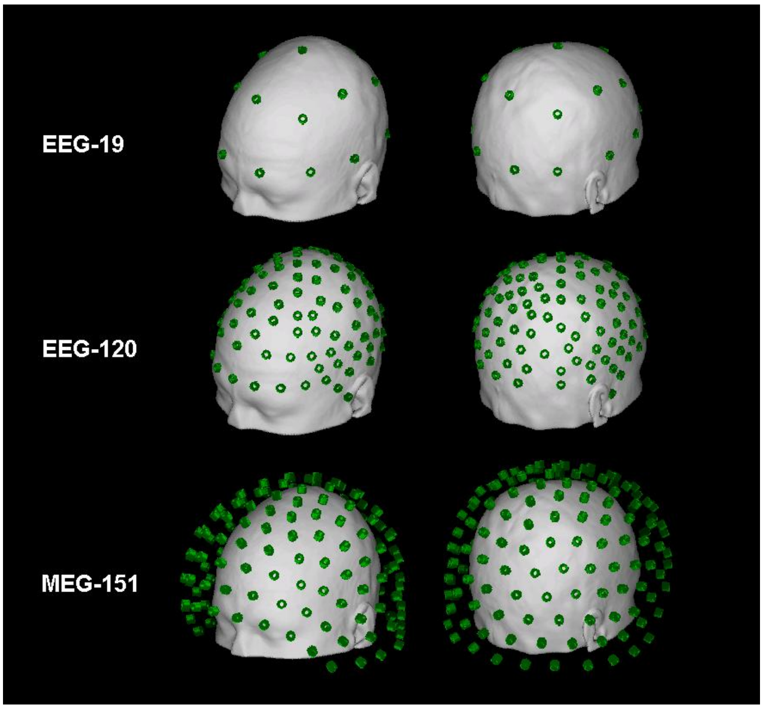


Figure 7.9: Different arrays of sensors used in the simulations. EEG-19 represents the 10/20 electrode system; EEG-120 is obtained by extending and refining the 10/20 system; and MEG-151 corresponds to the spatial configuration of MEG sensors in the helmet of the CTF System Inc.

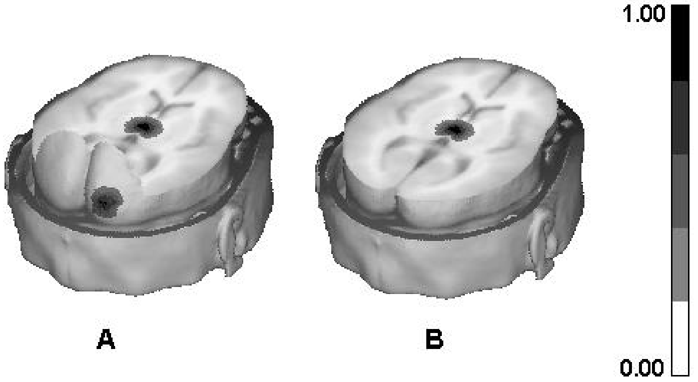


Figure 7.10: Spatial distributions of the simulated primary current densities. A) Simultaneous activation of two sources at different depths: one in the right Occipital Pole and the other in the Thalamus (OPR+TH). B) Simulation of a single source in the Thalamus (TH).

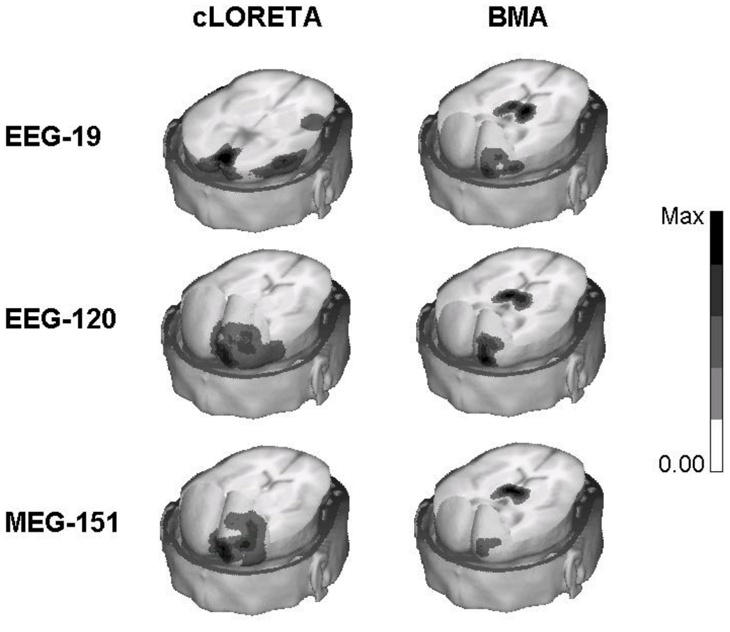


Figure 7.11: 3D reconstructions of the absolute values of BMA and cLORETA solutions for the OPR+TH source case. The first column indicates the array of sensors used in each simulated data set. The maximum of the scale is different for each case.

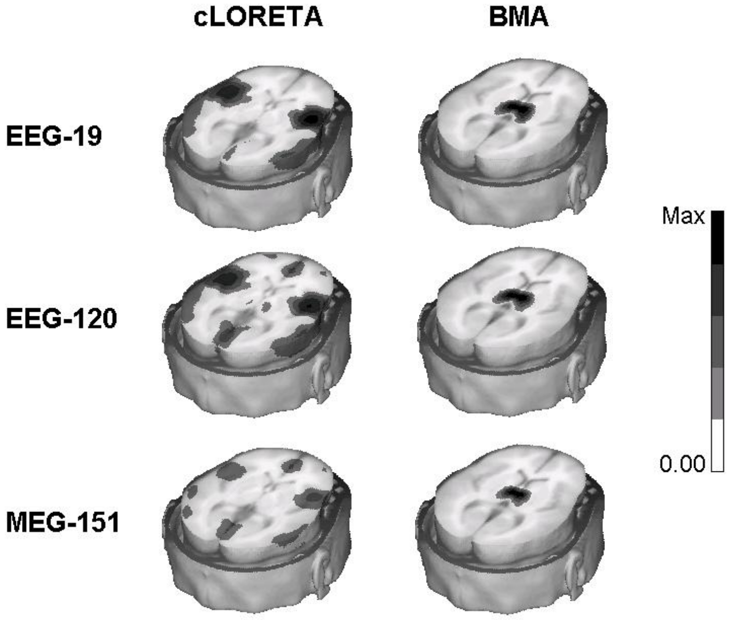


Figure 7.12: 3D reconstructions of the absolute values of BMA and cLORETA solutions for the TH source case. The first column indicates the array of sensors used in each simulated data set.

Chapter 8

Spectral Estimation

8.1 Contents

- Sinusoidal models
- Fourier transform
- Welch's method
- Multitaper method
- Multivariate spectral analysis
- Source reconstruction of MEG Gamma activity

8.1.1 Sines and cosines

Sines and cosines can be understood in terms of the vertical and horizontal displacement of a fixed point on a rotating wheel; the wheel has unit length and rotates *anti-clockwise*. The angle round the wheel is measured in degrees or radians ($0 - 2\pi$; for unit radius circles the circumference is 2π, radians tell us how much of the circumference we've got). If we go round the wheel a whole number of times we end up in the same place, eg.$\cos 4\pi = \cos 2\pi = \cos 0 = 1$. Frequency, f, is the number of times round the wheel per second. Therefore, given $x = \cos(2\pi ft)$, $x = 1$ at $t = 1/f, 2/f$ etc. For $x = \cos(2\pi ft + \Phi)$ we get a head start (lead) of Φ radians. Negative frequencies may be viewed as a wheel rotating *clockwise* instead of anti-clockwise.

8.1.2 Sampling and aliasing

If we assume we have samples of the signal every T_s seconds and in total we have N such samples then T_s is known as the sampling period and $F_s = 1/T_s$ is the sampling frequency in Hertz (Hz) (samples per second). The nth sample occurs at time $t[n] = nT_s = n/F_s$. The cosine of sampled data can be written

$$x[n] = \cos(2\pi f t[n]) \tag{8.1}$$

At a sampling frequency F_s the only *unique* frequencies are in the range 0 to $(F_s/2)$Hz. Any frequencies outside this range become *aliases* of one of the unique frequencies.

For example, if we sample at 8Hz then a -6Hz signal becomes indistinguishable from a 2Hz signal. More generally, if f_0 is a unique frequency then its aliases have frequencies given by

$$f = f_0 + kF_s \tag{8.2}$$

where k is any positive or negative integer, eg. for $f_0 = 2$ and $F_s = 8$ the two lowest frequency aliases, given by $k = -1$ and $k = 1$, are -6Hz and 10Hz.

8.2 Sinusoidal models

If our time series has a periodic component in it we might think about modelling it with the equation

$$x[n] = R_0 + Rcos(2\pi f t[n] + \Phi) + e[n] \tag{8.3}$$

where R_0 is the offset (eg. mean value of x[n]), R is the amplitude of the sine wave, f is the frequency and Φ is the phase. Because of the trig identity

$$\cos(A + B) = \cos A \cos B - \sin A \sin B \tag{8.4}$$

the model can be written in an alternative form

$$x[n] = R_0 + a\cos(2\pi f t[n]) + b\sin(2\pi f t[n]) + e[n] \tag{8.5}$$

where $a = R\cos(\Phi)$ and $b = -R\sin(\Phi)$. This is the form we consider for subsequent analysis.

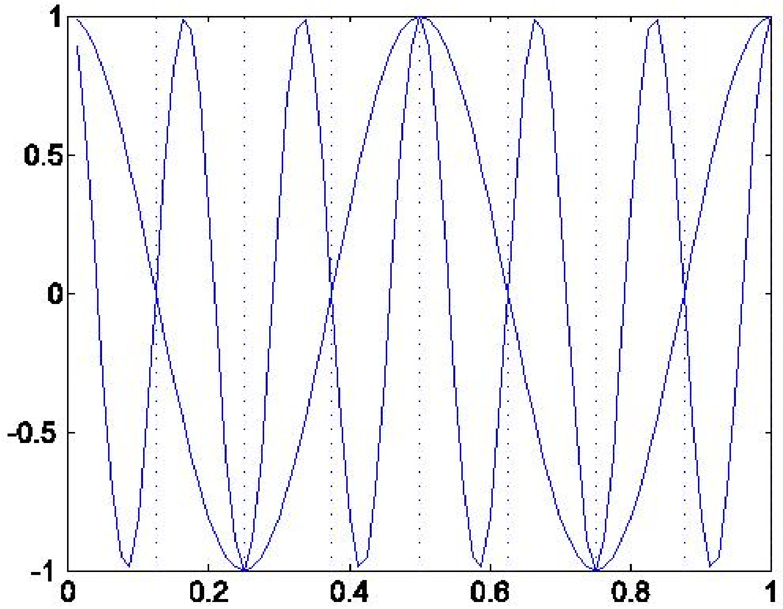


Figure 8.1: **Aliases** *The figure shows a 2Hz cosine wave and a -6Hz cosine wave as solid curves. At sampling times given by the dotted lines, which correspond to a sampling frequency of 8Hz, the* $-6Hz$ *signal is an alias of the 2Hz signal. Other aliases are given by equation 8.2.*

8.2.1 Fitting the model

If we let $\boldsymbol{x} = [x(1), x(2), ..., x(N)]^T$, $\boldsymbol{w} = [R_0, a, b]^T$, $\boldsymbol{e} = [e_1, e_2, ..., e_N]^T$ and

$$\boldsymbol{A} = \begin{bmatrix} 1 & cos2\pi ft[1] & sin2\pi ft[1] \\ 1 & cos2\pi ft[2] & sin2\pi ft[2] \\ 1 & cos2\pi ft[3] & sin2\pi ft[3] \\ .. & .. & .. \\ 1 & cos2\pi ft[N] & sin2\pi ft[N] \end{bmatrix} \tag{8.6}$$

then the model can be written in the matrix form

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{w} + \boldsymbol{e} \tag{8.7}$$

which is in the standard form of a multivariate linear regression problem. The solution is therefore

$$\boldsymbol{w} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{x} \tag{8.8}$$

8.2.2 But sinewaves are orthogonal

We restrict ourselves to a frequency f_p which is an integer multiple of the *base frequency*

$$f_p = pF_b \tag{8.9}$$

where $p = 1..N/2$ and

$$f_b = \frac{F_s}{N} \tag{8.10}$$

eg. for $F_s = 100$ and $N = 100$ (1 seconds worth of data), $f_b = 1Hz$ and we can have f_p from 1Hz up to 50Hz[1]. The *orthogonality* of sinewaves is expressed in the following equations

$$\sum_{n=1}^{N} \cos 2\pi f_k t[n] = \sum_{n=1}^{N} \sin 2\pi f_k t[n] = 0 \tag{8.11}$$

$$\sum_{n=1}^{N} \cos 2\pi f_k t[n] \sin 2\pi f_l t[n] = 0 \tag{8.12}$$

[1] To keep things simple we don't allow f_p where $p = N/2$; if we did allow it we'd get N and 0 in equations 8.13 and 8.14 for the case $k = l$. Also we must have N even.

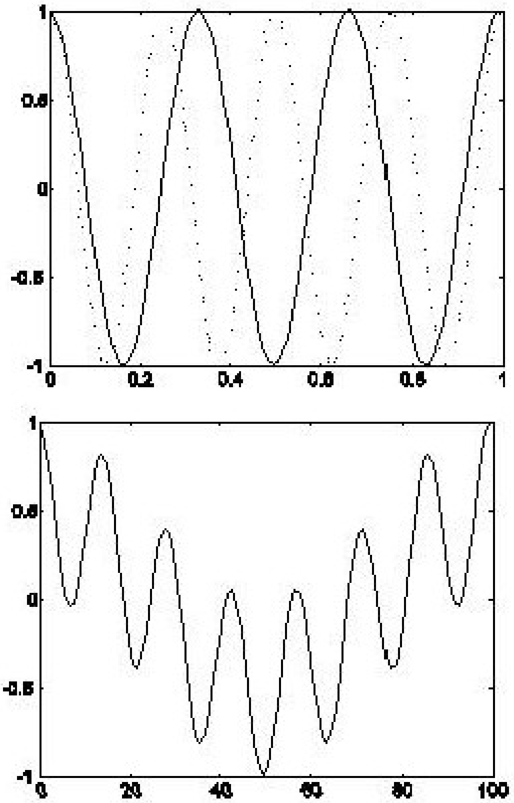


Figure 8.2: **Orthogonality of sinewaves** *Figure (top) shows* $\cos 2\pi 3 f_b t[n]$ *and* $\cos 2\pi 4 f_b t[n]$*, cosines which are 3 and 4 times the base frequency* $f_b = 1Hz$*. For any two integer multiples* k, l *we get* $\sum_{n=1}^{N} cos 2\pi f_k t[n] cos 2\pi f_l t[n] = 0$*. This can be seen from Figure (bottom) which shows the product* $cos 2\pi 3 f_b t[n] cos 2\pi 4 f_b t[n]$*. Because of the trig identity* $cos A cos B = 0.5 cos(A+B) + 0.5 cos(A-B)$ *this looks like a 7Hz signal superimposed on a 1Hz signal. The sum of this signal over a whole number of cycles can be seen to be zero; because each cos term sums to zero. If, however, k or l are not integers the product does not sum to zero and the orthogonality breaks down.*

$$\sum_{n=1}^{N} \cos 2\pi f_k t[n] \cos 2\pi f_l t[n] = \begin{array}{ll} 0 & k \neq l \\ N/2 & k = l \end{array} \qquad (8.13)$$

$$\sum_{n=1}^{N} \sin 2\pi f_k t[n] \sin 2\pi f_l t[n] = \begin{array}{ll} 0 & k \neq l \\ N/2 & k = l \end{array} \qquad (8.14)$$

The results depend on the fact that all frequencies that appear in the above sums are integer multiples of the base frequency; see figure 8.2.

This property of sinewaves leads to the result

$$\boldsymbol{A}^T \boldsymbol{A} = \boldsymbol{D} \qquad (8.15)$$

where $\boldsymbol{D}$ is a diagonal matrix. The first entry is N (from the inner product of two columns of 1's of length N; the 1's are the coefficients of the constant term R_0) and all the other entries are $N/2$. A matrix $\boldsymbol{Q}$ for which

$$\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{D} \tag{8.16}$$

is said to be orthogonal. Therefore our $\boldsymbol{A}$ matrix is orthogonal. Hence

$$\boldsymbol{w} = \boldsymbol{D}^{-1}\boldsymbol{A}^T\boldsymbol{x} \tag{8.17}$$

which is simply a *projection* of the signal onto the basis matrix, with some pre-factor . Given that $\boldsymbol{w} = [a, b, R_0]^T$ we can see that, for example, a is computed by simply projecting the data onto the second column of the matrix $\boldsymbol{A}$, eg.

$$a = \frac{2}{N}\sum_{n=1}^{N} \cos(2\pi f t) x_t \tag{8.18}$$

Similarly,

$$b = \frac{2}{N}\sum_{n=1}^{N} \sin(2\pi f t) x_t \tag{8.19}$$

$$R_0 = \frac{1}{N}\sum_{n=1}^{N} x_t \tag{8.20}$$

We applied the simple sinusoidal model to a 'sunspot data set' as follows. We chose 60 samples between the years 1731 and 1790 (because there was a fairly steady mean level in this period). The sampling rate $F_s = 1$Year. This gives a base frequency of $f_b = 1/60$. We chose our frequency $f = pf_b$ with p=6; giving a complete cycle once every ten years. This gave rise to the following estimates; $R_0 = 53.64$, $a = 39.69$ and $b = -2.36$.

8.2.3 Fourier Series

We might consider that our signal consists of lots of periodic components in which case the *multiple sinusoidal model* would be more appropriate

$$x(t) = R_0 + \sum_{k=1}^{p} R_k \cos(2\pi f_k t + \Phi_k) + e_t \tag{8.21}$$

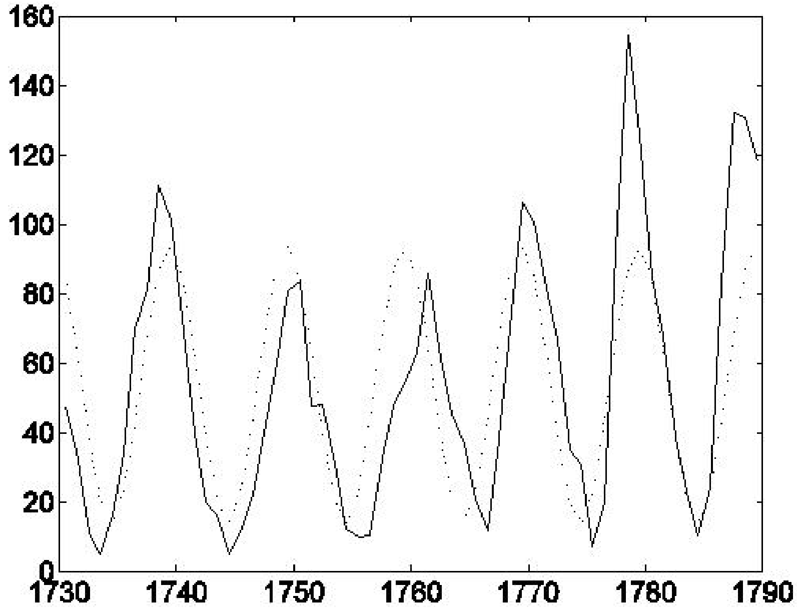


Figure 8.3: *Sunspot index (solid line) and prediction of it from a simple sinusoidal model (dotted line).*

where there are p sinusoids with different frequencies and phases. In a discrete Fourier series there are $p = N/2$ such sinusoids having frequencies

$$f_k = \frac{kF_s}{N} \tag{8.22}$$

where $k = 1..N/2$ and F_s is the sampling frequency. Thus the frequencies range from F_s/N up to $F_s/2$. The Fourier series expansion of the signal $x(t)$ is

$$x(t) = R_0 + \sum_{k=1}^{N/2} R_k \cos(2\pi f_k t + \Phi_k) \tag{8.23}$$

Notice that there is no noise term. Because of the trig identity

$$\cos(A + B) = \cos A \cos B - \sin A \sin B \tag{8.24}$$

this can be written in the form

$$x(t) = a_0 + \sum_{k=1}^{N/2} a_k \cos(2\pi f_k t) + b_k \sin(2\pi f_k t) \tag{8.25}$$

where $a_k = R_k \cos(\Phi_k)$ and $b_k = -R_k \sin(\Phi_k)$. Alternatively, we have $R_k^2 = a_k^2 + b_k^2$ and $\Phi = \tan^{-1}(b_k/a_k)$. Equivalently, we can write the nth sample as

$$x[n] = a_0 + \sum_{k=1}^{N/2} a_k \cos(2\pi f_k t[n]) + b_k \sin(2\pi f_k t[n]) \tag{8.26}$$

where $t[n] = nT_s$.

The important things to note about the sinusoids in a Fourier series are (i) the frequencies are *equally* spread out, (ii) there are $N/2$ of them where N is the number of samples, (iii) Given F_s and N the frequencies are *fixed.* Also, note that in the Fourier series 'model' there is *no noise.*

The Fourier coefficients can be computed by a generalisation of the process used to compute the coefficients in the simple sinusoidal model.

$$a_k = \frac{2}{N} \sum_{n=1}^{N} \cos(2\pi f_k t[n]) x[n] \tag{8.27}$$

Similarly,

$$b_k = \frac{2}{N} \sum_{n=1}^{N} \sin(2\pi f_k t[n]) x[n] \tag{8.28}$$

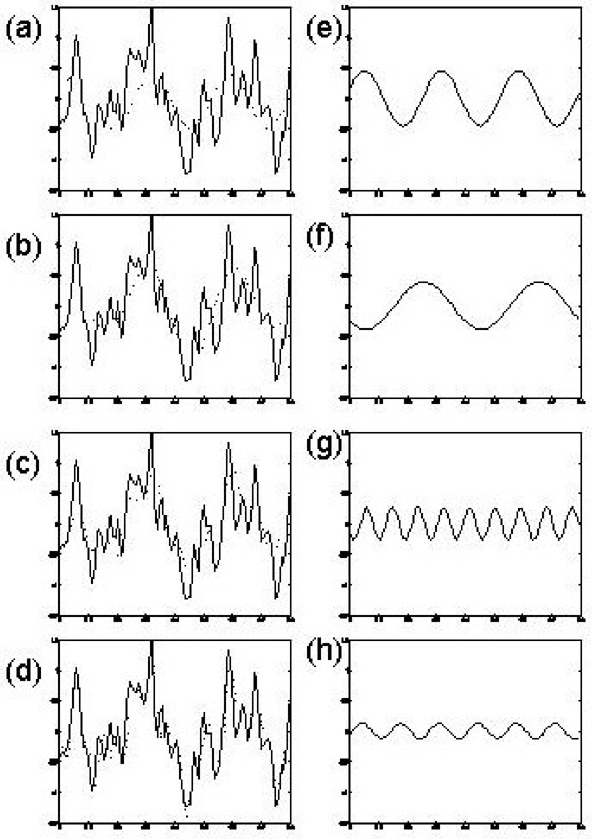


Figure 8.4: *Signal (solid line) and components of the Fourier series approximation* $\sum_{k=1}^{p} R_k cos(2\pi f_k + \Phi_k)$ *(dotted lines) with (a)* $p = 1$, *(b)* $p = 2$, *(c)* $p = 3$ *and (d)* $p = 11$ *where we have ordered the components according to amplitude. The corresponding individual terms are (e)* $R^2 = 0.205, f = 3.75$ *and* $\Phi = 0.437$, *(f)* $R^2 = 0.151$, $f = 2.5$ *and* $\Phi = 0.743$, *(g)* $R^2 = 0.069$, $f = 11.25$ *and* $\Phi = 0.751$ *and (h)* $R^2 = 0.016$, $f = 7.5$ *and* $\Phi = -0.350$.

$$a_0 = \frac{1}{N} \sum_{n=1}^{N} x[n] \tag{8.29}$$

These equations can be derived as follows. To find, for example, a_k, multiply both sides of equation 8.26 by $\cos(2\pi f_k t[n])$ and sum over n. Due to the orthogonality property of sinusoids (which still holds as all frequencies are integer multiples of a base frequency) all terms on the right go to zero except for the one involving a_k. This just leaves $a_k(N/2)$ on the right giving rise to the above formula.

8.3 Complex numbers

8.3.1 Power series

A function of a variable x can often be written in terms of a series of powers of x. For the sin function, for example, we have

$$\sin x = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + ... \tag{8.30}$$

We can find out what the appropriate coefficients are as follows. If we substitite $x = 0$ into the above equation we get $a_0 = 0$ since $sin 0 = 0$ and all the other terms disappear. If we now *differentiate* both sides of the equation and substitute $x = 0$ we get $a_1 = 1$ (because $\cos 0 = 1 = a_1$). Differentiating twice and setting $x = 0$ gives $a_2 = 0$. Continuing this process gives

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + ... \tag{8.31}$$

Similarly, the series representations for $cos x$ and e^x can be found as

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + ... \tag{8.32}$$

and

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + ... \tag{8.33}$$

More generally, for a function $f(x)$ we get the general result

$$f(x) = f(0) + x f'(0) + \frac{x^2}{2!} f''(0) + \frac{x^3}{3!} f'''(0) + ... \tag{8.34}$$

where $f'(0)$, $f''(0)$ and $f'''(0)$ are the first, second and third derivatives of $f(x)$ evaluated at $x = 0$. This expansion is called a *Maclaurin series.*

So far, to calculate the coefficients in the series we have differentiated and substituted $x = 0$. If, instead, we substitute $x = a$ we get

$$f(x) = f(a) + (x-a) f'(a) + \frac{(x-a)^2}{2!} f''(a) + \frac{(x-a)^3}{3!} f'''(a) + ... \tag{8.35}$$

which is called a *Taylor series.*

8.3.2 Complex numbers

Very often, when we try to find the roots of an equation, we may end up with our solution being the square root of a negative number. For example, the quadratic equation

$$ax^2 + bx + c = 0 \tag{8.36}$$

has solutions

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{8.37}$$

If $b^2 - 4ac < 0$ we need the square root of a negative number. To handle this, mathematicians have defined the number

$$i = \sqrt{-1} \tag{8.38}$$

allowing all square roots of negative numbers to be defined in terms of i, eg $\sqrt{-9} = \sqrt{9}\sqrt{-1} = 3i$. These numbers are called *imaginary numbers* to differentiate them from *real numbers.*

Finding the roots of equations, eg. the quadratic equation above, requires us to combine imaginary numbers and real numbers. These combinations are called *complex numbers.* For example, the equation

$$x^2 - 2x + 2 = 0 \tag{8.39}$$

has the solutions $x = 1 + i$ and $x = 1 - i$ which are complex numbers.

A complex number $z = a + bi$ has two components; a real part and an imaginary part which may be written

$$\begin{aligned} a &= Re\{z\} \qquad (8.40) \\ b &= Im\{z\} \end{aligned}$$

The *absolute value* of a complex number is

$$R = Abs\{z\} = \sqrt{a^2 + b^2} \tag{8.41}$$

and the *argument* is

$$\theta = Arg\{z\} = \tan^{-1}\left(\frac{b}{a}\right) \tag{8.42}$$

The two numbers $z = a + bi$ and $z^* = a - bi$ are known as *complex conjugates*; one is the complex conjugate of the other. When multiplied together they form a real number. The roots of equations often come in complex conjugate pairs.

8.3.3 Complex vectors

The transpose, x^T, becomes a 'Hermitian' transpose, x^H, which is the usual transpose but the elements become conjugates. This means that the length of a vector (squared) is now $x^H x$ instead of $x^T x$.

8.3.4 Complex exponentials

If we take the exponential function of an imaginary number and write it out as a series expansion, we get

$$e^{i\theta} = 1 + \frac{i\theta}{1!} + \frac{i^2\theta^2}{2!} + \frac{i^3\theta^3}{3!} + ... \tag{8.43}$$

By noting that $i^2 = -1$ and $i^3 = i^2 i = -i$ and similarly for higher powers of i we get

$$e^{i\theta} = \left[1 - \frac{\theta^2}{2!} + ...\right] + i\left[\frac{\theta}{1!} - \frac{\theta^3}{3!} + ...\right] \tag{8.44}$$

Comparing to the earlier expansions of $\cos\theta$ and $\sin\theta$ we can see that

$$e^{i\theta} = \cos\theta + i\sin\theta \tag{8.45}$$

which is known as *Euler's formula*. Similar expansions for $e^{-i\theta}$ give the identity

$$e^{-i\theta} = \cos\theta - i\sin\theta \tag{8.46}$$

We can now express the sine and cosine functions in terms of complex exponentials

$$\begin{aligned} \cos\theta &= \frac{e^{i\theta} + e^{-i\theta}}{2} \\ \sin\theta &= \frac{e^{i\theta} - e^{-i\theta}}{2i} \end{aligned} \tag{8.47}$$

8.3.5 DeMoivre's theorem

By using the fact that

$$e^{i\theta}e^{i\theta} = e^{i\theta + i\theta} \tag{8.48}$$

(a property of the exponential function and exponents in general eg. $5^3 5^3 = 5^6$) or more generally

$$(e^{i\theta})^k = e^{ik\theta} \tag{8.49}$$

we can write

$$(cos\theta + i\sin\theta)^k = cosk\theta + isink\theta \tag{8.50}$$

which is known as *DeMoivre's theorem.*

8.3.6 Argand diagrams

Any complex number can be represented as a complex exponential

$$a + bi = Re^{i\theta} = R(cos\theta + i\sin\theta) \tag{8.51}$$

and drawn on an *Argand diagram.* Multiplication of complex numbers is equivalent to rotation in the complex plane (due to DeMoivre's Theorem).

$$(a + bi)^2 = R^2 e^{i2\theta} = R^2(cos2\theta + i\sin 2\theta) \tag{8.52}$$

8.4 Discrete Fourier Transform

Fourier series can be expressed in terms of complex exponentials. This representation leads to an efficient method for computing the coefficients. We can write the cosine terms as complex exponentials

$$a_k \cos(2\pi f_k t[n]) = a_k \frac{\exp(i2\pi f_k t[n]) + \exp(-i2\pi f_k t[n])}{2} \tag{8.53}$$

where $i^2 = -1$. Picture this as the addition of two vectors; one above the real axis and one below. Together they make a vector on the real axis which is then halved.

We can also write the sine terms as

$$b_k \sin(2\pi f_k t[n]) = b_k \frac{\exp(i2\pi f_k t[n]) - \exp(-i2\pi f_k t[n])}{2i} \tag{8.54}$$

Picture this as one vector above the real axis minus another vector below the real axis. This results in a purely imaginary (and positive) vector. The result is halved and then multiplied by the vector $\exp(3\pi/2)$ ($-i$, from multplying top and bottom by i) which provides a rotation to the real axis.

Adding them (and moving i to the numerator by multiplying b_k top and bottom by i) gives

$$\frac{1}{2}(a_k - b_k i)\exp(i2\pi f_k t[n]) + \frac{1}{2}(a_k + b_k i)\exp(-i2\pi f_k t[n]) \qquad (8.55)$$

Note that a single term at frequency k has split into a complex combination (the coefficients are complex numbers) of a positive frequency term and a *negative frequency* term. Substituting the above result into equation 8.26 and noting that $f_k t[n] = kn/N$ we get

$$x[n] = a_0 + \frac{1}{2}\sum_{k=1}^{N/2}(a_k - b_k i)\exp(i2\pi kn/N) + \frac{1}{2}\sum_{k=1}^{N/2}(a_k + b_k i)\exp(-i2\pi kn/N) \qquad (8.56)$$

If we now let

$$\tilde{X}(k) = \frac{N}{2}(a_k - b_k i) \qquad (8.57)$$

and note that for real signals $\tilde{X}(-k) = \tilde{X}^*(k)$ (negative frequencies are reflections across the real plane, ie. conjugates) then the $(a_k + b_k i)$ terms are equivalent to $\tilde{X}(-k)$. Hence

$$x[n] = a_0 + \frac{1}{2N}\sum_{k=1}^{N/2}\tilde{X}(k)\exp(i2\pi kn/N) + \frac{1}{2N}\sum_{k=1}^{N/2}\tilde{X}(k)\exp(-i2\pi kn/N) \qquad (8.58)$$

Now, because $\tilde{X}(N-k) = \tilde{X}(-k)$ (this can be shown by considering the Fourier transform of a signal $x[n]$ and using the decomposition $\exp(-i2\pi(N-k)n/N) = \exp(-i2\pi N/N)\exp(i2\pi kn/N)$ where the first term on the right is unity) we can write the second summation as

$$x[n] = a_0 + \frac{1}{2N}\sum_{k=1}^{N/2}\tilde{X}(k)\exp(i2\pi kn/N) + \frac{1}{2N}\sum_{k=N/2}^{N-1}\tilde{X}(k)\exp(-i2\pi(N-k)n/N \qquad (8.59)$$

Using the same exponential decomposition allows us to write

$$x[n] = a_0 + \frac{1}{N}\sum_{k=1}^{N-1}\tilde{X}(k)\exp(i2\pi kn/N) \qquad (8.60)$$

If we now let $X(k+1) = \tilde{X}(k)$ then we can absorb the constant a_0 into the sum giving

$$x[n] = \frac{1}{N} \sum_{k=1}^{N} X(k) \exp(i 2\pi (k-1)n/N) \tag{8.61}$$

which is known as the *Inverse Discrete Fourier Transform* (IDFT). The terms $X(k)$ are the *complex valued Fourier coefficients*. We have the relations

$$\begin{aligned} a_0 &= Re\{X(1)\} \qquad (8.62)\\ a_k &= \frac{2}{N} Re\{X(k+1)\} \\ b_k &= \frac{-2}{N} Im\{X(k+1)\} \end{aligned}$$

The complex valued Fourier coefficients can be computed by first noting the orthogonality relations

$$\sum_{n=1}^{N} \exp(i2\pi(k-1)n/N) \;=\; \begin{array}{ll} N & k = 1, \pm(N+1), \pm(N+2) \\ 0 & otherwise \end{array}$$

If we now multiply equation 8.61 by $\exp(-i2\pi ln/N)$, sum from 1 to N and re-arrange we get

$$\begin{aligned} X(k) &= \sum_{n=1}^{N} x(n) \exp(-i2\pi(k-1)n/N) \qquad (8.63) \\ &= DFT(x) \end{aligned}$$

which is the Discrete Fourier Transform (DFT).

8.4.1 Power Spectral Density

The power in a signal is given by

$$P_x = \sum_{n=1}^{N} |x[n]|^2 \tag{8.64}$$

We now derive an expression for P_x in terms of the Fourier coefficients. If we note that $|x[n]|$ can also be written in its conjugate form (the conjugate form has the same magnitude; the phase is different but this does'nt matter as we're only interested in magnitude)

$$|x[n]| = \frac{1}{N} \sum_{k=1}^{N} X^*(k) \exp(-i2\pi(k-1)n/N) \tag{8.65}$$

then we can write the power as

$$P_x = \sum_{n=1}^{N} |x[n] \frac{1}{N} \sum_{k=1}^{N} X^*(k) \exp(-i2\pi(k-1)n/N)| \tag{8.66}$$

If we now change the order of the summations we get

$$P_x = \frac{1}{N} \sum_{k=1}^{N} |X^*(k) \sum_{n=1}^{N} x(n) \exp(-i2\pi(k-1)n/N)| \tag{8.67}$$

where the sum on the right is now equivalent to $X(k)$. Hence

$$P_x = \frac{1}{N} \sum_{k=1}^{N} |X(k)|^2 \tag{8.68}$$

We therefore have an equivalence between the power in the time domain and the power in the frequency domain which is known as *Parseval's relation*. The quantity

$$P_x(k) = |X(k)|^2 \tag{8.69}$$

is known as the *Power Spectral Density* (PSD).

8.4.2 Filtering

The filtering process

$$x[n] = \sum_{l=-\infty}^{\infty} x_1(l) x_2(n-l) \tag{8.70}$$

is also known as *convolution*

$$x[n] = x_1(n) * x_2(n) \tag{8.71}$$

We will now see how it is related to frequency domain operations. If we let $w = 2\pi(k-1)/N$, multiply both sides of the above equation by $\exp(-iwn)$ and sum over n the left hand side becomes the Fourier transform

$$X(w) = \sum_{n=-\infty}^{\infty} x[n] \exp(-iwn) \tag{8.72}$$

and the right hand side (RHS) is

$$\sum_{n=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x_1(l) x_2(n-l) \exp(-iwn) \tag{8.73}$$

Now, we can re-write the exponential term as follows

$$\exp(-iwn) = \exp(-iw(n-l)) \exp(-iwl) \tag{8.74}$$

Letting $n' = n - l$, we can write the RHS as

$$\sum_{l=-\infty}^{\infty} x_1(l) \exp(-iwl) \sum_{n'=-\infty}^{\infty} x_2(n') \exp(-iwn') = X_1(w) X_2(w) \tag{8.75}$$

Hence, the filtering operation is equivalent to

$$X(w) = X_1(w) X_2(w) \tag{8.76}$$

which means that convolution in the time domain is equivalent to multiplication in the frequency domain. This is known as the *convolution theorem.*

8.4.3 Autocovariance and Power Spectral Density

The autocovariance of a signal is given by

$$\sigma_{xx}(n) = \sum_{l=-\infty}^{\infty} x(l) x(l-n) \tag{8.77}$$

Using the same method that we used to prove the convolution theorem, but noting that the term on the right is $x(l-n)$ not $x(n-l)$ we can show that the RHS is equivalent to

$$X(w) X(-w) = |X(w)|^2 \tag{8.78}$$

which is the Power Spectral Density, $P_x(w)$. Combining this with what we get for the left hand side gives

$$P_x(w) = \sum_{n=-\infty}^{\infty} \sigma_{xx}(n) \exp(-iwn) \tag{8.79}$$

which means that the PSD is the Fourier Transform of the autocovariance. This is known as the *Wiener-Khintchine Theorem.* This is an important result. It means that the PSD can be estimated from the autocovariance and vice-versa. It also means that the PSD and the autocovariance contain the same information about the signal.

It is also worth noting that since both contain no information about the phase of a signal then the signal cannot be uniquely constructed from either. To do this we need to know the PSD *and* the *Phase spectrum* which is given by

$$\Phi(k) = \tan^{-1}(\frac{b_k}{a_k}) \tag{8.80}$$

where b_k and a_k are the real Fourier coefficients.

We also note that the Fourier transform of a symmetric function is real. This is because symmetric functions can be represented entirely by cosines, which are themselves symmetric; the sinewaves, which constitute the complex component of a Fourier series, are no longer necessary. Therefore, because the autocovariance is symmetric the PSD is real.

8.4.4 The Periodogram

The periodogram of a signal x_t is a plot of the normalised power in the kth harmonic versus the frequency, f_k of the kth harmonic. It is calculated as

$$\begin{aligned} I(f_k) &= \frac{T_s}{N}(a_k^2 + b_k^2) \qquad (8.81) \\ &= \frac{T_s}{N}|\sum_{n=1}^{N} x(n)\exp(-i2\pi(k-1)n/N)|^2 \end{aligned}$$

where a_k and b_k are the Fourier coefficients.

The periodogram is a low bias (actually unbiased) but high variance [2] estimate of the power at a given frequency. This is therefore a problem if the number of data points is small; the estimated spectrum will be very spiky.

[2] It is an *inconsistent* estimator, because the variance does'nt reduce to zero as the number of samples tends to infinity.

To overcome this, a number of algorithms exist to smooth the periodogram ie. to reduce the variance. The Bartlett method, for example, takes an N-point sequence and subdivides it into K nonoverlapping segments and calculates $I(f_k)$ for each. The final periodogram is just the average over the K estimates. This results in a reduction in variance by a factor K at the cost of reduced spectral resolution (by a factor K).

8.4.5 Modified periodograms

Other methods modify the periodogram by using a time domain window, h, also known as a data taper, such that the modified periodogram is given by

$$\begin{aligned} I(f_k) &= \frac{T_s}{N}(a_k^2 + b_k^2) \qquad (8.82) \\ &= \frac{T_s}{N}|\sum_{n=1}^{N} h(n)x(n)\exp(-i2\pi(k-1)n/N)|^2 \end{aligned}$$

For example, the matlab `psd` function chooses h to be a Hanning window.

Welch's method (`pwelch.m` in matlab) takes overlapping segments, and averages modified periodograms.

8.4.6 Multi-tapering

This uses multiple data tapers. The tapers are orthogonal to each other. Given a length N sequence

- Choose the desired spectral resolution, W, eg. $\pm 2Hz$
- Get corresponding Slepian sequences, h_i, $i = 1..2NW - 1$. These provide an eigenbasis in local frequency space ($\pm W$) for finite length data sequences.
- Sum modified periodograms

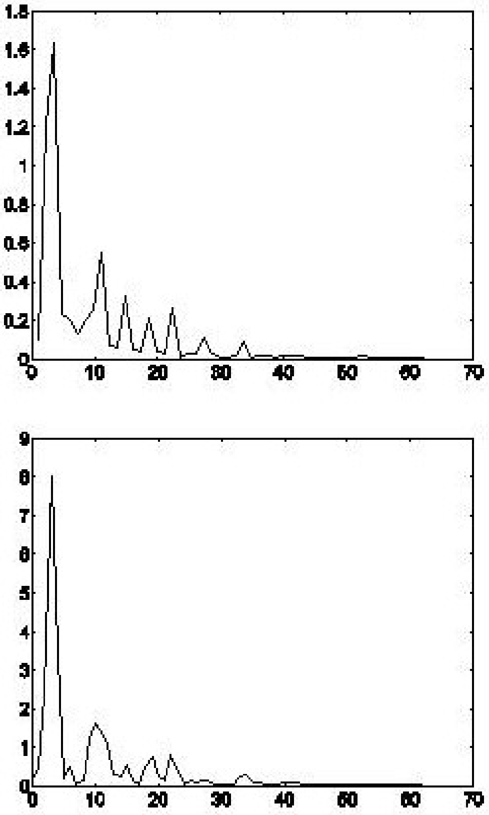


Figure 8.5: *Periodogram (top) and Welch's modified Periodogram (bottom).*

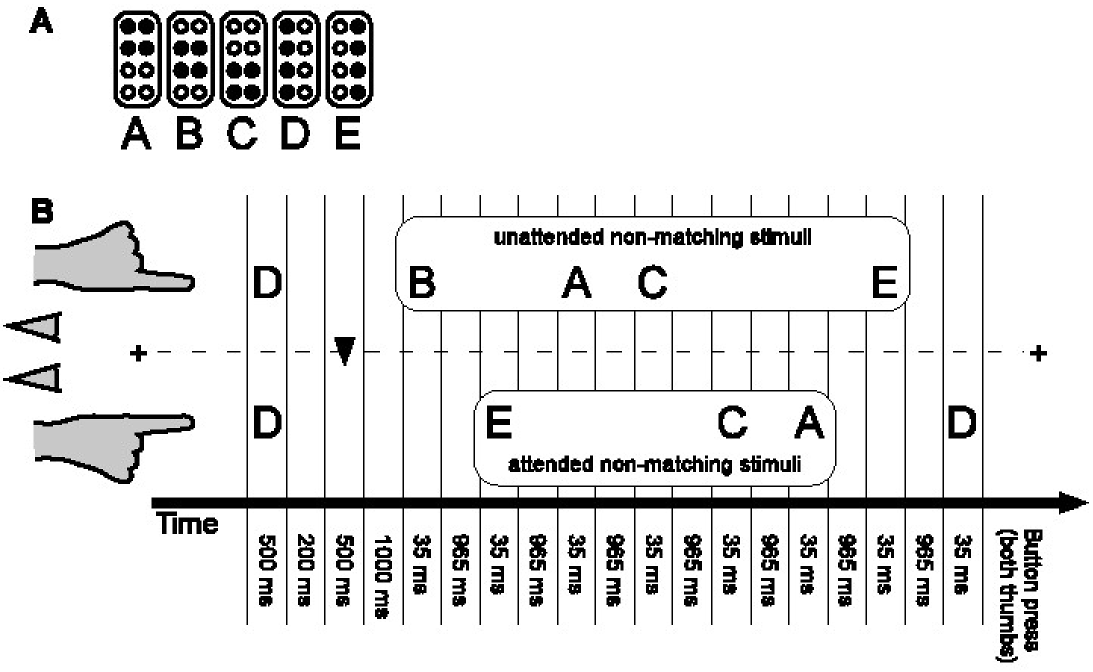


Figure 8.6: Delayed match-to-sample task.

This is implemented in eg. matlab's `pmtm.m` function. For theory see [40], and for application to brain imaging data see [31].

See eg. `sunspot_spectra.m`.

8.4.7 MEG data

Bauer et al. [4] estimate spectral density of MEG data during a delayed match-to-sample task. For each subject they computed spectra during periods when non-matching stimuli were correctly rejected. Spectra in range 40-180Hz, were computed using a multitaper method with 200ms windows and a spectral resolution of $W = 10Hz$.

They then computed z-scores for each time frequency bin by comparing with a baseline period. These z-scores were then averaged over subject.

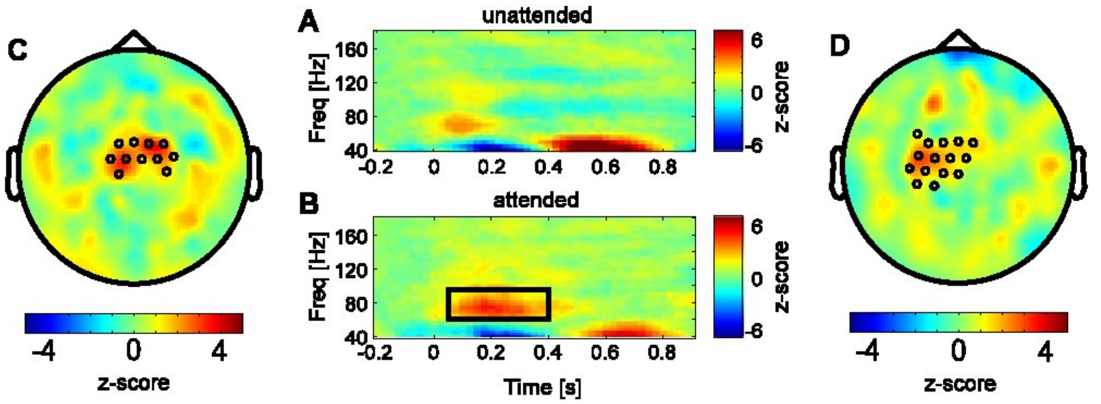


Figure 8.7: Spectra for attended and unattended side, averaged over electrode set shown in (C) for left finger stimulation and (B) right finger stimulation.

8.5 Multiple time series

8.5.1 Cross-correlation

Given *two* time series x_t and y_t we can delay x_t by T samples and then calculate the *cross-covariance* between the pair of signals. That is

$$\sigma_{xy}(T) = \frac{1}{N-1} \sum_{t=1}^{N} (x_{t-T} - \mu_x)(y_t - \mu_y) \tag{8.83}$$

where μ_x and μ_y are the means of each time series and there are N samples in each. The function $\sigma_{xy}(T)$ is the *cross-covariance* function. The *cross-correlation* is a normalised version

$$r_{xy}(T) = \frac{\sigma_{xy}(T)}{\sqrt{\sigma_{xx}(0)\sigma_{yy}(0)}} \tag{8.84}$$

where we note that $\sigma_{xx}(0) = \sigma_x^2$ and $\sigma_{yy}(0) = \sigma_y^2$ are the variances of each signal. Note that

$$r_{xy}(0) = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{8.85}$$

which is the correlation between the two variables. Therefore unlike the autocorrelation, r_{xy} is not, generally, equal to 1.

The cross-correlation is a normalised cross-covariance which, assuming zero mean signals, is given by

$$\sigma_{xy}(T) = < x_{t-T} y_t > \tag{8.86}$$

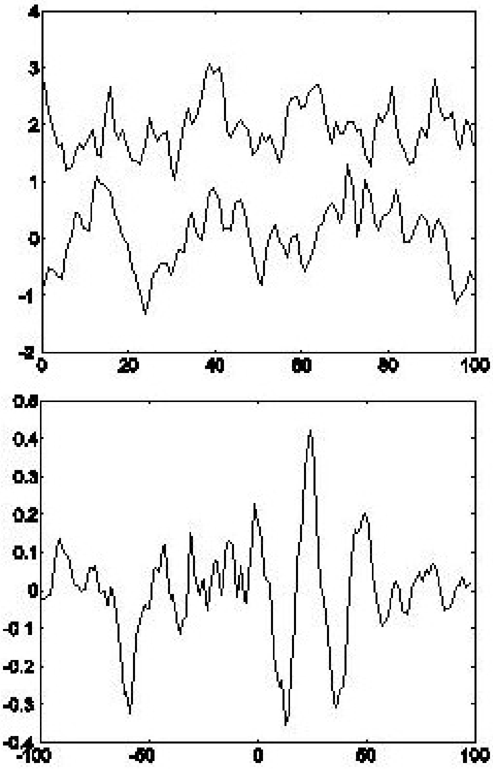


Figure 8.8: *Top plot: Signals x_t (top curve) and y_t (bottom curve). Bottom plot: Cross-correlation function $r_{xy}(T)$. A lag of T denotes the top series, x, lagging the bottom series, y. Notice the big positive correlation at a lag of 25.*

and for negative lags

$$\sigma_{xy}(-T) = < x_{t+T} y_t > \tag{8.87}$$

Subtracting T from the time index now gives

$$\sigma_{xy}(-T) = < x_t y_{t-T} > \tag{8.88}$$

which is different to $\sigma_{xy}(T)$. To see this more clearly we can subtract T once more from the time index to give

$$\sigma_{xy}(-T) = < x_{t-T} y_{t-2T} > \tag{8.89}$$

Hence, the cross-covariance, and therefore the cross-correlation, is an *asymmetric* function (the autocorrelation is symettric).

To summarise: moving signal A right (forward in time) and multiplying with signal B is not the same as moving signal A left and multiplying with signal B; unless signal A equals signal B.

8.5.2 Cross Spectral Density

Just as the Power Spectral Density (PSD) is the Fourier transform of the auto-covariance function we may define the Cross Spectral Density (CSD) as the Fourier transform of the cross-covariance function

$$P_{12}(w) = \sum_{n=-\infty}^{\infty} \sigma_{x_1 x_2}(n) \exp(-iwn) \tag{8.90}$$

Note that if $x_1 = x_2$, the CSD reduces to the PSD. Now, the cross-covariance of a signal is given by

$$\sigma_{x_1 x_2}(n) = \sum_{l=-\infty}^{\infty} x_1(l) x_2(l-n) \tag{8.91}$$

Substituting this into the earlier expression gives

$$P_{12}(w) = \sum_{n=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x_1(l) x_2(l-n) \exp(-iwn) \tag{8.92}$$

By noting that

$$\exp(-iwn) = \exp(-iwl) \exp(iwk) \tag{8.93}$$

where $k = l - n$ we can see that the CSD splits into the product of two integrals

$$P_{12}(w) = X_1(w) X_2(-w) \tag{8.94}$$

where

$$X_1(w) = \sum_{l=-\infty}^{\infty} x_1(l) \exp(-iwl) \tag{8.95}$$

$$X_2(-w) = \sum_{k=-\infty}^{\infty} x_2(k) \exp(+iwk)$$

For real signals $X_2^*(w) = X_2(-w)$ where * denotes the complex conjugate. Hence, the cross spectral density is given by

$$P_{12}(w) = X_1(w) X_2^*(w) \tag{8.96}$$

This means that the CSD can be evaluated in one of two ways (i) by first estimating the cross-covariance and Fourier transforming or (ii) by taking the Fourier transforms of each signal and multiplying (after taking the conjugate of one of them). A number of algorithms exist which enhance the spectral estimation ability of each method. These algorithms are basically extensions of the algorithms for PSD estimation, for example, for type (i) methods we can perform Blackman-Tukey windowing of the cross-covariance function and for type (ii) methods we can employ Welch's algorithm for averaging modified periodograms before multiplying the transforms. See Carter [8] for more details.

The CSD is complex because the cross-covariance is asymmetric (the PSD is real because the auto-covariance is symmetric; in this special case the Fourier transorm reduces to a cosine transform).

8.5.3 PSD matrix

The frequency domain characteristics of a multivariate time-series (eg. two or more) may be summarised by the power spectral density *matrix* (Marple, 1987[28]; page 387). For d time series

$$\boldsymbol{P}(f) = \begin{pmatrix} P_{11}(f) & P_{12}(f) & \cdots & P_{1d}(f) \\ P_{12}(f) & P_{22}(f) & \cdots & P_{2d}(f) \\ \multicolumn{4}{c}{\dots\dots\dots\dots\dots\dots\dots\dots\dots} \\ P_{1d}(f) & P_{2d}(f) & \cdots & P_{dd}(f) \end{pmatrix} \tag{8.97}$$

where the diagonal elements contain the spectra of individual channels and the off-diagonal elements contain the cross-spectra. The matrix is called a *Hermitian matrix* because the elements are complex numbers.

8.5.4 Coherence and Phase

The *complex coherence function* is given by (Marple 1987; p. 390)

$$r_{ij}(f) = \frac{P_{ij}(f)}{\sqrt{P_{ii}(f)}\sqrt{P_{jj}(f)}} \tag{8.98}$$

The coherence, or *mean squared coherence* (MSC), between two channels is given by

$$r_{ij}^2(f) = \mid r_{ij}(f) \mid^2 \tag{8.99}$$

The phase spectrum, between two channels is given by

$$\theta_{ij}(f) = tan^{-1}\left[\frac{Im(r_{ij}(f))}{Re(r_{ij}(f))}\right] \tag{8.100}$$

The MSC measures the linear correlation between two time series at each frequency and is directly analagous to the squared correlation coefficient in linear regression. As such the MSC is intimately related to *linear filtering*, where one signal is viewed as a filtered version of the other. This can be interpreted as a linear regression at each frequency. The optimal regression coefficient, or linear filter, is given by

$$H(f) = \frac{P_{xy}(f)}{P_{xx}(f)} \tag{8.101}$$

This is analagous to the expression for the regression coefficient $a = \sigma_{xy}/\sigma_{xx}$ (see first lecture). The MSC is related to the optimal filter as follows

$$r^2_{xy}(f) = |H(f)|^2 \frac{P_{xx}(f)}{P_{yy}(f)} \tag{8.102}$$

which is analagous to the equivalent expression in linear regression $r^2 = a^2(\sigma_{xx}/\sigma_{yy})$.

At a given frequency, if the phase of one signal is *fixed* relative to the other, then the signals can have a high coherence at that frequency. This holds even if one signal is entirely out of phase with the other (note that this is different from adding up signals which are out of phase; the signals cancel out. We are talking about the coherence *between* the signals).

At a given frequency, if the phase of one signal changes relative to the other then the signals will not be coherent at that frequency. The time over which the phase relationship is constant is known as the *coherence time*. See [**?**], for an example.

8.5.5 Welch's method for estimating coherence

Algorithms based on Welch's method (such as the cohere function in the matlab system identification toolbox) are widely used [8] [43]. The signal is split up into a number of segments, N, each of length T and

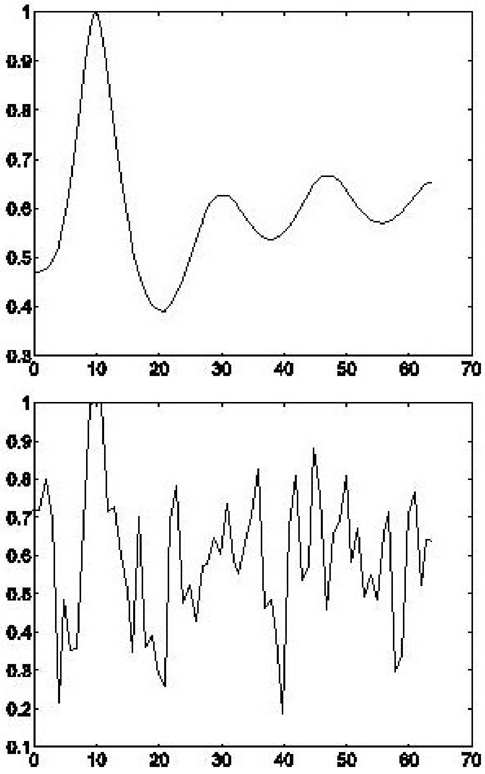


Figure 8.9: *Coherence estimates using (bottom) Welch's method and (top) MAR model*

the segments may be overlapping. The complex coherence estimate is then given as

$$\hat{r}_{ij}(f) = \frac{\sum_{n=1}^{N} X_i^n(f)(X_j^n(f))^*}{\sqrt{\sum_{n=1}^{N} X_i^n(f)^2}\sqrt{\sum_{n=1}^{N} X_j^n(f)^2}} \tag{8.103}$$

where n sums over the data segments. This equation is exactly the same form as for estimating correlation coefficients (see chapter 1). Note that if we have only $N = 1$ data segment then the estimate of coherence will be 1 regardless of what the true value is (this would be like regression with a single data point). Therefore, we need a number of segments.

Note that this only applies to Welch-type algorithms which compute the CSD from a product of Fourier transforms. We can trade-off good spectral resolution (requiring large T) with low-variance estimates of coherence (requiring large N and therefore small T). We can also overlap segments.

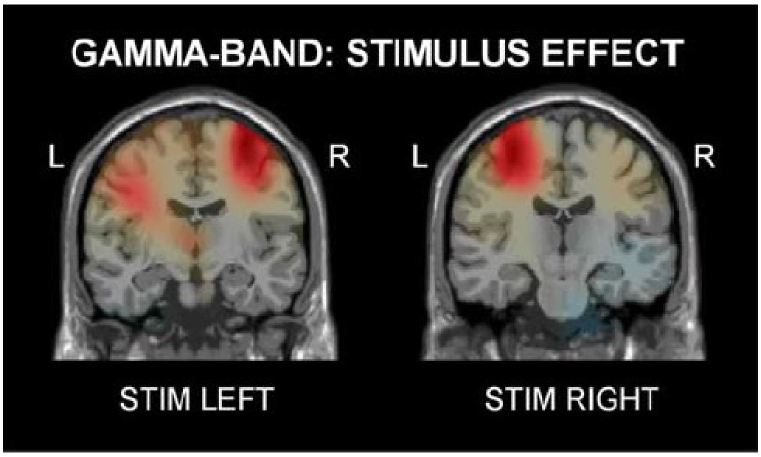


Figure 8.10: Source analysis of gamma power (60-95Hz) versus baseline

8.6 Source reconstruction of MEG Gamma activity

Specify frequency-domain model, and reconstruct using maximum likelihood estimator

$$\begin{aligned} y_f &= Lr_f + e_f \qquad (8.104) \\ \hat{r}_f &= (L^T P_f^{-1} L)^{-1} L^T P_f^{-1} y_f \end{aligned}$$

where L is the lead-field and P_f is the PSD matrix over sensors computed for time-period of interest.

This can be augmented to a Bayesian estimator in the usual way. This leads to the Dynamic Imaging of Coherent sources (DICs) algorithm [19].

We can write the reconstructed activity as

$$\hat{r}_f = w_{rf} y_f \qquad (8.105)$$

The power is then given by

$$p_{rf} = w_{rf}^* P_f w_{rf} \qquad (8.106)$$

8.7 Further topics

- AR/MAR - parametric

- Subspace methods - small number of modes
- Wavelets - nonstationarity

Chapter 9

Approximate Bayesian Inference

9.1 Contents

- Laplace approximation
- Kullback-Liebler divergence
- Variational Bayes
- Application: Single subject fMRI with GLM-AR models
- Expectation Maximisation
- Mixture models
- Application: Identifying degenerate systems

9.2 Laplace approximation

Laplace's method approximates the integral of a function $\int f(\theta)d\theta$ by fitting a Gaussian at the maximum $\hat{\theta}$ of $f(\theta)$, and computing the volume of the Gaussian. The covariance of the Gaussian is determined by the Hessian matrix of $\log f(\theta)$ at the maximum point $\hat{\theta}$ [25].

The term 'Laplace approximation' is used for the method of approximating a posterior distribution with a Gaussian centered at the Maximum a Posterior (MAP) estimate. This is the application of Laplace's method with $f(\theta) = p(Y|\theta)p(\theta)$.

9.3 Kullback-Liebler divergence

For densities $q(\theta)$ and $p(\theta)$ the Relative Entropy or Kullback-Liebler (KL) divergence from q to p is [11]

$$KL[q||p] = \int q(\theta) \log \frac{q(\theta)}{p(\theta)} d\theta \tag{9.1}$$

The KL-divergence satisfies the Gibb's inequality [26]

$$KL[q||p] \geq 0 \tag{9.2}$$

with equality only if $q = p$. In general $KL[q||p] \neq KL[p||q]$, so KL is not a distance measure.

9.4 Variational Bayes

Given a probabilistic model of some data, the log of the 'evidence' or 'marginal likelihood' can be written as

$$\begin{aligned} \log p(Y) &= \int q(\theta) \log p(Y) d\theta \\ &= \int q(\theta) \log \frac{p(Y,\theta)}{p(\theta|Y)} d\theta \\ &= \int q(\theta) \log \left[\frac{p(Y,\theta) q(\theta)}{q(\theta) p(\theta|Y)} \right] d\theta \\ &= F + KL(q(\theta)||p(\theta|Y)) \end{aligned} \tag{9.3}$$

where $q(\theta)$ is considered, for the moment, as an arbitrary density. We have

$$F = \int q(\theta) \log \frac{p(Y,\theta)}{q(\theta)} d\theta, \tag{9.4}$$

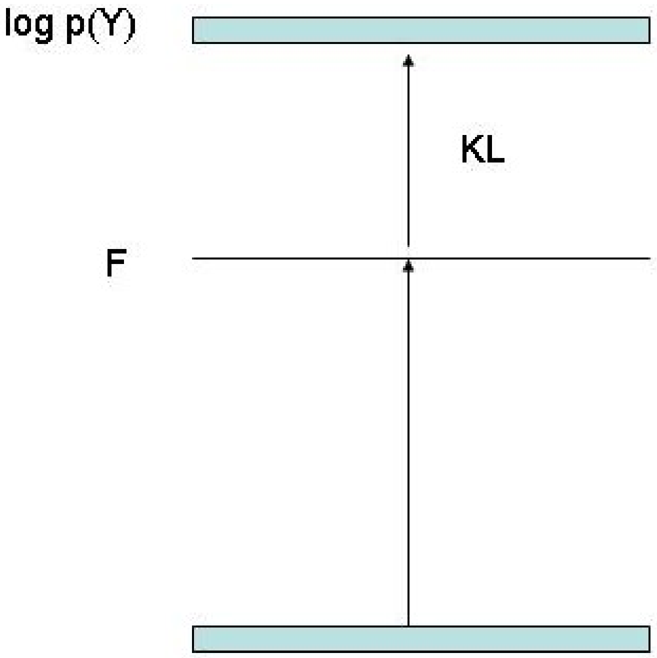


Figure 9.1: *The negative variational free energy, F, provides a lower bound on the log-evidence of the model with equality when the approximate posterior equals the true posterior.*

which in statistical physics is known as the *negative* variational free energy. The second term in equation 9.3 is the KL-divergence between the density $q(\theta)$ and the true posterior $p(\theta|Y)$. Equation 9.3 is the fundamental equation of the VB-framework and is shown graphically in Figure 9.1. Because KL is always positive, due to the Gibbs inequality, F provides a lower bound on the model evidence. Moreover, because KL is zero when two densities are the same, F will become equal to the model evidence when $q(\theta)$ is equal to the true posterior. For this reason $q(\theta)$ can be viewed as an *approximate posterior*.

9.4.1 Example

The solid lines in Figure 9.2 show a posterior distribution p which is a Gaussian mixture density comprising two modes.

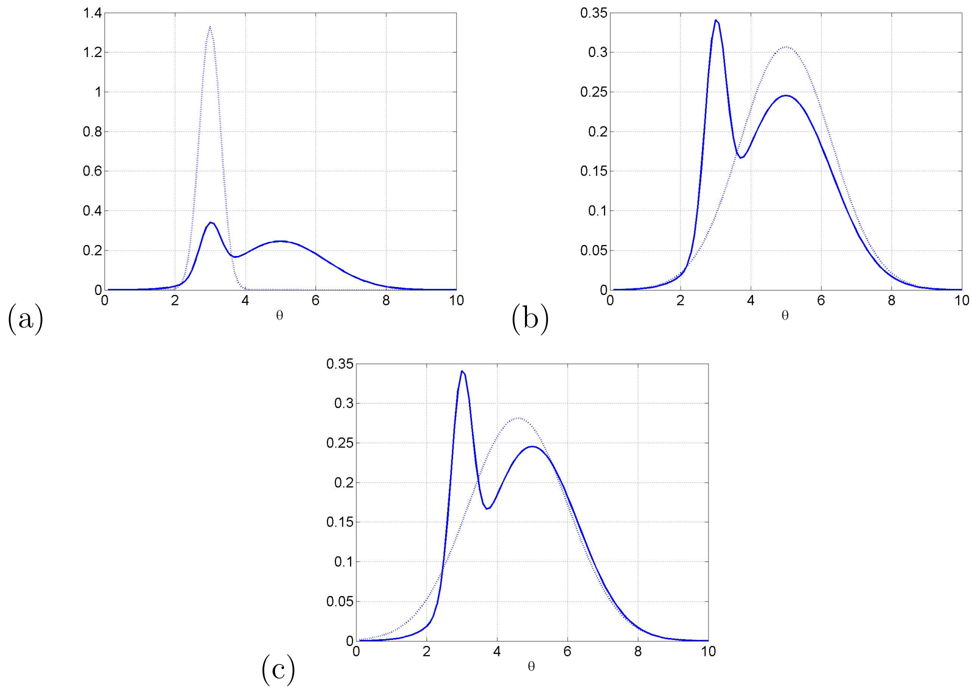


Figure 9.2: *Probability densities* $p(\theta)$ *(solid lines) and* $q(\theta)$ *(dashed lines) for a Gaussian mixture* $p(\theta) = 0.2 \times \mathsf{N}(m_1, \sigma_1^2) + 0.8 \times \mathsf{N}(m_2, \sigma_2^2)$ *with* $m_1 = 3, m_2 = 5, \sigma_1 = 0.3, \sigma_2 = 1.3$*, and a single Gaussian* $q(\theta) = \mathsf{N}(\mu, \sigma^2)$ *with (a)* $\mu = \mu_1, \sigma = \sigma_1$ *which fits the first mode, (b)* $\mu = \mu_2, \sigma = \sigma_2$ *which fits the second mode and (c)* $\mu = 4.6, \sigma = 1.4$ *which is moment-matched to* $p(\theta)$*.*

The first contains the Maximimum A Posteriori (MAP) value and the second contains the majority of the probability mass.

The Laplace approximation to p is therefore given by a Gaussian centred around the first, MAP mode. This is shown in Figure 9.2(a).

Figure 9.2(b) shows a Laplace approximation to the second mode, which could arise if MAP estimation found a local, rather than a global, maximum. Finally, Figure 9.2(c) shows the minimum KL-divergence approximation, assuming that q is a Gaussian. This is a fixed-form VB approximation, as we have fixed the form of the approximating density (ie. q is a Gaussian). This VB solution corresponds to a density q which is moment matched to p.

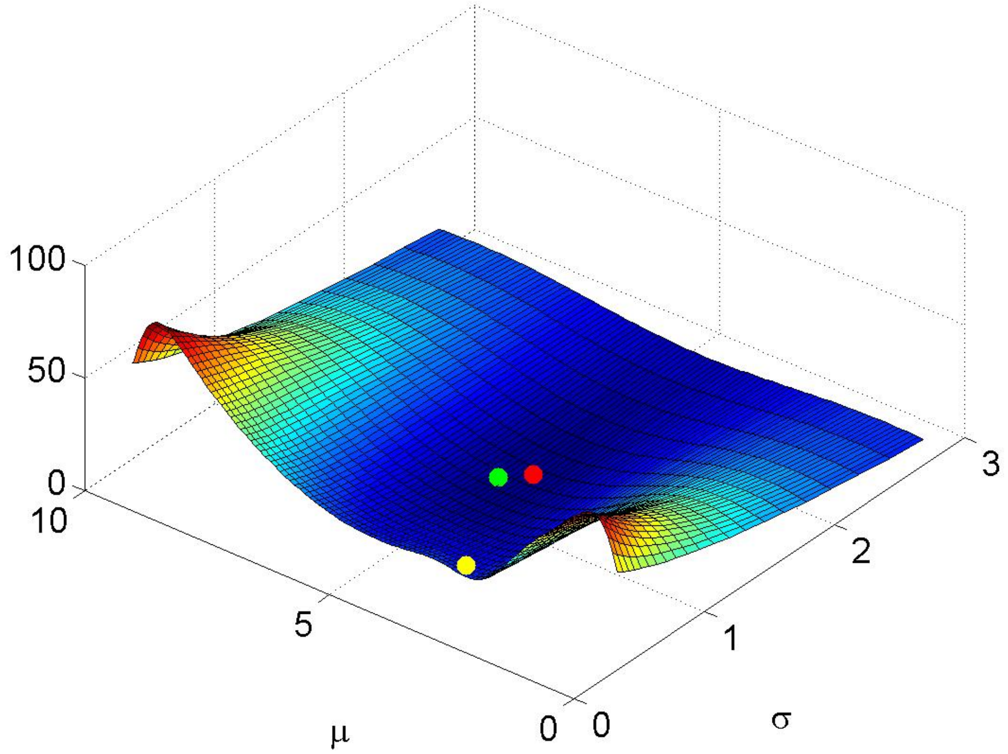


Figure 9.3: *KL-divergence, $KL(q||p)$ for p as defined in Figure 9.2 and q being a Gaussian with mean μ and standard deviation σ. The KL-divergences of the approximations in Figure 9.2 are (a) 11.73 for the first mode (yellow ball), (b) 0.93 for the second mode (green ball) and (c) 0.71 for the moment-matched solution (red ball).*

Figure 9.3 plots $KL[q||p]$ as a function of the mean and standard deviation of q, showing a minimum around the moment-matched values. These KL values were computed by discretising p and q and approximating equation 9.1 by a discrete sum. The MAP mode, maximum mass mode and moment-matched solutions have $KL[q||p]$ values of 11.7, 0.93 and 0.71 respectively. This shows that low KL is achieved when q captures most of the probability mass of p and, minimum KL when q is moment-matched to p. The figure also shows that, for reasonable values of the mean and standard deviation, there are no local minima. This is to be contrasted with the posterior distribution itself which has two maxima, one local and one global.

This example provides a good motivation for VB. But in higher dimensions due to (i) nature of KL and (ii) factorisations (see later) VB is not so optimal. See Minka [30] and Mackay [26] for further details.

9.4.2 Nonlinear functions of parameters

Capturing probability mass is particularly important if one is interested in nonlinear functions of parameter values, such as model predictions. Figures 9.4 and 9.5 show histograms of model predictions for squared and logistic-map functions indicating that VB predictions are qualitatively better than those from the Laplace approximation.

Often in Bayesian inference, one quotes posterior exceedance probabilities. For the squared function, Laplace says 5% of samples are above $g = 12.2$. But in the true density, 71% of samples are. For the logisitic function 62% are above Laplace's 5% point. The percentage of samples above VB's 5% points

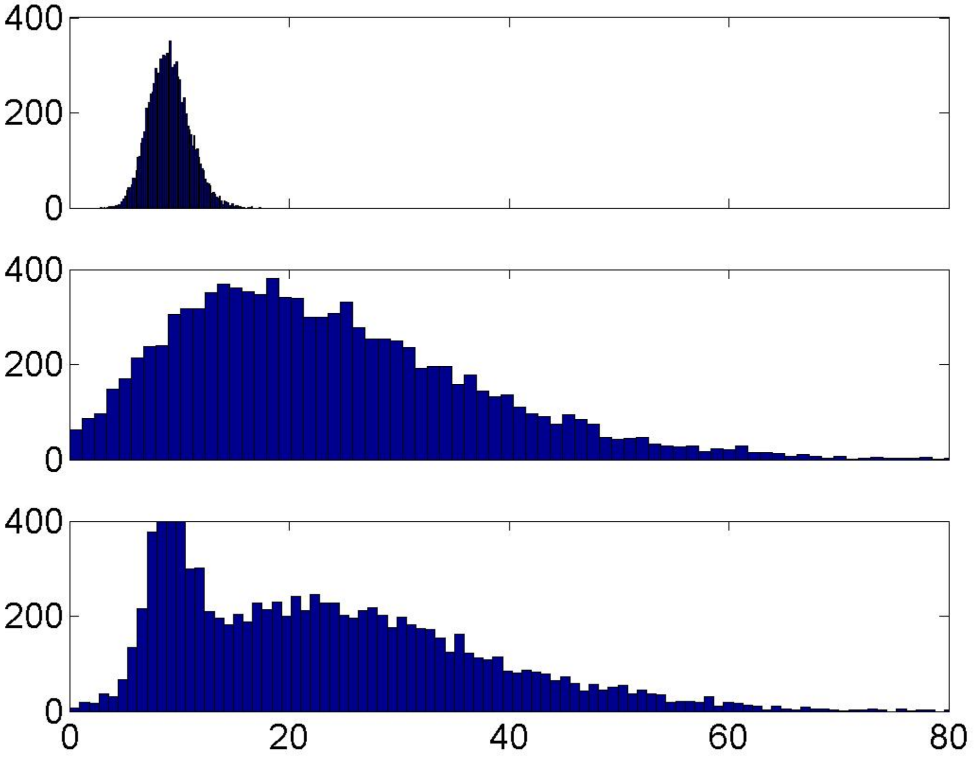


Figure 9.4: *Histograms of 10,000 samples drawn from $g(\theta)$ where the distribution over θ is from the Laplace approximation (top), VB approximation (middle) and true distribution, p, (bottom) for $g(\theta) = \theta^2$.*

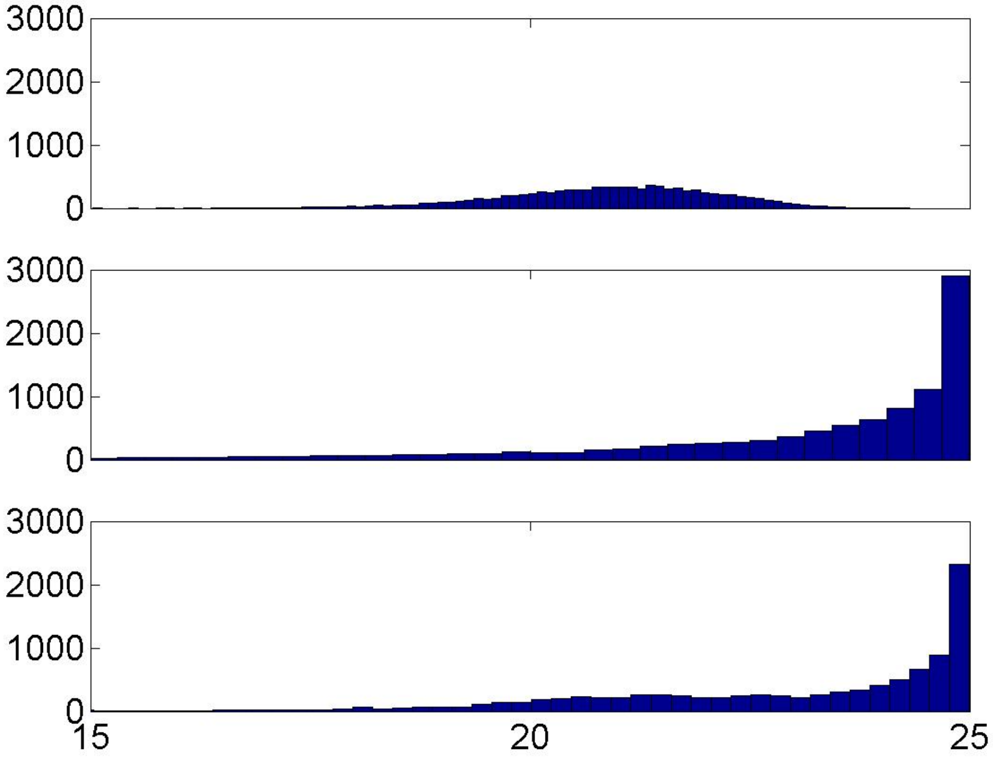


Figure 9.5: *Histograms of 10,000 samples drawn from $g(\theta)$ where the distribution over θ is from the Laplace approximation (top), VB approximation (middle) and true distribution, p, (bottom) for $g(\theta) = \theta * (10 - \theta)$. This is akin to a logistic map function encountered in dynamical systems [32].*

are 5.1% for the squared function and 4.2% for the logistic-map function. So for this example, Laplace can tell you the posterior exceedance probability is 5% when, in reality it is an order of magnitude greater. This is not the case for VB.

9.4.3 Factorised Approximations

To obtain a practical learning algorithm we must also ensure that the integrals in F are tractable. One generic procedure for attaining this goal is to assume that the approximating density factorizes over groups of parameters. In physics, this is known as the mean field approximation. Thus, we consider:

$$q(\theta) = \prod_i q(\theta_i) \tag{9.5}$$

where θ_i is the ith group of parameters. We can also write this as

$$q(\theta) = q(\theta_i) q(\theta_{\backslash i}) \tag{9.6}$$

where $\theta_{\backslash i}$ denotes all parameters *not* in the ith group. The distributions $q(\theta_i)$ which maximise F can then be shown to be

$$q(\theta_i) = \frac{\exp[I(\theta_i)]}{Z} \tag{9.7}$$

where Z is the normalisation factor needed to make $q(\theta_i)$ a valid probability distribution and

$$I(\theta_i) = \int q(\theta_{\backslash i}) \log p(Y, \theta) d\theta_{\backslash i} \tag{9.8}$$

For proof see [36].

9.4.4 Model Inference

As we have seen earlier, the negative free energy, F, is a lower bound on the model evidence. If this bound is tight then F can be used as a surrogate for the model evidence and so allow for Bayesian model selection and averaging. Earlier, the negative free energy was written

$$F(m) = \int q(\theta|m) \log \frac{p(Y, \theta|m)}{q(\theta|m)} d\theta \tag{9.9}$$

By using $p(Y, \theta|m) = p(Y|\theta, m)p(\theta|m)$ we can express it as the sum of two terms

$$F(m) = \int q(\theta|m) \log p(Y|\theta, m) d\theta - KL[q(\theta|m)||p(\theta|m)] \tag{9.10}$$

where the first term is the average likelihood of the data and the second term is the KL between the approximating posterior and the *prior*.

9.4.5 KL for Gaussians

The KL divergence for Normal densities $q(x) = \mathsf{N}(\mu_q, \Sigma_q)$ and $p(x) = \mathsf{N}(\mu_p, \Sigma_p)$ is

$$\begin{aligned} KL_N(\mu_q, \Sigma_q; \mu_p, \Sigma_p) &= 0.5 \log \frac{|\Sigma_p|}{|\Sigma_q|} + 0.5Tr(\Sigma_p^{-1}\Sigma_q) \quad (9.11) \\ &+ 0.5(\mu_q - \mu_p)^T \Sigma_p^{-1} (\mu_q - \mu_p) - \frac{d}{2} \end{aligned}$$

where $|\Sigma_p|$ denotes the determinant of the matrix Σ_p. The KL will tend to increase with the dimension of x.

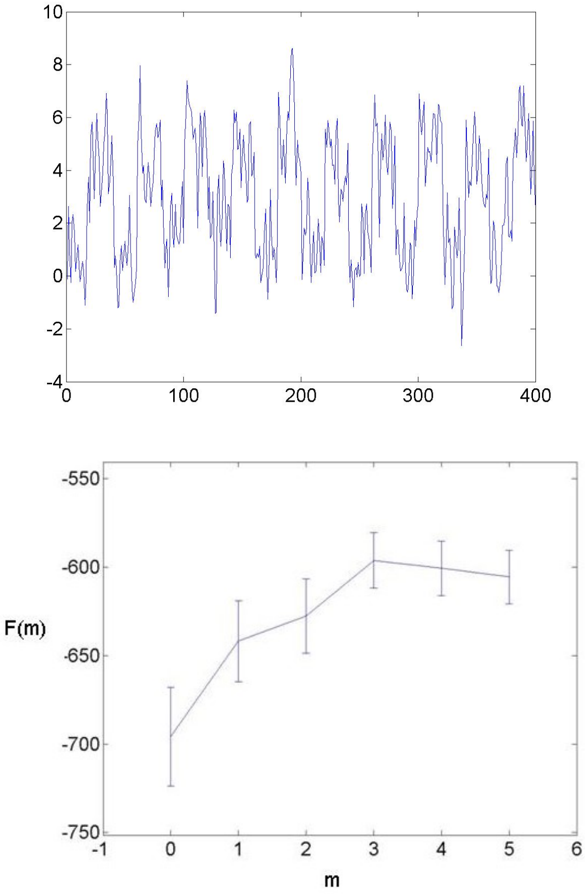
9.5 Single-subject fMRI: GLM-AR models

We generated data from a GLM-AR model having two regression coefficients and three autoregressive coefficients

$$y_t = x_t w + e_t \tag{9.12}$$

$$e_t = \sum_{j=1}^{m} a_j e_{t-j} + z_t \tag{9.13}$$

where x_t is a two-element row vector, the first element flipping between a '-1' and '1' with a period of 40 scans (ie. 20 -1's followed by 20 1's) and the second element being '1' for all t. The two corresponding entries in w reflect the size of the activation, $w_1 = 2$, and the mean signal level, $w_2 = 3$. We used an AR(3) model for the errors with parameters $a_1 = 0.8$, $a_2 = -0.6$ and $a_3 = 0.4$. See [38] for further details.



(a)

(b)

Figure 9.6: The figures show (a) an example time series from a GLM-AR model with AR model order $m = 3$ and (b) a plot of the average negative free energy $F(m)$, with error bars, versus m. This shows that $F(m)$ picks out the correct model order.

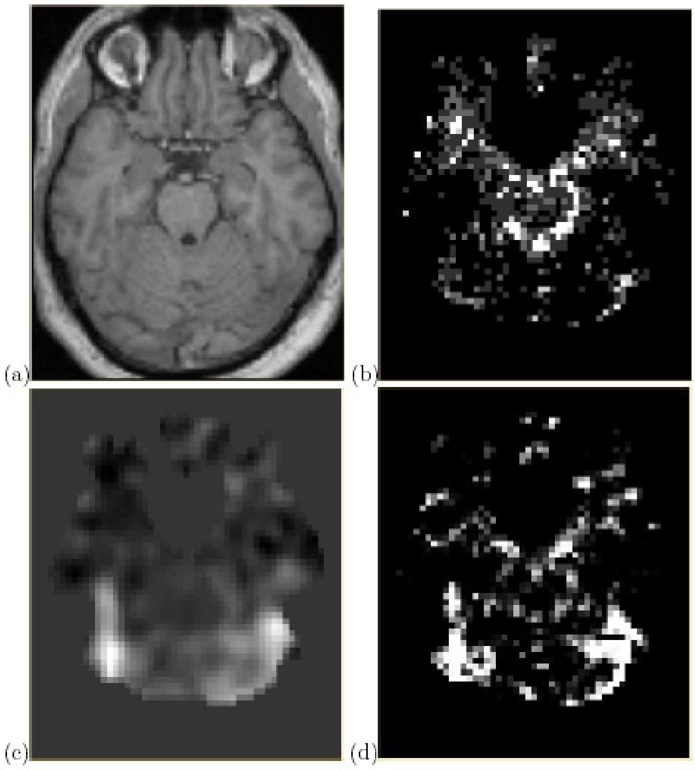


Figure 9.7: Face data: plot (b) shows argmax $F(m)$ as a function of voxel with $m = 0$ in black and $m = 3$ in white.

9.6 Mixture models

9.6.1 EM for mixture models

In this context EM is a maximum-likelihood algorithm for models with observed variables Y and hidden variables H. Hidden variable denotes which Gaussian is used to generate a data point. Select Gaussian k with probability k. That Gaussian has parameters μ_k and Σ_k.

Now, repeat 'VB derivation' but with eveything conditioned on parameters $\beta = \{\mu_k, \Sigma_k, \pi_k\}$ and replace θ with H. This gives

$$\log p(Y|\beta) = F_{EM} + KL[q(H)||p(H|Y,\beta)] \tag{9.14}$$

where

$$F_{EM} = \int q(H) \log \frac{p(H, Y|\beta)}{q(H)} dH \tag{9.15}$$

This gives rise to the following algorithm.

- E-Step: Set $q(H) = p(H|Y, \beta)$. This sets the KL term to zero. This can be done by letting

$$\begin{aligned} q(h_n) &= p(h_n|y_n, \beta) && (9.16) \\ &= \frac{p(y_n|h_n, \beta)p(h_n|\beta)}{p(y_n|\beta)} && (9.17) \end{aligned}$$

  for all data points n. This is just Bayes rule. Write $\gamma_n^k = q(h_n = k)$, the responsibilies ie. the probability that data point n was generated from the kth Gaussian.

- M-step: Now, as $KL = 0$, $F_{EM} = \log p(Y|\beta)$, so we can maximise the likelihood wrt. β by maximising F_{EM} wrt. β. We have

$$\begin{aligned} F_{EM} &= \sum_k \sum_n \gamma_k^n \log p(y_n|h_n = k)p(h_n = k) && (9.18) \\ &= \sum_k \sum_n \gamma_k^n \log p(y_n|h_n = k) + \sum_k \sum_n \gamma_k^n p(h_n = k) \end{aligned}$$

  Setting the derivatives $dF_{EM}/d\beta$ to zero gives the following updates

$$\begin{aligned} \mu_k &= \frac{\sum_n \gamma_n^k y_n}{\sum_n \gamma_n^k} && (9.19) \\ \Sigma_k &= \frac{\sum_n \gamma_n^k (y_n - \mu_k)(y_n - \mu_k)^T}{\sum_n \gamma_n^k} \\ \pi_k &= \frac{\sum_n \gamma_n^k}{N} \end{aligned}$$

See netlab demo `demgmm1.m`.

9.6.2 VB for mixture models

Allows for priors on model parameters eg. means of Gaussians. Provides approximation to model evidence based on the negative free energy. See Attias [2] and tech report `vbgmm.ps` for details.

9.6.3 Cross-modal priming fMRI

Mixture models have been applied to an analysis of intersubject variability in fMRI data. Model comparisons based on VB identified two overlapping degenerate neuronal systems in subjects performing a crossmodal priming task [33].

SVD was applied to contrast images from 17 subjects and the first 5 spatial modes were used. A cluster analysis was then implemented in this 5-dimensional space.

Due to the problem of local maxima the cluster analysis was run 10,000 times. On 9,308 the evidence (as approximated using $F(m)$) for the 2-cluster model was higher. This was also the case if 2, 3 or 4 spatial modes were used.

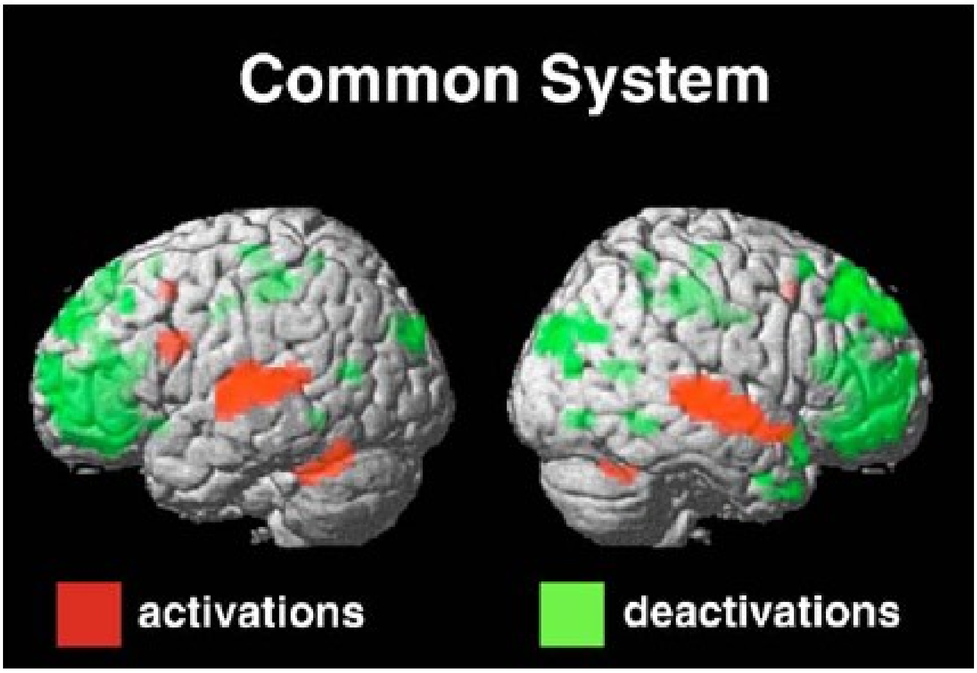
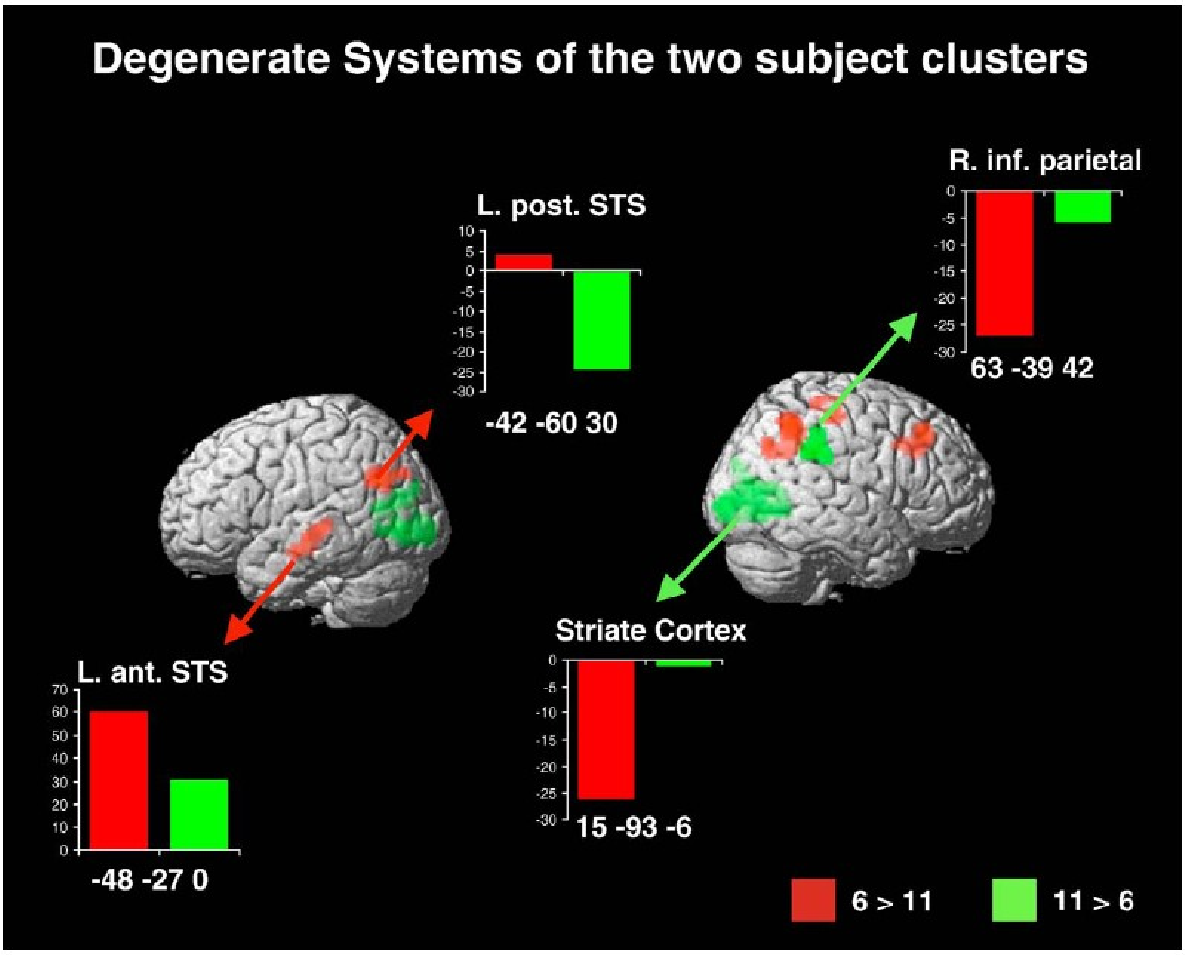


Fig. 1. Activations (red) and deactivations (green) for semantic decisions on crossmodal compound trials (for all normal subjects) relative to fixation are rendered on an averaged normalized brain. Height threshold: $P < 0.05$ corrected. Extent threshold > 3 voxels.



Degenerate Systems of the two subject clusters

Fig. 2. Semantic decisions on crossmodal compound trials. Differential activation across groups is rendered on an averaged normalized brain. Heigh $P < 0.01$ uncorrected. Extent threshold > 50 voxels. Red = 6 > 11 subjects. Green = 11 > 6 subjects. Parameter estimates for 6 subject cluster (r subject cluster (green) during semantic decisions on crossmodal stimuli. The bar graphs represent the size of the effect in adimensional units (corres percent whole brain mean).

Chapter 10

Nonlinear Models

10.1 Contents

- Central Limit Theorem
- Independent Component Analysis
- Application: Removing EEG artefacts
- Discriminant analysis
- Application: Estimating perceptual state from fMRI

10.2 Central Limit Theorem

Given n samples from *any* probability distribution, the distribution of the sample mean becomes Gaussian as $n \rightarrow \infty$. For proof see [46]. A caveat is that the sample variance must be finite.

More formally, if $y_1, y_2, ..., y_n$ are Independent and Identically Distributed (IID) random variables with $E[y_i] = \mu$ and $Var[y_i] = \sigma^2 < \infty$ and

$$\begin{aligned} \bar{y}_n &= \frac{1}{n} \sum_{i=1}^{n} y_i \qquad (10.1) \\ u_n &= \sqrt{n} \left(\frac{\bar{y}_n - \mu}{\sigma} \right) \end{aligned}$$

then $p(u_n)$ converges to a standard Gaussian density as $n \rightarrow \infty$. This is the Central Limit Theorem (CLT).

The CLT can be extended to Independent and Non-Identically Disributed (IND) random variables, as long as $E[y_i]$ and $Var[y_i]$ are finite.

This implies that if you have a Gaussian observation then its a 'mixture' (average or weighted average) of non-Gaussian signals. ICA attempts to find the underlying signals by looking for projections of the observations that are most non-Gaussian. This is implemented either (i) informally, by maximising an index of non-Gaussianity such as kurtosis, $E[(y_i - \mu)^4]$ (a Gaussian has zero kurtosis) or (ii) formally by specifying a probability model where the sources are non-Gaussian.

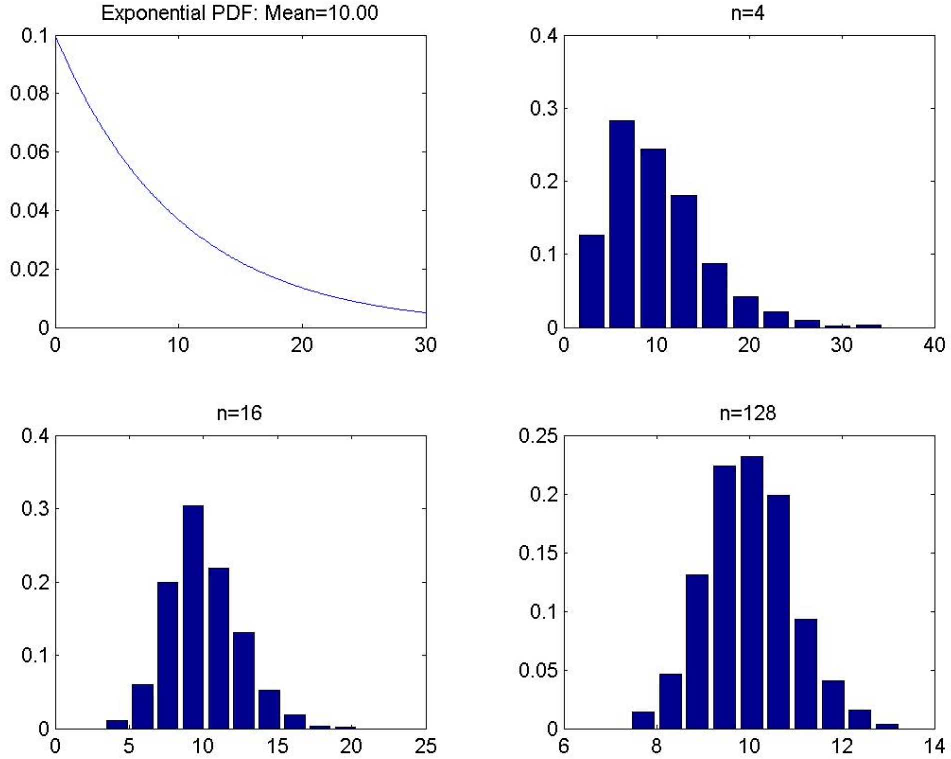


Figure 10.1: Take n samples from an exponential PDF, compute the sample mean. Do this multiple times to get an empirical estimate of the distribution of the sample mean. As n increases, the distribution becomes Gaussian.

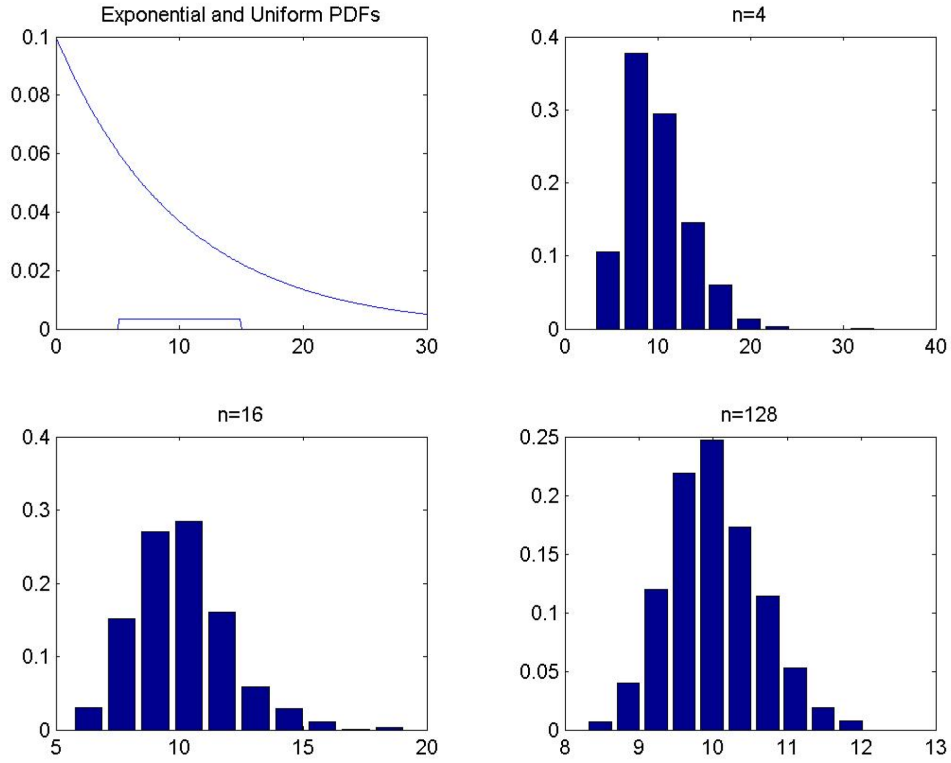


Figure 10.2: Take $n/2$ samples from an exponential PDF and $n/2$ from a uniform PDF, compute the sample mean. Do this multiple times to get an empirical estimate of the distribution of the sample mean. As n increases, the distribution becomes Gaussian.

10.3 Independent Component Analysis

In Independent Component Analysis (ICA) an M-dimensional vector observation y is modelled as

$$y = X\beta \tag{10.2}$$

where X is an unknown mixing matrix and β an unknown M-dimensional source vector. The matrix X is therefore $M \times M$. If we know $p(\beta)$, then using the method of transforming probability densities we can write the likelihood of an observation as

$$p(y) = \frac{p(\beta)}{|\det X|} \tag{10.3}$$

The determinant measures the volume of a matrix. So Eq 3. takes into account volumetric changes in the transformation, so that probability mass is preserved as we transform β into y.

ICA assumes the sources to be Independent (this is the I in ICA)

$$p(\beta) = \prod_{i=1}^{M} p_s(\beta_i) \tag{10.4}$$

We can therefore write the likelihood as

$$p(y) = \frac{\prod_{i=1}^{M} p_s(\beta_i)}{|\det X|} \tag{10.5}$$

The log-likelihood is then given by

$$\log p(y) = -\log|\det X| + \sum_{i=1}^{M} \log p_s(\beta_i) \tag{10.6}$$

We can write the unknown sources as

$$\begin{aligned} \beta &= X^{-1} y \qquad (10.7) \\ &= Ay \end{aligned}$$

where $A = X^{-1}$ is the inverse mixing matrix. We can also write $\beta_i = \sum_{j=1}^{M} A_{ij} y_j$ and express the log-likelihood as

$$\log p(y) = \log |\det A| + \sum_{i=1}^{M} \log p_s(\sum_{j=1}^{M} A_{ij} y_j) \tag{10.8}$$

The log-likelihood is now a function of the data and the inverse mixing matrix.

If we have $n = 1..N$ independent samples of data, Y, the likelihood is

$$\log p(Y) = N \log |\det A| + \sum_{n=1}^{N} \sum_{i=1}^{M} \log p_s(\sum_{j=1}^{M} A_{ij} y_{nj}) \tag{10.9}$$

We can find A by giving this function to any optimisation algorithm. As elements of A become co-linear $|\det A| \rightarrow 0$, and the likelihood reduces. Maximising the likelihood therefore encourages sources to be different (via the first term) and encourages them to be similar to p_s ie. non-Gaussian (via the second term).

10.3.1 Source densities

Different ICA models result from different assumptions about the source densities $p_s(\beta_i)$. One possibility is the generalised exponential family. Another is the 'inverse cosh' density

$$\begin{aligned} p_s(\beta_i) &= \frac{1}{cosh(\beta_i)} \qquad (10.10) \\ &= \frac{1}{\exp \beta_i + \exp -\beta_i} \end{aligned}$$

This latter choice gives rise to the original 'Infomax' algorithm [5].

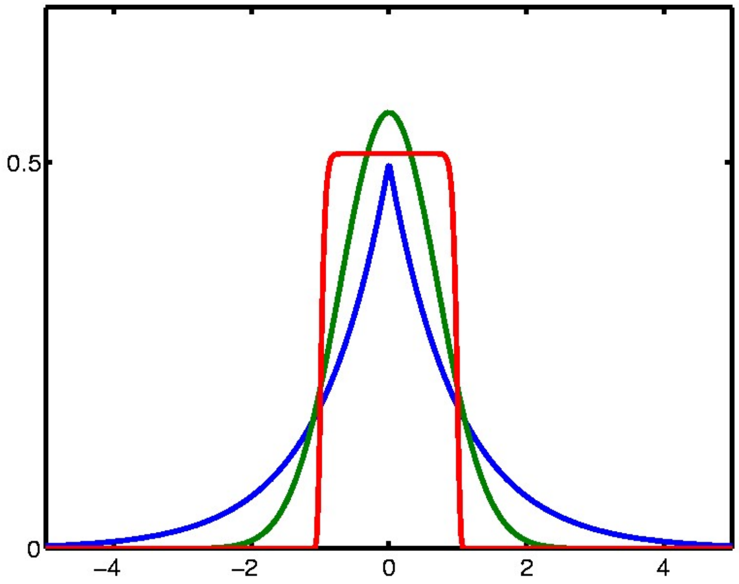


Figure 10.3: Generalised exponential densities $p_s(\beta_i) \propto \exp\left(-|\frac{\beta_i}{\sigma}|^R\right)$with $R = 1$ (Blue, 'Laplacian density'), $R = 2$ (Green, 'Gaussian density'), $R > 20$ (Red, 'Uniform density'). The parameter σ defines the width of the density.

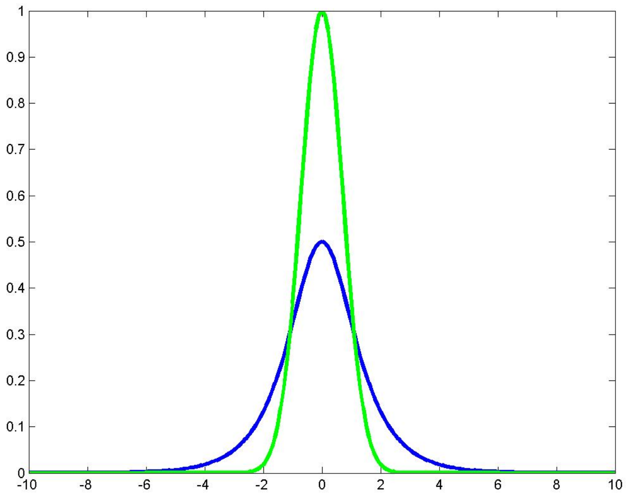


Figure 10.4: Inverse Cosh, $\frac{1}{\exp\beta_i + \exp -\beta_i}$ (Blue) and Gaussian, $\exp -\beta_i^2$ (Green)

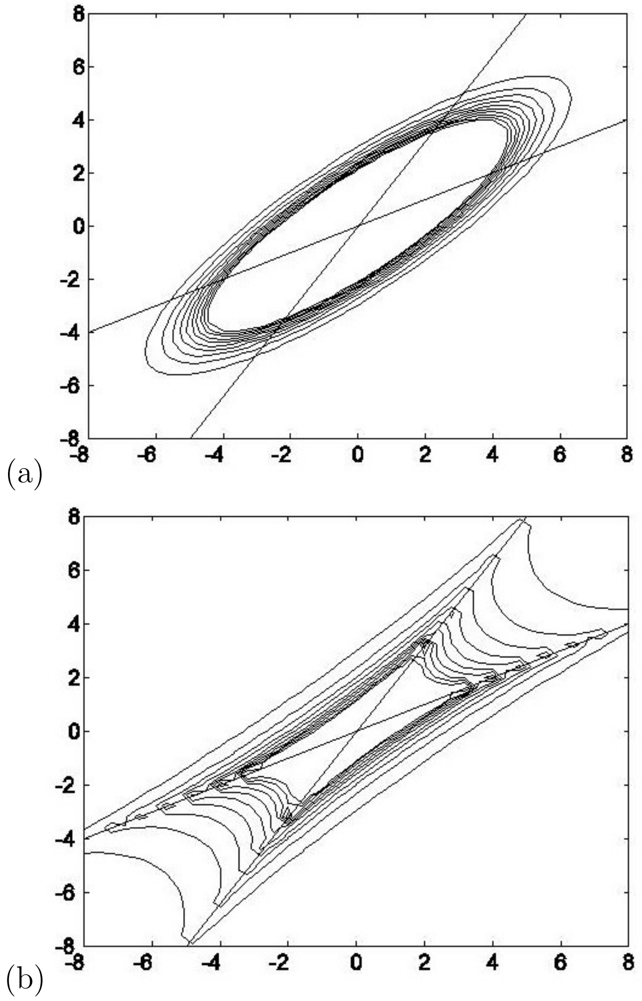


Figure 10.5: Probability contours, $p(y)$, from 2D-ICA models with (a) Gaussian sources and (b) Heavy-tailed sources. The mixing matrices X are the same.

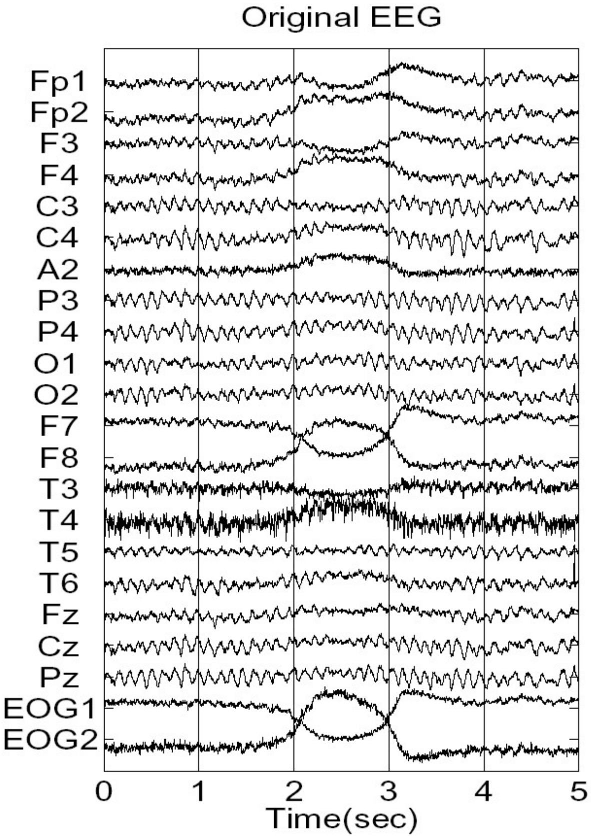


Figure 10.6: Original 5-second EEG record, containing prominent slow eye-movement (seconds 2 to 4).

10.3.2 Removing EEG artefacts

Jung et al. [23] use ICA to remove artefacts from EEG data recorded from 20 scalp electrodes placed according to the 10/20 system and 2 EOG electrodes, all references to the left mastoid. The sampling rate was 256Hz. An ICA decomposition was implemented by applying an extended Infomax algorithm to 10-second EEG epochs to produce sources with time series that are maximally independent.

This artefact removal method compares favourably to PCA and filtering approaches, and approaches for eye-movement correction based on dipole models and regression [23]. It has been incorporated in the EEGLAB available from `http://sccn.ucsd.edu/eeglab`

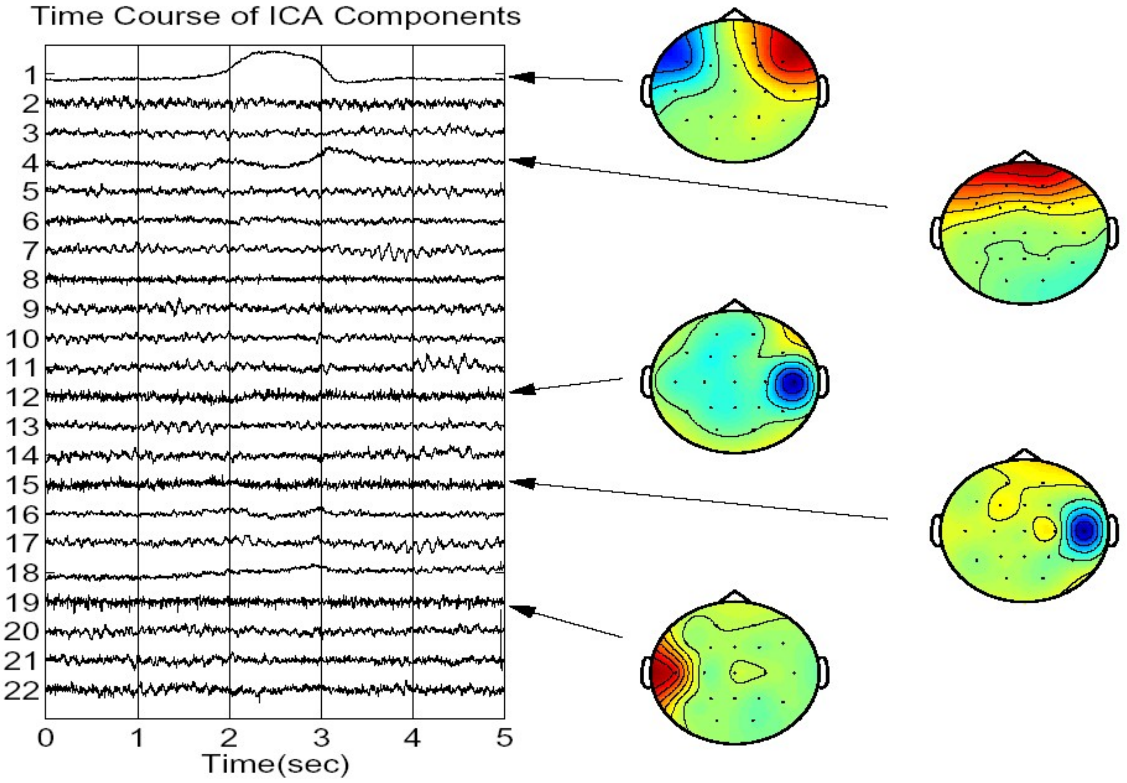


Figure 10.7: Left panel: Time course of source estimates β_{ni} for n=1..N samples (N=5 x 256), and i=1..22 sources. Right panel: Spatial topographies (rows of mixing matrix X) for 5 selected components. The top two components account for eye movement and the bottom three for muscle activity over fronto-temporal regions.

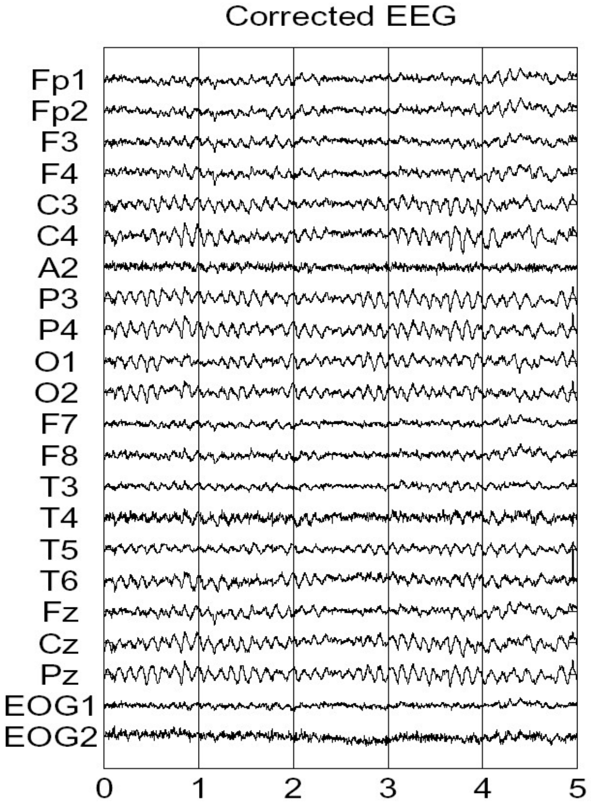


Figure 10.8: Corrected EEG formed by subtracting five selected components from original data. This data is free from EOG and muscle artifacts. We can now see activity in T3/T4 that was previously masked by muscle artifact.

10.4 Discriminant analysis

10.4.1 Linear decision boundary

The aim of discriminant analysis is to estimate class label $y = \{1, 2\}$ given multivariate data x. This could be eg. $y = 1$ for patients and $y = 2$ for controls. One approach is to use labelled data to form a likelihood model for each class, $p(x|y)$. New data points are then assigned to the class with the highest likelihood. Another way of saying this is to form the Likelihood Ratio (LR)

$$LR_{12} = \frac{p(x|y = 1)}{p(x|y = 2)} \tag{10.11}$$

and assign to class 1, if LR_{12} is greater than one. According to the Neymann-Pearson Lemma (see eg. [12]) this test has the highest sensitivity, for any given level of specificity. Any monotonic function of LR_{12} will provide as good a test as the likelihood, and the logarithm is often used.

If, additionally, we have prior probabilities for each category, $p(y)$ then the optimal decision is to assign to the class with the highest posterior probability. For class 1 we have

$$p(y = 1|x) = \frac{p(x|y = 1)p(y = 1)}{p(x|y = 1)p(y = 1) + p(x|y = 2)p(y = 2)} \tag{10.12}$$

For equal priors this reduces to an LR test.

A simple likelihood model for each class is a Gaussian. The above posterior probability is then the same as the 'responsibilty' in a Gaussian mixture model (see last lecture). It can be re-written as

$$\begin{aligned} p(y=1|x) &= \frac{1}{1+\frac{p(x|y=2)p(y=2)}{p(x|y=1)p(y=1)}} \\ &= \frac{1}{1+exp(-a)} \\ &= g(a) \end{aligned} \tag{10.13}$$

where $g(a)$ is the 'sigmoid' or 'logistic' function and

$$a = \log\left(\frac{p(x|y=1)p(y=1)}{p(x|y=2)p(y=2)}\right) \tag{10.14}$$

For Gaussians with *equal covariances* $\Sigma_1 = \Sigma_2 = \Sigma$ we have

$$\begin{aligned} \log p(x|y=1) &= \frac{d}{2}\log 2\pi + \frac{1}{2}\log|\Sigma| \\ &- \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) \\ \log p(x|y=2) &= \frac{d}{2}\log 2\pi + \frac{1}{2}\log|\Sigma| \\ &- \frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2) \end{aligned} \tag{10.15}$$

This gives

$$a = w^T x + w_0 \tag{10.16}$$

where

$$\begin{aligned} w &= \Sigma^{-1}(\mu_1 - \mu_2) \\ w_0 &= -\frac{1}{2}\mu_1^T\Sigma\mu_1 + \frac{1}{2}\mu_2^T\Sigma\mu_2 + \log\frac{p(y=1)}{p(y=2)} \end{aligned} \tag{10.17}$$

This approach is known as logistic discrimination or logistic classification. The decision boundary is given by $a = 0$.

10.4.2 Nonlinear decision boundary

If the Gaussians do not have equal covariance then the decision boundary becomes quadratic. If Gaussians are not good models of the class probability densities then another approach is required eg. Nearest Neighbour classifiers, or Multi-Layer Perceptrons (MLPs). An MLP comprises nested logistic functions.

A two-layer MLP is given by

$$\begin{aligned} p(y=1|x) &= g\left(\sum_{h=1}^{H} w_h^{(2)} z_h\right) \qquad (10.18) \\ z_h &= g\left(\sum_{d=1}^{D} w_{hd}^{(1)} x_d)\right) \end{aligned}$$

with D is the dimension of the input x, H is the number of 'hidden units' in the 'first layer', and z_h is the output of the hth unit. Superscripts 1 and 2 denote 1st and 2nd layer weights. This allows for classification using arbitrary decision boundaries. There is no closed form for the parameters w, but they can be estimated using an optimisation algorithm as described in [6]. This is an example of an Artificial Neural Network (ANN).

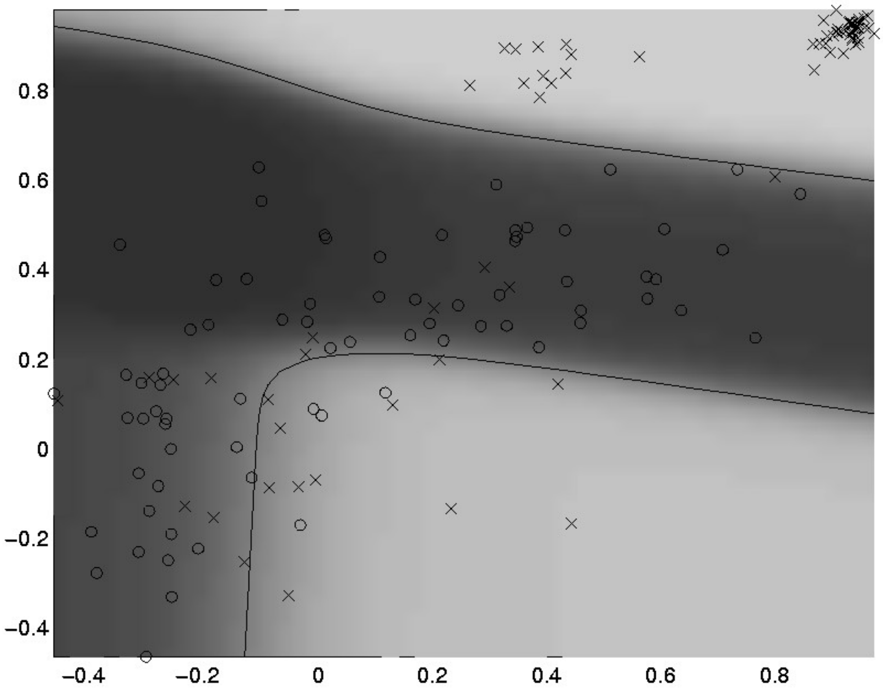


Figure 10.9: Tremor data. Crosses represent data points from patients $y = 1$, circles data points from normal subjects, $y = 2$. The solid line shows the overall decision boundary formed by an MLP with three hidden units. The shade of gray codes the output, $p(y = 1|x)$ and the features, x, are from autoregressive modelling of EMG data.

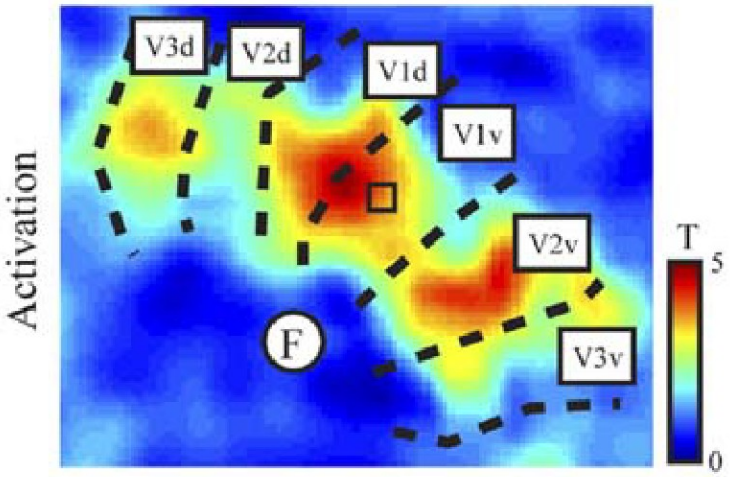


Figure 10.10: Functional localiser (4Hz reversing checkerboard) is used to select 50 visually responsive voxels in each region.

10.5 Estimating perceptual state from fMRI

Haynes and Rees [21] used Linear Discriminant Analysis (LDA) to classify perceptual state during binocular rivalry from fMRI data.

Retinotopic mapping and functional localisers (reversing checkerboard stimuli) were used to identify the V1, V2, V3 and V5 regions of visual cortex. The 50 most visually responsive voxels in each region were then selected for subsequent analysis.

Subjects then viewed rivalrous stimuli, and pressed buttons to indicate perceptual state, $y_t = 1$ for red percept and $y_t = 2$ for blue percept. Activity in selected voxels x_t were then used to estimate y_t. The labels y_t were time-shifted to accomodate the delay in the hemodynamic response.

Estimates of perceptual state were then formed using

$$\begin{aligned} \hat{y}_t &= w^T x_t \\ w &= \Sigma^{-1}(\mu_{red} - \mu_{blue}) \end{aligned} \tag{10.19}$$

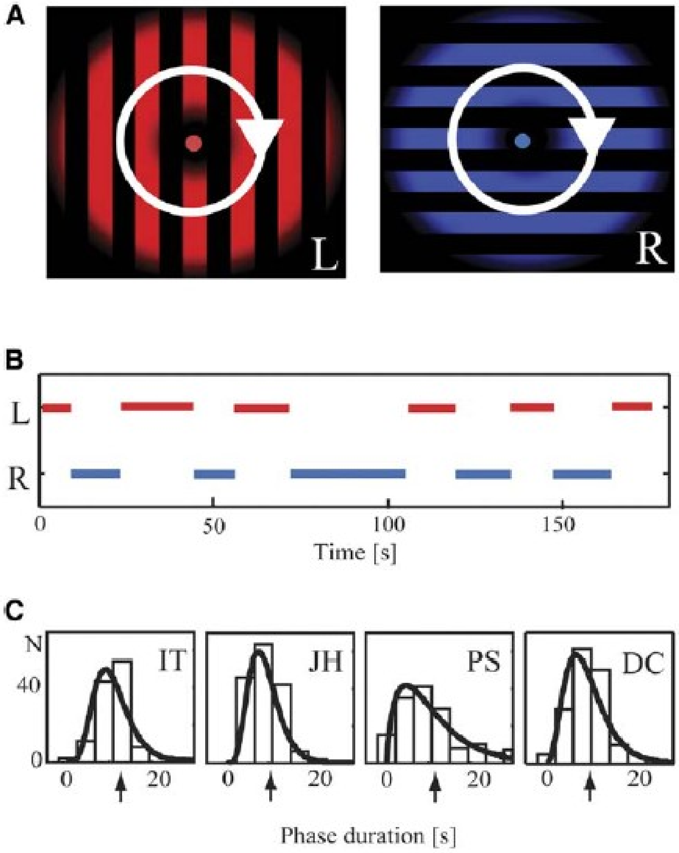


Figure 10.11: A: Superimposed gratings viewed through red/blue filtering glasses. B: Subjects pressed buttons indicating perceptual state. C: Percept durations for four subjects.

where Σ is the within group sample covariance (estimated from both red and blue fMRI samples) and m_{red} and m_{blue} are the mean fMRI vectors for each condition. These estimates were then time-shifted, by convolving with a 'Canonical HRF' before comparison with true values. Cross-validation was used to assess accuracy.

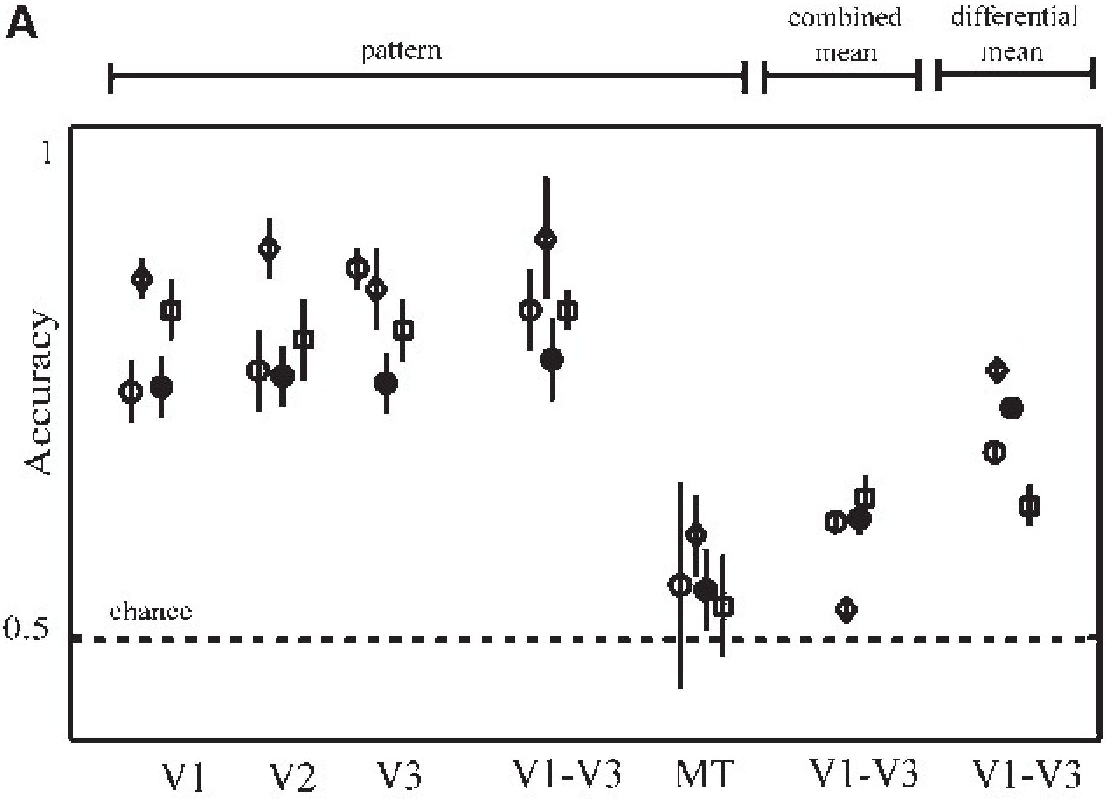


Figure 10.12: Accuracy by region assessed using cross-validation.

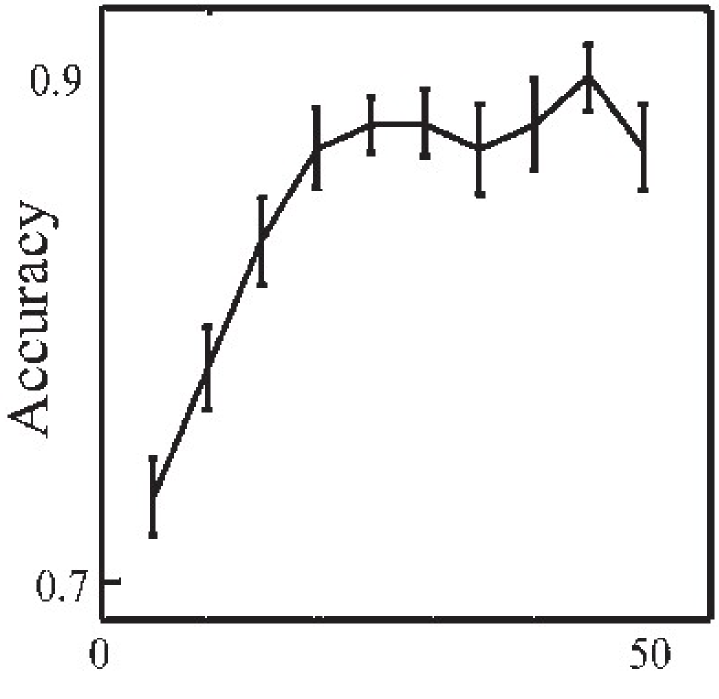


Figure 10.13: Accuracy by number of voxels assessed using cross-validation.

Bibliography

[1] R.J. Adler. *The geometry of random fields.* John Wiley, New York., 1981.

[2] H. Attias. Inferring Parameters and Structure of Latent Variable Models by Variational Bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999.

[3] S. Baillet, J.C. Mosher, and R.M. Leahy. Electromagnetic Brain Mapping. *IEEE Signal Processing Magazine*, pages 14–30, November 2001.

[4] M. Bauer, R. Oostenveld, M. Peeters, and P. Fries. Tactile spatial attention enhances gamma-band activity in somatosensory cortex and reduces low-frequency activity in parieto-occipital areas. *The Journal of Neuroscience*, 26(2):490–501, 2006.

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

[6] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1995.

[7] K.A. Bollen. *Structural Equations with Latent Variables.* John Wiley, New York, 1989.

[8] G. Clifford Carter. Coherence and time delay estimation. *Proceedings of the IEEE*, 75(2):236–255, 1987.

[9] C. Chatfield. *An Introduction to Multivariate Analysis*. Chapman and Hall, 1991.

[10] R. Christensen. *Plane Answers to Complex questions: the theory of linear models*. Springer-Verlag, New York, US., 2002.

[11] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

[12] P. Dayan and L.F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, 2001.

[13] B. Efron and R.J. Tibshirani. *An introduction to the bootstrap*. Chapman and Hall, 1993.

[14] K.J. Friston, K.J. Worsley R.S.J. Frackowiak, J.C. Mazziotta, and A.C. Evans. Assessing the significance of focal activations using their spatial extent. *Human Brain Mapping*, 1:214–220, 1994.

[15] K.J. Friston, D.E. Glaser, R.N.A. Henson, S.J. Kiebel, C. Phillips, and J. Ashburner. Classical and Bayesian inference in neuroimaging: Applications. *NeuroImage*, 16:484–512, 2002.

[16] K.J. Friston, O. Josephs, G. Rees, and R. Turner. Nonlinear event-related responses in fMRI. *Magnetic Resonance in Medicine*, 39:41–52, 1998.

[17] K.J. Friston, W.D. Penny, C. Phillips, S.J. Kiebel, G. Hinton, and J. Ashburner. Classical and Bayesian inference in neuroimaging: Theory. *NeuroImage*, 16:465–483, 2002.

[18] K.J. Friston, K.E. Stephan, T.E. Lund, A. Morcom, and S.J. Kiebel. Mixed-effects and fMRI studies. *NeuroImage*, 24:244–252, 2005.

[19] J. Gross, J. Kujala, M. Hammalainen, L. Timmerman, A. Schnitzler, and R. Salmelin. Dynamic imaging of coherent sources: studying neural interactions in the brain. *Proceedings of the National Academy of Sciences*, 98(2):694–699, 2001.

[20] D. A. Harville. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, 72(358):320–338, 1977.

[21] J.D. Haynes and G. Rees. Predicting the stream of consciousness from activity in human visual cortex. *Current Biology*, 15:1301–1307, 2005.

[22] R.N.A. Henson, Y. Goshen-Gottstein, T. Ganel, L.J. Otten, A. Quayle, and M.D. Rugg. Electrophysiological and hemodynamic correlates of face perception, recognition and priming. *Cerebral Cortex*, 13:793–805, 2003.

[23] T. Jung, S. Makeig, C. Humphries, T. Lee, M. J. McKeown, V. Iragui, and T. J. Sejnowski. Removing Electroencephalographic Artifacts by Blind Source Separation. *Psychophysiology*, 1999.

[24] D.G. Kleinbaum, L.L. Kupper, and K.E. Muller. *Applied Regression Analysis and Other Multivariable Methods.* PWS-Kent, Boston, 1988.

[25] D. J. C. MacKay. Choice of basis for Laplace approximations. *Machine Learning*, 33:77–86, 1998.

[26] D.J.C Mackay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, Cambridge, 2003.

[27] J.R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics.* John Wiley, 1997.

[28] S.L. Marple. *Digital spectral analysis with applications.* Prentice-Hall, 1987.

[29] J. Mattout, C. Phillips, W.D. Penny, M. Rugg, and K.J. Friston. Meg source localisation under multiple constraints: an extended Bayesian framework. *NeuroImage*, 2005.

[30] T. Minka. Divergence measures and message passing. Technical report, Microsoft Research Ltd, Cambridge, UK, 2005. MSR-TR-2005-173.

[31] P.P. Mitra and B. Pesaran. Analysis of dynamic brain imaging data. *Biophysical Journal*, 76:691–708, 1999.

[32] T. Mullin. *The Nature of Chaos.* Oxford Science Publications, 1993.

[33] U. Noppeney, W. D. Penny, C. J. Price, G. Flandin, and K. J. Friston. Identification of degenerate neuronal systems based on intersubject variability. *Neuroimage*, 30:885–890, 2006.

[34] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, Singapore, 3 edition, 1991.

[35] J. Pearl. Graphs, causality, and structural equation models. *Sociological Methods and Research*, 27:226–284, 1998.

[36] W. Penny, S. Kiebel, and K. Friston. Variational bayes. In K. Friston, J. Ashburner, S. Kiebel, T. Nichols, and W. Penny, editors, *Statistical Parametric Mapping: The analysis of functional brain images*. Elsevier, London, 2006.

[37] W.D. Penny, A.P. Holmes, and K.J. Friston. Random effects analysis. In R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, K.J. Friston, C.J. Price, S. Zeki, J. Ashburner, and W.D. Penny, editors, *Human Brain Function*. Academic Press, 2nd edition, 2003.

[38] W.D. Penny, S.J. Kiebel, and K.J. Friston. Variational Bayesian Inference for fMRI time series. *NeuroImage*, 19(3):727–741, 2003.

[39] W.D. Penny, K.E. Stephan, A. Mechelli, and K.J. Friston. Comparing Dynamic Causal Models. *NeuroImage*, 22(3):1157–1172, 2004.

[40] D.B. Percival and A.T. Walden. *Spectral analysis for physical applications: multitaper and conventional univariate techniques*. Cambridge University Press, UK., 1993.

[41] R. Poldrack. Can cognitive processes be inferred from neuroimaging data ? *Trends in Cognitive Sciences*, 2006. In Press.

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

[43] J.C. Shaw and D. Simpson. ”eeg coherence: Caution and cognition”. *British Psychological Society Quaterly*, 30/31, 1997.

[44] G. Strang. *Linear algebra and its applications*. Harcourt Brace, 1988.

[45] N. Trujillo-Barreto, E. Aubert-Vazquez, and P. Valdes-Sosa. Bayesian model averaging in EEG/MEG imaging. *Neuroimage*, 21:1300–1319, 2004. In Press.

[46] D.D. Wackerley, W. Mendenhall, and R.L. Scheaffer. *Mathematical statistics with applications*. Duxbury Press, 1996.

[47] S. Weisberg. *Applied Linear Regression*. John Wiley, 1980.

[48] K. J. Worsley. The geometry of random images. *Chance*, 9(1):27–40, 1996.

[49] K. J. Worsley, M. Andermann, T. Koulis, D. MacDonald, and A. C. Evans. Detecting changes in non-isotropic images. *Human Brain Mapping*, 8(2):98–101, 1999.

[50] Keith J. Worsley, S. Marrett, P. Neelin, A.C. Vandal, K.J. Friston, and A.C. Evans. A unified statistical approach for determining significant signals in images of cerebral activation. *Human Brain Mapping*, 4:58–73, 1996.

[51] K.J. Worsley, A.C. Evans, S. Marrett, and P. Neelin. A three dimensional statistical analysis for CBF activation studies in human brain. *Journal of Cerebral Blood Flow and Metabolism*, 12:900–918, 1992.