Maximum likelihood from spatial random effects models via the stochastic approximation expectation maximization algorithm

Hongtu Zhu · Minggao Gu · Bradley Peterson

Received: March 2004 / Accepted: August 2006 / Published online: 27 January 2007

Abstract We introduce a class of spatial random effects models that have Markov random fields (MRF) as latent processes. Calculating the maximum likelihood estimates of unknown parameters in SREs is extremely difficult, because the normalizing factors of MRFs and additional integrations from unobserved random effects are computationally prohibitive. We propose a stochastic approximation expectation-maximization (SAEM) algorithm to maximize the likelihood functions of spatial random effects models. The SAEM algorithm integrates recent improvements in stochastic approximation algorithms; it also includes components of the Newton-Raphson algorithm and the expectation-maximization (EM) gradient algorithm. The convergence of the SAEM algorithm is guaranteed under some mild conditions. We apply the SAEM algorithm to three examples that are representative of real-world applications: a state space model, a noisy Ising model, and segmenting magnetic resonance images (MRI) of the human brain. The SAEM algorithm gives satisfactory results in finding the maximum likelihood estimate of spatial random effects models in each of these instances.

H. Zhu (✉)
Department of Biostatistics and Biomedical Research Imaging Center, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-7420, USA
e-mail: htzhu@email.unc.edu

H. Zhu · B. Peterson
Department of Psychiatry, Columbia University and New York State Psychiatric Institute, 1051 Riverside Drive, Unit 74, New York, New York 10032, USA

M. Gu
Department of Statistics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, P.R. China

Keywords Expectation maximization · Markov chain Monte Carlo · Markov random fields · Spatial random effects models · Stochastic approximation

1 Introduction

Spatial random effects models, also called hidden Markov models, represent a natural extension of Markov random fields (Besag, 1986, 1974). Spatial random effects models are very useful for accommodating overdispersion among outcomes (Zeger et al., 1988) and for interpolating or smoothing spatial and image data (Diggle et al., 1998). Special classes of spatial random effects models include: generalized linear mixed models, such as those used in biomedical studies (Breslow and Clayton, 1993; Lee and Nelder, 1996); spatial generalized linear mixed models (SGLMM), as used in geostatistics (Christensen and Waagepetersen, 2002; Zhang, 2002); and noisy Gaussian Markov random fields (GMRF), as applied for image segmentation and restoration in image analysis (Saquib et al., 1998; Rajapakse et al., 1997; Marroquin et al., 2003).

Because of the utility of spatial random effects models, developing procedures for estimating the maximum likelihood estimate of spatial random effects models has been an issue of central importance (Marroquin et al., 2003; Qian and Titterington, 1991). Spatial random effects models involve latent MRFs, whose normalizing factors are notorious for their computational complexity. This complexity makes calculating the maximum likelihood estimate of MRFs, and therefore the maximum likelihood estimate of spatial random effects models, prohibitively difficult. A second issue is the additional integrations found in spatial random effects models. Most existing procedures for approximating the maximum likelihood estimate of MRFs include the

Monte Carlo likelihood inference (Geyer and Thompson, 1992), Monte Carlo Newton-Raphson sampling (Penttinen, 1984), numerical approximations (Pettitt et al., 2003), and the stochastic approximation algorithm (Younes, 1989; Moyeed and Baddeley, 1991; Gu and Zhu, 2001). However, these algorithms cannot be applied directly to calculating the maximum likelihood estimate of spatial random effects models because they do not account for the presence in spatial random effects models of additional integrations from unobserved random effects (Qian and Titterinton, 1991). Rydén (1997) proposed a stochastic approximation algorithm for recursive estimation of hidden Markov models, Delyon et al. (1999) proposed a stochastic approximation EM algorithm for curved exponential families with random effects, and Zhang (2002) proposed a Monte Carlo EM algorithm for computing the maximum likelihood estimate of spatial generalized linear mixed models. However, neither of these models contains any normalizing factors that are intractable; the estimation algorithms for these models therefore cannot be applied to spatial random effects models.

We propose an SAEM algorithm for computing the maximum likelihood estimate of spatial random effects models, and we give a proof of its convergence under some conditions. Examples of a state space model, a noisy Ising model, and image segmentation in MRI illustrate the effective performance of our SAEM algorithm in calculating the maximum likelihood estimate of spatial random effects models.

2 Spatial random effect models

2.1 Definition of spatial random effects models

We consider a data set that is composed of a response $y_j(s_i)$ and covariate vector $\boldsymbol{x}_j(s_i)$ for $j = 1, \ldots, m_i$ at a site $s_i \in S$ for $i = 1, \ldots, n$, where $S = \{s_i : i = 1, \ldots, n\}$ is a known discrete index set. For instance, in image processing, s_i represents the location of a particular voxel/pixel. Furthermore, we assume that there is an unobserved $d \times 1$ random effect vector $\boldsymbol{b}(s_i)$ for each $\boldsymbol{y}(s_i) = (y_1(s_i), \ldots, y_{m_i}(s_i))^T$. Spatial random effects models are defined as follows.

(i) Conditional on $\boldsymbol{b} = (\boldsymbol{b}(s_1), \ldots, \boldsymbol{b}(s_n))^T$, the components of $\boldsymbol{Y} = (\boldsymbol{y}(s_1), \ldots, \boldsymbol{y}(s_n))^T$ are mutually independent, and the conditional density of $\boldsymbol{y}(s_i)$ given $\boldsymbol{b}$ is a member of the exponential family (McCullagh and Nelder, 1989) given by

$$p(\boldsymbol{y}(s_i)|\boldsymbol{b};\alpha,\beta)=\prod_{j=1}^{m_i}\exp\{\phi_j(s_i)[y_j(s_i)\theta_j(s_i)-a(\theta_j(s_i))] +c(y_j(s_i),\phi_j(s_i))\}, \qquad (1)$$

where $\phi_j(s_i) = \phi_j(\alpha, \boldsymbol{b}(s_i))$, α is an unknown $q_1 \times 1$ parameter vector, and $a(\cdot)$ and $c(\cdot)$ are known continuously differentiable functions. For a known link function $h_1(\cdot)$,

$$\mu_j(s_i) = E[y_j(s_i)|\boldsymbol{b}] = h_1(\boldsymbol{x}_j(s_i)^T\beta, \boldsymbol{b}(s_i)), \qquad (2)$$

where β is a $q_2 \times 1$ parameter vector.

(ii) The joint distribution of random effects $\boldsymbol{b}$ has the Gibbs form:

$$p(\boldsymbol{b}|\tau) = \exp\{-U(\boldsymbol{b})^T h_2(\tau) - \log C(\tau)\}, \qquad (3)$$

where $h_2(\cdot)$ is a known function, τ is a $q_3 \times 1$ vector characterizing the granularity of MRF, and $U(\boldsymbol{b})^T h_2(\tau)$ is a potential (or energy) function, which exhibits the interaction between random effects (Besag, 1974). In addition, the normalizing factor $C(\tau)$, called a partition function, has the form

$$C(\tau) = \int_{\boldsymbol{b}\in\mathcal{B}} \exp\{-U(\boldsymbol{b})^T h_2(\tau)\} m(d\boldsymbol{b}), \qquad (4)$$

where $\mathcal{B}$ is the minimal sample space of $\boldsymbol{b}$ and $m(d\boldsymbol{b})$ is either the Dirac's delta measure or $d\boldsymbol{b}$.

The likelihood function of observed data $\boldsymbol{Y} = \boldsymbol{y}_o$ for an spatial random effects model is given by

$$L(\xi;\boldsymbol{y}_o) = \int_{\mathcal{B}} \exp\{-U(\boldsymbol{b})^T h_2(\tau) - \log C(\tau)\} \prod_{i=1}^{n} p(\boldsymbol{y}(s_i)|\boldsymbol{b};\alpha,\beta) m(d\boldsymbol{b}), \qquad (5)$$

where $\xi^T = (\alpha^T, \beta^T, \tau^T)$ is a $q \times 1$ $(q = q_1 + q_2 + q_3)$ vector of unknown parameters. Because the integration above is usually of very high dimension and/or $C(\tau)$ is difficult to obtain analytically, evaluating $L(\xi; \boldsymbol{y}_o)$ is computationally prohibitive.

2.2 Examples of spatial random effects models

We examine three examples of spatial random effects models:

Example 1 (*Generalized linear mixed models*). Generalized linear mixed models usually assume that random effects $\boldsymbol{b}(s_i)$ are normally distributed with zero mean and covariance matrix Σ_b and $\boldsymbol{b}(s_i)$ and $\boldsymbol{b}(s_{i'})$ are independent of each other for $s_i \neq s_{i'}$. See, for example, Breslow and Clayton (1993), Aitkin (1996), and Zhu and Lee (2002), among many others. For generalized linear mixed models, $s_i \in S$ can represent either a subject or a cluster in a longitudinal study or a family in a familial study.

In particular, $U(\boldsymbol{b})^T h_2(\tau) = 0.5 \sum_{i=1}^n \boldsymbol{b}(s_i)^T \Sigma_b^{-1} \boldsymbol{b}(s_i)$ and $\log C(\tau) = 0.5n \log |\Sigma_b| + 0.5nd \log(2\pi)$, where τ contains all unknown parameters in Σ_b.

Example 2 (*State space models*). State space models represent further extensions of generalized linear mixed models by considering time series dependence among random effects $\boldsymbol{b}(s_i)$ (Chan and Ledolter, 1995; Durbin and Koopman, 1997, 2000). In this case, each s_i denotes a time point such that $s_1 < s_2 < \cdots < s_n$. For example, in the time series of count data considered in Section 4.1, $\boldsymbol{y}(s_i)$ follows the Poisson distribution with mean $\mu(s_i) = \exp(\boldsymbol{x}(s_i)^T \beta + \boldsymbol{b}(s_i))$. Assume that $\boldsymbol{b}(s_1)$ is given and $\{\boldsymbol{b}(s_i)\}$ is a stationary Gaussian AR(1) process, that is, $\boldsymbol{b}(s_i) = \rho \boldsymbol{b}(s_{i-1}) + \epsilon_i$, where $\{\epsilon_i\}$ is identically and independently distributed as $N(0, \sigma_\epsilon^2)$. Thus, $\tau = (\rho, \sigma_\epsilon^2)^T$, $U(\boldsymbol{b})^T h_2(\tau) = \sum_{i=1}^{n-1} [\boldsymbol{b}(s_{i+1}) - \rho \boldsymbol{b}(s_i)]^2 / (2\sigma_\epsilon^2)$, and $\log C(\tau) = [(n-1)/2] \log(\sigma_\epsilon^2)$.

Example 3 (*Spatial random effects models for image segmentation*). Image segmentation is among the several important image processes that have been modeled by using spatial random effects models since the seminal papers by Geman and Geman (1984) and Besag (1974). See, for example, Winkler (1995), Li (2001), Rajapakse et al. (1997), and Marroquin et al. (2003), among many others. For image segmentation, $s_i \in S$ denotes either a pixel site or a line site in a pixelated image, $\boldsymbol{Y}$ denotes the observed image, and each $\boldsymbol{b}(s_i)$ in $\boldsymbol{b}$ denotes the true identity at the voxel s_i. The purpose of image segmentation is to classify $\boldsymbol{Y}$ into M nonoverlapping regions $\{R_1, \ldots, R_M\}$.

A simple example of spatial random effects models for image segmentation (Qian and Titterington, 1991; Besag, 1986; Derin and Elliott, 1987) assumes that $Y_i | \boldsymbol{b}(s_i) \sim N(\mu(\boldsymbol{b}(s_i)), \sigma^2)$ for $i = 1, \ldots, n$, and $p(\boldsymbol{b}|\tau) = \exp\{\tau \sum_{s_i \sim s_j} \delta(\boldsymbol{b}(s_i), \boldsymbol{b}(s_j)) - \log C(\tau)\}$, where $\boldsymbol{b}(s_i)$ takes value from 1 to M, the summation is over nearest-neighbor pairs $s_i \sim s_j$, and $\delta(x, z)$ is the Kronecker function equaling to 1 when $x = z$ and 0 otherwise. The potential function $U(\boldsymbol{b}) = -\sum_{s_i \sim s_j} \delta(\boldsymbol{b}(s_i), \boldsymbol{b}(s_j))$ and the normalizing factor $C(\tau) = \sum_{\boldsymbol{b}} \exp(-U(\boldsymbol{b})\tau)$, which involves M^n terms.

We consider a generalization of a spatial random effects model for image segmentation of MRI from Zhang et al. (2001) as follows. The observation $\boldsymbol{y}(s_i)$ at a particular voxel s_i can be modeled as

$$\boldsymbol{y}(s_i) = \boldsymbol{x}_0(s_i)^T \beta_0 + \sum_{k=1}^M [\boldsymbol{x}_1(s_i)^T \beta(k) + \epsilon_k(s_i)] \delta(\boldsymbol{b}(s_i), k), \tag{6}$$

where $\boldsymbol{x}_0(s_i)$ and $\boldsymbol{x}_1(s_i)$ are, respectively, covariate vectors characterizing common and individual features at the voxel s_i, $\boldsymbol{b}(s_i) \in \{1, \ldots, M\}$, $\epsilon_k(s_i) \sim N(0, e^{\sigma_k})$, and β_k is the parameter vector associated with the class R_k. We further assume that the joint distribution of the label fields $\boldsymbol{b}$ is given by $p(\boldsymbol{b}|\tau) = \exp\{\sum_{i=1}^n \tau_1(\boldsymbol{b}(s_i)) + \sum_{s_i \sim s_j} \tau_2 \delta(\boldsymbol{b}(s_i), \boldsymbol{b}(s_j)) - \log C(\tau)\}$, where $\tau_1(\boldsymbol{b}(s_i))$ may depend on the value of $\boldsymbol{b}(s_i)$ and $\tau_1(1)$ is set to zero to avoid redundancy, τ_2 controls the granularity of MRF, and $\tau = (\tau_1(2), \ldots, \tau_1(M), \tau_2)^T$. The potential function $U(\boldsymbol{b})^T \tau = (-\sum_{i=1}^n \delta(\boldsymbol{b}(s_i), 2), \ldots, -\sum_{i=1}^n \delta(\boldsymbol{b}(s_i), M), -\sum_{s_i \sim s_j} \delta(\boldsymbol{b}$ $(s_i), \boldsymbol{b}(s_j))\tau$ and the normalizing factor $C(\tau)$ is obtained by summing all possible configurations $\boldsymbol{b}$, $C(\tau) = \sum_{\boldsymbol{b}} \exp(-U(\boldsymbol{b})^T \tau)$. If $\tau_2 = 0$, then $\log C(\tau) = n \log[1 + \sum_{k=2}^M \exp(\tau_1(k))]$ and the above spatial random effects model reduces to a mixture linear regression model (Zhang et al., 2001; Zhu and Zhang, 2004).

3 SAEM algorithm

Under the spatial random effects model specified by (1), (2), and (3), the maximum likelihood estimate of ξ, denoted by $\hat{\xi} = (\hat{\alpha}, \hat{\beta}, \hat{\tau})$, is defined by

$$L(\hat{\xi}; \boldsymbol{y}_o) = \max_{\xi} L(\xi; \boldsymbol{y}_o). \tag{7}$$

Because $L(\xi; \boldsymbol{y}_o)$ in (5) is computationally intractable, it is infeasible to maximize the likelihood function of observed data directly. Instead, we consider the first-order and second-order partial derivatives of the log-likelihood function in order to use gradient-type algorithms, such as the Newton-Raphson and Gauss-Newton algorithms (Ortega, 1990).

3.1 First-order and second-order derivatives of the log-likelihood function

The first-order and second-order derivatives of the log-likelihood functions can be derived by using the log-likelihood functions of complete data, denoted by $l_c(\xi; \boldsymbol{b}, \boldsymbol{y}_o)$, which is given by

$$\sum_{i=1}^n \sum_{j=1}^{m_i} \{\phi_j(s_i)[y_j(s_i)\theta_j(s_i) - a(\theta_j(s_i))] + c(y_j(s_i), \phi_j(s_i))\} \\ -U(\boldsymbol{b})^T h_2(\tau) - \log C(\tau). \tag{8}$$

From the missing information principle, the first-order derivative of $L(\xi; \boldsymbol{y}_o)$, called the score function, can be written as

$$s_\xi(\xi; \boldsymbol{y}_o) = \partial_\xi \log L(\xi; \boldsymbol{y}_o) = E[S_\xi(\xi; \boldsymbol{b}) | \boldsymbol{y}_o, \xi], \tag{9}$$

where $S_\xi(\xi; \boldsymbol{b}) = \partial_\xi l_c(\xi; \boldsymbol{b}, \boldsymbol{y}_o)$ and $E[\cdot | \boldsymbol{y}_o, \xi]$ denotes that the expectation is taken with respect to the conditional distribution $p(\boldsymbol{b}|\boldsymbol{Y} = \boldsymbol{y}_o, \xi)$. In addition, we use ∂ and ∂^2 to denote the first-order and second-order derivatives with respect to a parameter vector, say, $\partial_\xi a(\xi) = \partial a(\xi)/\partial \xi$ and $\partial^2_\xi a(\xi) = \partial^2 a(\xi)/\partial \xi \partial \xi^T$. To calculate the second-order derivative of the log-likelihood function, we apply Louis's (1982) formula to obtain

$$-\partial^2_\xi \log L(\xi; \boldsymbol{y}_o) = E[I_{\xi\xi}(\xi; \boldsymbol{b}) - S_\xi(\xi; \boldsymbol{b})^{\otimes 2}|\boldsymbol{y}_o, \xi] + s_\xi(\xi; \boldsymbol{y}_o)^{\otimes 2}, \tag{10}$$

where for vector $\boldsymbol{a}$, $\boldsymbol{a}^{\otimes 2} = \boldsymbol{a}\boldsymbol{a}^T$ and $I_{\xi\xi}(\xi; \boldsymbol{b}) = -\partial^2_\xi l_c(\xi; \boldsymbol{b}, \boldsymbol{y}_o)$ denotes the information matrix for complete data.

We obtain explicit forms of the first-order and second-order derivatives of $l_c(\xi; \boldsymbol{b}, \boldsymbol{y}_o)$. By differentiating $l_c(\xi; \boldsymbol{b}, \boldsymbol{y}_o)$ with respect to ξ, we obtain

$$\begin{aligned}
S_\alpha(\xi; \boldsymbol{b}) &= \sum_{i=1}^{n} \sum_{j=1}^{m_i} \partial_\alpha \phi_j(s_i)[y_j(s_i)\theta_j(s_i) - a(\theta_j(s_i)) \\
&\quad + \partial_{\phi_j} c(y_j(s_i), \phi_j(s_i))], \\
S_\beta(\xi; \boldsymbol{b}) &= \sum_{i=1}^{n} \sum_{j=1}^{m_i} \phi_j(s_i) e_j(s_i) \partial_\beta \theta_j(s_i), \text{ and} \\
S_\tau(\xi; \boldsymbol{b}) &= -U(\boldsymbol{b})^T \partial_\tau h_2(\tau) - \partial_\tau \log C(\tau),
\end{aligned} \tag{11}$$

where $e_j(s_i) = y_j(s_i) - \mu_j(s_i)$. With some algebraic manipulation, we obtain

$$\begin{aligned}
I_{\beta\beta}(\xi; \boldsymbol{b}) &= \sum_{i=1}^{n} \sum_{j=1}^{m_i} \left\{\phi_j(s_i)\partial_\beta \theta_j(s_i)^T \left[\partial^2_{\theta_j} a(\theta_j(s_i))\right] \partial_\beta \theta_j(s_i) \right. \\
&\quad \left. -\phi_j(s_i) e_j(s_i) \partial^2_\beta \theta_j(s_i)\right\}, \\
I_{\beta\alpha}(\xi; \boldsymbol{b}) &= -\sum_{i=1}^{n} \sum_{j=1}^{m_i} \partial_\alpha \phi_j(s_i) e_j(s_i) \partial_\beta \theta_j(s_i), \\
I_{\tau\tau}(\xi; \boldsymbol{b}) &= \partial^2_\tau [U(\boldsymbol{b})^T h_2(\tau)] + \partial^2_\tau \log C(\tau), \\
I_{\alpha\alpha}(\xi; \boldsymbol{b}) &= -\sum_{i=1}^{n} \sum_{j=1}^{m_i} \partial^2_\alpha \phi_j(s_i)\{[y_j(s_i)\theta_j(s_i) - a(\theta_j(s_i))] \\
&\quad + \partial_{\phi_j} c(y_j(s_i), \phi_j(s_i))\} \\
&\quad - \sum_{i=1}^{n} \sum_{j=1}^{m_i} \partial_\alpha \phi_j(s_i)^T \partial^2_{\phi_j} c(y_j(s_i), \phi_j(s_i)) \partial_\alpha \phi_j(s_i), \\
I_{\beta\tau}(\xi; \boldsymbol{b}) &= 0, \text{ and } I_{\alpha\tau}(\xi; \boldsymbol{b}) = 0.
\end{aligned} \tag{12}$$

Because $I_{\beta\alpha}(\xi; \boldsymbol{b})$ is close to zero at the maximum likelihood estimate, we set $I_{\beta\alpha}(\xi; \boldsymbol{b}) = 0$ in the SAEM algorithm, which leads to a stable algorithm when the initial parameters are far from the maximum likelihood estimate.

To calculate the score function in (9) and the information matrix in (10), we need to calculate $\partial_\tau \log C(\tau)$ and $\partial^2_\tau \log C(\tau)$. Following Gelman and Meng (1998), we have

$$\begin{aligned}
&\partial_\tau \log C(\tau) = -E_\tau \left[U(\boldsymbol{b})^T \partial_\tau h_2(\tau)\right] \quad \text{and} \\
&\partial^2_\tau \log C(\tau) = -E_\tau [J(\tau; \boldsymbol{b})] - \{\partial_\tau \log C(\tau)\}^{\otimes 2},
\end{aligned} \tag{13}$$

where $J(\tau; \boldsymbol{b}) = \partial^2_\tau [U(\boldsymbol{b})^T h_2(\tau)] - [U(\boldsymbol{b})^T \partial_\tau h_2(\tau)]^{\otimes 2}$ and E_τ is taken with respect to the MRF (3). One way to calculate $\partial_\tau \log C(\tau)$ and $\partial^2_\tau \log C(\tau)$ is to use numerical integration by using Eq. (13); however, the numerical integration is accurate only in a few special cases. Another way is to resort to Monte Carlo methods (Liu, 2001; Møller, 1999; Roberts and Casella, 1999; Gu and Zhu, 2001). If we can simulate $\{\boldsymbol{b}_k : k = 1, \ldots, N_k\}$ from the MRF (3), then we can use Eq. (13) to obtain the Monte Carlo approximation of $\partial_\tau \log C(\tau)$ and $\partial^2_\tau \log C(\tau)$. Moreover, Eq. (13) is also the basis for the method of path sampling for estimating $C(\tau)$ at any τ (Gelman and Meng, 1998; Huang and Ogata, 2001; Pettitt et al., 2003). Explicitly, the path sampling method is based on the following formula:

$$\begin{aligned}
\log C(\tau^{**}) - \log C(\tau^*) &= \int_{\tau^*}^{\tau^{**}} \partial_\tau \log C(\tau) d\tau \\
&= -\int_{\tau^*}^{\tau^{**}} E_\tau [U(\boldsymbol{b})^T \partial_\tau h_2(\tau)] d\tau.
\end{aligned}$$

Thus, the Monte Carlo methods can be used to approximate $E_\tau[U(\boldsymbol{b})^T \partial_\tau h_2(\tau)]$ at each τ and then estimate $\log[C(\tau^{**})] - \log[C(\tau^*)]$.

We use Eqs. (11), (12), and (13) to calculate the first-order and second-order derivatives of the likelihood functions of observed data. The score function can be written as $(S_\alpha(\xi; \boldsymbol{b})^T, S_\beta(\xi; \boldsymbol{b})^T, [S_{\tau,1} - S_{\tau,2}]^T)^T$, where $S_{\tau,2} = \partial_\tau \log C(\tau)$ and $S_{\tau,1} = -E_\xi[U(\boldsymbol{b})^T \partial_\tau h_2(\tau)|\boldsymbol{y}_o, \xi]$. We define

$$I_1(\xi; \boldsymbol{b}) = \begin{pmatrix} I_{\alpha\alpha}(\xi; \boldsymbol{b}) & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & I_{\beta\beta}(\xi; \boldsymbol{b}) & \boldsymbol{0} \\ \boldsymbol{0} & 0 & \partial^2_\tau [U(\boldsymbol{b})^T h_2(\tau)] \end{pmatrix} \text{ and}$$

$$I_2(\xi; \boldsymbol{b}) = -\begin{pmatrix} S_\alpha(\xi; \boldsymbol{b}) \\ S_\beta(\xi; \boldsymbol{b}) \\ -U(\boldsymbol{b})^T \partial_\tau h_2(\tau) \end{pmatrix}^{\otimes 2}.$$

The information matrix $-\partial^2_\xi \log L(\xi; \boldsymbol{y}_o)$ is given by

$$\begin{aligned}
&E_\xi[I_1(\xi; \boldsymbol{b})|\boldsymbol{y}_o, \xi] + \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -E_\tau [J(\tau; \boldsymbol{b})] - (S_{\tau,2})^{\otimes 2} \end{pmatrix} \\
&\quad + E_\xi[I_2(\xi; \boldsymbol{b})|\boldsymbol{y}_o, \xi] \\
&\quad + \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -(S_{\tau,2})^{\otimes 2} + S_{\tau,1} S^T_{\tau,2} + S_{\tau,2} S^T_{\tau,1}, \end{pmatrix} + s_\xi(\xi; \boldsymbol{y}_o)^{\otimes 2}.
\end{aligned} \tag{14}$$

3.2 Basic steps of the SAEM algorithm

At the k-th iteration, ξ^k is the current estimate of $\hat{\xi}$; $\boldsymbol{h}^k$, the current estimate of $s_\xi(\hat{\xi}; \boldsymbol{y}_o)$; $S^k_{\tau,1}$ is the current estimate of $E_{\hat{\xi}}[-U(\boldsymbol{b})^T \partial_\tau h_2(\tau)|\boldsymbol{y}_o, \hat{\xi}]$; $S^k_{\tau,2}$, the current estimate of $-\partial_\tau \log C(\hat{\tau})$; $\boldsymbol{\Gamma}^k_1$, the current estimate of $E_{\hat{\xi}}[I_1(\hat{\xi}; \boldsymbol{b})|\boldsymbol{y}_o, \hat{\xi}]$; $\boldsymbol{\Gamma}^k_2$, the current estimate of $E_{\hat{\xi}}[I_2(\hat{\xi}; \boldsymbol{b})|\boldsymbol{y}_o, \hat{\xi}]$; and $\boldsymbol{\Gamma}^k_3$, the current estimate of $E_{\hat{\tau}}[J(\hat{\tau}; \boldsymbol{b})]$. We assume that $\Pi_\tau(\cdot, \cdot)$ is the Markov transition probability of the Metropolis-Hasting (MH) algorithm used to simulate from the MRF (3), and $\Pi_{\boldsymbol{y}_o,\xi}(\cdot, \cdot)$ is the transition probability of the MH algorithm used to simulate from the conditional distribution of $\boldsymbol{b}$ given $\boldsymbol{y}_o$ and ξ.

Step 1. At the kth iteration, set $\boldsymbol{b}_{k,0} = \boldsymbol{b}_{k-1,N_{k-1}}$ and $\boldsymbol{b}_{y,k,0} = \boldsymbol{b}_{y,k-1,N_{k-1}}$. Generate $\boldsymbol{b}_k = (\boldsymbol{b}_{k,1}, \ldots, \boldsymbol{b}_{k,N_k})$ and $\boldsymbol{b}_{y,k} = (\boldsymbol{b}_{y,k,1}, \ldots, \boldsymbol{b}_{y,k,N_k})$ from the transition probabilities $\Pi_{\tau^{k-1}}(\boldsymbol{b}_{k,i-1}, \cdot)$ and $\Pi_{\boldsymbol{y}_o,\xi^{k-1}}(\boldsymbol{b}_{y,k,i-1}, \cdot)$, respectively.

Step 2. Update the seven estimates as follows:

$$\begin{cases} \xi^k = \xi^{k-1} + \gamma_k [\boldsymbol{\Gamma}(t)^k]^{-1} \overline{H}(\xi^{k-1}; \boldsymbol{b}_k, \boldsymbol{b}_{y,k}), \\ \boldsymbol{h}^k = \boldsymbol{h}^{k-1} + \gamma_k (\overline{H}(\xi^{k-1}; \boldsymbol{b}_k, \boldsymbol{b}_{y,k}) - \boldsymbol{h}^{k-1}), \\ \boldsymbol{\Gamma}^k_1 = \boldsymbol{\Gamma}^{k-1}_1 + \gamma_k \big(\overline{I}_1(\xi^{k-1}; \boldsymbol{b}_{y,k}) - \boldsymbol{\Gamma}^{k-1}_1\big), \\ \boldsymbol{\Gamma}^k_2 = \boldsymbol{\Gamma}^{k-1}_2 + \gamma_k \big(\overline{I}_2(\xi^{k-1}; \boldsymbol{b}_{y,k}) - \boldsymbol{\Gamma}^{k-1}_2\big), \\ \boldsymbol{\Gamma}^k_3 = \boldsymbol{\Gamma}^{k-1}_3 + \gamma_k \big(\overline{J}(\tau^{k-1}; \boldsymbol{b}_k) - \boldsymbol{\Gamma}^{k-1}_3\big), \\ S^k_{\tau,1} = S^{k-1}_{\tau,1} + \gamma_k \left(-\overline{U}(\boldsymbol{b}_{y,k})^T \partial_\tau h_2(\tau^{k-1}) - S^{k-1}_{\tau,1}\right), \\ S^k_{\tau,2} = S^{k-1}_{\tau,2} + \gamma_k \left(\widehat{U}(\boldsymbol{b}_k) \partial_\tau h_2(\tau^{k-1}) - S^{k-1}_{\tau,2}\right), \end{cases} \tag{15}$$

where $t \in [0, 1]$, $\overline{J}(\tau; \boldsymbol{b}_k) = N_k^{-1} \sum_{i=1}^{N_k} J(\tau; \boldsymbol{b}_{k,i})$,

$$\overline{I}_1(\xi; \boldsymbol{b}_{y,k}) = \sum_{i=1}^{N_k} I_1(\xi; \boldsymbol{b}_{y,k,i})/N_k,$$

$$\overline{I}_2(\xi; \boldsymbol{b}_{y,k}) = \sum_{i=1}^{N_k} I_2(\xi; \boldsymbol{b}_{y,k,i})/N_k,$$

$$\overline{U}(\boldsymbol{b}_{y,k})^T \partial_\tau h_2(\tau) = N_k{}^{-1} \sum_{i=1}^{N_k} U(\boldsymbol{b}_{y,k,i})^T \partial_\tau h_2(\tau),$$

$$\widehat{U}(\boldsymbol{b}_k)^T \partial_\tau h_2(\tau) = N_k{}^{-1} \sum_{i=1}^{N_k} U(\boldsymbol{b}_{k,i})^T \partial_\tau h_2(\tau),$$

$$\overline{H}(\xi; \boldsymbol{b}_k, \boldsymbol{b}_{y,k}) = \Bigg(\frac{1}{N_k} \sum_{i=1}^{N_k} S_\alpha(\xi; \boldsymbol{b}_{y,k,i})^T, \quad \frac{1}{N_k} \sum_{i=1}^{N_k} S_\beta(\xi; \boldsymbol{b}_{y,k,i})^T, \quad [-\overline{U}(\boldsymbol{b}_{y,k}) + \widehat{U}(\boldsymbol{b}_k)]^T \partial_\tau h_2(\tau)\Bigg)^T.$$

In addition, $\boldsymbol{\Gamma}(t)^k$ is a current estimate of $E[I_{\xi\xi}(\xi; \boldsymbol{b}) - t S_\xi(\xi; \boldsymbol{b})^{\otimes 2}|\boldsymbol{y}_o, \xi] + s_\xi(\xi; \boldsymbol{y}_o)^{\otimes 2}$ given by

$$\boldsymbol{\Gamma}^k_1 + [\boldsymbol{h}^k]^{\otimes 2} + t\boldsymbol{\Gamma}^k_2 + \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{\Gamma}^k_3 - (1+t)(S^k_{\tau,2})^{\otimes 2} + t S^k_{\tau,1} S^{kT}_{\tau,2} + t S^k_{\tau,2} S^{kT}_{\tau,1}, \end{pmatrix}. \tag{16}$$

Finally, the gain constants sequence $\{\gamma_k\}$ satisfies the following conditions:

$$0 \le \gamma_k \le 1 \text{ for all } k, \quad \sum_{k=1}^{\infty} \gamma_k = \infty \text{ and } \sum_{k=1}^{\infty} \gamma_k^2 < \infty. \tag{17}$$

An important feature of the SAEM algorithm is that it uses a gain constants sequence $\{\gamma_k\}$ to handle the noise in approximating $\partial_\xi \log L(\xi; \boldsymbol{y}_o)$ and $\partial^2_\xi \log L(\xi; \boldsymbol{y}_o)$ in Step 2 (Robbins and Monro, 1951; Lai, 2003). In principle, the choice of N_k should not affect the convergence of the stochastic approximation algorithm, but a good choice of N_k can improve the performance of the SAEM algorithm. At the end of the SAEM algorithm, $\boldsymbol{\Gamma}^k(1)$, $\boldsymbol{\Gamma}^k(0)$, and $\boldsymbol{\Gamma}^k(0) - \boldsymbol{\Gamma}^k(1)$ can be used to estimate the observed-data, completed-data and missing-data information matrix, respectively (Louis, 1982). For some models, $\boldsymbol{\Gamma}^k(1)$ may not be positive definite, but the corresponding $\boldsymbol{\Gamma}^k(0)$ is positive definite (Lange, 1995). Based on three examples in Section 4, we suggest using $\boldsymbol{\Gamma}^k(0)$ in the SAEM algorithm.

3.3 Two stages of the SAEM algorithm

Following Gu and Zhu (2001), our procedure to find $\hat{\xi}$ defined by (7) is composed of two stages. The main idea of the two stages is based on the observation that, if the starting point is not in the neighborhood of the maximum likelihood estimate, then the SAEM algorithm will usually converge slowly. In Stage I, we use a large gain constants sequence so that the parameter will move quickly into the vicinity of the maximum likelihood estimate. In Stage II, we use a small gain constants sequence to stabilize the algorithm in the neighborhood of the maximum likelihood estimate and an off-line averaging method to achieve the optimal convergence rate.

The main procedure is implemented as follows.

Stage I. Iterate Steps 1 and 2 with $i = 1, \ldots, K_1$ and the gain constants are defined by

$$\gamma_i = \gamma_{1i} = b_1/(i^{a_1} + b_1 - 1), \quad i = 1, \ldots, K_1,$$

where $K_1 \geq K_0$ is determined by

$$K_1 = \inf\left\{K \geq K_0 : \Delta_i^{(1)} = \left\| \sum_{i=K-K_0+1}^{K} \mathrm{Sign}(\xi^i - \xi^{i-1})/K_0 \right\| \leq \eta_1 \right\}. \tag{18}$$

Function Sign(z) is a vector of 1, 0, or -1 according to whether each component of z is positive, zero, or negative, respectively. Integers b_1 and K_0, real number $a_1 \in (0, 1)$, and η_1 are pre-assigned constants. We choose a_1 to be close to 0.5 and b_1 to be relatively large (e.g., $a_1 = 0.3$ and $b_1 = 5$) to obtain large gain constants. Also, we choose a relatively small value of η_1 and K_0 (e.g., $\eta_1 = 0.1$ and $K_0 = 100$) to ensure that the estimates ξ^is start to move around a certain point, possibly the maximum likelihood estimate.

Stage II. Take the seven estimates in the last iteration of Stage I as their initial values in Stage II. We iterate Steps 1 and 2 with $i = 1, \ldots, K_2$ and the gain constants are defined by

$$\gamma_i = \gamma_{2i} = b_2/(i^{a_2} + b_2 - 1), \quad i = 1, \ldots, K_2,$$

where integer b_2 and $a_2 \in (1/2, 1]$ are preassigned. We choose a_2 close to 1 and a small integer for b_2 (e.g., $a_2 = 0.8$, $b_2 = 2$) to obtain small gain constants and to stabilize the algorithm. We use an off-line averaging procedure at the same time. We set the initial estimates as $\tilde{\xi}^0 = \xi^{K_1}$, $\tilde{\boldsymbol{h}}^0 = \boldsymbol{h}^{K_1}$, $\tilde{\boldsymbol{\Gamma}}_1^0 = \boldsymbol{\Gamma}_1^{K_1}$, $\tilde{\boldsymbol{\Gamma}}_2^0 = \boldsymbol{\Gamma}_2^{K_1}$, $\tilde{\boldsymbol{\Gamma}}_3^0 = \boldsymbol{\Gamma}_3^{K_1}$, $\tilde{S}_{\tau,1}^0 = \tilde{S}_{\tau,1}^{K_1}$, $\tilde{S}_{\tau,2}^0 = \tilde{S}_{\tau,2}^{K_1}$, and $\tilde{\boldsymbol{\Gamma}}^0(t) = \boldsymbol{\Gamma}^{K_1}(t)$, and then update eight estimates as follows:

$$\begin{aligned}
\tilde{\xi}^i &= \tilde{\xi}^{i-1} + (\xi^i - \tilde{\xi}^{i-1})/i,\\
\tilde{\boldsymbol{h}}^i &= \tilde{\boldsymbol{h}}^{i-1} + (\boldsymbol{h}^i - \tilde{\boldsymbol{h}}^{i-1})/i,\\
\tilde{S}_{\tau,m'}^i &= \tilde{S}_{\tau,m'}^{i-1} + (S_{\tau,m'}^i - \tilde{S}_{\tau,m'}^i)/i,\\
\tilde{\boldsymbol{\Gamma}}_m^i &= \tilde{\boldsymbol{\Gamma}}_m^{i-1} + (\boldsymbol{\Gamma}_m^i - \tilde{\boldsymbol{\Gamma}}_m^{i-1})/i, \quad \text{and}\\
\tilde{\boldsymbol{\Gamma}}^i(t) &= \tilde{\boldsymbol{\Gamma}}^{i-1}(t) + [\boldsymbol{\Gamma}^i(t) - \tilde{\boldsymbol{\Gamma}}^{i-1}(t)]/i,
\end{aligned} \tag{19}$$

where $m = 1, 2, 3$ and $m' = 1, 2$. Theoretically, this off-line averaging procedure automatically leads to an optimal convergence without estimating the information matrix (Polyak, 1990; Polyak and Juditski, 1992). The stopping rule of Stage II is defined by

$$K_2 = \inf\left\{i : \quad \Delta_i^{(2)} \leq \eta_2\right\}, \tag{20}$$

where $\Delta_i^{(2)} = \tilde{\boldsymbol{h}}^{iT}[\tilde{\boldsymbol{\Gamma}}^i(1)]^{-1}\tilde{\boldsymbol{h}}^i + \mathrm{tr}\{[\tilde{\boldsymbol{\Gamma}}^i(1)]^{-1}\hat{\Sigma}\}/i$, in which $\hat{\Sigma}$ denotes an estimate of Σ, the covariance matrix of Monte Carlo error. A rough estimate of Σ can be achieved by taking the sample covariance of $\overline{H}(\xi^{k-1}; \boldsymbol{b}_k, \boldsymbol{b}_{y,k})$. The value of η_2 is usually taken to be around 0.002 to ensure small values of $s_\xi(\hat{\xi}; \boldsymbol{y}_o)$ as the convergence criterion of the SAEM algorithm (Gu and Zhu, 2001). At the K_2-th iteration, we use the off-line average $(\tilde{\xi}^{K_2}, \tilde{\boldsymbol{\Gamma}}^{K_2}(1))$ as our final estimate of $(\hat{\xi}, -\partial_\xi^2 \log L(\hat{\xi}; \boldsymbol{y}_o))$.

3.4 Convergence of the SAEM algorithm

We first establish the convergence of an algorithm, which is an approximation to the SAEM algorithm. Note that the maximum likelihood estimate $\hat{\xi}$, defined by (7), can be obtained as a solution to

$$s_\xi(\hat{\xi}, \boldsymbol{y}_o) = H(\hat{\xi}) = E[S_\xi(\hat{\xi}; \boldsymbol{b})|\boldsymbol{y}_o, \hat{\xi}] = \mathbf{0}. \tag{21}$$

Without loss of generality, we assume that $\partial_\tau \log C(\tau)$ and $\partial_\tau^2 \log C(\tau)$ can be evaluated analytically so that we can omit the step of sampling from the MRF (3). We also define $G_t(\xi, \boldsymbol{b}) = I_{\xi\xi}(\xi, \boldsymbol{b}) - tS_\xi(\xi; \boldsymbol{b})^{\otimes 2}$ and $G_t(\xi) = E[G_t(\xi, \boldsymbol{b})|\boldsymbol{y}_o, \hat{\xi}]$. In principle, Step 2 of the SAEM algorithm is equivalent to

$$\begin{cases}
\xi^k = \xi^{k-1} + \gamma_k[\boldsymbol{\Gamma}^k(t)]^{-1}\overline{H}(\xi^{k-1}; \boldsymbol{b}_{y,k}),\\
\boldsymbol{\Gamma}^k(t) = \boldsymbol{\Gamma}^{k-1}(t) + \gamma_k[\overline{G}_t(\xi^k, \boldsymbol{b}_{y,k}) - \boldsymbol{\Gamma}^{k-1}(t)],
\end{cases} \tag{22}$$

where $\overline{H}(\xi^{k-1}; \boldsymbol{b}_{y,k}) = \sum_{l=1}^{N_k} S_\xi(\xi; \boldsymbol{b}_{y,k,l})/N_k$ and $\overline{G}_t(\xi, \boldsymbol{b}_{y,k}) = \sum_{l=1}^{N_k} G_t(\xi; \boldsymbol{b}_{y,k,l})/N_k$. The basic iteration in (22) can be further viewed as an approximation to

$$\begin{cases}
\overline{\xi}^k = \overline{\xi}^{k-1} + \gamma_k[\overline{\boldsymbol{\Gamma}}^k(t)]^{-1}H(\overline{\xi}^{k-1}),\\
\overline{\boldsymbol{\Gamma}}^k(t) = \overline{\boldsymbol{\Gamma}}^{k-1}(t) + \gamma_k[G_t(\overline{\xi}^{k-1}) - \overline{\boldsymbol{\Gamma}}^{k-1}(t)],
\end{cases} \tag{23}$$

where we use notation $\overline{\xi}^k$ and $\overline{\boldsymbol{\Gamma}}^k(t)$ to represent the estimates generated from (23). The algorithm in (22) can be convergent only if (23) is convergent.

We establish the geometric convergence of (23) in Theorem 1. A detailed proof is given in Zhu and Gu (2005). The proof follows the general arguments for showing the convergence of the Newton-Raphson algorithm (Stoer and Bulisch, 1980). Let $\|\cdot\|$ be a norm on R^q, the norm of a $q \times q$ matrix A is defined as $\|A\| = \max_{\boldsymbol{x}:\|\boldsymbol{x}\|=1} \|A\boldsymbol{x}\|$, where $\boldsymbol{x} = (x_1, \ldots, x_q)^T$ is a vector. We assume that functions $H(\xi)$ and $G_t(\xi)$ are both differentiable on a convex set $C_0 \subset R^q$.

Lemma 1. *Assume that $\hat{\xi} \in C_0$ is a root of $H(\xi)$ and $\{\xi : \|\xi - \hat{\xi}\| \leq c_a\} \subset C_0$ for some c_a. Suppose that the sequence $\{\overline{\xi}^k, k > 0\}$ is defined by (23) with initial value $\overline{\xi}^0 \in C_0$. Assume that*

(a) $\|\partial_\xi H(\xi) - \partial_\xi H(\xi')\| \le c_\eta \|\xi - \xi'\|$ *for every* $\xi, \xi' \in C_0$*;*
(b) $\|\overline{\xi}^0 - \hat{\xi}\| \le c_a$*;*
(c) $\|[\overline{\boldsymbol{\Gamma}}^k(t)]^{-1}\| \le c_b$, *for* $k = 0, 1, \ldots$;
(d) $\|\boldsymbol{I}_q - B\| \le 1 - \lambda$, *where* $\boldsymbol{I}_q$ *is a* $q \times q$ *identity matrix,* $1 > \lambda \ge 0$ *and* $B = -[\overline{\boldsymbol{\Gamma}}^k(t)]^{-1}\partial_\xi H(\hat{\xi})$*;*
(e) $\delta = \lambda - c_a c_b c_\eta > 0$.

Then for each $k \ge 0$,

$$\|\overline{\xi}^k - \hat{\xi}\| \le \exp\{-\delta T_k\}\|\overline{\xi}^0 - \hat{\xi}\|, \tag{24}$$

where $T_0 = 0$ *and* $T_k = \sum_{i=1}^k \gamma_i$.

Theorem 1 (Geometric convergence of the algorithm (23)). *Suppose that* $\overline{\xi}^0 \in C_0$ *and* $\overline{\boldsymbol{\Gamma}}^0(t)$ *is a positive definite matrix and that* $\{\gamma_k, k \ge 1\}$ *is a sequence of positive numbers such that* $\gamma_k \le 1$. *For sequence* $\{\overline{\xi}^k, \overline{\boldsymbol{\Gamma}}^k(t), k \ge 0\}$ *as defined in* (23) *with initial values* $\overline{\xi}^0$ *and* $\overline{\boldsymbol{\Gamma}}^0(t)$, *assume that Assumption* (a) *to* (c) *in Lemma* 1 *are valid. Further assume that*

(f) $\|G_t(\xi) - G_t(\xi')\| \le c_d\|\xi - \xi'\|$ *for every* $\xi, \xi' \in C_0$*;*
(g) $\|G_t(\xi)^{-1}\| \le c_b$ *for all* $\xi \in C_0$*;*
(h) $\|G_t^{-1}(\xi)[G_t(\xi) + \partial_\xi H(\xi)]\| \le 1 - \lambda$ *for all* $\xi \in C_0$, *where* $1 \ge \lambda > 0$*;*
(i) $\|G_t(\hat{\xi}) - \overline{\boldsymbol{\Gamma}}^0(t)\| \le c_a c_d$*;*
(j) $\delta' = 1 - c_a c_b c_d + (1 - \lambda)(1 - c_a c_b c_d)^{-1} > 0$.

Then for each $k \ge 0$,

$$\|\overline{\xi}^k - \hat{\xi}\| \le \exp\{-\delta' T_k\}\|\overline{\xi}^0 - \hat{\xi}\|, \tag{25}$$

and if $T_k \ge 3$, *then*

$$\begin{aligned}\|\overline{\boldsymbol{\Gamma}}^k(t) - G_t(\hat{\xi})\| \le \exp\{-\delta' T_k/2\}[\|G_t(\hat{\xi}) \\ -\overline{\boldsymbol{\Gamma}}^0(t)\| + 2c_d\|\hat{\xi} - \overline{\xi}^0\|].\end{aligned} \tag{26}$$

Theorem 1 generalizes Ostrowski's Theorem to establish a geometric convergence of algorithm (23) (Ortega, 1990). If $\gamma_k = 1$ for all k, then algorithm (23) reduces to a standard iterative method for solving $s_\xi(\hat{\xi}, \boldsymbol{y}_0) = 0$. Under almost the same conditions as in Theorem 1, Ostrowski's Theorem states that $\hat{\xi}$ is a *point of attraction*. That is, there is an open neighborhood C_0 of $\hat{\xi}$ such that whenever $\overline{\xi}^0 \in C_0$, the iterations in algorithm (23) are well defined and the sequence $\overline{\xi}^k$ converges to $\hat{\xi}$ (Ortega, 1990, p. 144–145). Theorem 1 for algorithm (23) is more general because $\{\gamma_k : k \ge 1\}$ is a sequence of positive numbers satisfying $\gamma_k \le 1$.

Next, we provide a convergence theorem of the algorithm (22). When the functions $H(\xi)$ and $G_t(\xi)$ cannot be explicitly calculated, we may substitute them with Monte Carlo estimates, which results to the algorithm given by (22). We follow the ordinary differential equation (ODE) method (see Chapter 2 of Benveniste et al., 1990). Suppose that we have a vector function $\xi(v)$ and a matrix function $\boldsymbol{\Gamma}(v)$, $v \ge 0$, satisfying the ordinary differential equations

$$\frac{d\xi(v)}{dv} = \boldsymbol{\Gamma}(v)^{-1}H(\xi(v)), \quad \frac{d\boldsymbol{\Gamma}(v)}{dv} = G_t(\xi(v)) - \boldsymbol{\Gamma}(v), \tag{27}$$

with the initial condition $\xi(0) = \xi^0$ and $\boldsymbol{\Gamma}(0) = \boldsymbol{\Gamma}^0$. It is shown in Benveniste et al. (1990) that the function $\xi(v)$ is closely related to the estimates ξ^k produced by (22) with the same initial vector. It is easy to see that $(\hat{\xi}, G_t(\hat{\xi}))$ is a *stability point* of the above differential equation and all eigenvalues of $G_t(\hat{\xi})^{-1}\partial_\xi H(\hat{\xi})$ have negative real parts. A set D is called *a domain of attraction* of a stability point $(\hat{\xi}, G_t(\hat{\xi}))$ if the solution of (27) with $(\xi(0), \boldsymbol{\Gamma}(0)) \in D$ remains indefinitely in D and converges to $(\hat{\xi}, G_t(\hat{\xi}))$. Theorem 1 guarantees the existence of such a set D.

Assume that the transition probability $\Pi(\cdot, \cdot|\xi)$ satisfies the conditions (C.3)−(C.6) given in Gu and Kong (1998). We assume that (C.7) holds for the functions $S_\xi(\xi; \boldsymbol{b})$, $S_\xi(\xi; \boldsymbol{b})^{\otimes 2}$, and $I_{\xi\xi}(\xi; \boldsymbol{b})$. Under these conditions, the results considered in Gu and Kong (1998) hold in our case. Therefore, we have the following theorem.

Theorem 2. *Assume that the conditions* (C.1)–(C.7) *in Gu and Kong (1998) are valid. If* $\{(\xi^k, \boldsymbol{\Gamma}^k(t)), k \ge 1\}$ *from the SAEM algorithm* (22) *is a bounded sequence and visits infinitely often a compact subset of the domain of attraction of the stability point* $(\hat{\xi}, G_t(\hat{\xi}))$ *of the differential Eq.* (27) *almost surely, then*

$$\xi^k \to \hat{\xi} \quad and \quad \boldsymbol{\Gamma}^k(t) \to G_t(\hat{\xi}) \quad almost\ surely.$$

4 Examples

We used one simulation study and two real datasets to illustrate the performance of the SAEM algorithm. All computations were done in C on a SUN HPC4500 workstation. In all examples, the convergence criterion in (18) and (20) was used in Stages I and II, respectively, and (K_0, η_1, η_2) was set at (100, 0.1, 0.001).

4.1 State space model

State space models have received much consideration from both classical and Bayesian perspectives; see Durbin and Koopman (1997, 2000) and references therein. One special state space model considered in Durbin and Koopman (2000) assumes that the latent process $\{\boldsymbol{b}(s)\}$ satisfies $\boldsymbol{b}(s) = \boldsymbol{u}(s)\boldsymbol{b}(s-1) + \boldsymbol{r}(s)\epsilon(s)$, where $\epsilon(s) \sim p(\cdot|\tau)$ and both $\boldsymbol{u}(s)$ and $\boldsymbol{r}(s)$ may depend on unknown parameters. Given $\{\boldsymbol{b}(s)\}$, the observations $\boldsymbol{y}(s)$ are conditionally independent and follow the distribution (1) with $\mu(s) = \boldsymbol{x}(s)^T\beta + \boldsymbol{b}(s)$. Although the maximum likelihood estimate has good theoretical properties (Ledet and Petersen, 1999), the log-likelihood function of this model does not have a simple closed form and so the maximum likelihood estimate is usually intractable (Chan and Ledolter, 1995; Durbin and Koopman, 2000). We apply the SAEM algorithm for finding the maximum likelihood estimate of the state space model.

The Polio Incidence data reported in Zeger (1988) gave the monthly number of cases of poliomyelitis from January 1970-December 1983. This data is seasonal. Following Chan and Ledolter (1995), we model the dataset by

$$\boldsymbol{y}(s)|\boldsymbol{b}(s) \sim \text{Possion}(\mu(s)), \quad \log\mu(s) = \boldsymbol{x}(s)^T\beta + \boldsymbol{b}(s),$$

and $\boldsymbol{b}(s) = \rho\boldsymbol{b}(s-1) + \epsilon(s)$ for $s = 1, \ldots, 168$, where $\epsilon \sim N(0, \exp(\sigma_\epsilon))$ and $\boldsymbol{x}(s)$ is given by $(1, s/1000, \cos(2\pi s/12), \sin(2\pi s/12), \cos(2\pi s/6), \sin(2\pi s/6))^T$.

To sample $\boldsymbol{b} = \{\boldsymbol{b}(s)\}$ conditional on $\boldsymbol{y}_o = \{\boldsymbol{y}(s)\}$, we used the random-walk Metropolis algorithm to sample from the full conditional densities $p(\boldsymbol{b}(s)|\text{all other } \boldsymbol{b}(t), \boldsymbol{y}_o)$ (Chan and Ledolter, 1995; Eq. (9)) as follows. At the rth iteration of the Metropolis algorithm with a current value $\boldsymbol{b}(s)^{(r)}$, a new candidate $\boldsymbol{b}(s)^*$ is generated from $N[\boldsymbol{b}(s)^{(r)}, \sigma^2]$ and the probability of accepting this new candidate is

$$\min\left\{1, \frac{p(\boldsymbol{b}(s)^*|\text{all other } \boldsymbol{b}(t), \boldsymbol{y}_o)}{p(\boldsymbol{b}(s)^{(r)}|\text{all other } \boldsymbol{b}(t), \boldsymbol{y}_o)}\right\}.$$

The σ^2 is chosen to be 1.0 so that the average acceptance rate is approximately 0.44.

We applied the SAEM algorithm with $(a_1, b_1; a_2, b_2) = (0.2, 4; 0.8, 2)$ and $N_k = 40$ to find the maximum likelihood estimate. The initial value for $\xi = (\beta, \rho, \sigma_\epsilon)$ was set to be $\xi^0 = \mathbf{0}$. Figure 1(a) gives the plot of $\Delta_i^{(2)}$ in Stage II of the SAEM algorithm with $t = 0$, and Fig. 1(b) shows the estimates $(\beta_1^k, \tilde{\beta}_1^k)$ and $(\rho^k, \tilde{\rho}^k)$ at each iteration of the SAEM algorithm with $t = 0$. Figure 1(a) shows that the SAEM algorithm converges very quickly. Figure 1(b) shows that a large gain constants sequence in Stage I can force the estimates into a small neighborhood of $\hat{\xi}$, with all parameters oscillating around the maximum likelihood estimate after the 200-th iteration. Because the SAEM algorithm with $t = 1$ shows similar behavior, we omit it.

To make a comparison, we ran the Monte Carlo EM algorithm with $\gamma_k = 1$ and $N_k = 2k^2 + 40$ for 100 iterations and also included the maximum likelihood estimate in Table 1 (Chan and Ledolter, 1995; Wei and Tanner, 1990; Zhang, 2002). The initial value for $\xi = (\beta, \rho, \sigma_\epsilon)$ was set to be $\xi^0 = \mathbf{0}$. Figure 1(c) shows the relative change between consecutive estimates $\Delta_k = \|\xi^k - \xi^{k-1}\|/8$, which is a common stopping criterion for the Monte Carlo EM algorithm. It also shows that with increasing N_k, all Δ_k still oscillate around zero and do not show clear convergence. The Monte Carlo EM algorithm also converges to the maximum likelihood estimate; however, the Monte Carlo EM algorithm requires more computational time. Figure 1(d) presents the three elements of ξ^k at each iteration of the Monte Carlo EM algorithm. All those estimates oscillate around the maximum likelihood estimate even for a large N_k.

The maximum likelihood estimates obtained from the SAEM and Monte Carlo EM algorithms are included in Table 1. We observed that the maximum likelihood estimates obtained from all algorithms are close to each other; however, the SAEM algorithm with $t = 0$ apparently outperforms that with $t = 1$ in terms of computer time and the number of iterations.

4.2 Noisy Ising model

The Ising model is a well-known MRF with a binary random variable $\boldsymbol{b}(i, j) \in \{0, 1\}$ at each site (i, j) on a regular

Table 1 Model fits to polio incidence data

Iter/time		β_1	β_2	β_3	β_4	β_5	β_6	ρ	σ_ϵ
				SAEM with $t = 0$					
1923	EST	0.228	−3.717	0.166	−0.483	0.412	−0.011	0.650	−1.267
350s	SE	0.125	1.346	0.090	0.115	0.101	0.098	0.060	0.110
				SAEM with $t = 1$					
2470	EST	0.241	−3.786	0.163	−0.482	0.414	−0.011	0.639	−1.238
532s	SE	0.126	1.347	0.090	0.116	0.105	0.101	0.058	0.111
				Monte Carlo EM algorithm					
100	EST	0.242	−3.787	0.162	−0.479	0.414	−0.010	0.649	−1.270
3392s	SE	0.125	1.346	0.090	0.115	0.101	0.098	0.059	0.110

Iter denotes the number of iterations; EST denotes the estimates; SE denotes the standard errors of estimates.

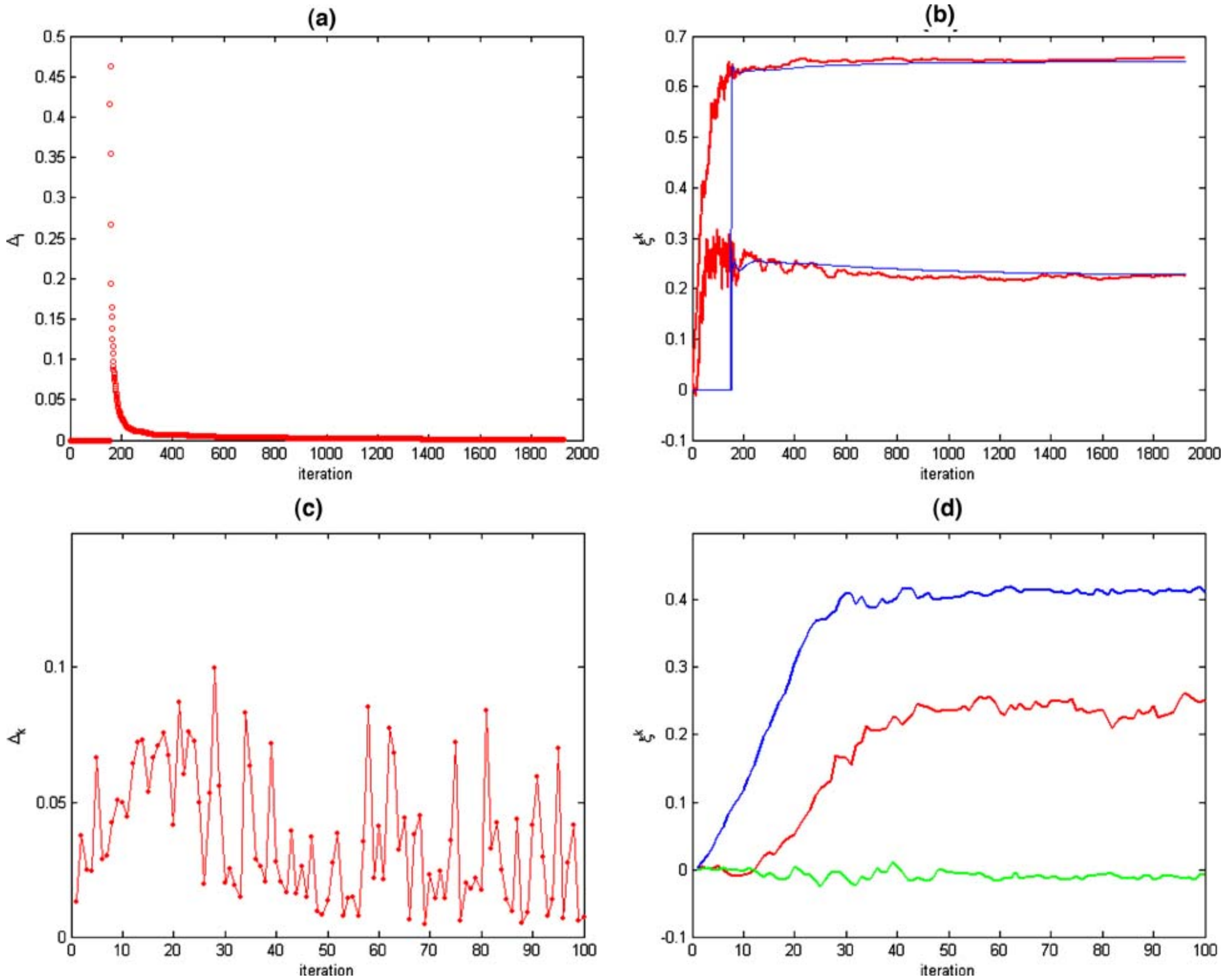

Fig. 1 Polio Incidence data: (a) $\Delta_k^{(2)}$ at each iteration of Stage II of the SAEM algorithm with $t = 0$; (b) (β_1^k, ρ^k) (red lines) and $(\tilde{\beta}_1^k, \tilde{\rho}^k)$ (blue lines) at each iteration of the SAEM algorithm with $t = 0$; (c) Δ_k at each iteration of the Monte Carlo EM algorithm; (d) β_1^k (red line), β_5^k (blue line), and β_6^k (green line) at each iteration of the Monte Carlo EM algorithm

$M_0 \times N_0$ lattice $\mathcal{Z}^2_{M_0,N_0}$. The two-dimensional Ising model without external field term can be written as

$$p(\boldsymbol{b}|\theta) \propto \exp\left\{\tau \sum_{nn} \delta(\boldsymbol{b}(i, j), \boldsymbol{b}(u, v))\right\}, \tag{28}$$

where nn means that the summation is over all the pairs of first-order neighboring points on the plane (Besag, 1974). The potential function is $-\tau \sum_{nn} \delta(\boldsymbol{b}(i, j), \boldsymbol{b}(u, v))$ and the normalizing factor $C(\tau)$ is obtained by summing over all possible configurations $\boldsymbol{b}$. Because the parameter $\tau > 0$ measures the degree of homogeneity in neighborhood sites, this distribution invites clustering of like-valued pixels. We consider a noisy version of the true scene as our data $\{\boldsymbol{y}(i, j) : (i, j) \in \mathcal{Z}^2_{M_0,N_0}\}$ with

$$\boldsymbol{y}(i, j) = \boldsymbol{b}(i, j) + \epsilon(i, j), \tag{29}$$

where errors $\epsilon(i, j)$ are identically and independently distributed as Gaussian noise with mean zero and variance $\exp(\sigma)$.

To evaluate the usefulness of the proposed algorithm, we consider the following simulation study for the noisy Ising model. In this simulation study, the Ising model is set on a 30 × 30 square lattice on the plane. The periodic boundary for the square lattice is assumed. Swendsen and Wang's (1987) algorithm was used to simulate the process $\{\boldsymbol{b}(i, j) : i, j = 1, \ldots, 30\}$. This algorithm uses auxiliary bond variables and is designed to speed up simulations with very large Ising models (Hidgon, 1998). The key idea of Swendsen and Wang's (1987) algorithm is summarized as follows: Let $\boldsymbol{u} = \{\boldsymbol{u}((i, j), (u, v)) : (i, j) \sim (u, v)\}$ be a set of auxiliary variables, the joint distribution $p(\boldsymbol{u}, \boldsymbol{b})$ has the distribution (28) as the marginal distribution for $\boldsymbol{b}$ and $p(\boldsymbol{u}|\boldsymbol{b})$ and $p(\boldsymbol{b}|\boldsymbol{u})$ are easy to sample from. Following Hidgon (1998), we choose

$$p(\boldsymbol{u}, \boldsymbol{b}) \propto I(0 \le \boldsymbol{u}((i, j), (u, v)) \le \exp(\tau\delta(\boldsymbol{b}(i, j), \boldsymbol{b}(u, v)))),$$

where $I(\cdot)$ is an indicator function. A Gibbs sampler is then used to sample iteratively from $p(\boldsymbol{u}|\boldsymbol{b})$ and $p(\boldsymbol{b}|\boldsymbol{u})$.

The initial state of the process is taken at random such that $\boldsymbol{b}(i, j)$ is independently $\{0, 1\}$ with equal probability. Swendsen and Wang's (1987) algorithm was repeated 4000 times to ensure that the equilibrium states were achieved and Gaussian noise with mean zero and variance $\exp(-0.5)$ was added to produce a "noisy" dataset.

We simulated 500 datasets for each parameter value $\tau_0 \in \{0.2, 0.4, 0.6, 0.8\}$. Based on the simulated datasets, we applied the SAEM algorithm with $t = 0$ as described in Section 3 to obtain the maximum likelihood estimate of $\xi = (\tau, \sigma)$. The starting value of ξ^0 was taken to be $(0.1, 0.0)$. The SAEM algorithm with $(a_1, b_1; a_2, b_2) = (0.4, 5; 0.8, 2)$ converged quickly. In order to estimate $\partial_\tau \log C(\tau)$ and $\partial_\tau^2 \log C(\tau)$, the Metropolis algorithm (Gu and Zhu, 2001) was applied at each site (i, j) to generate $\boldsymbol{b}$ from (28) in each iteration of the SAEM algorithm.

To simulate the random sample from $\boldsymbol{b}$ conditional on $\boldsymbol{y}_o$, we used the following Metropolis algorithm. Let the current value of the process at site (l, j) be $\boldsymbol{b}(l, j)$ and the current value of the potential function be U. Take the alternative value $\boldsymbol{b}(l, j)^*$ at the site (l, j), which leads to the value of the potential function U^*, the probability of accepting this new candidate $\boldsymbol{b}(l, j)^*$ and U^* is

$$\min\{1, \exp\{(U - U^*) + 0.5[\boldsymbol{y}(l, j) - \boldsymbol{b}(l, j)]^2 \\ \times \exp(-\sigma) - 0.5[\boldsymbol{y}(l, j) - \boldsymbol{b}(l, j)^*]^2 \exp(-\sigma)\}\}.$$

Moreover, each site was selected at random with $1/(30 \times 30)$ probability not according to the lexicographical order. For example, if site (1,1) was selected, we ran the above mentioned Metropolis procedure at the site (1,1) with other sites unchanged. Thus, only the value at one site can change from $\boldsymbol{b}_{k,i-1}$ to $\boldsymbol{b}_{k,i}$. The number N_k was set at $N_k = 20000$. Compared with the total number of sites $30 \times 30 = 900$, $N_k = 20000$ is not exceedingly large.

To illustrate the performance of the SAEM algorithm, we calculated the bias, the mean of the standard deviation estimates, and the root mean-square error obtained from the 500 estimates. We also obtained the mean of the number of iterations for each estimate and the average CPU time for each estimate. The results are summarized in Table 2. The performance of the SAEM algorithm was very good, with all relative efficiencies (the ratio of the mean of the standard deviation estimates and the root mean-square error) close to 1.0. The computational time decreased with the values of β (the higher β, the stronger spatial aggregation). In addition, according to our experience (not presented here), the SAEM algorithm could converge very slow when $-\sigma$ is extremely high (e.g., $\sigma = -1.4$).

4.3 Segmentation of brain magnetic resonance images

Magnetic resonance imaging (MRI) of the brain provides detailed information about brain tissues (i.e., White Matter (WM), Gey Matter (GM), or Cerebro Spinal Fluid (CSF)). Such information is central for quantitative studies of certain illnesses, pre- and intra- operative guidance for therapeutic intervention, and advanced morphometric techniques for research, among other uses. Developing methods for assigning each voxel of MRI to a specific tissue have been an active field. See, for example, Zhang et al. (2001), Rajapakse et al. (1997), and Marroquin et al. (2003), among many others. Statistical models, including spatial random effects model (6), have been used to account for noise inherent in the signal intensities of MRI and accurately assigning tissues to voxels in MRIs.

To demonstrate the application of spatial random effects model (6) in image segmentation, we used a single slice of a MRI volume, which was generated by using the Brainweb MRI simulator (Kwan et al., 1999). The image grid on the slice is 181×181 and the in-plane spatial resolution is 1 × 1 mm. For simplicity, only two major tissues (WM and GM) on the slice were considered here. The "anatomical model" in Fig. 2(a) shows true tissue identity (WM or GM) in each voxel of the slice; 5904 voxels contain WM and 6083 voxels contain GM. Figure 2(a) shows that the same tissues cluster together with GM encircling WM. The simulated MRI slices with the 0% and 12% noise levels are, respectively, shown in Fig. 2(b) and (c). Mechanism for simulating MRI data and adding physical noise has been reported in Kwan et al. (1999).

Table 2 Bias, RMS, SD, and EFF of the maximum likelihood estimates of the noisy Ising model

	β				$\sigma = -0.5$					
True	Bias	RMS	SD	EFF	Bias	RMS	SD	EFF	AVEN	AVET
0.20	0.005	0.138	0.143	0.968	0.006	0.063	0.063	0.995	2130	720s
0.40	0.008	0.111	0.117	0.946	0.004	0.062	0.064	0.968	1767	605s
0.60	0.014	0.077	0.084	0.915	0.007	0.061	0.059	1.035	1325	480s
0.80	0.010	0.041	0.043	0.961	0.006	0.058	0.057	1.043	1429	498s

True denotes the true value of parameters; Bias denotes the bias of the mean of estimates; RMS denotes the root-mean-square error; SD denotes the mean of standard deviation estimates; AVEN denotes the average of the number of iterations for each estimate; AVET denotes the mean CPU time; and EFF denotes the ratio of SD and RMS.

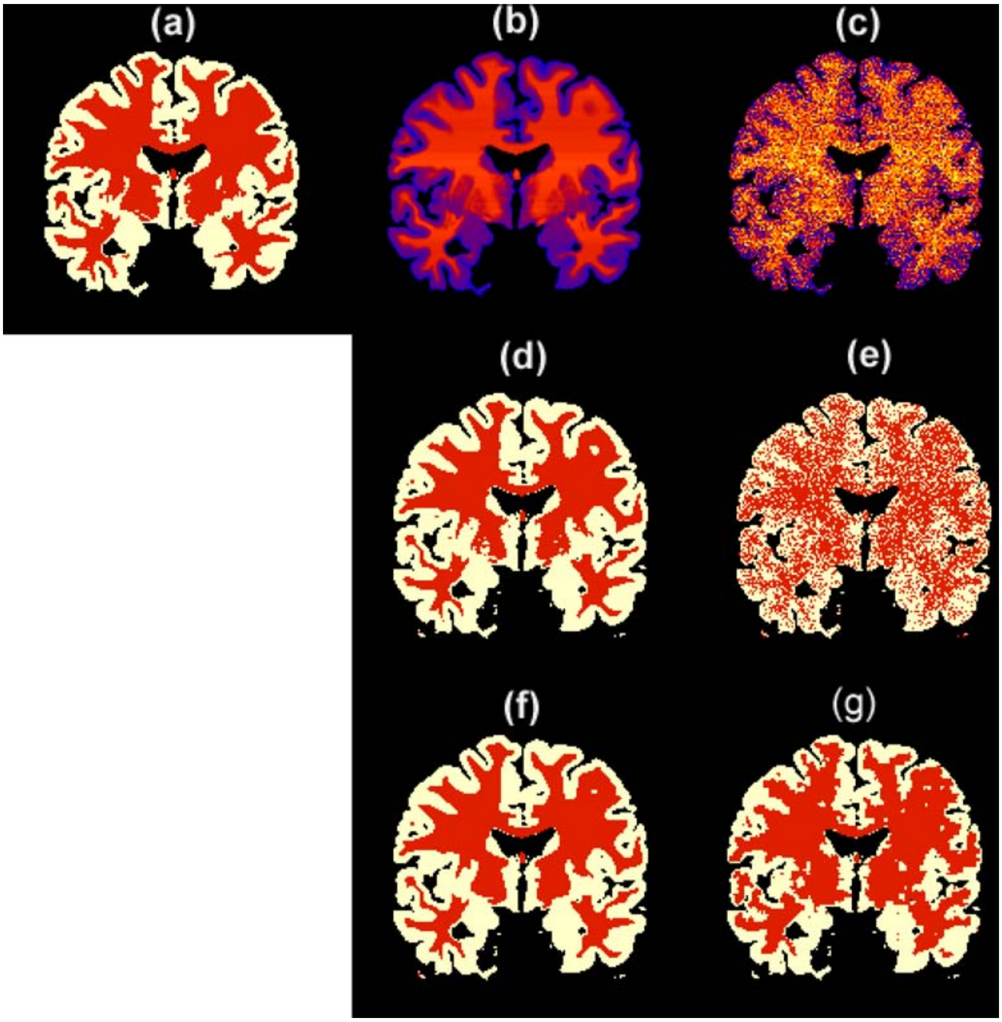

Fig. 2 Segmentation of MRI data: an anatomical model giving true identity (White Matter (colored red) or Grey Matter (colored white)) in each voxel of the MRI slice (a); the simulated MRI slice with the 0 percent noise (b, d, and f); and the simulated MRI slice with the 12 percent noise (c, e, and g). For the MRI slice with 0 percent noise, we show the raw data in (b), the estimated anatomical model from mixture model in (d), and the estimated anatomical model from spatial random effects model and by using ICM in (f). For the MRI slice with 12 percent noise, we show the raw data in (c), the estimated anatomical model from mixture model in (e), and the estimated anatomical model from spatial random effects model and by using ICM in (g)

Given the ground truth in Fig. 2(a), we performed some preliminary analyses on the MRI slice with different noise levels. For the slice with the 0% noise level, we calculated the mean and standard deviation of image intensities for WM as (702.462, 31.335), and those for GM as (530.245, 51.133). However, for the slice with the 12% noise level, the mean and standard deviation of image intensities for WM and GM are, respectively, (809.397, 173.675) and (612.819, 171.809).

We used a mixture normal model to cluster two tissues (GM and WM) on the two simulated MRI slices. The signal intensity $\boldsymbol{y}(s_i)$ at each voxel s_i can be written as

$$\boldsymbol{y}(s_i) = \sum_{k=1}^{2} [\beta(k) + \epsilon_k(s_i)]\delta(\boldsymbol{b}(s_i), k), \tag{30}$$

where $\beta(1)$ and $\beta(2)$ denote, respectively, the mean signal intensities of WM and GM, $\epsilon_k(s_i)$ follows the normal distribution with zero mean and variance $\exp(\sigma_k)$ for $k = 1$ and 2, and $b(s_i) = 1$ (or 2) represents unknown tissue type WM (or GM). Furthermore, all $\boldsymbol{b}(s_i)$ are binary variables and identically and independently distributed. In addition, $P(\boldsymbol{b}(s_i) = 1) = 1/[1 + \exp(\tau_1(2))]$ and $P(\boldsymbol{b}(s_i) = 2) = \exp(\tau_1(2))/[1 + \exp(\tau_1(2))]$. The unknown parameter vector ξ for the mixture model is given by $\xi = (\beta(1), \sigma_1, \beta(2), \sigma_2, \tau_1(2))^T$. Given the maximum likelihood estimate $\hat{\xi}$, we can obtain an estimate of the conditional probability $P(\boldsymbol{b}(s_i) = 1|\hat{\xi})$ at each s_i as follow:

$$\frac{\phi(\boldsymbol{y}(s_i); \hat{\beta}(1), \hat{\sigma}_1)}{\phi(\boldsymbol{y}(s_i); \hat{\beta}(1), \hat{\sigma}_1) + \phi(\boldsymbol{y}(s_i); \hat{\beta}(2), \hat{\sigma}_2)\exp(\hat{\tau}_1(2))},$$

where $\phi(\boldsymbol{y}(s_i);\beta,\sigma)=\exp(-0.5\sigma-0.5(\boldsymbol{y}(s_i)-\beta)^2/\exp(\sigma))$. Then, we can compute the sum of the correct prediction (SCP) as follow:

$$\mathrm{SCP}=\sum_{i=1}^{n}\delta\big(\boldsymbol{b}(s_i)^{true},\boldsymbol{b}(s_i)^{pred}\big) \tag{31}$$

where $n=11987$, $\boldsymbol{b}(s_i)^{true}$ denotes the true $\boldsymbol{b}(s_i)$, and $\boldsymbol{b}(s_i)^{pred}$ equals 1 when $\mathrm{P}(\boldsymbol{b}(s_i)=1|\hat{\xi})>0.5$ and 2 otherwise.

For the MRI slice with the 0% noise level, we applied the expectation-maximization algorithm to find $\hat{\xi}=(713.068, 5.983, 548.409, 8.355, 0.385)^T$ with $\mathrm{SCP}=906$. For the MRI slice with the 12% noise level, $\hat{\xi}$ is $(808.591, 10.358, 604.351, 10.183, -0.062)^T$ with $\mathrm{SCP}=3371$. We present $\boldsymbol{b}^{pred}$ for both MRI slices in Fig. 2(d) and (e). As expected, higher levels of noise leads to larger SCPs. Moreover, for data with high levels of noise, the mixture model cannot recover the cohesion of the same tissues.

We applied model (6) to cluster two tissues (WM and GM) on the MRI slices with differing noise levels, and we calculated $\hat{\xi}$ by using the SAEM algorithm. Because S in Fig. 2(a) is an irregular lattice, we consider the joint distribution of the site responses $\boldsymbol{b}=\{\boldsymbol{b}(s_i):s_i\in S\}$ conditional upon responses $\boldsymbol{b}^O=\{\boldsymbol{b}(s_i):s_i\in S^O\}$, where S^O and S denote the set of all sites forming the outside of S and the set of all sites of S, respectively. Furthermore, we assume that all $\boldsymbol{b}(s_i)$ for $s_i\in S^O$ equal 2, which requires that the tissues close to the boundary are GM, although we only use information from the voxels near the boundary of S. We assume model (31), but the joint distribution of the latent field $\boldsymbol{b}$ given $\boldsymbol{b}^O$ can be written as

$$\exp\left\{\tau_1(2)\sum_{i=1}^{n}\delta(\boldsymbol{b}(s_i),2)+\tau_2\sum_{s_i\sim s_j}\delta(\boldsymbol{b}(s_i),\boldsymbol{b}(s_j))\right.$$
$$\left.-\log C(\tau_1(2),\tau_2)\right\}, \tag{32}$$

where the first-order neighboring correlation is used (Huffer and Wu, 1998). The unknown parameter vector ξ is $(\beta_1,\sigma_1,\beta_2,\sigma_2,\tau_1(2),\tau_2)^T$.

To estimate $\partial_\tau\log C(\tau)$ and $\partial_\tau^2\log C(\tau)$, we applied the Metropolis algorithm at each site s_i to generate $\boldsymbol{b}$ from (32) in each iteration of the proposed algorithm (Gu and Zhu, 2001). To simulate the random sample from $\boldsymbol{b}$ conditional on $\boldsymbol{y}_o$ and $\boldsymbol{b}^O$, the following Metropolis algorithm was used. We call it Sampling Method (I). Let the current value of the process at site s_i be $\boldsymbol{b}(s_i)$ and the current value of the potential function is denoted as U. Take the alternative value $\boldsymbol{b}(s_i)^*$ at the site s_i which leads to the value of the potential function U^*, the probability of accepting this new candidate $\boldsymbol{b}(s_i)^*$ and U^* is

$$\min\{1,\exp\{(U-U^*)+0.5[y(s_i)-\beta(k)]^2e^{-\sigma_k}$$
$$-0.5[y(s_i)-\beta(m)]^2e^{-\sigma_m}+0.5[\sigma_k-\sigma_m]\}\},$$

where $\boldsymbol{b}(s_i)=k$ and $\boldsymbol{b}(s_i^*)=m$. Moreover, each site was selected at random with $1/n=1/11987$ probability. The number N_k was set at $N_k=60,000$, which is about five times the number of sites in S.

For the two simulated MRI slices with differing noise levels, we applied the SAEM algorithm with $(a_1,b_1;a_2,b_2)=(0.2,5;0.8,2)$ and $t=0$ to find the maximum likelihood estimate of spatial random effects model (6). For the MRI slice with the 0% noise level, ξ^0 was set at $(713.07, 5.98, 548.41, 8.36, 0.39, 0.0, 1.0)^T$ and the SAEM algorithm took 4032 iterations to obtain

$$\hat{\xi}^T=(710.78, 6.18, 541.70, 8.17, -3.62, 1.84)$$

with standard errors $(0.84, 0.02, 0.39, 0.03, 0.02, 0.01)$. Given the maximum likelihood estimate $\hat{\xi}$, we applied the iterated conditional modes (ICM) to obtain an estimate of $\boldsymbol{b}^{pred}$ and the value of SCP as 695. Recall that $\mathrm{SCP}=906$ based on the mixture model. We present $\boldsymbol{b}^{pred}$ in Fig. 2(f). For the MRI slice with the 12% noise level, the SAEM algorithm starts from initial value $\xi^0=(713.07, 5.98, 548.41, 8.36, 0.39, 0.0, 1.0)^T$ and converges to $\hat{\xi}^T=(823.79, 10.19, 582.06, 10.04, -3.50, 1.77)$ with standard errors $(2.89, 0.02, 2.90, 0.02, 0.01, 0.01)$ in 2393 iterations. We then applied ICM to estimate $\boldsymbol{b}^{pred}$; see Fig. 2(g). Moreover, the value of $\mathrm{SCP}=1463$ is much smaller than $\mathrm{SCP}=3371$ based on the mixture model described above. It reveals that introducing spatial correlation truly improves performance of the segregation methods.

For illustration, we present the performance of the SAEM algorithm for the MRI slice with the 12% noise level in Fig. 3. Figure 3(a) gives $\Delta_i^{(1)}$ and $\Delta_i^{(2)}$ at each iteration of each stage of the SAEM algorithm with $t=0$. It indicates that all parameters oscillate around the maximum likelihood estimate from the 600th iteration. Furthermore, starting from ξ^0, the estimates $(\beta(1)^k,\beta(2)^k)$, $(\tau_1(2)^k,\tau_2^k)$, and (σ_1^k,σ_2^k) at each iteration are shown in Fig. 3(b)–(d), respectively. A large gain constants sequence in Stage I can force the estimates into a small neighborhood of $\hat{\xi}$ and all estimates start to oscillate in Stage II.

5 Discussion

We have introduced a class of spatial random effects models and provided the SAEM algorithm to calculate the maximum likelihood estimate of these spatial random effects models. Many issues, however, merit further research. For instance, the SAEM algorithm could be applied to problems of missing

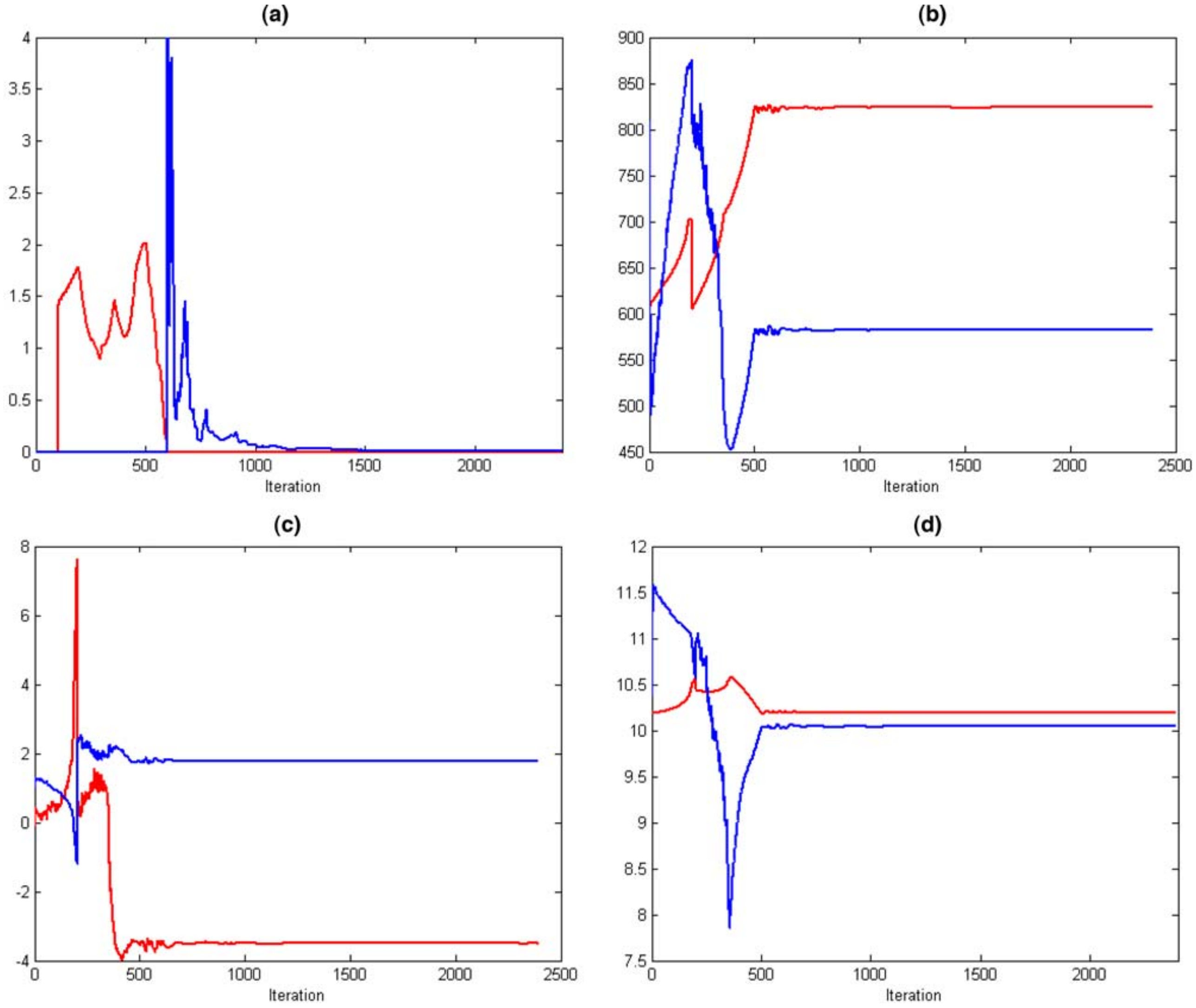

Fig. 3 Segmentation of MRI data: (a) $\Delta_k^{(1)}$ (blue line) and $\Delta_k^{(2)}$ (red line) at each iteration of Stages I and II of the SAEM algorithm; (b) β_1^k (red line) and β_2^k (blue line) at each iteration; (c) $\tau_1(2)^k$ (red line) and τ_2^k (blue line) at each iteration; (d) σ_1^k (red line) and σ_2^k (blue line) at each iteration

data, such as those found in generalized linear mixed measurement error models (Wang et al., 1998), parametric regression models with missing covariates (Horton and Laird, 1998), and generalized nonparametric mixed effects models (Karcher and Wang, 2001). The SAEM algorithm should provide an efficient algorithm for finding the maximum likelihood estimate of those models for missing data. *Pairwise interaction Markov Random Fields* are another widely used class of distributions in spatial statistics (Besag et al., 1995); spatial random effects models, however, exclude this model, because we assume that the sites of latent process of ***b*** are predetermined and non-random. Future work applying SAEM algorithms to the pairwise interaction MRFs will need to develop an efficient algorithm for sampling from the latent process. Further research should also calculate Bayesian estimates of spatial random effects models. Here, sampling from the distribution of parameters associated with the normalizing factor of MRFs will be the primary difficulty (Liu, 2001).

Acknowledgments This work was supported in part by NSF grant SES-0643663 to Dr. Zhu, Hong Kong Research Grants Council competitive grant 216078 to Dr. Gu, NIDA grant DA017820, NIMH grants MH068318 and K0274677 to Dr. Peterson, by the Suzanne Crosby Murphy Endowment at Columbia University Medical Center, and by the Thomas D. Klingenstein and Nancy D. Perlman Family Fund. We thank the Editor, an Associate Editor, and three anonymous referees for valuable suggestions, which greatly helped improve our presentation. Thanks to Dr. Jason Royal for his invaluable editorial assistance.

References

Aitkin M. 1996. A general maximum likelihood analysis of overdispersion in generalized linear models. Statistics and Computing 6: 251–262.

Benveniste A., Métivier M., and Priouret P. 1990. Adaptive Algorithms and Stochastic Approximations. Springer-Verlag, New York.

Besag J.E. 1974. Spatial interaction and the statistical analysis of lattice systems (with discussion). Journal of Royal Statistical Society, Series B 36: 192–236.

Besag J.E. 1986. On the statistical analysis of dirty pictures (with discussion). Journal of Royal Statistical Society, Series B 48: 259–302.

Besag J.E., Green P., Higdon D., and Mengersen K. 1995. Bayesian computation and stochastic systems (with discussion). Statistical Science 10: 3–66.

Breslow N.E. and Clayton D.G. 1993. Approximate inference in generalized linear mixed models. Journal of American Statistical Association 88: 9–25.

Chan K.S. and Ledolter J. 1995. Monte Carlo EM estimation for time series models involving counts. Journal of American Statistical Association 90: 242–252.

Christensen O.F. and Waagepetersen R.P. 2002. Bayesian prediction of spatial count data using generalized linear mixed models. Biometrics 58: 280–286.

Delyon B., Lavielle E., and Moulines E. 1999. Convergence of a stochastic approximation version of the EM algorithm. Annals of Statistics 27: 94–128.

Derin H. and Elliott H. 1987. Modeling and segmentation of noisy and textured images using Gibbs random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 9: 39–55.

Diggle P.J., Tawn J.A., and Moyeed R.A. 1998. Model-based geostatistics (with discussion). Applied Statistics 47: 299–350.

Durbin J. and Koopman S.J. 1997. Monte Carlo maximum likelihood estimation for non-Gaussian state space models. Biometrika 84: 669–684.

Durbin J. and Koopman S.J. 2000. Time series analysis of non-Gaussian observations based on state space models from both classical and Bayesian perspectives (with discussion). Journal of Royal Statistical Society, Series B 62: 3–56.

Gelman A. and Meng X.L. 1998. Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. Statistical Science 13: 163–185.

Geman S. and Geman D. 1984. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6: 721–741.

Geyer C.J. and Thompson E.A. 1992. Constrained Monte Carlo maximum likelihood for dependent data (with discussion). Journal of Royal Statistical Society, Series B 54: 657–699.

Gu M.G. and Kong F.H. 1998. A stochastic approximation algorithm with Markov chain Monte Carlo method for incomplete data estimation problems. In: Proceeding of National Academic Science of USA 95: 7270–7274.

Gu M.G. and Zhu H.T. 2001. Maximum likelihood estimation for spatial models by Markov chain Monte Carlo stochastic approximation. Journal of Royal Statistical Society, Series B 63: 339–355.

Higdon D.M. 1998. Auxiliary variable methods for Markov chain Monte Carlo with applications. Journal of American Statistical Association 93: 585–595.

Horton N.J. and Laird N.M. 1998. Maximum likelihood analysis of generalized linear models with missing covariates. Statistical Methods in Medical Research 8: 37–50.

Huang F. and Ogata Y. 2001. Comparison of two methods for calculating the partition functions of various spatial statistical models. The Australian and New Zealand Journal of Statistics 43: 47–65.

Huffer F.W. and Wu H.L. 1998. Markov chain Monte Carlo for autologistic regression models with application to the distribution of plant species. Biometrics 54: 509–524.

Jens L.J. and Niels V.P. 1999. Asymptotic normality of the maximum likelihood estimator in state space models. Annals of Statistics 27: 514–535.

Karcher P. and Wang Y. 2001. Generalized nonparametric mixed effects models. Journal of Computational and Graphical Statistics 10: 641–655.

Kwan R.K.S., Evans A.C., and Pike G.B. 1999. MRI simulation-based evaluation of image-processing and classification methods. IEEE Transactions on Medical Imaging 18: 1085–1097.

Lai T.L. 2003. Stochastic approximation. Annals of Statistics 31: 391–406.

Lange K. 1995. A gradient algorithm locally equivalent to the EM algorithm. Journal of Royal Statistical Society, Series B 55: 425–437.

Lee Y. and Nelder J.A. 1996. Hierarchical generalized linear models (with discussion). Journal of Royal Statistical Society, Series B 58: 619–678.

Li S.Z. 2001. Markov Random Field Modeling in Image Analysis. Springer-Verlag, Tokyo.

Liu J. 2001. Monte Carlo Strategies in Scientific Computing. Springer, New York.

Louis T.A. 1982. Finding the observed information matrix when using the EM algorithm. Journal of Royal Statistical Society, Series B 44: 190–200.

Marroquin J.L., Santana E.A., and Botello S. 2003. Hidden Markov measure field models for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 25: 1380–1397.

McCullagh P. and Nelder J.A. 1989. Generalized Linear Models (2nd edn.). Chapman and Hall, London.

Møller J. 1999. Markov chain Monte Carlo and spatial point processes. In W.S. Kendall, O.E. Barndorff-Nielsen, and M.C. van Lieshout (Eds.), Stochastic Geometry: Likelihood and Computation, Chapman and Hall, London.

Moyeed R.A. and Baddeley A.J. 1991. Stochastic approximation of the maximum likelihood estimate for a spatial point pattern. Scandinavian Journal of Statistics 18: 39–50.

Ortega J.M. 1990. Numerical Analysis: A Second Course. Society for Industrial and Academic Press, Philadelphia.

Penttinen A. 1984. Modelling interaction in spatial point patterns: parameter estimation by the maximum likelihood method. Jy. Stud. Comput. Sci. Econometr. Statist. 7.

Pettitt A.N., Friel N., and Reeves R. 2003. Efficient calculation of the normalisation constant of the autologistic model on the lattice. Journal of Royal Statistical Society, Series B 65: 235–247.

Polyak B.T. 1990. New stochastic approximation type procedures. Autom. Telem. pp. 98–107. (English translation in Automat. Remote Contr. 51).

Polyak B.T. and Juditski A.B. 1992. Acceleration of stochastic approximation by averaging. SIAM Journal of Control and Optimization 30: 838–855.

Qian W. and Titterington D.M. 1991. Estimation of parameters in hidden Markov models. Philosophical Transactions of the Royal Society of London, Series A 337: 407–428.

Rajapakse J.C., Giedd J.N., and Rapoport J.L. 1997. Statistical approach to segmentation of single-channel cerebral MR images. IEEE Transactions on Medical Imaging 16: 176–186.

Robbins H. and Monro S. 1951. A stochastic approximation method. Annals of Mathematical Statistics 22: 400–407.

Robert C.P. and Casella G. 1999. Monte Carlo Statistical Methods. Springer-Verlag, New York.

Rydén T. 1997. On recursive estimation for hidden Markov models. Stochastic Processes and their Applications 66: 79–96.

Saquib S.S., Bouman C.A., and Sauer K. 1998. ML parameter estimation for Markov random fields with applications to Bayesian tomography. Transactions on Image Processing 7: 1029–1044.

Stoer J. and Bulisch R. 1980. Introduction to Numerical Analysis. Springer-Verlag, New York.

Swendsen R.H. and Wang J.S. 1987. Nonuniversal critical dynamics in Monte Carlo simulation. Physics Review Letters 58: 86–88.

Wang N., Lin X., Gutierrez R.G., and Carroll R.J. 1998. Bias analysis and SIMEX approach in generalized linear mixed measurement error models. Journal of American Statistical Association 93: 249–261.

Wei G.C.G. and Tanner M.A. 1990. A Monte Carlo implementation of the EM algorithm and the Poor man's data augmentation algorithm. Journal of American Statistical Association 85: 699–704.

Winkler G. 1995. Image Analysis, Random Fields and Dynamic Monte Carlo Methods: A Mathematical Introduction. Springer-Verlag, Berlin Heidelberg.

Younes L. 1989. Parameter estimation for imperfectly observed Gibbsian fields. Probability Theory Related Feilds 82: 625–645.

Zeger S.L. 1988. A regression model for time series of counts. Biometrika 75: 621–629.

Zeger S.L., Liang K.Y., and Albert P.S. 1988. Models for longitudinal data: a generalized estimating equation approach. Biometrics 44: 1049–1060.

Zhang H. 2002. On estimation and prediction for spatial generalized linear mixed models. Biometrics 56: 129–136.

Zhang Y., Brady M., and Smith S. 2001. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Transactions on Medical Imaging 15: 45–57.

Zhu H.T. and Gu M.G. 2005. Maximum likelihood from spatial random effects models via the stochastic approximation expectation maximization algorithm (supplement). Technical report, Department of Biostatistics, University of North Carolina at Chapel Hill.

Zhu H.T. and Lee S.Y. 2002. Analysis of generalized linear mixed models via a stochastic approximation algorithm with Markov chain Monte Carlo method. Statistics and Computing 12: 175–183.

Zhu H.T. and Zhang H.P. 2004. Hypothesis testing in a class of mixture regression models. Journal of Royal Statistical Society, Series B: 66: 3–16.