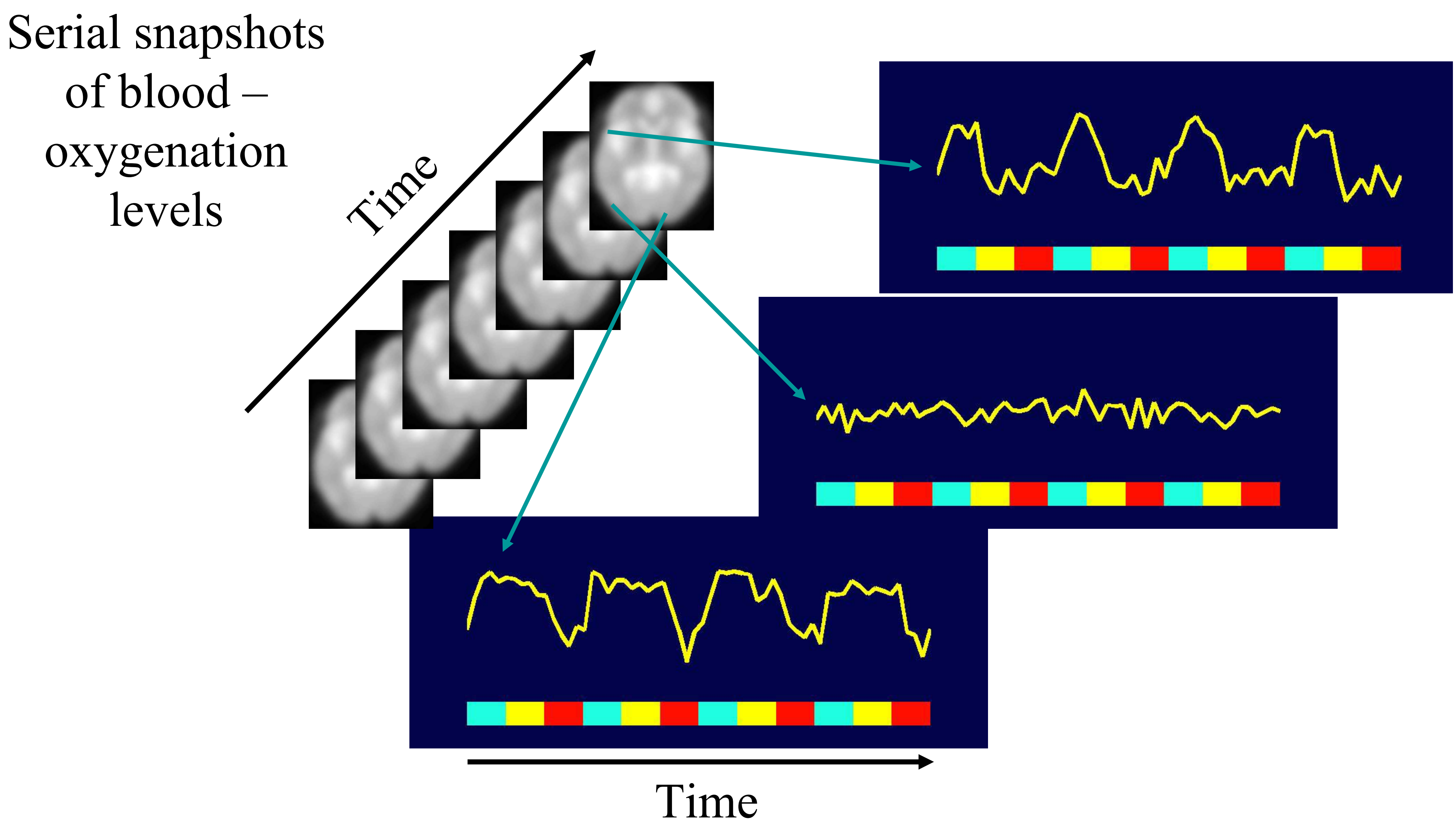
Generative Models for Brain Imaging

Will Penny

Wellcome Department of Imaging Neuroscience, UCL
http://www.fil.ion.ucl.ac.uk/~wpenny

Natural Computing Applications Forum (NCAF), ‘Cognitive Systems – from Neural Imaging to Neurocomputing’, 18-19 May, 2005, York University, UK.

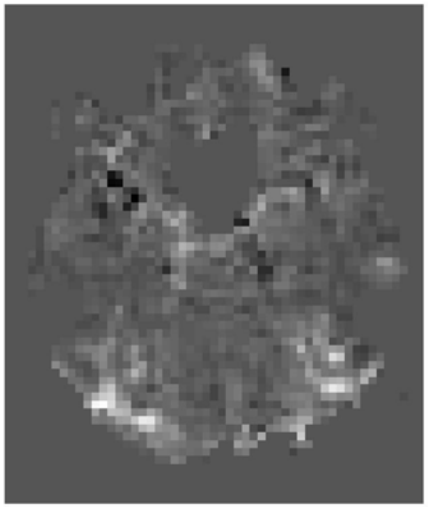
2D fMRI time-series



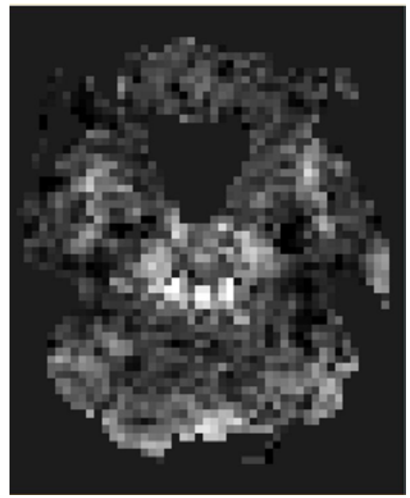
Motivation

Even without applied spatial smoothing, activation maps (and maps of eg. AR coefficients) have spatial structure

Contrast



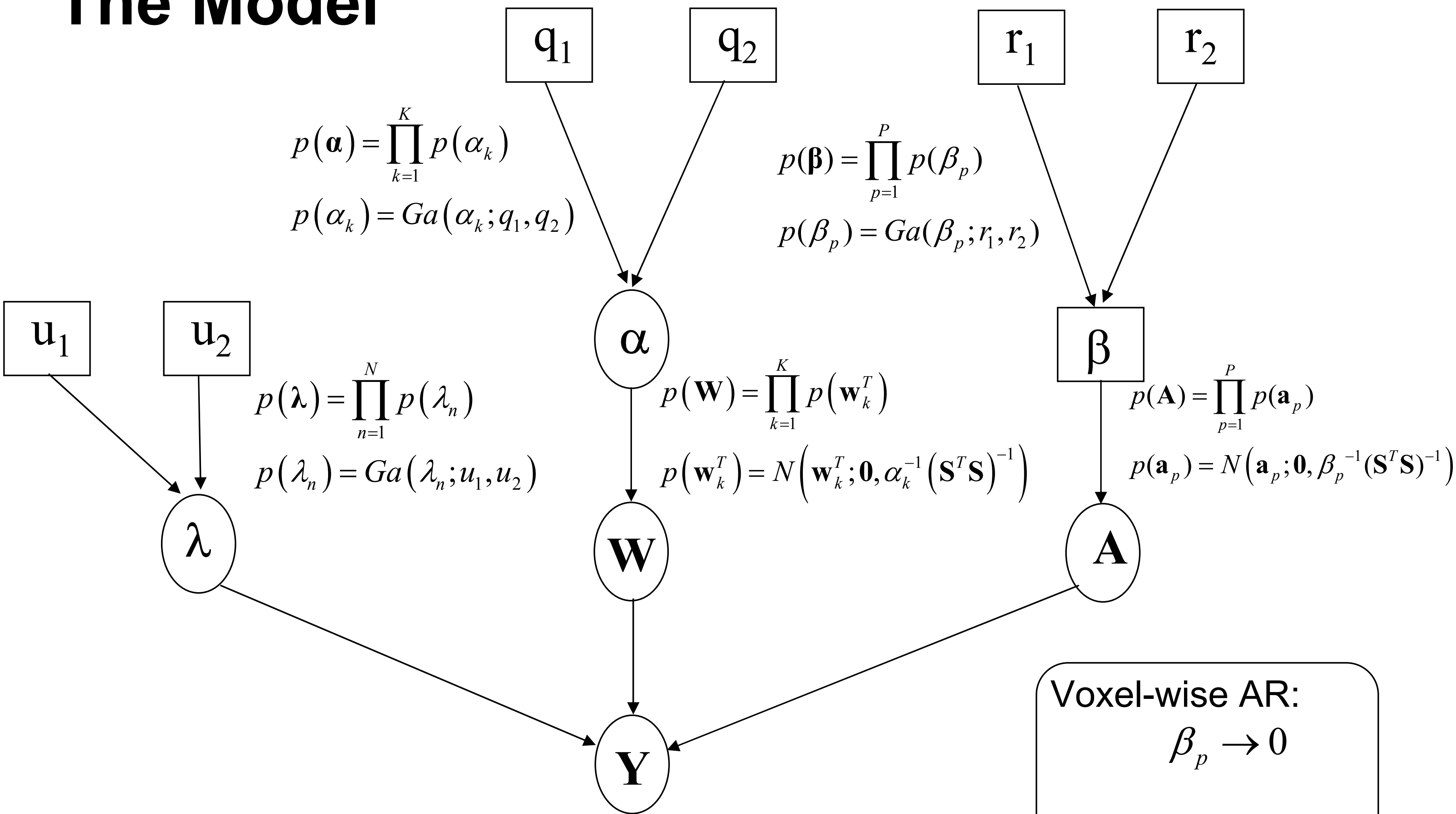
AR(1)



We can increase the sensitivity of our inferences by smoothing *data* with Gaussian kernels (SPM2). This is worthwhile, but crude. Can we do better with a spatial model (SPM5) ?

Aim: For SPM5 to remove the need for spatial smoothing just as SPM2 removed the need for temporal smoothing

The Model



$$p(\boldsymbol{\alpha}) = \prod_{k=1}^{K} p(\alpha_k)$$

$$p(\alpha_k) = Ga(\alpha_k; q_1, q_2)$$

$$p(\boldsymbol{\beta}) = \prod_{p=1}^{P} p(\beta_p)$$

$$p(\beta_p) = Ga(\beta_p; r_1, r_2)$$

$$p(\boldsymbol{\lambda}) = \prod_{n=1}^{N} p(\lambda_n)$$

$$p(\lambda_n) = Ga(\lambda_n; u_1, u_2)$$

$$p(\mathbf{W}) = \prod_{k=1}^{K} p(\mathbf{w}_k^T)$$

$$p(\mathbf{w}_k^T) = N\left(\mathbf{w}_k^T; \mathbf{0}, \alpha_k^{-1}\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\right)$$

$$p(\mathbf{A}) = \prod_{p=1}^{P} p(\mathbf{a}_p)$$

$$p(\mathbf{a}_p) = N\left(\mathbf{a}_p; \mathbf{0}, \beta_p^{-1}(\mathbf{S}^T\mathbf{S})^{-1}\right)$$

Y=XW+E

[TxN] [TxK] [KxN] [TxN]

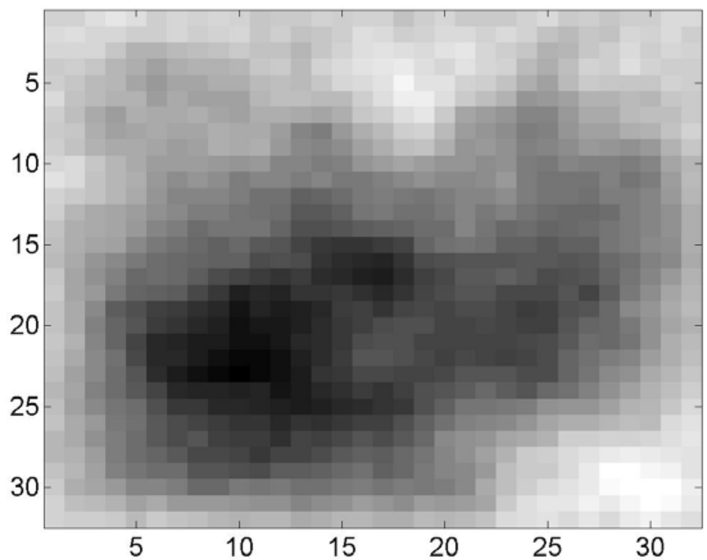
Voxel-wise AR:

$$\beta_p \to 0$$

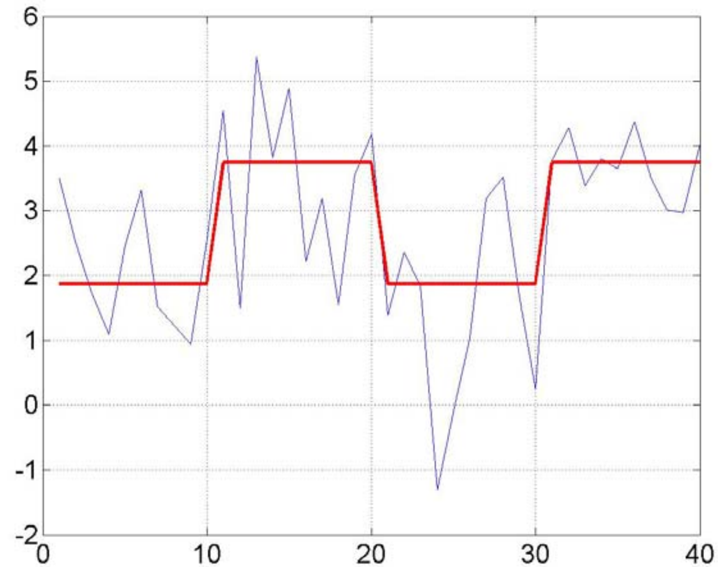
Spatial pooled AR:

$$\beta_p \to \infty$$

Synthetic Data 1 : from Laplacian Prior



reshape(w_1,32,32)



t

Prior, Likelihood and Posterior

In the prior, **W** factorises over k and **A** factorises over p:

$$p(\mathbf{W},\mathbf{A},\boldsymbol{\lambda},\boldsymbol{\alpha},\boldsymbol{\beta}) = \left(\prod_k p(\mathbf{w}_k \mid \alpha_k)\, p(\alpha_k \mid q_1, q_2)\right)\left(\prod_p p(\mathbf{a}_p \mid \beta_p)\, p(\beta_p \mid r_1, r_2)\right)$$
$$\left(\prod_n p(\lambda_n \mid u_1, u_2)\right)$$

The likelihood factorises over n:

$$p(\mathbf{Y} \mid \mathbf{W},\mathbf{A},\boldsymbol{\lambda}) = \prod_n p(\mathbf{y}_n \mid \mathbf{w}_n, \mathbf{a}_n, \lambda_n)$$

The posterior over W therefore does'nt factor over k or n. It is a Gaussian with an NK-by-NK full covariance matrix. This is unwieldy to even store, let alone invert ! So exact inference is intractable.

Variational Bayes

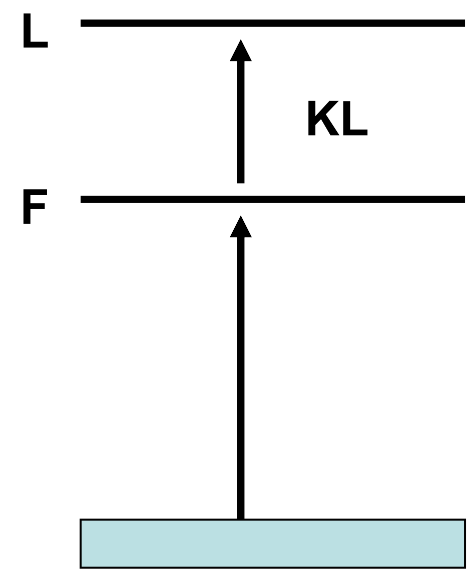
$$p(\mathbf{Y}) = \frac{p(\mathbf{Y},\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \mathbf{Y})}$$

$$\log p(\mathbf{Y}) = \log p(\mathbf{Y},\boldsymbol{\theta}) - \log p(\boldsymbol{\theta} \mid \mathbf{Y})$$

$$\log p(\mathbf{Y}) = \int q(\boldsymbol{\theta}) \log p(\mathbf{Y},\boldsymbol{\theta}) d\boldsymbol{\theta} - \int q(\boldsymbol{\theta}) \log p(\boldsymbol{\theta} \mid \mathbf{Y}) d\boldsymbol{\theta}$$

$$\log p(\mathbf{Y}) = \int q(\boldsymbol{\theta}) \log \frac{p(\mathbf{Y},\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} + \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \mathbf{Y})} d\boldsymbol{\theta}$$

$$L = F + KL$$



Variational Bayes

If you assume posterior factorises

$$q(\boldsymbol{\theta}) = \prod_i q(\boldsymbol{\theta}_i)$$

then F can be maximised by letting

$$q(\boldsymbol{\theta}_i) = \frac{\exp[I(\boldsymbol{\theta}_i)]}{\int \exp[I(\boldsymbol{\theta}_i)]d\boldsymbol{\theta}_i}$$

where

$$I(\boldsymbol{\theta}_i) = \int q(\boldsymbol{\theta}_{/i}) \log p(\mathbf{Y}, \boldsymbol{\theta}) d\boldsymbol{\theta}_{/i}$$

Variational Bayes

In the prior, **W** factorises over k and **A** factorises over p:

$$p(\mathbf{W},\mathbf{A},\boldsymbol{\lambda},\boldsymbol{\alpha},\boldsymbol{\beta}) = \left(\prod_k p(\mathbf{w}_k \mid \alpha_k)\, p(\alpha_k \mid q_1, q_2)\right)\left(\prod_p p(\mathbf{a}_p \mid \beta_p)\, p(\beta_p \mid r_1, r_2)\right)$$
$$\left(\prod_n p(\lambda_n \mid u_1, u_2)\right)$$

In *chosen* approximate posterior, **W** and **A** factorise over n:

$$q(\mathbf{W},\mathbf{A},\boldsymbol{\lambda},\boldsymbol{\alpha} \mid \mathbf{Y}) = \left(\prod_k q(\alpha_k \mid \mathbf{Y})\right)\left(\prod_p q(\beta_p \mid \mathbf{Y})\right)\left(\prod_n q(\mathbf{w}_n \mid \mathbf{Y})\, q(\mathbf{a}_n \mid \mathbf{Y})\, q(\lambda_n \mid \mathbf{Y})\right)$$

So, in the posterior for W we only have to store and invert N K-by-K covariance matrices.

Updating approximate posterior

Regression coefficients, **W**

$$q(\mathbf{w}_n) = N\left(\mathbf{w}_n; \hat{\mathbf{w}}_n, \hat{\mathbf{\Sigma}}_n\right)$$

$$\widehat{\mathbf{w}}_n = \hat{\mathbf{\Sigma}}_n \left(\overline{\lambda_n} \tilde{\mathbf{b}}_n^{\ T} + \mathbf{r}_n\right)$$

$$\hat{\mathbf{\Sigma}}_n = \left(\overline{\lambda_n} \tilde{\mathbf{A}}_n + \mathbf{B}_{nn}\right)^{-1}$$

$$\mathbf{B} = \mathbf{H}\left(diag\left[\boldsymbol{\alpha}\right] \otimes \mathbf{S}^T\mathbf{S}\right)\mathbf{H}^T$$

$$\mathbf{r}_n = -\sum_{i=1, i\neq n}^{N} \mathbf{B}_{ni} \hat{\mathbf{w}}_i$$

AR coefficients, **A**

$$q(\mathbf{a}_n) = N(\mathbf{a}_n; \mathbf{m}_n, \mathbf{V}_n)$$

$$\mathbf{V}_n = \left(\lambda_n \widetilde{\mathbf{C}}_n + \mathbf{J}_{nn}\right)^{-1}$$

$$\mathbf{m}_n = \mathbf{V}_n \left(\lambda_n \tilde{\mathbf{d}}_n + \mathbf{j}_n\right)$$

$$\mathbf{J} = \mathbf{H}_a \left(diag(\overline{\boldsymbol{\beta}}) \otimes \mathbf{S}^T\mathbf{S}\right)\mathbf{H}_a^{\ T}$$

$$\mathbf{j}_n = -\sum_{i=1, i\neq n}^{N} \mathbf{J}_{ni} \mathbf{m}_i$$

Spatial precisions for **W**

$$q(\boldsymbol{\alpha}) = \prod_{k=1}^{K} q(\alpha_k)$$

$$q(\alpha_k) = Ga(\alpha_k; g_k, h_k)$$

$$\frac{1}{g_k} = \frac{1}{2}\left[Tr\left(\hat{\mathbf{\Sigma}}_k \mathbf{S}^T\mathbf{S}\right) + \widehat{\mathbf{w}}_k^{\ T} \mathbf{S}^T\mathbf{S}\widehat{\mathbf{w}}_k\right] + \frac{1}{q_1}$$

$$h_k = \frac{N}{2} + q_2$$

$$\overline{\alpha}_k = g_k h_k$$

Spatial precisions for **A**

$$q(\boldsymbol{\beta}) = \prod_{p=1}^{P} q(\beta_p)$$

$$q(\beta_p) = Ga(\beta_p; r_{1p}, r_{2p})$$

$$\frac{1}{r_{1p}} = \frac{1}{2}\left(Tr(\mathbf{V}_p \mathbf{S}^T\mathbf{S}) + \mathbf{m}_p^{\ T} \mathbf{S}^T\mathbf{S}\mathbf{m}_p\right) + \frac{1}{r_1}$$

$$r_{2p} = \frac{P}{2} + r_2$$

$$\overline{\beta}_p = r_{1p} r_{2p}$$

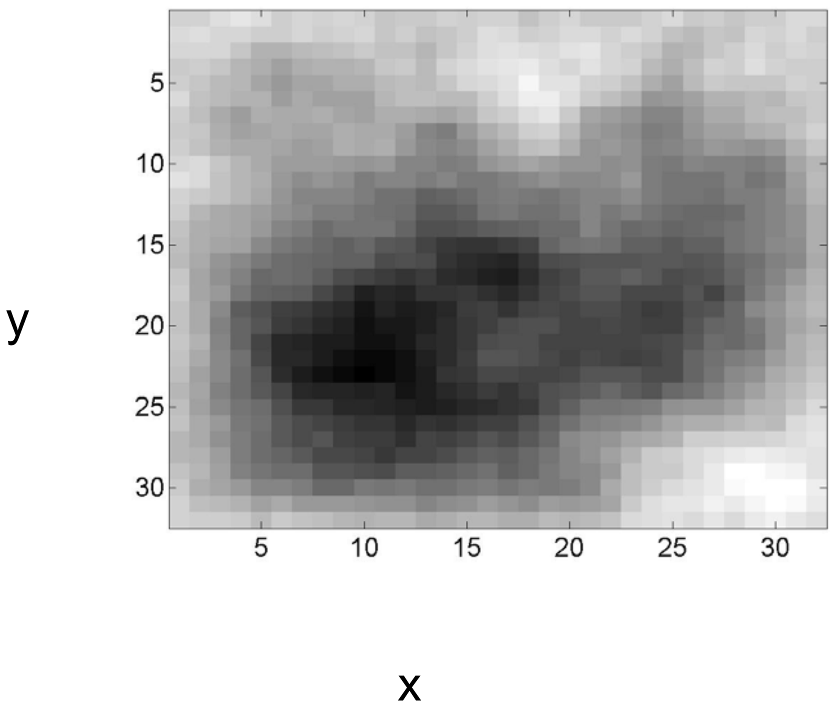
Observation noise

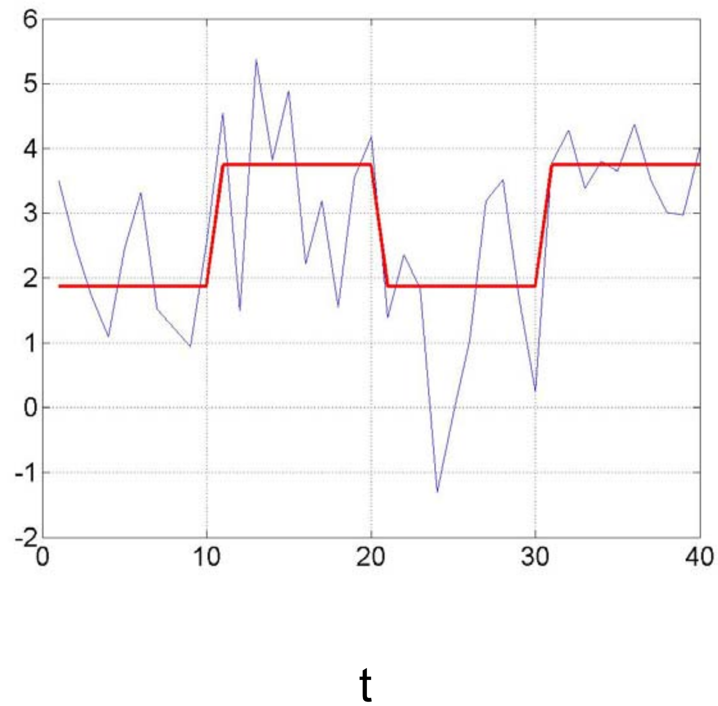
$$q(\lambda_n) = Ga(\lambda_n; b_n, c_n)$$

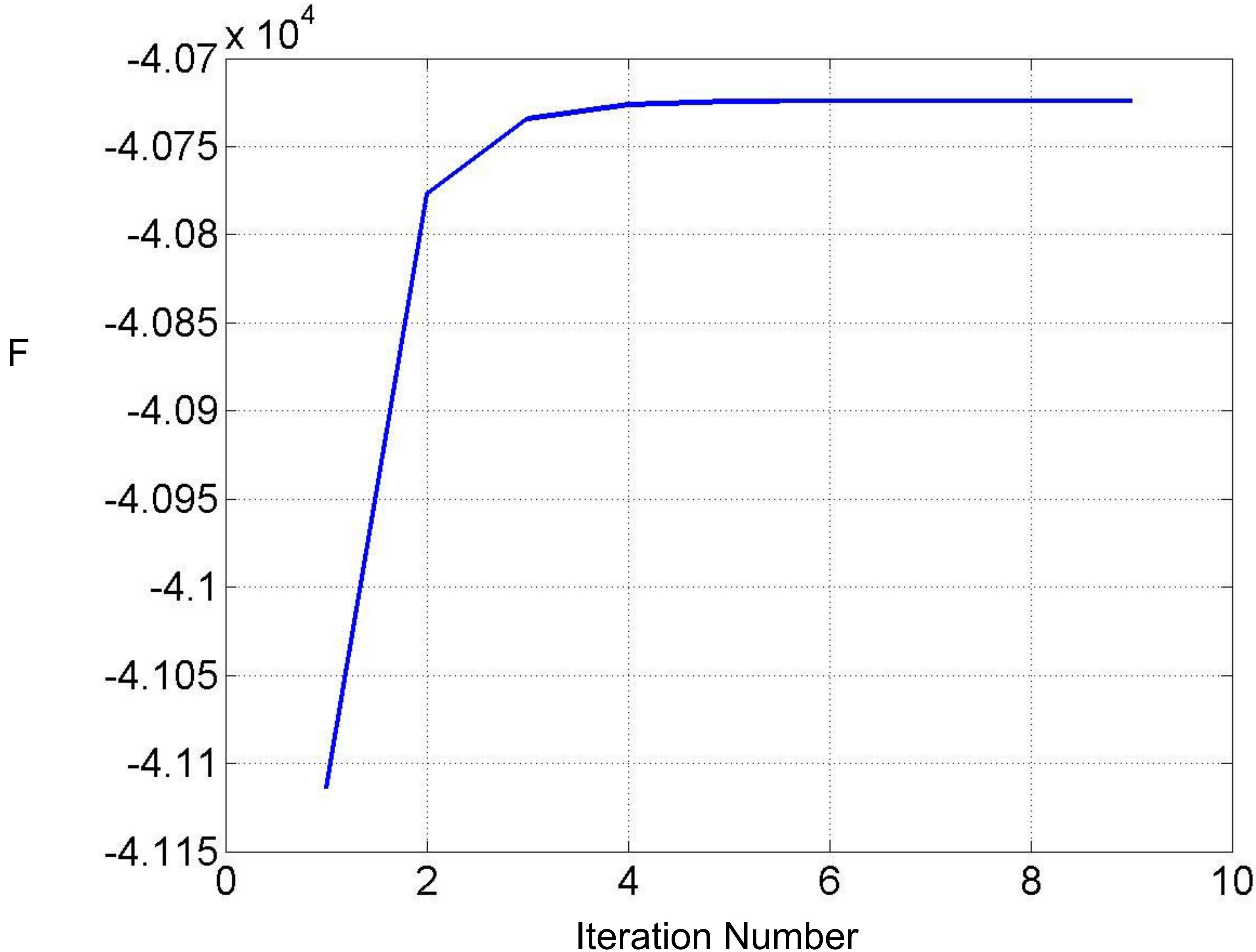
$$\frac{1}{b_n} = \frac{\widetilde{G}_n}{2} + \frac{1}{u_1}$$

$$c_n = \frac{T}{2} + u_2$$

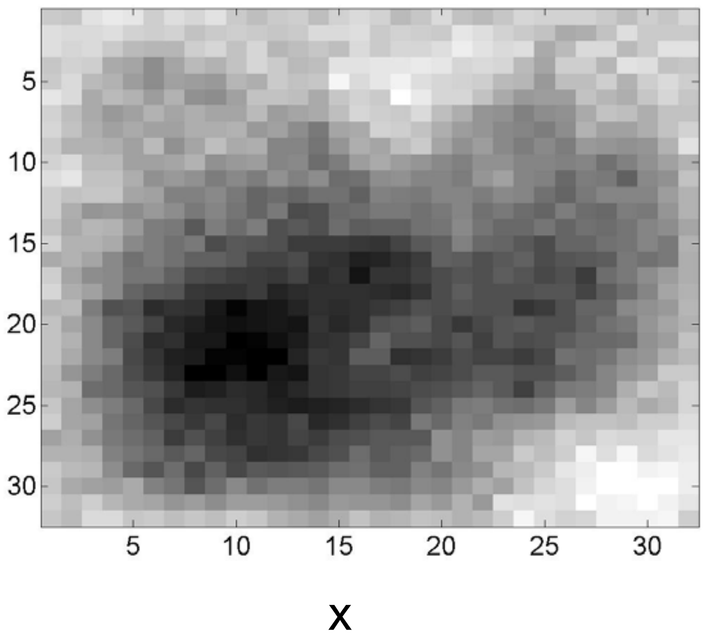
Synthetic Data 1 : from Laplacian Prior





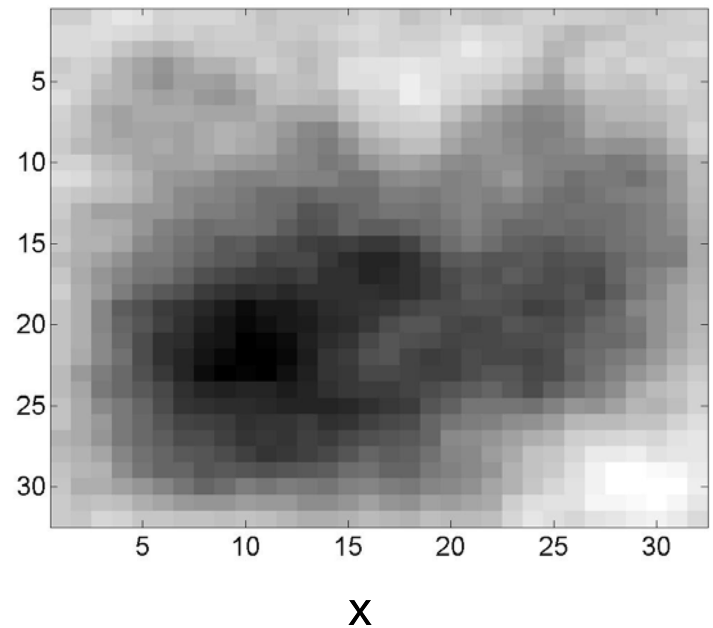

x 10^4
-4.07
-4.075
-4.08
-4.085
-4.09
-4.095
-4.1
-4.105
-4.11
-4.115
F
0
2
4
6
8
10
Iteration Number

Least Squares



Coefficients = 1024

VB – Laplacian Prior

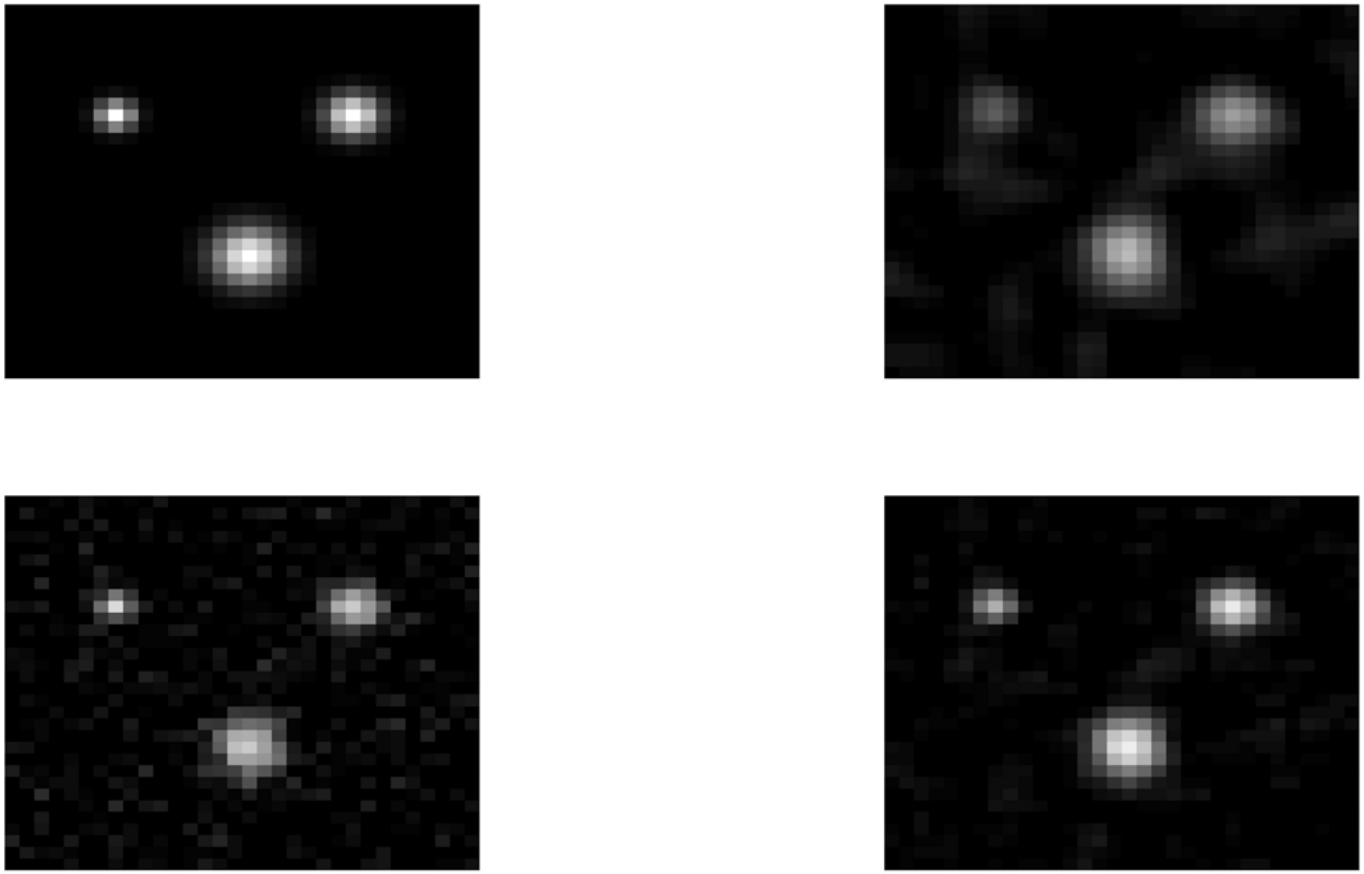


`Coefficient RESELS' = 366

Synthetic Data II : blobs

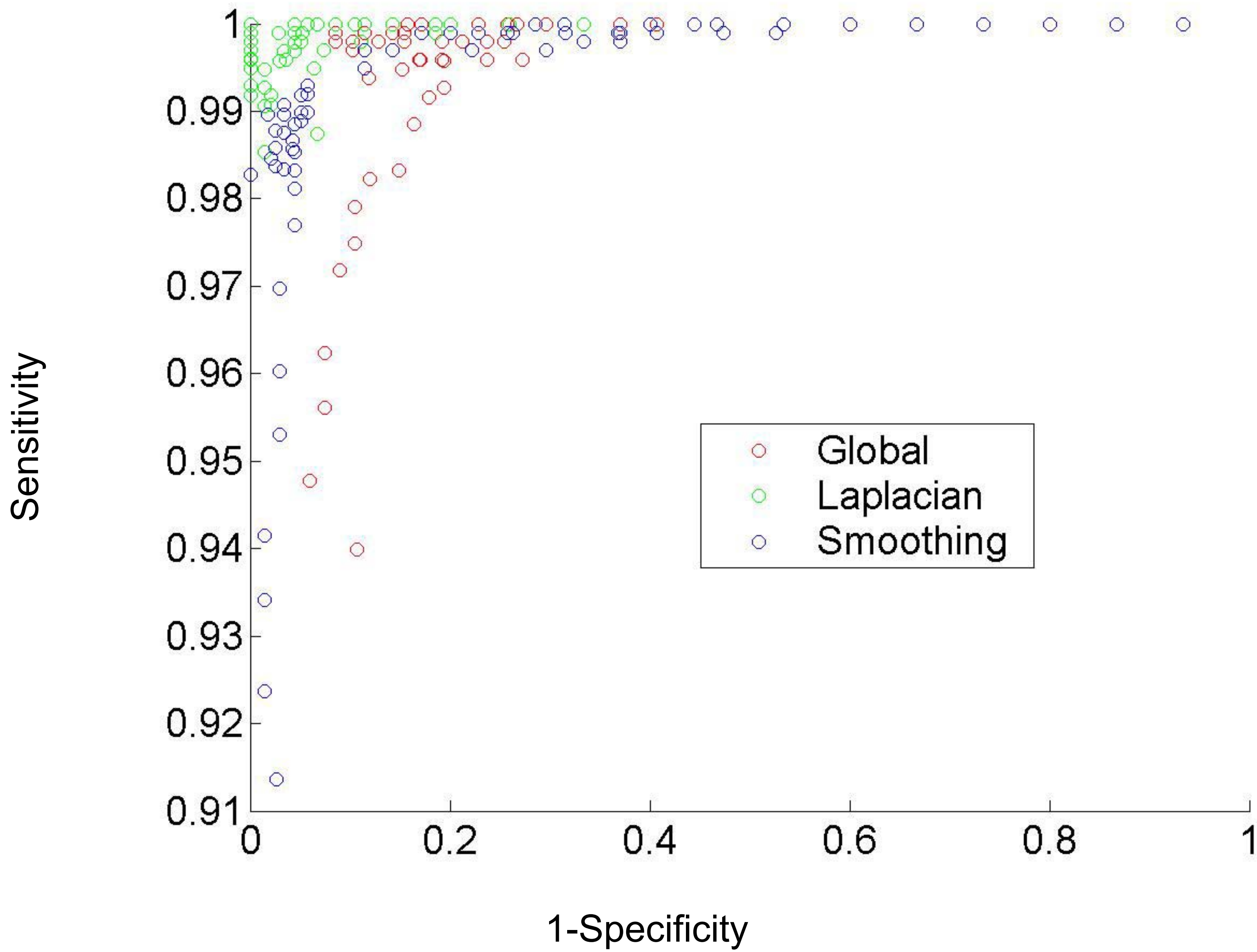
True

Smoothing



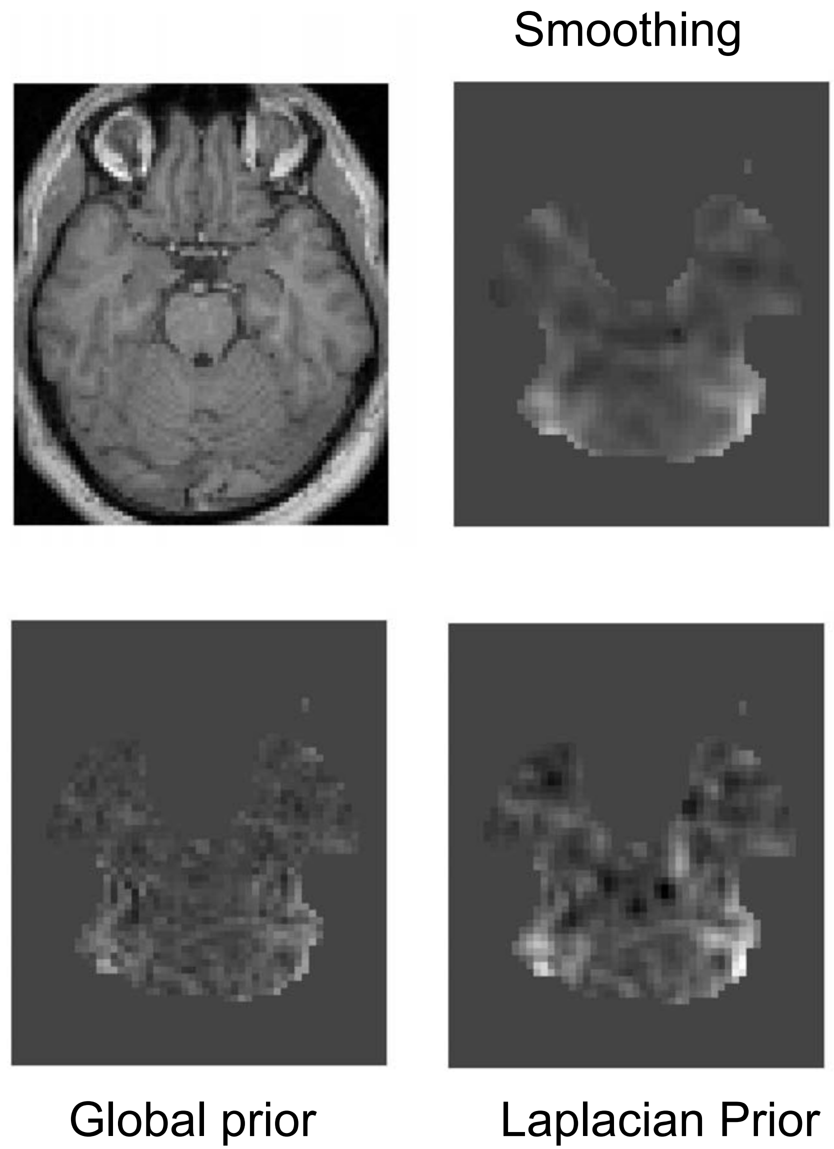
Global prior

Laplacian prior



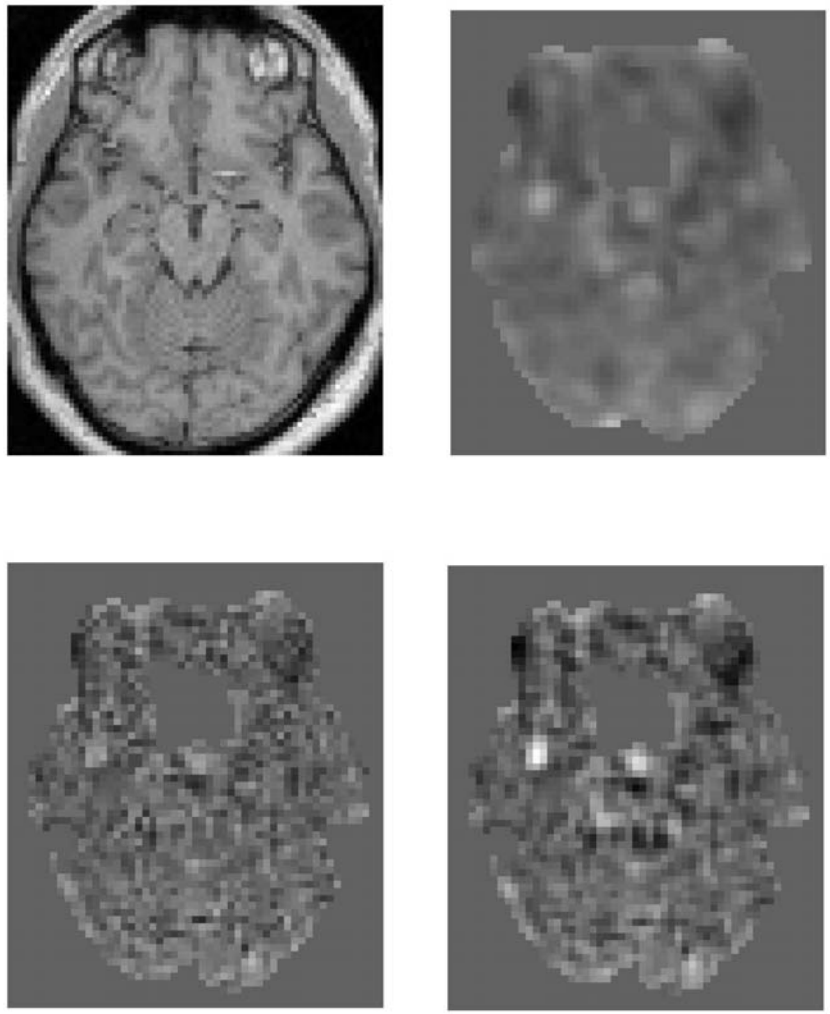
Sensitivity
1-Specificity
1
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0.92
0.91
0
0.2
0.4
0.6
0.8
1
Global
Laplacian
Smoothing

Event-related fMRI: Faces versus chequerboard



Event-related fMRI: Familiar faces versus unfamiliar faces

Smoothing



Global prior Laplacian Prior