An Approximate Gibbs Sampler for Adaptive Smoothing of Functional MR Images

M. G. Hughes[a], V. J. Schmid[b]

[a] *Fakultät für Informatik und Mathematik, Hochschule München, Lothstrasse 34, 80335 München, Germany*
[b] *Institut für Statistik, Ludwig-Maximilians Universität München, Ludwigstrasse 33, 80539 München, Germany*

Abstract

Adaptive smoothing allows to account for edges and other sharp features when smoothing images. Brezger et al. (2007) proposed a Bayesian latent adaptive Gaussian Markov random field model for the analysis of functional magnetic resonance imaging (fMRI) data. However, MCMC inference for their model includes computation of thousands of Cholesky decompositions of big matrices in each iteration. Therefore, we propose to use a much faster approximate Gibbs sampling algorithm. Similar to pseudo-likelihood procedures, we do not expect the algorithm to estimate the correct posterior, but to provide estimates of the adaptive Gaussian Markov random field similar to the computationally much more demanding correct algorithm. We validate the proposed technique on simulated and real data. The results demonstrate that the proposed approximate Gibbs sampler can indeed provide similar results for the latent adaptive Gaussian Markov random field.

Keywords: Adaptive Smoothing, Gaussian Markov Random Fields, Imaging, Latent models, Markov Chain Monte Carlo

1. Introduction

Gaussian Markov Random fields (GMRF) are frequently used in imaging applications, often as latent field in order to globally smooth the images (Winkler, 2002). Nevertheless, in medical imaging applications a global smoothing is not always desirable, as the tissue of interest typically contains sharp features like tissue edges (Schmid et al., 2006; Schmid, 2011). For the analysis of functional magnetic resonance imaging (fMRI), Brezger et al. (2007) proposed a latent adaptive GMRF approach, which allows to estimate local smoothing weights along with the latent GMRF, adaptive to the data in a fully Bayesian approach. However, the proposed Monte Carlo Markov chain (MCMC) algorithm is computationally very expensive, as it depends on the explicit computation of Eigenvalues of a high dimensional matrix for each weight in each iteration. In this paper, we propose an approximate MCMC algorithm for adaptive GMRF models and compare the results of the proposed approach with the original algorithm. Similar to pseudo- or quasi-likelihood procedures, we do not expect the algorithm to estimate the correct posterior, but to provide good estimates of the adaptive Gaussian Markov random field.

In Neuroimaging, fMRI is a standard tool for classifying brain regions according to their function. (Lange, 1996; Stippich, 2007). A great advantage of fMRI is the non-invasive procedure in which the images can be retrieved by using magnetic resonance (MR) scanners. The change of blood flow in the brain is measured while the subject undergoes some form of external stimulation. The neural system of the brain reacts by sending the information from neuron to neuron. Resulting from this are magnetic resonance images which visualize the change in blood oxygenation, the so called *Blood Oxygenation Level Dependent* (BOLD) effect. Neurologists are interested in which brain region reacts to what kind of external stimulus. These are referred

Email addresses: hughesgm@me.com (M. G. Hughes), volker.schmid@lmu.de (V. J. Schmid)

to as the Regions of Interest (ROI). Tests are performed on healthy subjects to see whether these respond equally and whether effects are significant. Similarly, patients who have been traumatized and are subject to a comatose condition are tested on reactions to external stimuli. Here fMRI analysis is of great interest in order to see to what extent the patient has been affected by the trauma and help provide treatment for the patients road to recovery. Other clinical usages of mapping the brain are to detect the effects of tumors, head and brain injuries or diseases such as Alzheimer's.

To obtain fMRI data, a time series of MR scans of the subjects is acquired, during which the subject is exposed to one or more external stimuli. One standard approach to analyze fMRI data are Statistical Parametric Maps (SPM) originally proposed by Friston et al. (1995). As an alternative, Posterior Probability Maps (PPM) based on Bayesian models where proposed (Gössl et al., 2001a). Both approaches are based on a general linear model (GLM) for fMRI data. We denote the observed MR signal at voxel $i = 1, \ldots, I$ at time $t = 1, \ldots, T$ as y_{it}. The external stimulus typically occurs in on-off intervals contained in a variable x_t. In order to account for the time delay d_i between external stimulus and the cerebral blood flow, a so-called *haemodynamic response function* $h(s; \theta)$, is used instead of the original stimulus (Gössl et al., 2001b). Therefore, the independent variable z_{it} is obtained through a convolution between the function h and the on/off stimulus x_{t-d_i-s}:

$$z_{it} = \sum_{s=0}^{t-d_i} h(s; \theta_i) x_{t-d_i-s}$$

The full regression model for the observed fMRI signal y_{it} is

$$y_{it} = \boldsymbol{u}_t^\top \boldsymbol{\alpha}_i + z_{it}\beta_i + \epsilon_{it}, \qquad \epsilon_{it} \sim \mathrm{N}(0, \sigma_i^2), \tag{1}$$

where $\epsilon_{it}, t = 1, \ldots, T, i = 1, \ldots, I$ are independent Gaussian errors, with variance σ_i^2. $\boldsymbol{u}_t$ is constituted by some basis functions, in order to account for arbitrary trends in the data.

For the spatial regression parameter β Gössl et al. (2001a) proposed to use a GMRF as prior distribution in an hierarchical Bayesian model, in order to smooth the effects, and, on the other hand, to "borrow strength" from neighboring voxels, hence to allow a more robust parameter estimation. Though, activation in the brain due to a specific stimulus is concentrated on specific, well defined regions and we are interested in finding borders of this regions, rather than to smooth over the field of view. Hence, adaptive smoothing was proposed by Brezger et al. (2007), where locally adaptive smoothing weights are estimated along with the regression parameters $\boldsymbol{\beta}$. However, MCMC inference for the proposed model is computationally very expensive. To this end, we propose to use an approximate MCMC algorithm, which is computationally far more efficient, but basically provides the same estimates of the latent adaptive GMRF as the algorithm proposed by Brezger et al. (2007).

This paper is organized as follows: Section 2 is dedicated to the specific theory of the adaptive GMRF models and the proposed approximate algorithm, implemented in the '*adaptsmoFMRI*' package (Hughes, 2012). In section 3, the proposed approximate algorithm is validated on simulated data. In section 4 the non-approximate and the approximate version of the algorithm are applied on a *in vivo* data set. The resulting estimates of the activation pattern is similar for both the simulated and the *in vivo* data sets. We close with conclusions and a discussion of the proposed algorithm.

2. Adaptive Spatial Smoothing Model

Following Brezger et al. (2007), we use a Bayesian hierarchical model to analyze the fMRI signal. The general linear model (1) is the first stage of the hierarchical model.

2.1. Specification of prior distributions

In the second stage of the hierarchical model, we formulate prior distributions for the unknown parameters α, β, and σ^2. The purpose of the term $\boldsymbol{u}_t^\top \boldsymbol{\alpha}_i$ is to extract certain types of trends in the data, so that only the 'true' effects are captured in β_i for each voxel. For $\boldsymbol{\alpha}_i$ we use diffuse priors $\mathrm{p}(\boldsymbol{\alpha}_i) \propto$ constant.

As prior model for the activation effect β we use a Gaussian Markov random field (Rue and Held, 2005). That is, the conditional distribution of β_i given its neighbors $\beta_{\partial i}$ is

$$\beta_i|\boldsymbol{\beta}_{i\neq j}, \tau^2, \boldsymbol{w} \sim \mathrm{N}\left(\sum_{i\sim j} \frac{w_{ij}\beta_j}{w_{i+}}, \frac{\tau^2}{w_{i+}}\right),$$

where $i \sim j$ means voxel i is neighbor of voxel j and $w_{i+} = \sum_{j:i\sim j} w_{ij}$ is the sum of locally adaptive smoothing weights between i and all nodes adjacent to i. Each weight w_{ij} determines the "smoothness" between i and j. High weights indicate that the activation β is similar in both voxels. In contrast small weights imply that the activation in both voxels can be seen as independent of each other.

From the full conditional the joint Gaussian distribution with precision matrix $\boldsymbol{K}$ can be derived:

$$p(\boldsymbol{\beta}|\boldsymbol{w}, \tau^2) \propto \tau^{(I-1)/2} \left(\prod_{i=1}^{I-1} \lambda_i\right)^{1/2} \exp\left(-\frac{1}{2\tau^2}\boldsymbol{\beta}^\top \boldsymbol{K}\boldsymbol{\beta}\right), \tag{2}$$

where $\underset{(I\times 1)}{\boldsymbol{\beta}} = (\beta_1, \ldots, \beta_I)^T$ and following $\underset{(I\times I)}{\boldsymbol{K}}$ has the following elements

$$K_{ij} = \begin{cases} w_{i+} & i = j \\ -w_{ij} & i \sim j, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Here, λ_i are the non-negative Eigenvalues of $\boldsymbol{K}$. That is, $\boldsymbol{\beta}|\boldsymbol{w}, \tau^2$ has an (improper) Gaussian distribution with expectation vector 0 and precision matrix $(1/\tau^2)\boldsymbol{K}$.

In most applications of GMRFs the weights w_{ij} are fixed (to one), wheras here the weights are treated as unknown parameters. With this, we can estimate the influence on each voxel on its neighbor, and, hence, adapt the smoothness of the latent GMRF locally according to the data. Note that $\boldsymbol{K}$ depends on the w_{ij}, and therefore the normalizing factor $\left(\prod_{i=2}^{I} \lambda_i\right)^{1/2}$ in equation (2) is not a constant, but it is dependent on the weights w_{ij}. Consequently, we have to define prior distributions on the weights w_{ij}. This prior distribution for w_{ij} is the key for the adaptiveness of the weights, as we allow them to vary stochastically. Following Brezger et al. (2007) and Rue and Held (2005) we postulate independent identically distributed gamma priors with parameters $\nu/2$ and $\nu/2$,

$$p(w_{ij}) \propto w_{ij}^{\nu/2-1} \exp\left(-w_{ij}\frac{\nu}{2}\right). \tag{4}$$

For the variance parameters $(\sigma_i^2,\ \tau)$ we assign a the conjugate inverse gamma prior distribution with parameters a, b and c, d, respectively.

2.2. Full conditional distributions and Markov Chain Monte Carlo algorithm

As in most Bayesian hierarchical models, we need a Markov Chain Monte Carlo (MCMC) algorithm to infer the posterior distribution. We use the following MCMC algorithm:

Step 1: Draw blocks $\boldsymbol{\alpha}^{(k)} = (\boldsymbol{\alpha}_1^{(k)}, \ldots, \boldsymbol{\alpha}_I^{(k)})$ from the Gaussian full conditionals, for $k = 1, \ldots, K$.

Step 2: Draw blocks $\boldsymbol{\beta}^{(k)} = (\beta_1^{(k)}, \ldots, \beta_I^{(k)}),\ k = 1, \ldots, K$ from the (multivariate) Gaussian full conditionals, given previous iterates for the weights $\boldsymbol{w}_{(k-1)}$.

Step 3: Draw weights $w_{ij}^{(k)},\ k = 1, \ldots, K$ as proposed below.

Step 4: Draw variance parameters $(\boldsymbol{\sigma}^{(k)})^2 = ((\sigma_1^{(k)})^2, \ldots, (\sigma_I^{(k)})^2$ and the hyper parameters $(\tau^{(k)})^2$ from their corresponding inverse gamma full conditionals, for $k = 1, \ldots, K$.

We denote (k) as the current iteration step and $(k-1)$ as the previous one. Steps 1, 2 and 4 are Gibbs steps, where the full conditionals are known standard distributions. In particular, with $\boldsymbol{\Sigma} = \text{diag}\left(\sigma_1^2, \ldots, \sigma_I^2\right)$, the full conditional for $\boldsymbol{\beta}$ is

$$p(\boldsymbol{\beta}|\boldsymbol{\alpha}, \boldsymbol{\Sigma}, \tau^2, \boldsymbol{w}, \boldsymbol{y}) \propto \exp\left(\boldsymbol{\beta}^\top [\boldsymbol{Z}^\top (\boldsymbol{\Sigma} \otimes \boldsymbol{I}_T)^{-1} (\boldsymbol{y} - \boldsymbol{U}\boldsymbol{\alpha})] - \frac{1}{2}\boldsymbol{\beta}^\top [\boldsymbol{Z}^\top (\boldsymbol{\Sigma} \otimes \boldsymbol{I}_T)^{-1} \boldsymbol{Z} + \tau^{-2}\boldsymbol{K}]\boldsymbol{\beta}\right), \tag{5}$$

where $\otimes$ denotes the Kronecker matrix product. This full conditional represents a multivariate normal distribution with expectation vector $\boldsymbol{\mu}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}} \left[\boldsymbol{Z}^\top (\boldsymbol{\Sigma} \otimes \boldsymbol{I}_T)^{-1} (\boldsymbol{y} - \boldsymbol{U}\boldsymbol{\alpha})\right]$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = \left[\boldsymbol{Z}^\top (\boldsymbol{\Sigma} \otimes \boldsymbol{I}_T)^{-1} \boldsymbol{Z} + \tau^{-2}\boldsymbol{K}\right]^{-1}$. Samples from this multivariate distribution can be computed efficiently following Rue and Held (2005).

2.3. Sampling the locally adaptive weight w_{ij}

In this paragraph, we describe step 3 in the MCMC algorithm as proposed by Brezger et al. (2007). The full conditional of the weights is given by

$$p(w_{ij}|\boldsymbol{\beta}, \tau^2, \boldsymbol{w_{-ij}}) \propto \left(\prod_{i=2}^{I} \lambda_i\right)^{1/2} (w_{ij})^{[\nu/2]-1} \exp\left(-w_{ij}\left[\frac{\nu}{2} + \frac{(\beta_i - \beta_j)^2}{2\tau^2}\right]\right), \tag{6}$$

where $\boldsymbol{w_{-ij}}$ is the vector of all weights but w_{ij}. This probability distribution function looks similar to a gamma distribution. Moreover, the "normalizing constant" $\left(\prod_{i=2}^{I} \lambda_i\right)^{1/2}$ depends on all w_{ij}, which not only makes (6) a non-standard distribution, but also shows that even given $\boldsymbol{\beta}$ and τ^2 all w_{ij} are *a posteriori* dependent. Brezger et al. (2007) propose to use a Metropolis-Hastings algorithm with a Gamma distribution with parameters $e' = \frac{\nu}{2}$ and $f' = \frac{\nu}{2} + \frac{(\beta_i^{(k)} - \beta_j^{(k)})^2}{2(\tau^{(k-1)})^2}$ as proposal. With this, the acceptance probability is

$$r\left(w_{ij}^*|w_{ij}^{(k-1)}\right) = \min\left\{1, \left(\prod_{i=2}^{I} \lambda_i^* / \prod_{i=2}^{I} \lambda_i^{(k-1)}\right)^{1/2}\right\} \tag{7}$$

with $*$ denoting the proposed variable from the proposal density and $(k-1)$ the current state.

2.4. Approximate Gibbs sampling algorithm

In the Metropolis-Hastings step above, the computation of the acceptance rate includes the computation of the Eigenvalues of $\boldsymbol{K}^*$ for each of the thousand of weights in each iteration. The sampling of the weights is therefore the part of the MCMC algorithm which dominates the computation time of the adaptive GMRF model. To this end, we propose to use an approximate Gibbs sampler for the local weights to speed up the computation.

For this, we use an approximation for the full conditional of w_{ij}. We approximate the product of the non-negative Eigenvalues $\prod_{i=2}^{I} \lambda_i^*$, which actually depends on w_{ij}, with a constant. *I.e.*, we assume that given all other weights the product of the non-negative Eigenvalues is independent of w_{ij}. This leads to the following approximation of the full conditional

$$p(w_{ij}|\boldsymbol{\beta}, \tau^2) \overset{a}{\propto} (w_{ij})^{[\nu/2]-1} \exp\left(-w_{ij}\left[\frac{\nu}{2} + \frac{(\beta_i - \beta_j)^2}{2\tau^2}\right]\right). \tag{8}$$

This is the probability density of a Gamma distribution with parameters e' and f'. In addition, the weights w_{ij} are assumed to be conditionally independent.

In the Metropolis-Hastings algorithm described above, the approximation, therefore, means that $\prod_{i=2}^{I} \lambda_i^* \approx \prod_{i=2}^{I} \lambda_i$, that is, the acceptance probability (7) simplifies to 1. The approximation therefore avoids the computation of the Eigenvalues for the update of each weight, which, in turn, leads to a immense reduction of computation time.

2.5. Implementation

The proposed methods were implemented in the R (R Core Team, 2012) package '*adaptsmoFMRI*' (Hughes, 2012). The package provides the necessary tools for computing the effects of external stimulus on the human brain, by including a GMRF model with adaptive weights as described above. In each iteration step the random numbers of the unknown variables building the Markov chain are drawn from their corresponding distributions, using random number generators provided in '*MCMCpack*' and '*mvtnorm*' (Martin et al., 2011; Genz et al., 2012). For efficient simulation we make use of the sparseness of the precision matrix using the '*Matrix*' package (Bates and Maechler, 2012). Functions for simulated data, as well as for real data are implemented. Furthermore, an advanced plot function, depending on the '*spatstat*' package (Baddeley and Turner, 2005), can be invoked for depicting the weights in between pixels.

To reduce the burden of computation time, various functions from the '*parallel*' package are implemented for parallel computation where applicable (Eugster et al., 2011). Especially, when using the non-approximate model for two covariates it is of substantial advantage to parallelize the Cholesky decomposition of each precision matrix with order $O(n)$. Here the overall elapsed time for execution is decreased by more than 30%.

3. Simulation study

In order to evaluate the proposed approximate algorithm, we generate simulated fMRI data by predefining an activated and non-activated area on a grid following Brezger et al. (2007).

3.1. Simulation setting

Fig. 1(c) shows the true activation area $\boldsymbol{\beta}$, a cylinder on a 20×20 grid. We use the following model to simulate the fMRI signal:

$$y_{it} = \beta_i z_{it} + \epsilon_{it}, \quad \epsilon_{it} \sim \mathrm{N}(0, \sigma_i^2), \quad t = 1, \dots 210, \quad i = 1, \dots, 400,$$

where

$$z_{it} = \begin{cases} -0.5 & \text{stimulus off} \\ 0.5 & \text{stimulus on,} \end{cases}$$

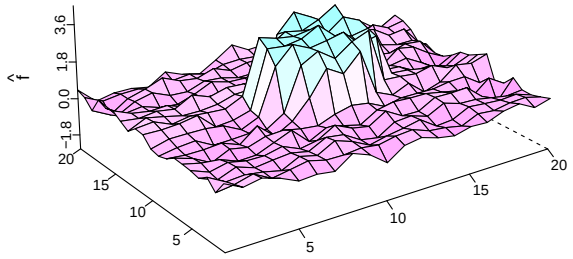
where the on-off periods are 30 seconds (equally) long to simulate a visual stimulus over a 210 seconds time period. We use pixel-specific variances as $\sigma_i^2 = 25 + 2\mathrm{N}(0, 1)$, '*to achieve a realistically low signal-to-noise ratio*', (Brezger et al., 2007).

For the analysis, we use hyper parameters $a = 0.001$ and $b = 30$ for the inverse gamma prior distribution of σ_i^2, $c = 1200$ and $d = 1$ for the inverse gamma prior distribution of τ^2, and $\nu = 1$ for the gamma prior distribution of w_{ij}. Fig. 1 depicts the estimated activation surfaces using the non-approximate and the approximate adaptive algorithm. Visually, the results from both algorithms do not differ from each other. In both cases the model captures the edges of the activated area and does not smooth over the edges by neglecting the difference from the change in height of the effects. Fig. 2 shows the scatter plot between $\hat{\beta}$ using the non-approximate and the approximate algorithm.

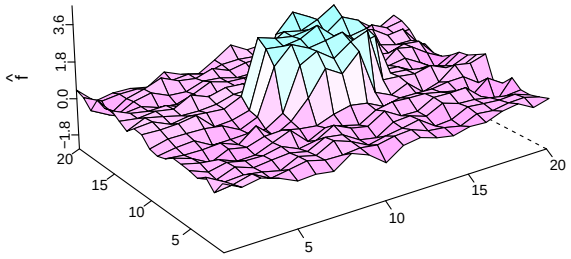
The mean-squared errors (MSE) of the estimated effects $\hat{\boldsymbol{\beta}}$ in reference to the true $\boldsymbol{\beta}$ is 0.101 for the approximate algorithm and 0.093 for the correct algorithm, *i.e.*, the difference between the two results is rather small. However, as can be expected, the non-approximate algorithms shows a slight improvement in accuracy.

Fig. 3 depicts maps of activated pixels from the non-approximate and the approximate algorithm, where activated pixels are defined as pixels, where $p(\beta_i > 0) > 0.95$. Again, both algorithms are close, where the approximate algorithms (Fig. 3(a)) is outperformed by the non-approximate algorithms (Fig. 3(b)) by only one pixel.

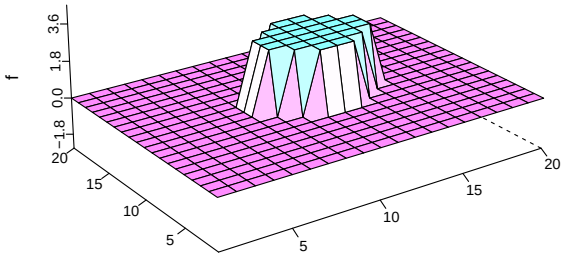
Completing the overview of the estimated models we plot the weights interacting between the pixels in Fig. 4 The weights enable the adaptiveness in the adaptive model, allowing a clear distinction between



(a)



(b)



(c)

Figure 1: Estimated surfaces $\hat{\boldsymbol{\beta}}$ using (a) non-approximate algorithm and (b) approximate algorithm compared with the (c) true surface $\boldsymbol{\beta}_i$.

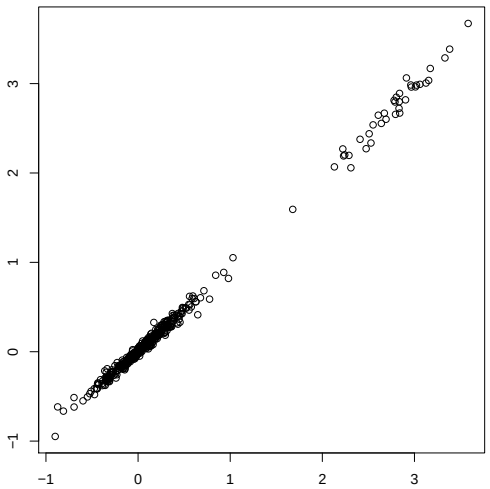
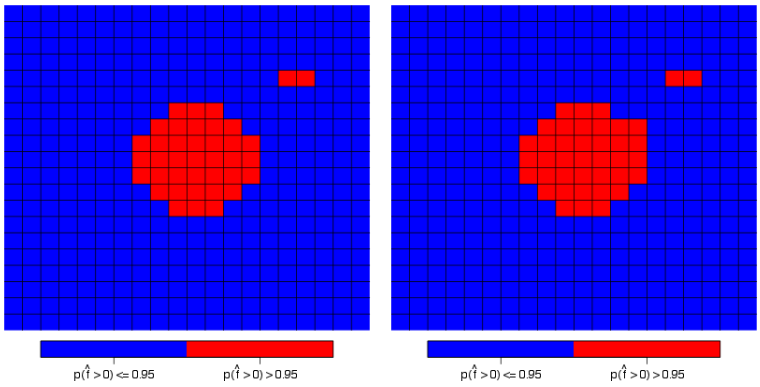


Figure 2: Scatter plot of $\hat{\boldsymbol{\beta}}^{non-approx.}$ vs. $\hat{\boldsymbol{\beta}}^{approx.}$.



(a) (b)

Figure 3: Posterior probabilities of the spatial effects, with red areas indicating that at least 95% of the sample estimates were positive: (a) non-approximate algorithm, (b) approximate algorithm.

activated and non (or less) activated areas. As expected, small weights occur where the difference between adjacent pixels is large, thus between areas with large and small effects. In Fig. 4 the red ring of weights defines the boundary between the activated cylinder and the area outside the cylinder. Moreover, when small weights are detected, two adjacent pixels (i and j) can be seen as independent of each other due to large differences between the estimates $\hat{\beta}_i$ and $\hat{\beta}_j$. Again, the results from both algorithms are very similar.

Fig. 5 depicts the marginal posteriors of the weights for three different locations (with small, medium and large weights). As can be seen, the approximation results in similar posterior densities, but with a slight underestimation of the posterior median with the approximate algorithm. The marginal posterior variance is slightly decreased in the approximate case, *i.e.*, the uncertainty in parameter estimation is slightly underestimated.

4. In vivo data set

We now evaluate the algorithms on a real fMRI data set. We use a previously analyzed data set (Bergerbest et al., 2004), publicly available at `www.fmridc.org` (Horn and Gazzaniga, 2013). In the auditory localizer scan, participants listened to three types of 20-sec blocks (10 time units, 1 unit = 2 sec): Six blocks of environmental sounds, six blocks of simple tones (simple tone data were not analyzed in this study), and six blocks of silence (scanner noise). Blocks were presented in a fixed pseudo-random order, which included all six possible orders of the three types of blocks, with the restriction that two blocks of the same type were not presented successively. Each of the six blocks of environmental sounds included eight pseudo randomly mixed sounds (2 sec each, 25% generated by an animal, a total of 48 sounds that were not used in the repetition priming scans). Sounds were separated by a 0.5-sec inter stimulus interval. Each block of tones included six low-pitched tones pseudo randomly mixed with two high-pitched tones. During the scanner noise blocks, no sounds were presented to the participants except for the background scanner noise, which was present for all blocks. Participants were asked to simply pay attention to the sounds and tones presented to them. In all sessions, participants were instructed to keep their eyes closed.

The examined subjects underwent an auditorial stimulus experiment while being scanned. Fig. 6(a) shows the convolved haemodynamic response function. One pixel with high dependencies in time obtained from an area of high activation is shown in Fig. 6(b), correspondingly a pixel with a medium dependence structure coming from a region with less activation (c) and a pixel with a time series of almost white noise (d), thus no detected activation.

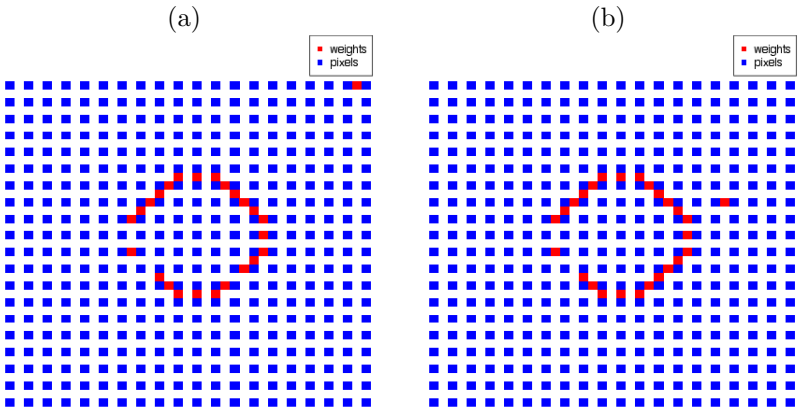


Figure 4: Boundary maps resulting from the adaptive Gauss prior: blue pixels correspond to the pixels $i = 1, \ldots, 400$ of simulated grid; red pixels depict the weights w_{ij} between pixel i and j with $\hat{w}_{ij} < 0.05$: (a) non-approximate algorithm, (b) approximate algorithm

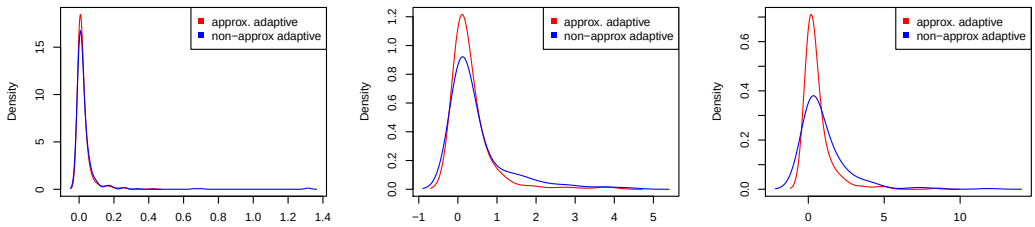


Figure 5: Kernel density estimates of the marginal posterior distributions resulting from the non-approximate algorithm (blue) with small ($\hat{w}_{234,235}^{non-approx.} = 0.024$, medium ($\hat{w}_{210,211}^{non-approx.} = 0.492$) and large ($\hat{w}_{82,83}^{non-approx.} = 0.997$) posterior means, and from the approximate algorithm (red) with small ($\hat{w}_{234,235}^{approx.} = 0.034$, medium ($\hat{w}_{210,211}^{approx.} = 0.327$) and large ($\hat{w}_{82,83}^{approx.} = 0.587$) posterior means.

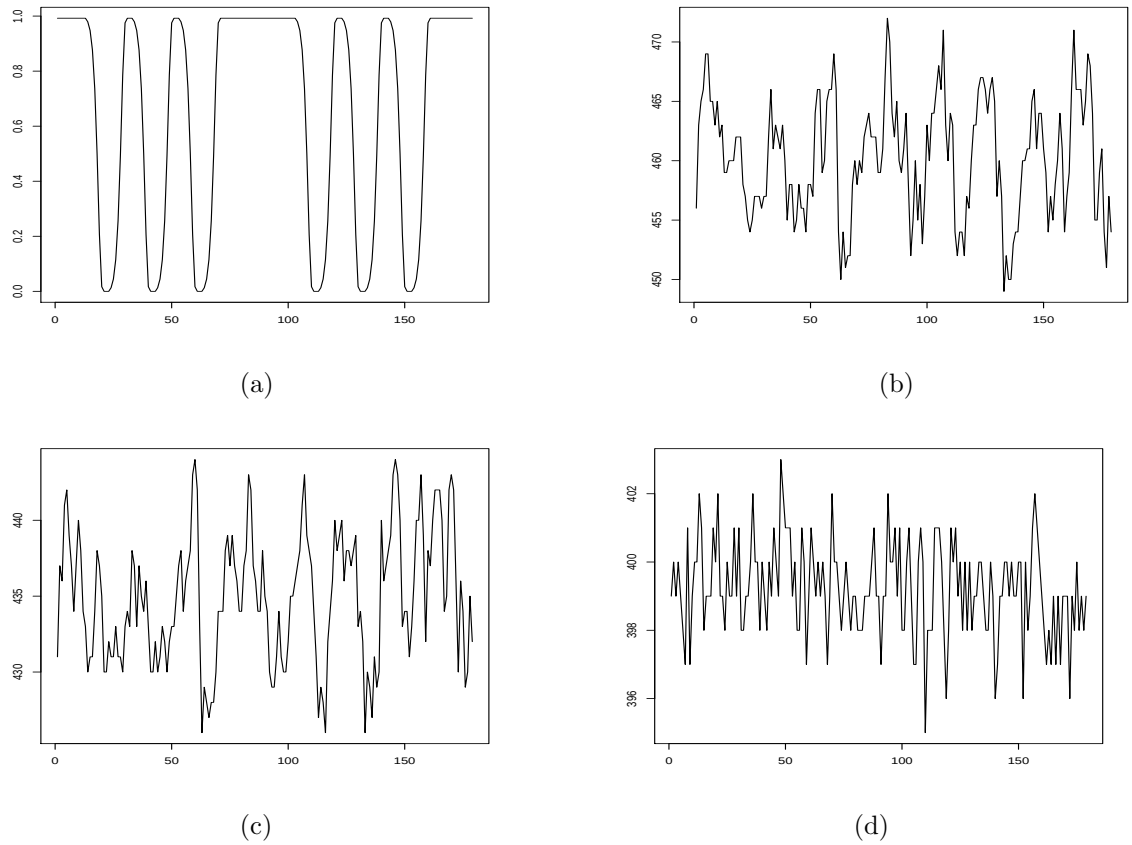


Figure 6: Time series of (a) the (transformed) on-off auditory stimulus, (b) strongly, (c) weakly and (d) non activated pixels representing the MR signal, with length $t = 1, \ldots, 179$.

We complement the model equation (1) with a time constant effect $\hat{\beta}_i$ (Brezger et al., 2007), by defining the baseline trend parametrically with

$$f_i(t) = \boldsymbol{u}_t^\top \boldsymbol{\alpha}_i = \alpha_{i0} + \alpha_{i1} t + \alpha_{i2} \sin\left(\frac{\pi t}{16}\right) + \alpha_{i3} \cos\left(\frac{\pi t}{25}\right) + \alpha_{i4} \cos\left(\frac{\pi t}{40}\right),$$

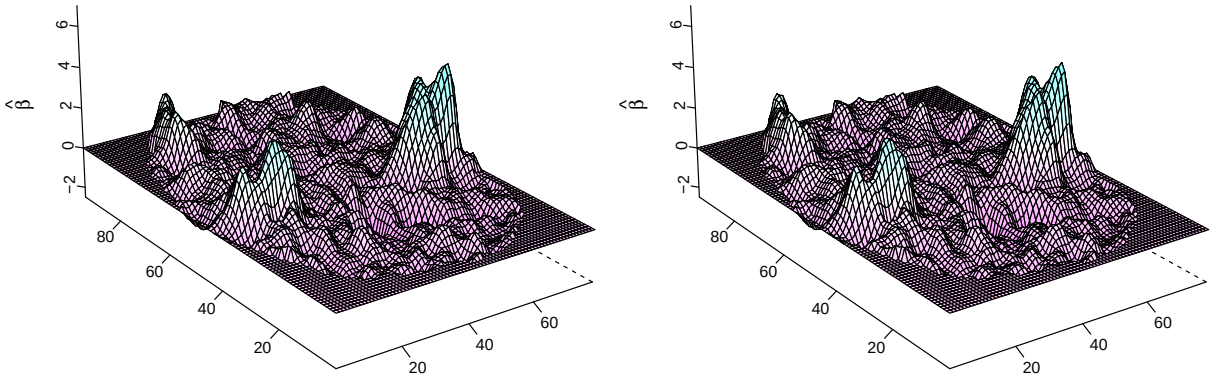
for $t = 1, \ldots, 179$, where i is the voxel.

For the Bayesian model we use non-informative prior scale and shape parameters, with $a = b = 0.001$ for the inverse gamma prior distribution of σ_i^2 and $c = d = 0.001$ for the inverse gamma prior distribution of τ^2 and the parameter of the gamma prior in the weight ν is set to 1.

In the following we show only results for one slice of the brain. The resulting surface of the posterior mean estimates of β can be seen in Fig. 7. The results from both algorithms show similar surfaces.

Fig. 9 depicts the activation map. Here, activation is defined by $\beta_i > 3.5$, where the threshold was chosen by visual judgment. The map was created using the `image()` function in the '*oro.nifti*' package (Whitcher et al., 2011).

Fig. 10 shows the boundary maps using the estimated weights. The weights are relatively large here, compared to the simulation study. This is due to the fact that the activated areas are relatively large on a relatively fine grid, hence the direct neighbors of estimated effects do not differ to a large extent and the descent in height only occurs gradually. The smallest posterior mean of the weights was 0.268 for the approximate algorithm and 0.323 for the non-approximate case, at the same location for both algorithms.



(a) (b)

Figure 7: Estimated surfaces ($\hat{\beta}_i,\ i = 1, \ldots, 5780$) in (a) approximate adaptive Gauss and (b) non-approximate adaptive Gauss.

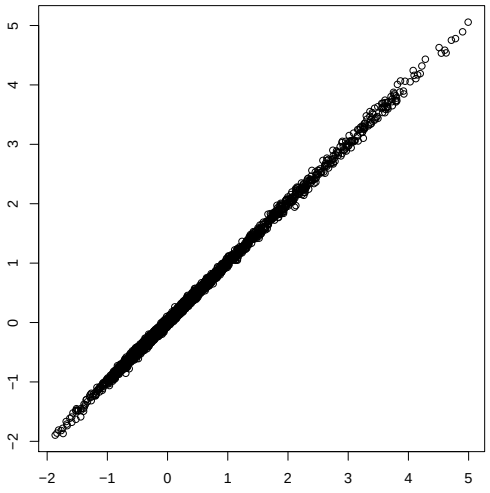
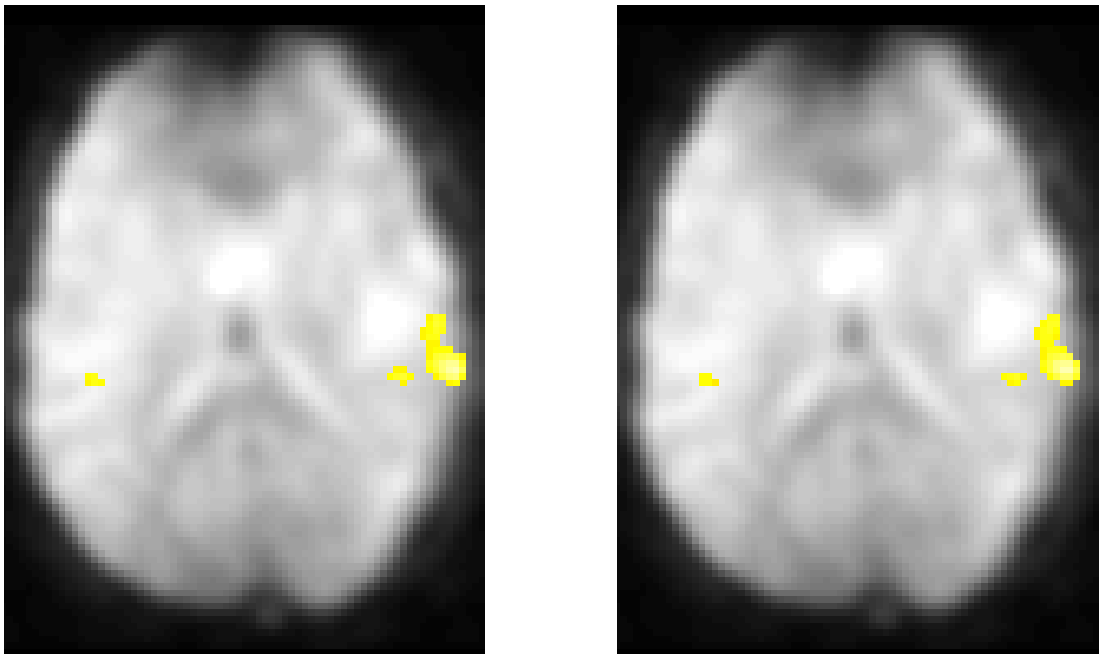


Figure 8: Scatter plot of $\hat{\boldsymbol{\beta}}^{non-approx.}$ vs. $\hat{\boldsymbol{\beta}}^{approx.}$.



(a) (b)

Figure 9: Map of activated pixels, with $\hat{\beta}_i > 3.5$; (a) non-approximate algorithm and (b) approximate algorithm.

The results of the real data also confirm that the difference between the approximate and non-approximate estimation is minimal, when comparing the plotted effects. Although, it seems the non-approximate case captures the high peaks slightly better with a maximum posterior mean estimate of 5.03 compared to 4.96 when approximating.

Fig. 11 shows the marginal posterior densities of three weights with different posterior means. Similar to the simulation study, there is only a slight underestimation of w using the approximate algorithm, with a slightly decreased posterior variance.

5. Conclusion

Adaptive GMRF smoothing allows not only to retain borders and other sharp features in the smoothness maps, it also allows to detect these borders via the estimated weights: By plotting the weights, one can detect differences from non-activated and activated areas easily. Even locating the exact pixels accountable for large discrepancy is feasible. Conclusively, the estimation with an adaptive Gauss prior allows a more accurate and reliable estimation.

Yet, the MCMC algorithm of the adaptive GMRF model has the big disadvantage of the necessity of computation of the Eigenvalues of a big matrix for each weight parameter. Here, we have proposed an approximate algorithm, which ignores the dependency between the weights in the update of each weight. The direct comparison of the results from the approximate algorithm and the non-approximate algorithm do not give reason to opt for the latter, as the differences between the results are small. Additionally, the computational burden is immense for real data on a large grid, amounting to days for the non-approximate case compared to a few hours when approximating.

The Metropolis-Hastings step can be sped up by the following methods (Brezger et al., 2007). The product of the non-negative Eigenvalues can be computed efficiently using the Cholesky decomposition of band matrices instead of the much more expensive Eigenvalue computation (Rue and Held, 2005). Secondly,

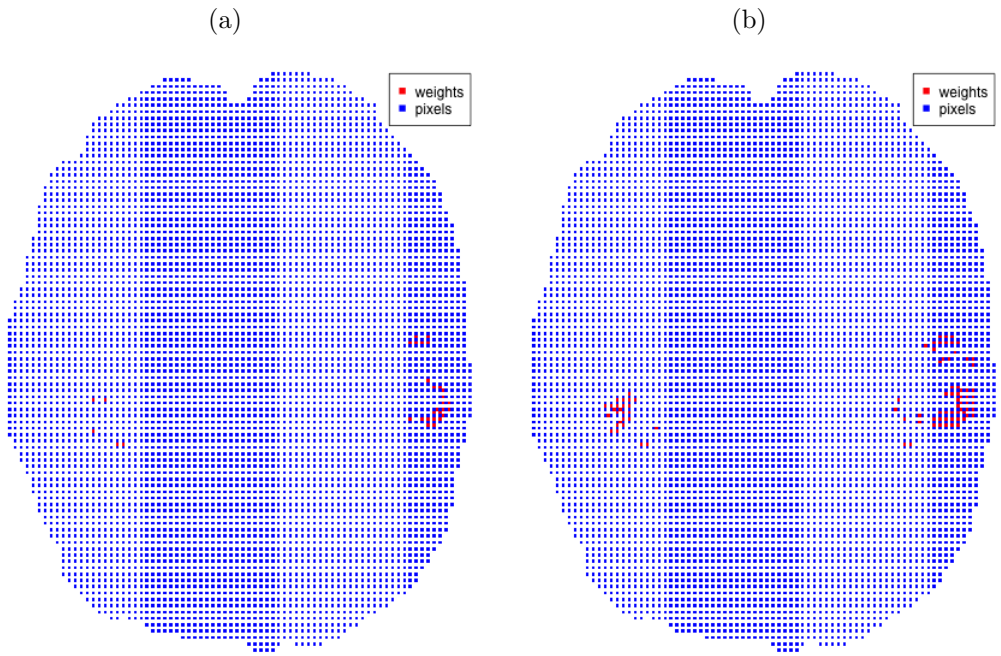


Figure 10: Boundary maps: blue pixels correspond to the original voxel grid, red pixels depict the weights w_{ij} between pixel i and j with $\hat{w}_{ij} < 0.175$. (a) non-approximate algorithm, (b) approximate algorithm.

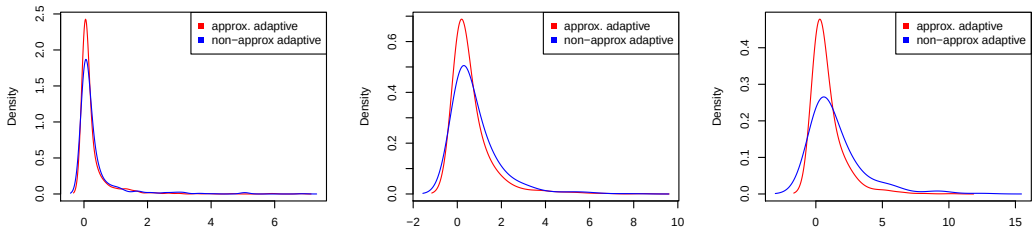


Figure 11: Kernel density estimates of the marginal posterior distributions resulting from the non-approximate algorithm (blue) with small ($\hat{w}^{non-approx.}_{5581,5635} = 0.332$, medium ($\hat{w}^{non-approx.}_{5675,5676} = 0.798$) and large ($\hat{w}^{non-approx.}_{1,22} = 1.498$) posterior means, and from the approximate algorithm (red) with small ($\hat{w}^{approx.}_{5581,5635} = 0.260$, medium ($\hat{w}^{approx.}_{5675,5676} = 0.620$) and large ($\hat{w}^{approx.}_{1,22} = 0.852$) posterior means.

the Eigenvalues change only from the position corresponding to the position where the weight is located, therefore the Cholesky decomposition can also start at the corresponding row. Finally, one can update a block of weights one at a time, but for this the acceptance rate has to be monitored closely. However, even with these efficient sampling techniques the computation time is still in the realm of days. For analyzing the *in vivo* data in section 4, the non-approximate model took on average 794.1 seconds per iteration (using block updates of 12 weights each in step 3), compared to 29.4 seconds for the approximate model in total elapsed time, being 27 times faster. The computations were conducted on a Mac Book Pro, with a 2.4 GHz Intel Core i5 processor and 8 GB 1333 MHz DDR3 RAM.

The general linear model equation (1) can be easily extended by adding a further covariate. For example, in an experiment with a visual and an auditory stimulus, the equation amounts to

$$y_{it} = \boldsymbol{u}_t^\top \boldsymbol{\alpha}_i + z_{it}^{vis.} \beta_i^{vis.} + z_{it}^{aud.} \beta_i^{aud.} + \epsilon_{it}, \qquad \epsilon_{it} \sim \mathrm{N}(0, \sigma_i^2), \tag{9}$$

where the covariate $z^{it,\,vis.}$ represents the haemodynamic response function of a visual stimulus and the covariate $z^{it,\,aud.}$ represents an auditory stimulus. This extension is implemented in the *adaptsmoFMRI* package. Equation (9) can, of course, be extended by any number of covariates of external stimuli. However, bare in mind that each additional independent variable increases the time of computation considerably.

Although the proposed approximate Gibbs sampler algorithm introduces an error in the MCMC sampling, and therefore in the estimation of the posterior, the application on simulated and real data shows that this error is rather small. Note, that the approximation is introduced when sampling the weights. This leads to an obvious error in the estimated posterior of the weights. Though, there is nearly no error on the posterior of the activation effects, which are the main interest in the fMRI analysis.

The proposed approximate Gibbs sampling algorithm therefore allows to efficiently estimate adaptive Gaussian Markov random fields. The application of this algorithm is not restricted to fMRI, but can also be adapted for more complex models, e.g. , for latent adaptive GMRFs in non-linear models, *e.g.* Dynamic contrast-enhanced MRI (Schmid et al., 2006) or for latent adaptive GMRF models in space-time interactions (Schmid, 2011). The algorithm can also easily be used on data with higher spatial resolution, *i.e.,* by including a lot more voxels, or even for applications in 3D imaging or data with higher dimensions.

Acknowledgements

The authors wish to thank the fMRI Data Center for providing data. Thanks to Ludwig Fahrmeir for input. VS was supported by a grant from the Deutsche Forschungsgemeinschaft (SCHM 2747/1-1).

References

Baddeley, A., Turner, R., 2005. Spatstat: an R package for analyzing spatial point patterns. Journal of Statistical Software 12, 1–42. ISSN 1548-7660.

Bates, D., Maechler, M., 2012. Matrix: Sparse and Dense Matrix Classes and Methods. URL: `http://cran.r-project.org/web/packages/Matrix/index.html`.

Bergerbest, D., Ghahremani, D.G., Gabrieli, J.D.E., 2004. Neural Correlates of Auditory Repetition Priming: Reduced fMRI Activation in the Auditory Cortex. Journal of Cognitive Neuroscience 16, 966–977. doi:`10.1162/0898929041502760`.

Brezger, A., Fahrmeir, L., Hennerfeind, A., 2007. Adaptive Gaussian Markov random fields with applications in human brain mapping. Journal of the Royal Statistical Society: Series C (Applied Statistics) 56, 327–345. doi:`10.1111/j.1467-9876.2007.00580.x`.

Eugster, M.J.A., Knaus, J., Porzelius, C., Schmidberger, M., Vicedo, E., 2011. Hands-on tutorial for parallel computing with R. Computational Statistics 26, 219–239. doi:`10.1007/s00180-010-0206-4`.

Friston, K., Holmes, A.P., Worsley, K.J., Poline, J.P., Frith, C.D., Frackowiak, R.S.J., 1995. Statistical Parametric Maps in Functional Imaging: A General Linear Approach. Human Brain Mapping 2, 189–210.

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., Hothorn, T., 2012. mvtnorm: Multivariate Normal and t Distributions. URL: `http://CRAN.R-project.org/package=mvtnorm`. r package version 0.9-9994.

Gössl, C., Auer, D.P., Fahrmeir, L., 2001a. Bayesian spatiotemporal inference in functional magnetic resonance imaging. Biometrics 57, 554–562. doi:`10.1111/j.0006-341X.2001.00554.x`.

Gössl, C., Fahrmeir, L., Auer, D.P., 2001b. Bayesian modeling of the hemodynamic response function in BOLD fMRI. NeuroImage 14, 140–8. doi:`10.1006/nimg.2001.0795`.

Horn, J.D.V., Gazzaniga, M.S., 2013. Why share data? lessons learned from the fmridc. NeuroImage doi:10.1016/j.neuroimage.2012.11.010. in print.

Hughes, M., 2012. adaptsmoFMRI: Adaptive Smoothing of FMRI Data (R package). URL: http://cran.r-project.org/web/packages/adaptsmoFMRI/index.html.

Lange, N., 1996. Statistical approaches to human brain mapping by functional magnetic resonance imaging. Statistics in Medicine 15, 389–428.

Martin, A.D., Quinn, K.M., Park, J.H., 2011. MCMCpack: Markov chain monte carlo in R. Journal of Statistical Software 42, 22.

R Core Team, 2012. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. URL: http://www.R-project.org/. ISBN 3-900051-07-0.

Rue, H., Held, L., 2005. Gaussian Markov Random Fields: Theory and Applications (Monographs on Statistics and Applied Probability). Chapman & Hall.

Schmid, V.J., 2011. Voxel-based adaptive spatio-temporal modelling of perfusion cardiovascular MRI. IEEE transactions on medical imaging 30, 1305–13. doi:10.1109/TMI.2011.2109733.

Schmid, V.J., Whitcher, B., Padhani, A.R., Taylor, N.J., Yang, G.Z., 2006. Bayesian methods for pharmacokinetic models in dynamic contrast-enhanced magnetic resonance imaging. IEEE Transactions on Medical Imaging 25, 1627–36.

Stippich, C., 2007. Introduction to presurgical functional MRI, in: Stippich, C. (Ed.), Clinical Functional MRI. Springer Berlin Heidelberg. chapter 1, pp. 1–7.

Whitcher, B., Schmid, V.J., Thornton, A., 2011. Working with the DICOM and NIfTI Data Standards in R. Journal of Statistical Software 44, 1–28.

Winkler, G., 2002. Image analysis, random fields and Markov chain Monte Carlo methods: A Mathematical Introduction. 2 ed., Springer, Berlin.