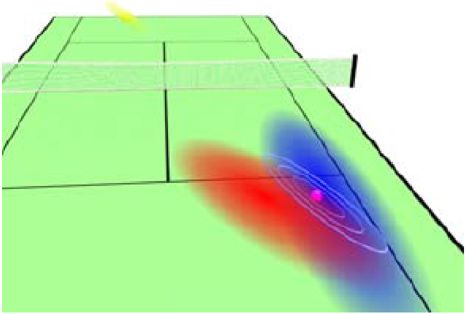
Empirical Bayes

Will Penny

Bayesian Inference Course,
WTCN, UCL, March 2013

Tennis

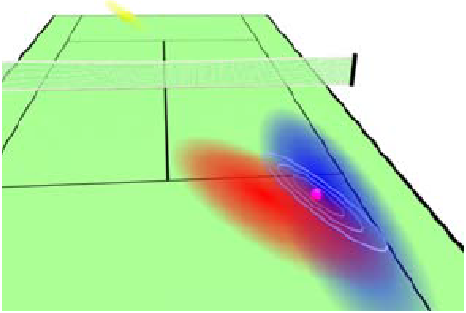
From Wolpert and Ghahramani (2006)



$$
\begin{aligned}
p(w) &= \mathrm{N}(w; \mu_w, C_w) \\
p(y|w) &= \mathrm{N}(y; Xw, C_y)
\end{aligned}
$$

Tennis

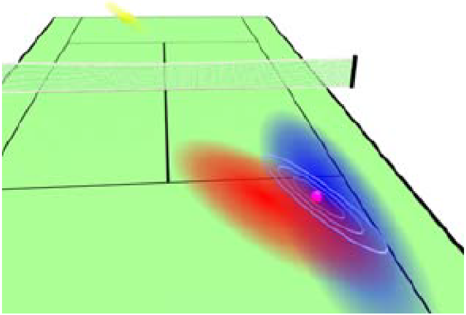
From Wolpert and Ghahramani (2006)



$$
\begin{aligned}
p(w|y) &= \mathrm{N}(w; m_w, S_w) \\
S_w^{-1} &= X^T C_y^{-1} X + C_w^{-1} \\
m_w &= S_w (X^T C_y^{-1} y + C_w^{-1} \mu_w)
\end{aligned}
$$

Tennis

From Wolpert and Ghahramani (2006)



$$
\begin{aligned}
p(w) &= \mathrm{N}(w; \mu_w, C_w) \\
p(y|w) &= \mathrm{N}(y; Xw, C_y)
\end{aligned}
$$

How can we estimate C_w and C_y from data ?

Covariances

Case I: Isotropic covariances

$$\begin{aligned} C_y &= \lambda_y^{-1} I_N \\ C_w &= \lambda_w^{-1} I_p \end{aligned}$$

and N data points and p parameters.

Case II: Linear covariances

$$\begin{aligned} C_y &= \sum_i \lambda_i Q_i \\ C_w &= \sum_{i'} \lambda_{i'} Q_{i'} \end{aligned}$$

where Q are known covariance basis functions.

Empirical Bayes

Hyperparameters, λ, can be estimated so as to maximise their evidence, $p(y|\lambda)$. This forms the basis of Empirical Bayes.

This is given by

$$\begin{aligned} p(y|\lambda) &= \int p(y,w|\lambda) dw \\ &= \int p(y|w,\lambda) p(w|\lambda) dw \end{aligned}$$

We then have

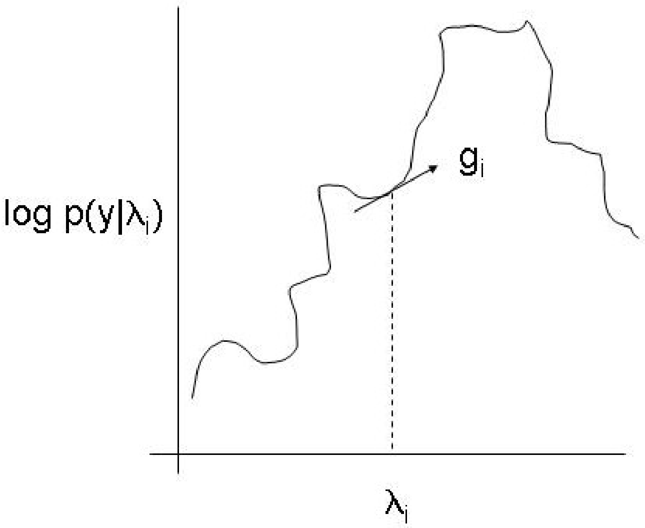
$$L(\lambda) = \log p(y|\lambda)$$

For linear models this can be derived as in Bishop (2006). See also next lecture.

In this formulation λ are not treated as random variables. There is no prior on them.

Gradient Ascent

Updating hyperparameters via gradient ascent of evidence.



Linear Models

The evidence for λ is composed of sum squared precision weighted prediction errors and Occam factors

$$\begin{aligned}
L(\lambda) &= -\frac{1}{2} e_y^T C_y^{-1} e_y - \frac{1}{2} \log |C_y| - \frac{N}{2} \log 2\pi \\
&- \frac{1}{2} e_w^T C_w^{-1} e_w - \frac{1}{2} \log \frac{|C_w|}{|S_w|}
\end{aligned}$$

where λ is a vector of hyperparameters that parameterise the covariances C_w and C_y. The prediction errors are the difference between what is expected and what is observed

$$\begin{aligned}
e_y &= y - X m_w \\
e_w &= m_w - \mu_w
\end{aligned}$$

Same as expression for model evidence in previous lecture.

Empirical Bayes

We iterate between finding the parameters w and hyperparameters λ. For linear Gaussian models this corresponds to computing the posterior over w

$$\begin{aligned} S_w^{-1} &= X^T C_y^{-1} X + C_w^{-1} \\ m_w &= S_w (X^T C_y^{-1} y + C_w^{-1} \mu_w) \end{aligned}$$

and then setting λ to maximise the model evidence.

$$\hat{\lambda} = \arg\max_{\lambda} L(\lambda)$$

These two steps are then iterated and can be thought of as E and M steps in an EM optimisation algorithm.

Isotropic Covariances

For a Bayesian GLM

$$\begin{aligned} y &= Xw + e_1 \\ w &= \mu_w + e_2 \end{aligned}$$

with isotropic covariances

$$\begin{aligned} C_y &= \lambda_y^{-1} I_N \\ C_w &= \lambda_w^{-1} I_p \end{aligned}$$

and N data points and p parameters. The equations for updating λ can be derived as shown in Chapter 10 of Bishop (2005).

Well-determined parameters

Define

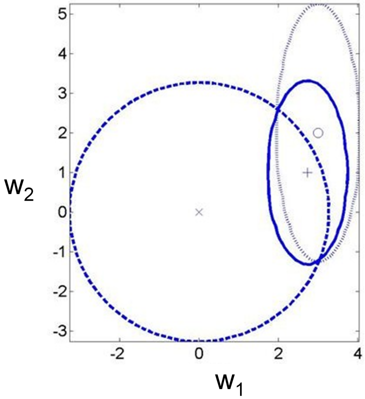
$$\gamma = \sum_{j=1}^{p} \frac{\alpha_j}{\alpha_j + \hat{\lambda}_w}$$

where α_j are eigenvalues of the data precision term $X^T C_y^{-1} X$. If $\alpha_j >> \hat{\lambda}_w$ for all j then $\gamma = p$. Parameters have all been determined by the data. So γ is equivalent to number of well-determined parameters.

Effectively,γ counts the number of parameters for which the data precision dominates the prior precision.

Well-determined parameters

Here $p = 2$ parameters. But $\gamma \approx 1$.



Only one of the parameters, w_1, is determined by the data.

M-Step

Then

$$\begin{aligned} \frac{1}{\hat{\lambda}_w} &= \frac{e_w^T e_w}{\gamma} \\ \frac{1}{\hat{\lambda}_y} &= \frac{e_y^T e_y}{N - \gamma} \end{aligned}$$

where the prediction errors are

$$\begin{aligned} e_y &= y - X m_w \\ e_w &= m_w - \mu_w \end{aligned}$$

This effectively partitions the degrees of freedom in the data into those for estimating the prior and the likelihood.

This is like cross-validation but without explicit separation of data.

Setting λ to maximise the *marginal* likelihood produces unbiased estimates of variances whereas ML estimation produces biased estimates.

EM Algorithm

E-Step:

$$
\begin{aligned}
S_w^{-1} &= \hat{\lambda}_y X^T X + \hat{\lambda}_w I_p \\
m_w &= S_w(\hat{\lambda}_y X^T y + \hat{\lambda}_w \mu_w)
\end{aligned}
$$

M-Step:

$$
\begin{aligned}
e_y &= y - X m_w \\
e_w &= m_w - \mu_w \\
\frac{1}{\hat{\lambda}_w} &= \frac{e_w^T e_w}{\gamma} \\
\frac{1}{\hat{\lambda}_y} &= \frac{e_y^T e_y}{N - \gamma}
\end{aligned}
$$

Shrinkage Priors

This numerical example caricatures the use of PEB for estimating effect sizes from brain imaging data (Friston and Penny, 2003).

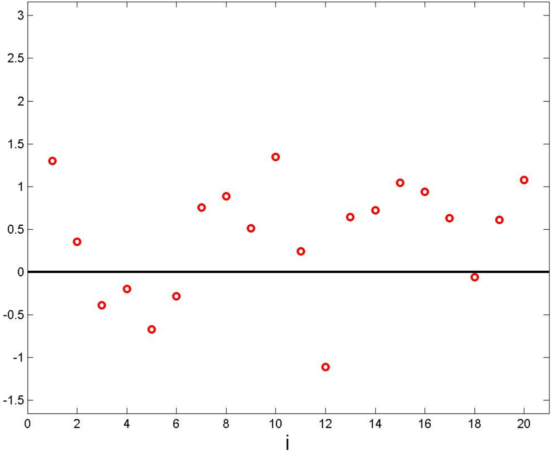
The approach uses a global ‘shrinkage prior’ which embodies a prior belief that across the brain

- the average effect is zero
- the variability of responses follows a Gaussian distribution

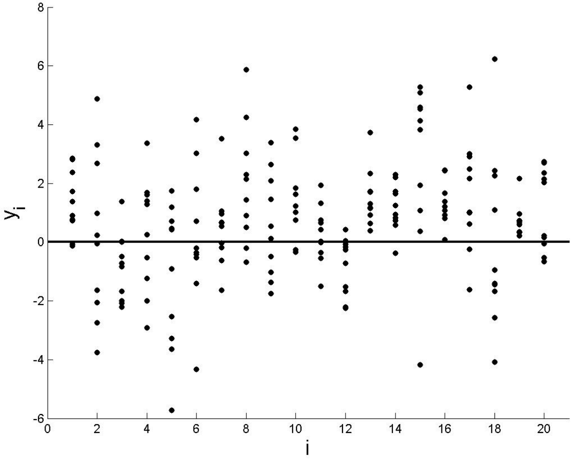
That is

$$p(\mu_i) = \mathrm{N}(\mu_i; 0, \alpha^{-1})$$

True effect sizes μ_i for $i = 1..20$ voxels generated from the prior $p(\mu_i|\alpha) = N(\mu_i; 0, \alpha^{-1})$ with $\alpha = 1$.

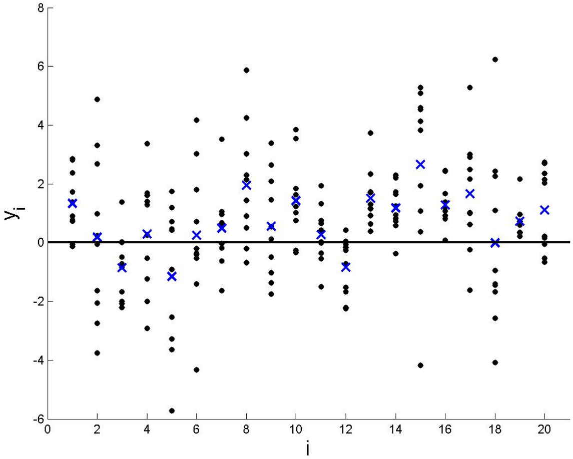


The black dots denote $N = 10$ data points at each voxel generated from the likelihood $p(y_i|\mu_i) = \mathrm{N}(y_i; \mu_i, \beta_i^{-1})$ with β_i drawn from a uniform distribution between 0.1 and 1.

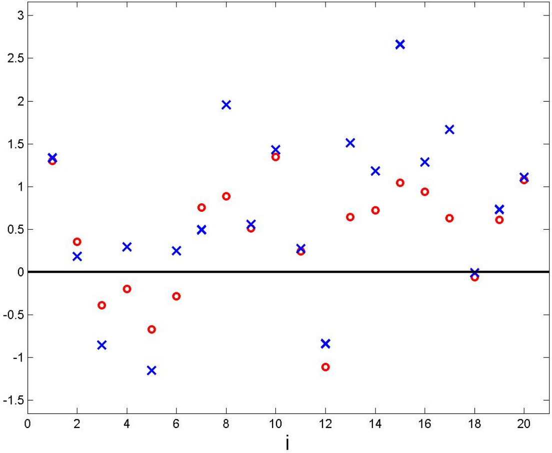


Thus some voxels, eg. voxels 2, 15 and 18, have noisier data than others.

Effect sizes were then estimated from this data using Maximum-Likelihood (ML) and PEB. ML estimates (blue) are given by $\mu_i^{ML} = \frac{1}{N} \sum_n y_{in}$.



True effect sizes (red) and ML estimates (blue)



EM/PEB algorithm

Initialise $\gamma_i = 1$, $\alpha = 0$.

- E-step:

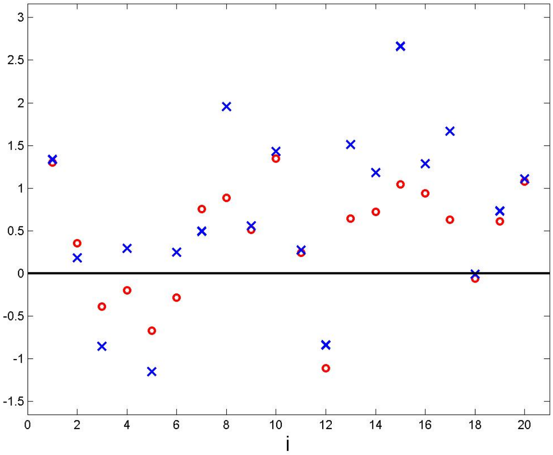
$$\mu_i = \frac{\gamma_i}{N} \sum_n y_{in}$$

- M-step:

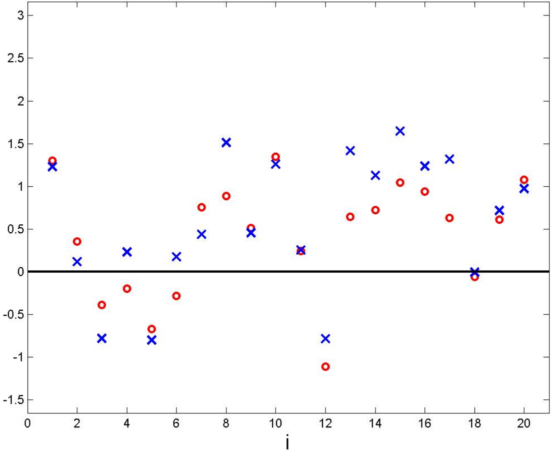
$$\begin{aligned} \frac{1}{\beta_i} &= \frac{1}{N-\gamma_i} \sum_n (y_{in} - \mu_i)^2 \\ \gamma_i &= \frac{N\beta_i}{N\beta_i + \alpha} \\ \frac{1}{\alpha} &= \frac{\sum_i \mu_i^2}{\sum_i \gamma_i} \end{aligned}$$

The E and M steps are then iterated.

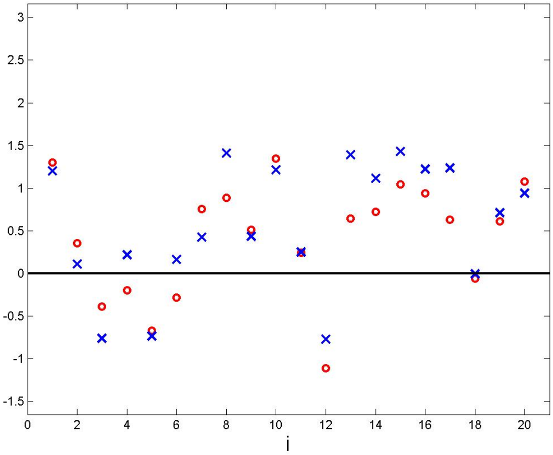
True effect sizes, μ_i (red circles) and estimated effect sizes, $\hat{\mu}_i$, (blue crosses) from PEB iteration number 1.



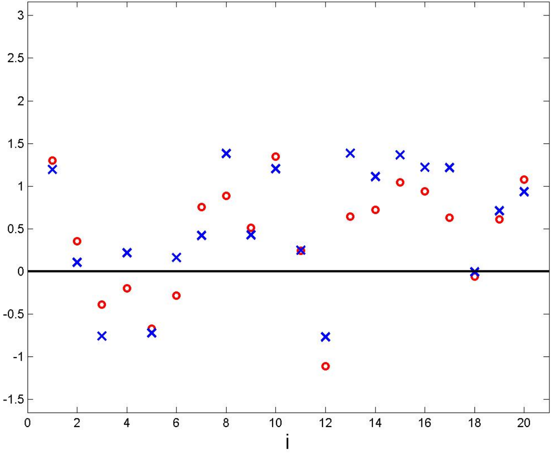
True effect sizes, μ_i (red circles) and estimated effect sizes, $\hat{\mu}_i$, (blue crosses) from PEB iteration number 3.



True effect sizes, μ_i (red circles) and estimated effect sizes, $\hat{\mu}_i$, (blue crosses) from PEB iteration number 5.



True effect sizes, μ_i (red circles) and estimated effect sizes, $\hat{\mu}_i$, (blue crosses) from PEB iteration number 7.



PEB versus ML

The corresponding estimates of α were 0, 0.82, 0.91 and 0.95, showing convergence to the true prior precision of 1.

The mean squared difference between the true and estimated effects across voxels is 0.71 for ML and $=$ 0.34 for PEB. That is, PEB estimates are twice as accurate on average.

PEB is only better 'on average'. It does better at most voxels at the expense of being worse at a minority, for example, voxel 2.

PEB can do better than ML because it uses more information - that effects have a mean of zero across the brain and follow a Gaussian variability profile.

Weighting of data and prior

The quantity

$$\gamma_i = \frac{N\beta_i}{N\beta_i + \alpha}$$

is the ratio of the data precision to the posterior precision.

A value of 1 indicates that the estimated effect is determined solely by the data, as in ML. A value of 0 indicates the estimate is determined solely by the prior.

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

$$\mu_i = \frac{\gamma_i}{N} \sum_n y_{in}$$

Empirical Bayes also known as 'Stein Estimators' (C. Stein, 1956).

Stein Estimators

EB more accurate than ML - the biggest result in postwar statistics (Efron/Muralidharan, Large Scale Simultaneous Inference, 2009).

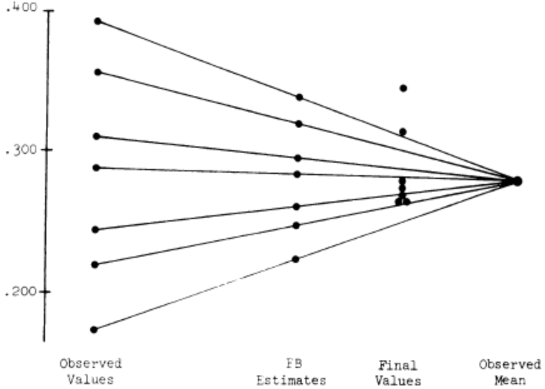


Figure 3. Graphical Display of the Baseball Data.

Regression to the mean (Casella, American Statistician, 1985).

Linear Covariances

For a Bayesian GLM

$$\begin{aligned} y &= Xw + e_1 \\ w &= \mu_w + e_2 \end{aligned}$$

with covariances

$$\begin{aligned} C_y &= \sum_i \lambda_i Q_i \\ C_w &= \sum_{i'} \lambda_{i'} Q_{i'} \end{aligned}$$

where Q are known covariance basis functions. The M-step is

$$\hat{\lambda} = \arg\max_{\lambda} L(\lambda)$$

Gradient Ascent

This maximisation is effected by first computing the gradient and curvature of $L(\lambda)$ at the current parameter estimate, λ^{old}

$$\begin{aligned} j_\lambda(i) &= \frac{dL(\lambda)}{d\lambda(i)} \\ H_\lambda(i,j) &= \frac{d^2L(\lambda)}{d\lambda(i)d\lambda(j)} \end{aligned}$$

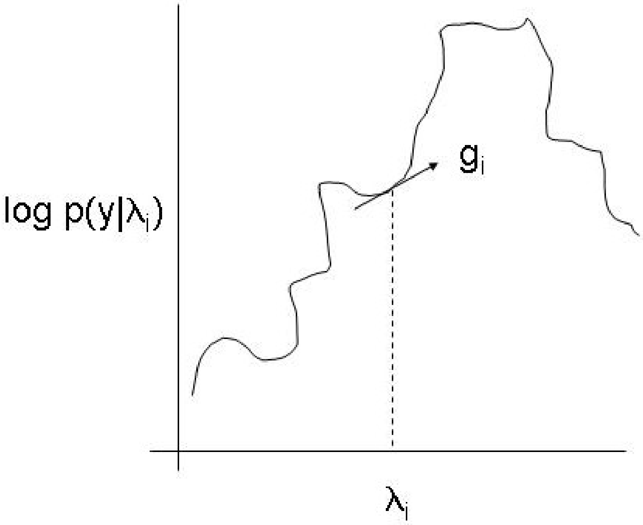
where i and j index the ith and jth parameters, j_λ is the gradient vector and H_λ is the curvature matrix (Friston et al. 2002). The new estimate is then given by

$$\lambda^{new} = \lambda^{old} - H_\lambda^{-1} j_\lambda$$

This is known as a Newton method in the optimisation literature (Press, 1988).

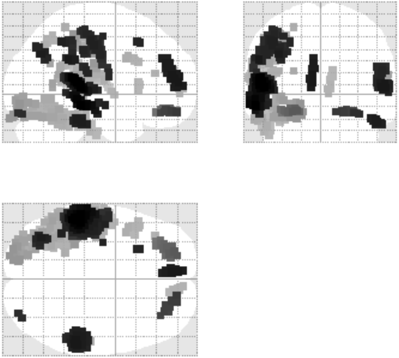
Gradient Ascent

Updating hyperparameters via gradient ascent of evidence.



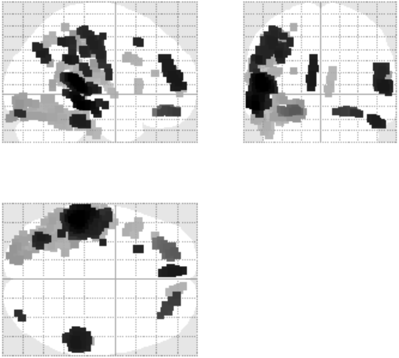
MEG Source Reconstruction

Implemented in SPM as the COH option. This is similar to the LORETA method. Here we set $\lambda_2 = 0.01$ (as prev lecture).



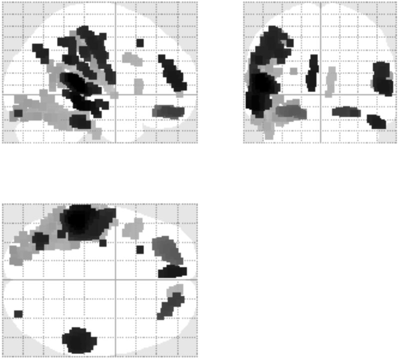
MEG Source Reconstruction

Here we set $\lambda_2 = 0.1$ (as prev lecture).



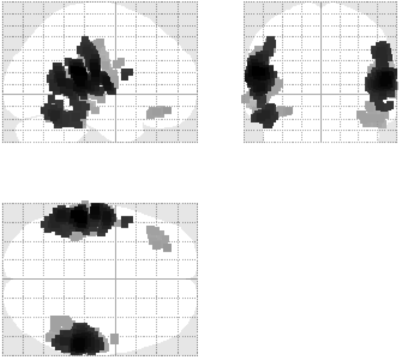
MEG Source Reconstruction

Here we set $\lambda_2 = 0.1$ (as prev lecture).



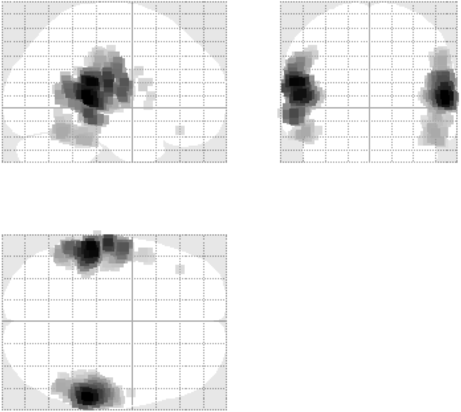
MEG Source Reconstruction

Here we set $\lambda_2 = 1$ (as prev lecture).

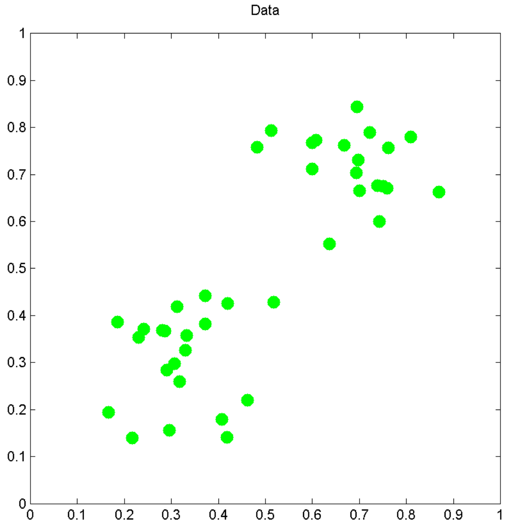


MEG Source Reconstruction

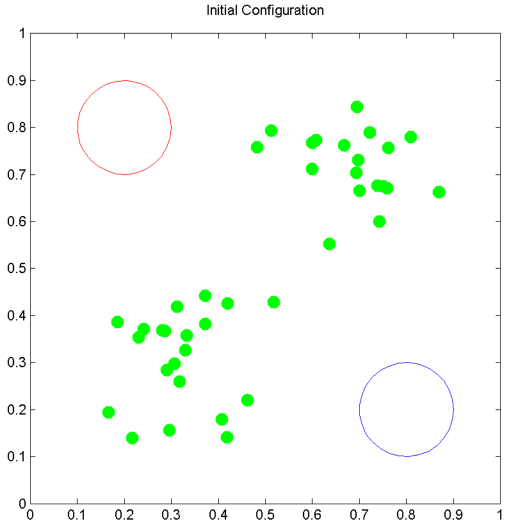
Hyperparameters set using Empirical Bayes.



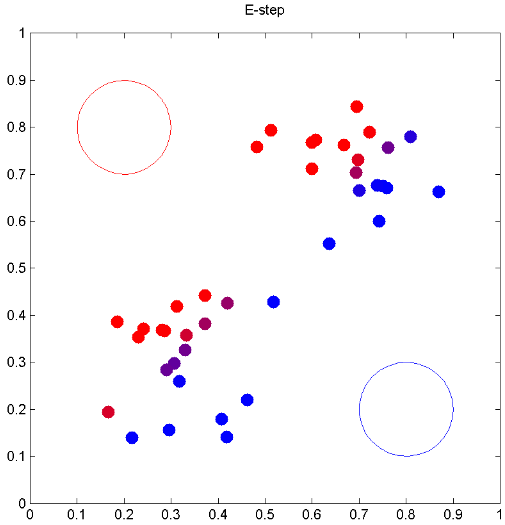
Clustering



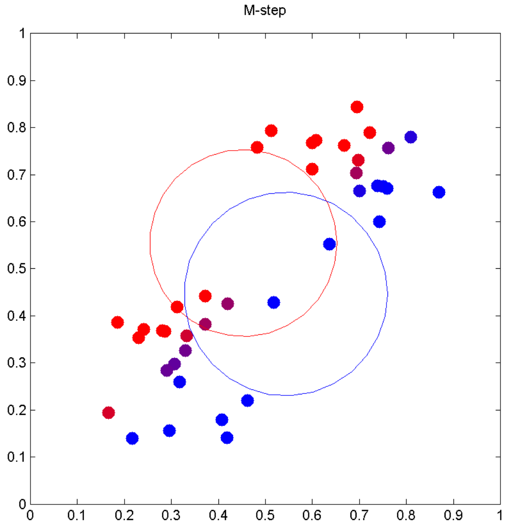
Clustering



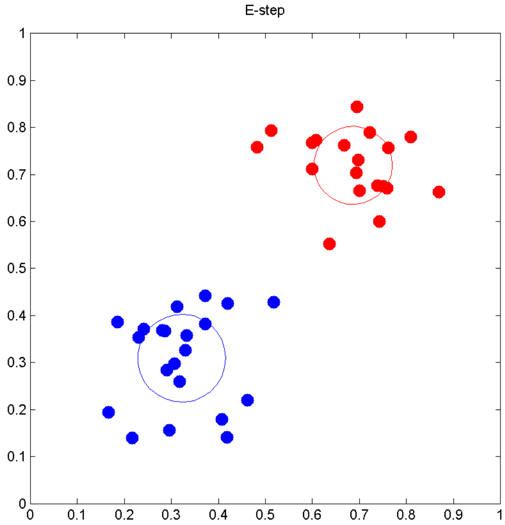
Clustering



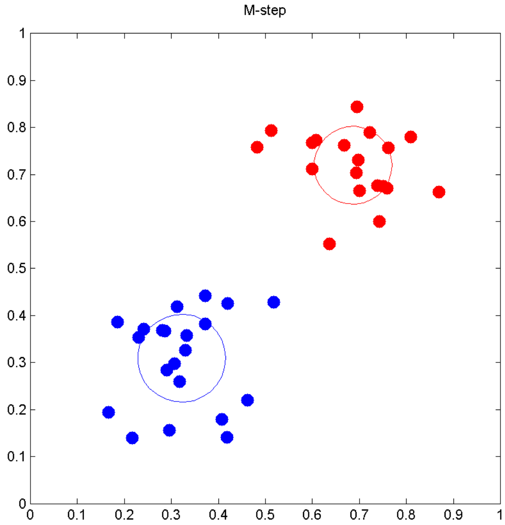
Clustering



Clustering

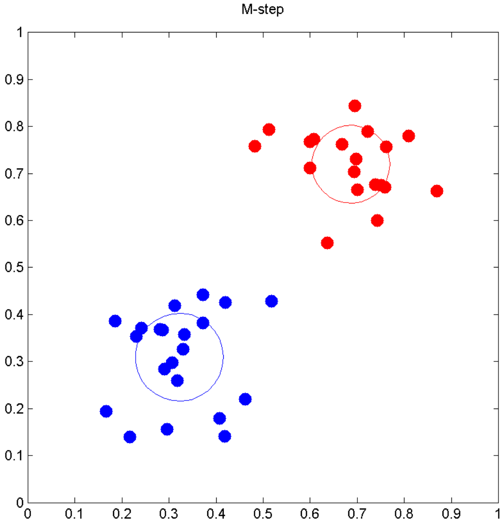


Clustering



Clustering

Gaussian Mixture Modelling in Netlab software.



Demo demgmm1.m

Sparse Coding

Learn a statistical model of natural image (patches)

$$y = Wx + e$$

Learn both W and x !

Visual Coding

We can also write

$$y = \sum_{i=1}^{p} w_i x_i + e$$

If there are d image patch elements then for $p > d$ we have an overcomplete basis. Usually $p < d$.

We wish to learn both w_i and x_i.

Sparse Coding

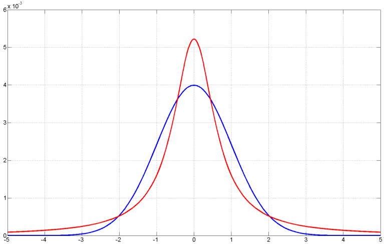
Olshausen and Field (1996) propose a sparse coding model of natural images. The likelihood is

$$p(y|W, x) = \mathrm{N}(y; Wx, \lambda_y^{-1} I)$$

But importantly, they also define a prior over coefficients

$$p(x) = \prod_i p(x_i)$$

where $p(x_i)$ is a *sparse* prior. This can be any distribution which is more peaked around zero than a Gaussian.



This means we expect most coefficients to be small, with a few being particularly large.

MAP Learning

Again, we need to learn both W and x. The posterior density is given by Bayes rule

$$p(W, x|y) = \frac{p(y|W, x)p(x)}{p(y)}$$

The Maximum A Posterior (MAP) estimate is given by

$$W_{MAP} = \arg\max_{w} p(W, x|y)$$

Because the maxima of $\log x$ is the same as the maximum of x we can also write

$$W_{MAP} = \arg\max_{W} L(W, x)$$

where

$$L = \log[p(y|W, x)p(x)]$$

is the joint log likelihood.

Learning

For the *i*th basis function

$$\tau_w \frac{dw_i}{dt} = \frac{dL}{dw_i}$$

This gives

$$\tau_w \frac{dw_i}{dt} = \lambda_y (y - Wx) x_i$$

which is simply the Delta rule.

Learning

For the ‘activations’, x, we have

$$\tau \frac{dx}{dt} = \frac{dL}{dx}$$

This gives

$$\tau \frac{dx}{dt} = \lambda_y W^T e - \sum_i g(x_i)$$

where

$$g(x_i) = \frac{d \log p(x_i)}{dx_i}$$

is the derivative of the log of the prior. Olshausen and Field have used a Cauchy density

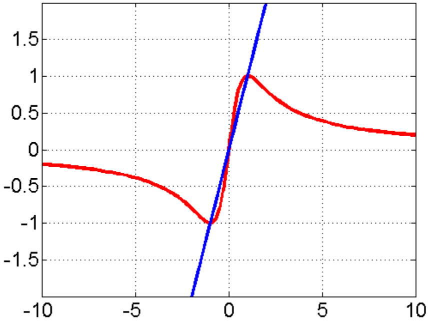
$$p(x) = \frac{1}{\pi(1 + x^2)}$$

Learning

This gives

$$\tau \frac{dx_i}{dt} = \lambda_y w_i^T e - g(x_i)$$

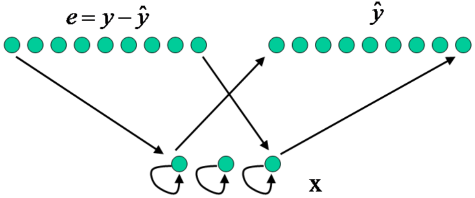
The figures shows $g(x_i) = x_i$ for Gaussian priors (blue) and $g(x_i) = 2x_i/(1+x_i^2)$ for Cauchy priors (red)



Self-Inhibition

In terms of the neural implementation we must add *self-inhibition* to the activation units, which is linear for Gaussian priors and nonlinear for Cauchy priors

$$\tau \frac{dx_i}{dt} = \lambda_y w_i^T e - g(x_i)$$



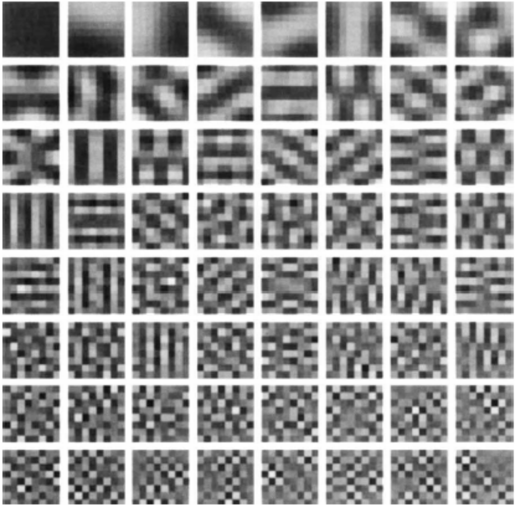
For Gaussian priors the amount of inhibition is proportional to the activation, whereas for Cauchy priors large activations are not inhibited.

Original Images

Ten images of natural scenes were low-pass filtered.

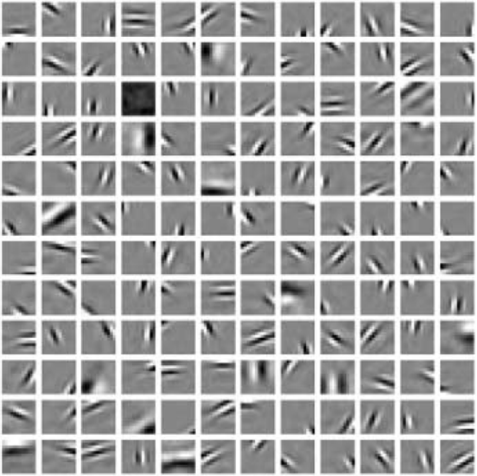
Principal Component Analysis

Receptive fields from PCA (Gaussian priors).



Receptive Fields from Sparse Coding

This produced receptive fields that are spatially localised, oriented and range over different spatial scales, much like the simple cells in V1.



References

G. Barnes (2010) MEG Source Localisation, SPM Manual, Chapter 35

C. Bishop (1995) Neural Networks for Pattern Recognition. OUP.

B. Carlin and T. Louis (1996). Bayes and Empirical Bayes Methods for Data Analysis, 2nd Edition, Chapman and Hall.

K. Friston et al. (2002) Neuroimage (16), 465-483

K. Friston and W. Penny (2003) Neuroimage (19) 1240-1249.

B. Olshausen and D. Field (1996) Nature 381, 607-609.

W. Press et al (1988) Numerical Recipes. Cambridge.

SPM Manual. http://www.fil.ion.ucl.ac.uk/spm/doc/