J. R. Statist. Soc. B (2001)
63, *Part* 2, *pp.* 339–355

Maximum likelihood estimation for spatial models by Markov chain Monte Carlo stochastic approximation

Ming Gao Gu

Chinese University of Hong Kong, People's Republic of China

and Hong-Tu Zhu

University of Victoria, Canada

[Received April 2000. Revised October 2000]

Summary. We propose a two-stage algorithm for computing maximum likelihood estimates for a class of spatial models. The algorithm combines Markov chain Monte Carlo methods such as the Metropolis–Hastings–Green algorithm and the Gibbs sampler, and stochastic approximation methods such as the off-line average and adaptive search direction. A new criterion is built into the algorithm so stopping is automatic once the desired precision has been set. Simulation studies and applications to some real data sets have been conducted with three spatial models. We compared the algorithm proposed with a direct application of the classical Robbins–Monro algorithm using Wiebe's wheat data and found that our procedure is at least 15 times faster.

Keywords: Auto-normal model; Ising model; Markov chain Monte Carlo methods; Off-line average; Spatial models; Stochastic approximation; Very-soft-core model

1. Introduction

Recently, there has been an increasing interest in modelling spatial data with interactions between points. Those include Strauss-type hard-core models (Strauss, 1975; Kelly and Ripley, 1976), inhomogeneous spatial Poisson processes (Baddeley and Turner, 2000), spatial lattice models (Besag, 1974; Strauss, 1977) and some pairwise interaction models (Besag, 1974; Ripley, 1977; Diggle *et al.*, 1994). Spatial statistical models consist of three seemingly distinct parts: problems with a continuous spatial index, problems with a lattice index and spatial point patterns. For a general introduction to statistical methodology for spatial models, see Ripley (1981), Diggle (1983), Stoyan *et al.* (1987), Cressie (1993) and Barndorff-Nielsen *et al.* (1999).

Owing to the intractable likelihood function, maximum likelihood estimation is rarely used for spatial models. A notable exception was Huffer and Wu (1998), where the Monte Carlo method of Geyer and Thompson (1992) was used; see also Geyer (1999). A direct computation of the maximum likelihood estimates by numerical approximations for some pairwise interaction models was developed by Ogata and Tanemura (1984). Other simulation-based methods include Monte Carlo Newton–Raphson sampling (Penttinen, 1984) and stochastic approximation (Younes, 1988, 1989; Moyeed and Baddeley, 1991). Owing to the difficulties

Address for correspondence: Ming Gao Gu, Department of Statistics, Chinese University of Hong Kong, Shatin, Hong Kong, People's Republic of China.
E-mail: minggao@cuhk.edu.hk

1369–7412/01/63339

that are encountered in directly computing the maximum likelihood estimate, the maximum pseudolikelihood estimator for spatial models was proposed as an alternative to maximum likelihood estimation (Besag, 1977; Goulard *et al.*, 1996; Baddeley and Turner, 2000). However, the maximum pseudolikelihood estimator is inefficient compared with the maximum likelihood estimate (Guyon, 1982; Pickard, 1982; Jensen and Møller, 1991; Comets, 1992; Mase, 1995). More recently, Huang and Ogata (1999) considered an approximate likelihood approach which is a combination of an initial maximum pseudolikelihood estimator and a one-step Monte Carlo Newton–Raphson method.

In this paper, we consider computing the maximum likelihood estimates of spatial models via an improved Markov chain Monte Carlo stochastic approximation algorithm. Younes (1988) first proposed the use of Markov chain Monte Carlo stochastic approximation for spatial statistical models. Moyeed and Baddeley (1991) have applied the Robbins and Monro (1951) type of algorithms to maximum likelihood estimation of the Strauss hard-core model. However, the use of such algorithms for spatial models is hampered by at least four problems: the convergence of the algorithm is slow unless a 'good' starting-point is used, one cannot estimate the information matrix, no practical stopping criterion was available and there was no central limit theorem for the convergence of the algorithm. However, stochastic approximation was recommended as a method to obtain a starting-point for the Monte Carlo likelihood method (Younes, 1989; Geyer, 1999).

Recent developments in stochastic approximation (Kushner and Yin, 1997) and in its application to general missing data problems (Gu and Kong, 1998; Delyon *et al.*, 1999) shed new light on how to apply it to spatial models. The major idea of Gu and Kong (1998) is that stochastic approximation can be used to compute the maximum likelihood estimate and the information matrix simultaneously, and the estimated information matrix at each iteration can be used to update the estimated maximum likelihood estimate; hence the optimal rate of convergence is achieved. In Delyon *et al.* (1999), two recent breakthroughs in stochastic approximation were introduced: the method of dynamic bounds (Chen *et al.*, 1988) and the method of off-line averaging (Polyak, 1990; Polyak and Juditski, 1992). The dynamic bounds method greatly reduces the conditions on the growth of the function for convergence. The off-line average method automatically gives the optimal rate of convergence without estimating the information matrix.

In this paper, we propose an algorithm which combines Markov chain Monte Carlo methods and these stochastic approximation type of methods. Moreover, inspired by recent developments in constant gain algorithms for time-varying dynamic systems, we propose a two-stage Markov chain Monte Carlo stochastic approximation algorithm. In stage I, a sequence of large gain constants is used to put our estimates quickly into the feasible region. In stage II, an optimal stochastic approximation procedure is carried out. A stopping criterion which depends on the desired precision of the estimate is built into this stage so stopping the program becomes automatic. Our algorithm has been successfully applied to three examples. At least for moderate sizes of data, our algorithm bears none of the unpleasant features which mark earlier applications of stochastic approximations to these models.

The paper is organized as follows. Section 2 introduces the spatial models and presents our Markov chain Monte Carlo stochastic approximation algorithm. A new stopping criterion is introduced in Section 3. Three spatial models are considered in Sections 4–6, where some simulation studies and real examples are analysed to illustrate our methodology. A comparison with the classical stochastic approximation is given in Section 7. A discussion is given in Section 8. The programs that were used in the analysis can be obtained from

`http://www.blackwellpublishers.co.uk/rss/`

2. The spatial models and the Markov chain Monte Carlo stochastic approximation algorithm

2.1. The spatial models

Assume that we have a pattern of points $\mathbf{X} = \{x_i \in A: i = 1, \ldots, n\}$ in a region $A \subset R^d$, where R^d is a d-dimensional Euclidean space. By *spatial model* we mean a statistical model for $\mathbf{X}$ with density

$$f(\mathbf{x}|\theta) = \exp[-Q(\mathbf{x};\, \theta) - \log\{C(\theta)\}], \tag{1}$$

where θ is a p-dimensional parameter vector, the potential function $Q(\cdot;\, \cdot)$ exhibits the interaction between components of $\mathbf{X}$ and the normalizing factor is

$$C(\theta) = \int_{A^n} \exp\{-Q(\mathbf{y};\, \theta)\}\, \mu(\mathrm{d}\mathbf{y}), \tag{2}$$

where $\mu(\mathrm{d}\mathbf{y})$ is either Dirac's delta measure $\delta_{\mathbf{y}}(\mathrm{d}\mathbf{y})$ or $\mathrm{d}\mathbf{y}$, according to whether $\mathbf{y}$ takes discrete or continuous values. To define the likelihood it is assumed that the *admissibility condition* $C(\theta) < \infty$ holds for a set of parameters. Thus, the log-likelihood of θ for the observation $\mathbf{X} = \mathbf{x}_0$ is

$$l(\theta;\, \mathbf{x}_0) = \log\{f(\mathbf{x}_0|\theta)\} = -Q(\mathbf{x}_0;\, \theta) - \log\{C(\theta)\}. \tag{3}$$

For simplicity of notation, we shall omit $\mathbf{x}_0$ in $l(\theta;\, \mathbf{x}_0)$ and $Q(\mathbf{x}_0;\, \theta)$.

Most parametric spatial models can be described by equation (1). The function $C(\theta)$ is also called the *partition function* in these models. As the integration in equation (2) is usually of very high dimension, the partition function generally admits no simple form.

2.2. The Markov chain Monte Carlo stochastic approximation algorithm

We wish to find the value $\hat{\theta} \in \Theta \subset R^p$ that maximizes $l(\theta)$, called the *maximum likelihood estimate*. Throughout the paper, we shall assume that the function $l(\theta)$ has a unique mode and that the maximum likelihood estimate always exists and is unique. Owing to the intractability of the partition function, a direct maximization of $l(\theta)$ is numerically infeasible.

For a smooth $l(\theta)$, its first and second derivatives are respectively

$$\begin{aligned} \nabla l(\theta) &= -\nabla Q(\theta) - \nabla[\log\{C(\theta)\}], \\ \nabla^2 l(\theta) &= -\nabla^2 Q(\theta) - \nabla^2[\log\{C(\theta)\}], \end{aligned} \tag{4}$$

where ∇ and ∇^2 are the first- and second-derivative operators with respect to θ. Thus, if we can calculate $\nabla l(\theta)$ and $\nabla^2 l(\theta)$ at each θ, we can expect to obtain the maximum likelihood estimate by the Newton–Raphson method. From equations (4), we need to calculate $\nabla[\log\{C(\theta)\}]$ and $\nabla^2[\log\{C(\theta)\}]$.

Using the identities $E_\theta[\nabla l(\theta;\, \mathbf{X})] = \mathbf{0}$ and $E_\theta[\nabla^2 l(\theta;\, \mathbf{X})] = -E_\theta[\nabla l(\theta;\, \mathbf{X})^{\otimes 2}]$, where E_θ denotes expectation with respect to density (1), we can show that

$$\begin{aligned} \nabla[\log\{C(\theta)\}] &= -E_\theta[\nabla Q(\mathbf{X};\, \theta)], \\ \nabla^2[\log\{C(\theta)\}] &= -E_\theta[\nabla^2 Q(\mathbf{X};\, \theta)] + E_\theta[\nabla Q(\mathbf{X};\, \theta)]^{\otimes 2} - \{E_\theta[\nabla Q(\mathbf{X};\, \theta)]\}^{\otimes 2}, \end{aligned} \tag{5}$$

where, for vector $\mathbf{a}$, $\mathbf{a}^{\otimes 2} = \mathbf{a}\mathbf{a}^{\mathrm{T}}$. One way to calculate $\nabla[\log\{C(\theta)\}]$ and $\nabla^2[\log\{C(\theta)\}]$ is to use numerical integration in equations (5). However, the numerical approximation is accurate only for some special cases, and it usually gives unstable estimates. Another way is to resort to Monte Carlo integration. So in principle we may obtain $\nabla[\log\{C(\theta)\}]$ and $\nabla^2[\log\{C(\theta)\}]$ by using the Monte Carlo approximation, if we can simulate $\{\mathbf{X}_\theta(t): t = 1, \ldots, T\}$ from model

(1). In the case of spatial models, random samples can be generated by Markov chain Monte Carlo methods, e.g. the Gibbs sampler, the Metropolis–Hastings algorithm, birth-and-death processes or the Metropolis–Hasting–Green algorithm; see Besag and Green (1993), Besag *et al.* (1995), Geyer (1999), Møller (1999), Robert and Casella (1999) and the references therein.

It should be noted that we introduce 'noise' in approximating $\nabla[\log\{C(\theta)\}]$ and $\nabla^2[\log\{C(\theta)\}]$ at each θ. The question is how closely one should approximate these two functions and how to handle the noise. The stochastic approximation algorithm, first proposed by Robbins and Monro (1951), provides a method of handling such noise and can be employed to find the maximum likelihood estimates of some spatial models. Early work in this area can be traced back to Younes (1988, 1989) and Moyeed and Baddeley (1991). However, as reported in Moyeed and Baddeley (1991), the results from a direct implementation of a Robbins–Monro algorithm to a single-parameter Strauss model were not satisfactory. See also Geyer (1999).

Using the fact that in most likelihood problems the information matrix can also be approximated by simulations, Gu and Kong (1998) proposed a Markov chain Monte Carlo stochastic approximation algorithm which uses the approximated information matrix in updating the next estimate. Although this algorithm improves on the performance of the classical Robbins–Monro algorithm, the direct application of this algorithm to spatial models is not satisfactory, as we found from simulation studies not reported in this paper with the models described in Sections 4–6. The problem is that the dimension of $\mathbf{X}$ is so large that the convergence of the algorithm is usually very slow, especially if the initial value is not close to the maximum likelihood estimate.

Another significant development in stochastic approximation is due to Polyak (1990) (see also Polyak and Juditski (1992) and Delyon *et al.* (1999)), who showed that, if we run an ordinary Robbins–Monro algorithm with a bigger gain constant sequence ($\gamma_k = k^{-\alpha}$, $\frac{1}{2} < \alpha < 1$, while $\gamma_k = k^{-1}$ is considered to be optimal) and offset the oscillation by off-line averaging, then the optimal rate of convergence is obtained by the averaged sequence. Again, simulation shows that the direct application of this idea to the spatial models discussed in Sections 4–6 is unsatisfactory. The problem is again the slowness of convergence to the maximum likelihood estimate if the initial value is not very close. Theoretically, from the proof of theorem 4 of Delyon *et al.* (1999) and chapter 11 of Kushner and Yin (1997), we see that the optimal rate of convergence is achieved only when the estimate is sufficiently close to the maximum likelihood estimate.

If the initial value is far from the maximum likelihood estimate, a large gain constant sequence can be used at first to force the estimates into a small neighbourhood of the maximum likelihood estimate. This idea of using larger gain constants when the current estimate is still far from the target can be traced back to Kesten (1957). Our proposed algorithm is also inspired by the development of constant gain algorithms for time-varying dynamic systems (Kushner and Yin, 1997; Duflo, 1997). Once the current estimate is close to the maximum likelihood estimate, then an optimal procedure such as the off-line average method can be used effectively.

Our algorithm has two stages of stochastic approximation. In stage I, we use a larger gain constant sequence, and in stage II we use the off-line average method of Polyak and Juditski (1992). In both stages, the estimated search direction method of Gu and Kong (1998) is employed.

We first introduce two basic steps in the stochastic approximation specialized to our model and notation. Let us keep in mind that θ^k is the current estimate of $\hat{\theta}$, $\mathbf{h}_k$ is the current estimate of $E_{\hat{\theta}}[\nabla Q(\mathbf{X};\, \hat{\theta})]$ and Γ_k is the current estimate of

$$-E_{\hat{\theta}}[\nabla^2 Q(\mathbf{X};\, \hat{\theta})] + E_{\hat{\theta}}[\nabla Q(\mathbf{X};\, \hat{\theta})]^{\otimes 2}.$$

We also assume that, for each θ, there is a Markov transition probability density $\Pi_\theta(\cdot, \cdot)$ such that the chain driven by this transition probability is aperiodic and irreducible with stationary distribution $f(\mathbf{x}|\theta)$.

Step 1: at the kth iteration, set $\mathbf{X}_{k,0} = \mathbf{X}_{k-1,m}$. For $i = 1, \ldots, m$, generate $\mathbf{X}_{k,i}$ from the transition probability density $\Pi_{\theta^{k-1}}(\mathbf{X}_{k,i-1}, \cdot)$.
Step 2: update θ^{k-1} to θ^k, $\mathbf{h}_{k-1}$ to $\mathbf{h}_k$ and Γ_{k-1} to Γ_k by

$$\left.\begin{aligned} \mathbf{h}_k &= \mathbf{h}_{k-1} + \gamma_k\{\bar{H}(\theta^{k-1}; \mathbf{X}_k) - \mathbf{h}_{k-1}\}, \\ \Gamma_k &= \Gamma_{k-1} + \gamma_k\{\bar{I}(\theta^{k-1}; \mathbf{X}_k) - \Gamma_{k-1}\}, \\ \theta^k &= \theta^{k-1} + \gamma_k\{\nabla^2 Q(\theta^{k-1}) + \Gamma_{k-1} - \mathbf{h}_{k-1}^{\otimes 2}\}^{-1}\{-\nabla Q(\theta^{k-1}) + \bar{H}(\theta^{k-1}; \mathbf{X}_k)\}, \end{aligned}\right\} \tag{6}$$

where $\mathbf{X}_k = (\mathbf{X}_{k,1}, \ldots, \mathbf{X}_{k,m})$,

$$\bar{H}(\theta^{k-1}; \mathbf{X}_k) = \frac{1}{m}\sum_{i=1}^{m} \nabla Q(\mathbf{X}_{k,i}; \theta^{k-1})$$

and

$$\bar{I}(\theta^{k-1}; \mathbf{X}_k) = -\frac{1}{m}\sum_{i=1}^{m} \nabla^2 Q(\mathbf{X}_{k,i}; \theta^{k-1}) + \frac{1}{m}\sum_{i=1}^{m} \{\nabla Q(\mathbf{X}_{k,i}; \theta^{k-1})\}^{\otimes 2}.$$

Stage I of the algorithm consists of choosing an initial point θ^0, an initial matrix Γ_0, an initial vector $\mathbf{h}_0$ and an initial spatial configuration $\mathbf{X}_{0,m}$ and of setting $k = 1$ followed by iterating steps 1 and 2 with $k = 1, \ldots, K_1$. The gain constants are defined by

$$\gamma_k = \gamma_{1k} = b_1/(k^{a_1} + b_1 - 1), \qquad k = 1, \ldots, K_1,$$

where $K_1 \geqslant K_0$ is determined by

$$K_1 = \inf\left\{K \geqslant K_0 : \left\| \sum_{k=K-K_0+1}^{K} \operatorname{sgn}(\theta^k - \theta^{k-1})/K_0 \right\| \leqslant \eta_1 \right\}, \tag{7}$$

where $\operatorname{sgn}(\theta)$ is a vector of 1, 0 or -1 according to whether the component of θ is positive, zero or negative respectively. Integers b_1, K_0 and real number $a_1 \in (0, \frac{1}{2})$ and η_1 are preassigned.

Stage II starts when stage I finishes and takes the final values of θ, $\mathbf{h}$, Γ and $\mathbf{X}$ of stage I as its initial values. The algorithm iterates steps 1 and 2 with $k = 1, \ldots, K_2$. The gain constants are defined by

$$\gamma_k = \gamma_{2k} = b_2/(k^{a_2} + b_2 - 1), \qquad k = 1, \ldots, K_2,$$

where the integer b_2 and the real number $a_2 \in (\frac{1}{2}, 1)$ are preassigned, and K_2 is defined in Section 3. At the same time, an averaging procedure is used,

$$\left.\begin{aligned} \tilde{\theta}^k &= \tilde{\theta}^{k-1} + (\theta^k - \tilde{\theta}^{k-1})/k, \\ \tilde{\mathbf{h}}_k &= \tilde{\mathbf{h}}_{k-1} + (\mathbf{h}_k - \tilde{\mathbf{h}}_{k-1})/k, \\ \tilde{\Gamma}_k &= \tilde{\Gamma}_{k-1} + (\Gamma_k - \tilde{\Gamma}_{k-1})/k, \end{aligned}\right\} \tag{8}$$

assuming $\tilde{\theta}^0 = 0$. After the K_2th iteration, we use the off-line average $(\tilde{\theta}^{K_2}, \tilde{\mathbf{h}}_{K_2}, \tilde{\Gamma}_{K_2})$ as our final estimate of $(\hat{\theta}, -\nabla[\log\{C(\hat{\theta})\}], \nabla^2[\log\{C(\hat{\theta})\}] + (\nabla[\log\{C(\hat{\theta})\}])^{\otimes 2})$. This is equivalent to averaging all the values (of stage II) up to K_2.

To ensure that the gain constant in stage I is large, we usually choose a_1 to be close to 0, b_1

to be relatively large and η_1 to be relatively small. For example, we may take $a_1 = 0.3$, $b_1 = 10$ and $\eta_1 = 0.1$. With these choices, the algorithm proposed will typically move quickly towards the feasible region. In stage II we use a_2 close to $\frac{1}{2}$, a small integer for b_2, say, $a_2 = 0.6$ and $b_2 = 1$. Coupling with the off-line averaging procedure (8), the algorithm will stabilize in the neighbourhood of the maximum likelihood estimate.

The existence of a finite value of K_1 in stage I can be argued from the point of view of the constant gain stochastic approximation algorithm. If $\gamma_{1k} = \gamma$ is a sufficiently small constant, then $(\mathbf{h}_k, \Gamma_k, \theta^k)$, $k = 1, 2, \ldots$ forms a recurrent Markov chain (theorem 8.1.5 of Duflo (1997)). In our case, since $\gamma_{1k} \to 0$, as $k \to \infty$, the recurrence is guaranteed.

A set of sufficient conditions to ensure root square convergence for $\tilde{\theta}^k$, $\tilde{\mathbf{h}}_k$ and $\tilde{\Gamma}_k$ in stage II was given in chapters 10 and 11 of Kushner and Yin (1997); see also Delyon *et al.* (1999). To be more specific, we have, under general conditions (theorem 10.8.1 and theorem 11.1.1 of Kushner and Yin (1997) or theorem 4 of Delyon *et al.* (1999)), as $k \to \infty$,

$$(\tilde{\theta}^k - \hat{\theta})\sqrt{k} \to \mathcal{N}[0, \{-\nabla^2 l(\hat{\theta})\}^{-1} \Sigma \{-\nabla^2 l(\hat{\theta})\}^{-1}], \tag{9}$$

where Σ is the covariance matrix in the central limit theorem

$$\frac{1}{\sqrt{k}} \sum_{j=1}^{k} (\bar{H}(\theta^{j-1}; \mathbf{X}_j) + \nabla[\log\{C(\theta^{j-1})\}]) \to \mathcal{N}(0, \Sigma), \tag{10}$$

as $k \to \infty$.

The choice of m should not affect the convergence of the procedure proposed. In general, a large m reduces the covariance Σ and the correlation between $\bar{H}(\theta^{k-1}; \mathbf{X}_k)$ and $\bar{H}(\theta^k; \mathbf{X}_{k+1})$. Therefore, a large m reduces the number of iterations that are required by the Markov chain Monte Carlo stochastic approximation algorithm to achieve convergence.

3. A stopping criterion

A standard stopping criterion used for the stochastic approximation procedure is to stop when the relative change in the parameter values from successive iterations is small. There are many problems with this since there is always a chance that the change in θ is small but the current estimate is still not close to the maximum likelihood estimate.

An important identity is that $\nabla l(\theta) = 0$ at the maximum likelihood estimate $\hat{\theta}$. It is natural to consider a stopping criterion which is based on small values of $\nabla l(\tilde{\theta}^k)$. At iteration k, define

$$\Delta_k = (\nabla l(\tilde{\theta}^k))^{\mathrm{T}} \{-\nabla^2 l(\hat{\theta})\}^{-1} (\nabla l(\tilde{\theta}^k)).$$

Ignoring high order terms, simple algebra shows that Δ_k is asymptotically equivalent to $(\tilde{\theta}^k - \hat{\theta})^{\mathrm{T}} \{-\nabla^2 l(\hat{\theta})\} (\tilde{\theta}^k - \hat{\theta})$, which is not affected by the scale since $\hat{\theta}$ has asymptotic variance $\{-\nabla^2 l(\hat{\theta})\}^{-1}$. We consider choosing K_2, the number of iterations of stage II, such that Δ_{K_2} is small. An estimate of $\nabla l(\tilde{\theta}^k)$ is $\nabla \tilde{l}(\theta^k) = -\nabla Q(\tilde{\theta}^k) + \tilde{\mathbf{h}}_k$ and an estimate of $-\nabla^2 l(\hat{\theta})$ is $-\nabla^2 l(\tilde{\theta}^{k-1}) = \nabla^2 Q(\tilde{\theta}^{k-1}) + \tilde{\Gamma}_{k-1} - \tilde{\mathbf{h}}_{k-1}^{\otimes 2}$. However, if we just use the natural estimate

$$\nabla \tilde{l}(\theta^k)^{\mathrm{T}} \{-\nabla^2 \tilde{l}(\theta^{k-1})\}^{-1} \nabla \tilde{l}(\theta^k)$$

of Δ_k as our criterion for convergence, then we are ignoring a possibly large Monte Carlo error.

To control the Monte Carlo estimation error, we argue as follows. Expressions (9) and (10) assert that $\nabla\tilde{l}(\theta^k)\sqrt{k}$ is asymptotically distributed as $\mathcal{N}(0, \Sigma)$. Therefore, the variance of $\nabla\tilde{l}(\theta^k)^{\mathrm{T}}\{-\nabla^2\tilde{l}(\theta^{k-1})\}^{-1}\nabla\tilde{l}(\theta^k)$ is asymptotically

$$2\,\mathrm{tr}[\{-\nabla^2\tilde{l}(\theta^{k-1})\}^{-1}\Sigma]^2/k^2,$$

where $\mathrm{tr}(A)$ denotes the trace of matrix A. See, for example, corollary 1.3 of section 2.5 of Searle (1971).

In practice let $\hat{\Sigma}$ denote an estimate of Σ. Then a convergence criterion can be based on

$$\hat{\Delta}_k = \nabla\tilde{l}(\theta^k)^{\mathrm{T}}\{-\nabla^2\tilde{l}(\theta^{k-1})\}^{-1}\nabla\tilde{l}(\theta^k) + \mathrm{tr}[\{-\nabla^2\tilde{l}(\theta^{k-1})\}^{-1}\hat{\Sigma}]/k. \tag{11}$$

Therefore, we define

$$K_2 = \inf(k\colon \hat{\Delta}_k \leqslant \eta_2),$$

where η_2 (usually taken to be around 0.001) is a preassigned small number.

An estimation of Σ can be performed in the following way. If m is large, we may expect the correlations between consecutive $\bar{H}(\theta^{j-1}; \mathbf{X}_j)$s to be small. So a natural estimate of Σ can be constructed by using the sample covariance of those values. A more precise estimate of Σ can certainly be used if we treat $\{\bar{H}(\theta^{j-1}; \mathbf{X}_j), j = 1, \ldots, k\}$ as a realization of a time series (Geyer, 1999). As we are dealing with the average of m values and this estimate is only used in the computation of the maximum likelihood estimate, a rough estimate should be enough to serve our purpose. Moreover, in each iteration, $\nabla^2 Q(\theta^{k-1}) + \Gamma_{k-1} - \mathbf{h}_{k-1}^{\otimes 2}$ can be used as a rough estimate of $-\nabla^2 l(\hat{\theta})$, which will save computation time, especially when the dimension of θ is large.

To illustrate the behaviour of the Markov chain Monte Carlo stochastic approximation algorithm proposed, two simulation studies and analyses of three real data sets in the literature will be discussed in Sections 4–6. All computations were done in the C language on a SUN hpc4500 workstation. In all the examples, the convergence criterion in equations (7) and (11) was used in stage I and II respectively and (K_0, η_1, η_2) was set to be (100, 0.1, 0.001).

4. Ising model

The Ising model is a discrete Markov random field model, placing a binary random variable $x(i, j)$ at each site (i, j) taking values in $\{-1, 1\}$ on a regular $M_0 \times N_0$ lattice $\mathcal{Z}^2_{M_0,N_0}$. Realizations $\mathbf{X} = \{x(i, j)\colon (i, j) \in \mathcal{Z}^2_{M_0,N_0}\}$ of the random field are configurations of pluses and minuses on $\mathcal{Z}^2_{M_0,N_0}$. The statistic that counts the excess of *like* over *unlike* nearest neighbour points on the lattice is defined as

$$V = V(\mathbf{X}) = \sum_{\mathrm{nn}} x(i, j)\, x(u, v),$$

where nn means that the summation is over all the pairs (i, j) and (u, v) such that the two points are nearest neighbours. The potential function is $Q(\mathbf{X}; \theta) = -\theta\, V(\mathbf{X})$ and the normalizing factor is obtained by summing over all possible configurations $\mathbf{X}$,

$$C(\theta) = \sum_{\mathbf{X}} \exp\{\theta\, V(\mathbf{X})\}.$$

In this model, $V(\mathbf{X})$ is the minimal sufficient statistic for θ. The sign of θ determines whether the Ising model is ferromagnetic or antiferromagnetic (attractive or repulsive). Let

$$m(\mathbf{X}) = \sum_{i=1}^{M_0}\sum_{j=1}^{N_0} x(i, j)$$

be the magnetic moment of configuration **X**. The spontaneous magnetization is defined by

$$\mathrm{SM}(\theta) = \sum_{\mathbf{X}} \frac{m(\mathbf{X}) \exp\{\theta\, V(\mathbf{X})\}}{M_0 N_0\, C(\theta)}.$$

When $|\theta|$ is smaller than the critical temperature near 0.44, $\mathrm{SM}(\theta)$ is zero (Brémaud, 1998).

To check the usefulness of the algorithm proposed, we performed a simulation study in which the Ising model was set on a 64×64 square lattice on the plane $\{x(l, j)\text{: } l, j = 1, \ldots, 64\}$. We assume a periodic boundary for the square lattice, which considers $\{x(i, 64), x(i, 1)\}$ and $\{x(64, j), x(1, j)\}$ as neighbours to each other. To simulate the process, the Metropolis algorithm with Gibbs dynamics (Müller, 1991) was used. Let the current value of the process at site (l, j) be $x(l, j)$ and the current total potential value be V. Take the alternative value $x(l, j)^* = -x(l, j)$ at the site (l, j), which leads to the potential value Q^*. Then, the Metropolis procedure at the present site (l, j) continues as follows:

(a) if $Q^* \leqslant Q$, replace $x(l, j)$ and Q by $x(l, j)^*$ and Q^* respectively;
(b) if $Q^* > Q$, generate a uniform(0, 1) random variable U and
  (i) if $U \leqslant \exp(Q - Q^*)$, set $x(l, j) = x(l, j)^*$ and $Q = Q^*$;
  (ii) otherwise, keep $x(l, j)$ and Q.

For each parameter value $\theta_0 \in \{0.00, \pm 0.20, \pm 0.40\}$, 500 data sets were simulated via the Metropolis algorithm as follows. Each site (l, j) was selected in lexicographical order. The initial state of the process was taken at random such that $x(i, j)$ is independently ± 1 with equal probability. The Metropolis algorithm was repeated at least 320×64^2 times (320 Monte Carlo steps). Then,

$$\mathrm{SM}_T(\theta) = \sum_{t=1}^{T} m(\mathbf{X}^t)/T$$

was used to assess the convergence of the Metropolis algorithm, where $\{\mathbf{X}^1, \ldots, \mathbf{X}^T\}$ $(T \geqslant 320)$ is the output from the Metropolis algorithm. When $|\mathrm{SM}_T(\theta)|$ is smaller than 0.001, we stopped the algorithm and declared that equilibrium had been achieved.

On the basis of the simulated data sets, we applied the Markov chain Monte Carlo stochastic approximation algorithm described in Section 2 to obtain the maximum likelihood estimate of the unknown parameter. The starting value of θ was taken to be 0.0 for all the true parameters θ_0. The two-stage Markov chain Monte Carlo stochastic approximation algorithm with $(a_1, b_1; a_2, b_2) = (0.3, 2; 0.8, 2)$ converged quickly. In each iteration, the same Metropolis algorithm was used to sample a random variable at each site (l, j); however, each site was selected at random with probability $1/(64 \times 64)$, not according to the lexicographical order. For example, if site (1, 1) was selected, we run the above-mentioned Metropolis procedure at the site (1, 1) with other sites unchanged. In other words, only the value at one site is possibly changed from $\mathbf{X}_{k,i-1}$ to $\mathbf{X}_{k,i}$. We set $m = 20000$. Compared with the total number of sites $64 \times 64 = 4096$, $m = 20000$ is not too large.

To illustrate the performance of the algorithm proposed, we calculated the bias, the mean of the standard deviation estimates and the root-mean-square error obtained from the 500 estimates. The mean of the number of iterations for each estimate and the average central processor unit (CPU) time for each estimate were also obtained. The results obtained are summarized in Table 1. It can be seen that the performance of the Markov chain Monte Carlo stochastic approximation algorithm proposed is almost perfect. All the relative efficiencies (the ratio of the mean of the standard deviation estimates and the root-mean-square error) are close to 1.0. For comparison, the maximum likelihood estimates obtained via the DALL optimization subroutine and the maximum pseudolikelihood estimates obtained based on 100 estimates, presented in Huang and Ogata (1999), are also included in Table 1. The DALL optimization program (Ishiguro and Akaike, 1989) is an implementation of Davidon's variance algorithm with a numerical derivative evaluation procedure. The performance of the Markov chain Monte Carlo stochastic approximation algorithm proposed is better since the efficiency coefficients are uniformly closer to 1 for the algorithm proposed than those for the DALL optimization subroutine and for the maximum pseudolikelihood estimate.

For an analysis of real data, we fitted the Ising model on a 125 × 12 rectangular lattice to transformed values of Wiebe's wheat data (Andrews and Herzberg, 1985). The value '1' denotes 'larger than or equal to the mean value' and '−1' denotes 'less than the mean value'. Fig. 1(a) depicts these data. An inspection reveals a strong degree of spatial correlation in the data. We also assume the periodic boundary condition. The two-stage Markov chain Monte Carlo stochastic approximation algorithm with $(a_1, b_1;\ a_2, b_2) = (0.3, 2;\ 0.8, 2)$ and $m = 5000$ was used to obtain the maximum likelihood estimate $\hat{\theta} = 0.372$ with an estimated standard deviation 0.012. The large value of $\hat{\theta}$ is consistent with the observation in Fig. 1(a). The algorithm was stopped at the 833rd iteration after 6 s. The same Metropolis algorithm as in the simulation was used. The likelihood function calculated via the Ogata–Tanemura method and the Onsager formula is presented in Fig. 1(b). The starting value for θ was set at -0.3, which is far from $\hat{\theta}$. Figs 1(c) and 1(d) show the convergence behaviour of θ^k, $\tilde{\theta}^k$ and $\hat{\Delta}_k$ at each iteration. Our algorithm is robust to the initial value of θ and can find the maximum likelihood estimate with high precision.

Table 1. Biases, standard deviations, root-mean-square errors and efficiency coefficients of the estimators of the Ising model†

Parameter	*Results for the following values of* θ_0*:*				
	−0.40	*−0.20*	*0.0*	*0.20*	*0.40*
Maximum likelihood estimates (stochastic approximation)					
Bias ($\times10^{-3}$)	−0.14	0.07	1.01	0.14	1.05
SD ($\times10^{-2}$)	0.68	1.00	1.10	1.00	0.67
RMS ($\times10^{-2}$)	0.68	1.04	1.15	0.92	0.70
EFF	1.00	0.96	0.96	1.08	0.98
AVEN	915	322	228	330	936
AVET (s)	30	11	7	12	31
Maximum likelihood estimates (DALL)					
Bias ($\times10^{-3}$)	0.94	−0.24	1.47	−0.60	−0.43
SD ($\times10^{-2}$)	0.71	0.96	1.08	1.11	0.68
RMS ($\times10^{-2}$)	0.78	0.89	1.03	1.35	0.70
EFF	0.91	1.08	1.05	0.82	0.97
Maximum pseudolikelihood estimates					
Bias ($\times10^{-3}$)	0.01	−0.57	−1.51	−0.11	−0.20
SD ($\times10^{-2}$)	1.22	1.10	1.08	1.19	1.21
RMS ($\times10^{-2}$)	3.93	1.35	1.04	1.70	3.89
EFF	0.31	0.82	1.04	0.70	0.31

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

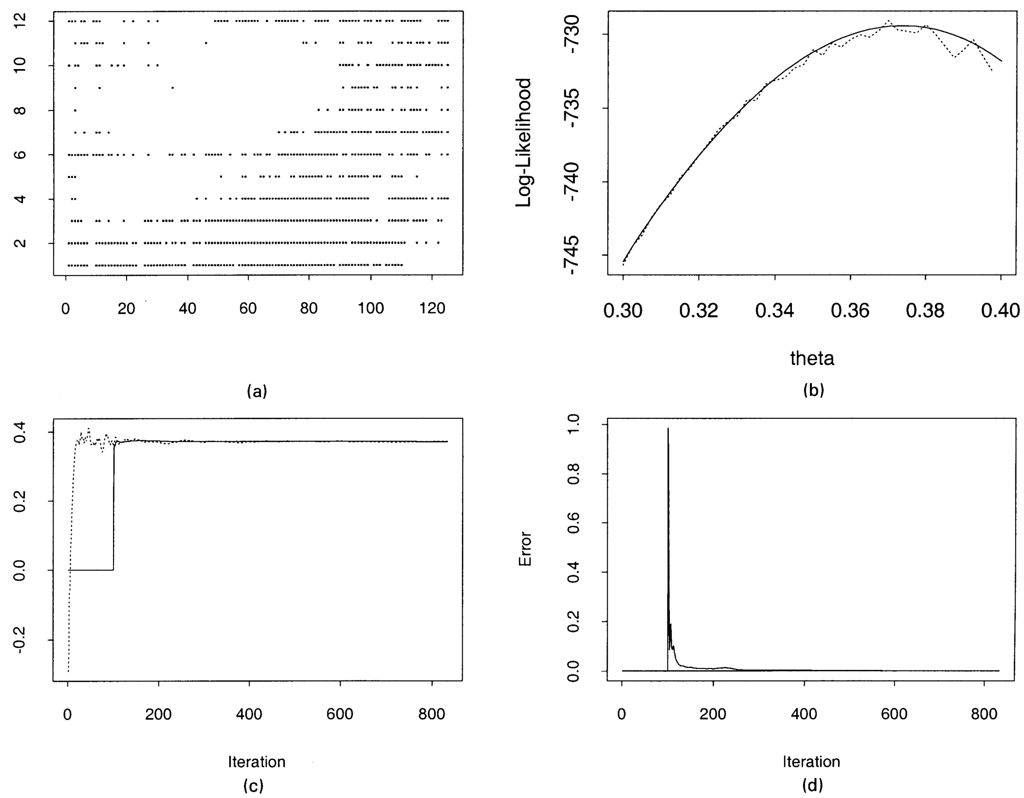


Fig. 1. Wiebe's wheat data: (a) transformed data on a 125 × 20 rectangular lattice (●, larger than or equal to the mean value; blank, less than the mean value); (b) log-likelihood function (———, calculated by using Onsager's formula; ··········, calculated by using Ogata and Tanemura's method); (c) θ^k (··········) and $\tilde{\theta}^k$ (———) at each iteration of the Markov chain Monte Carlo stochastic approximation algorithm; (d) $\hat{\Delta}_k$ at each iteration of the Markov chain Monte Carlo stochastic approximation algorithm

5. Autonormal model

Consider a Gaussian Markov random field $\mathbf{X} = \{x(i, j)\}$ on a lattice $\mathcal{Z}^2_{M_0,N_0}$, whose conditional probability at a site (i, j) in $\mathcal{Z}^2_{M_0,N_0}$ given the value $x(u, v)$ at all other sites is given by

$$f\{x(i, j)|x(u, v); (u, v) \neq (i, j)\} = (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{[x(i, j) - \mu_{i,j} - \sum_{(u,v)\neq(i,j)} \beta_{i,j;u,v}\{x(u, v) - \mu_{u,v}\}]^2}{2\sigma^2}\right), \tag{12}$$

where σ, $\mu_{i,j}$ and the $\beta_{i,j;u,v}$ are parameters, and the sum is taken for the neighbouring sites (u, v) of (i, j). The joint probability density of the process $\mathbf{X}$ (Besag, 1974) can be written as

$$f(\mathbf{X}|\mu, \sigma^2, B) = (2\pi\sigma^2)^{-M_0N_0/2}|B|^{1/2} \exp\left\{-\frac{(\mathbf{X}-\mu)^{\mathrm{T}}B(\mathbf{X}-\mu)}{2\sigma^2}\right\} \tag{13}$$

where $B = (-\beta_{i,j;u,v})$ with $-\beta_{i,j;i,j} = 1$ for $i = 1, \ldots, M_0$ and $j = 1, \ldots, N_0$. The maximum likelihood estimate for a general Markov random field model of form (12) is not easy to calculate because of the difficulty of evaluating the normalizing constant, since B is an

$(M_0N_0 \times M_0N_0)$-dimensional matrix (e.g. a 2048×2048 matrix corresponding to a small model on a 64×64 square lattice). If we assume a modulo boundary for the lattice process (as in the previous section), then $|B|$ admits a simpler form (Besag and Moran, 1975).

We conducted a simulation study of our proposed algorithm with the autonormal model. The Gaussian Markov process is set on a 64×64 square lattice on the plane $\{x(i, j)$: $i, j = 1, \ldots, 64\}$. To avoid edge effects, the periodic boundary for the square lattice is assumed. It is assumed that $\mu = 0$ and $\beta_{i,j;u,v} = \beta$ for the nearest neighbour sites of (i, j), and $\beta_{i,j;u,v} = 0$ for the other (u, v). In our simulations, β is set to 0.05, 0.11, 0.159 and 0.233, as only processes with $\beta < 0.25$ exist (Moran, 1973). The true value of $\sigma = \exp(\sigma^*)$ is set to 1.0, i.e. $\sigma^* = 0.0$. Thus, there are two parameters β and σ^* to be estimated.

The Gibbs algorithm as described in Huang and Ogata (1999) was used. Another approach to sample from Gaussian Markov random fields is given in Rue (2001). For each site (i, j), selected in lexicographical order, we generated a random variate $\epsilon(i, j)$ from $N(0, \sigma^2)$ and set $x(i, j) = \beta x(i, j)^* + \epsilon(i, j)$ where $x(i, j)^* = x(i-1, j) + x(i+1, j) + x(i, j-1) + x(i, j+1)$. To assess convergence of the Gibbs algorithm, we use the Gelman and Rubin (1992) method and choose to monitor sufficient statistics

$$s_1(\mathbf{X}) = \sum_{i=1}^{M_0} \sum_{j=1}^{N_0} \frac{x(i, j)^2}{M_0N_0},$$

$$s_2(\mathbf{X}) = \sum_{i=1}^{M_0} \sum_{j=1}^{N_0} \frac{x(i, j)\, x(i, j)^*}{M_0N_0}.$$

Starting from four quite different initial states of the process, four Gibbs algorithms were run. The Gibbs algorithm was repeated at least 320×64^2 (320 Monte Carlo steps) times. After that, we began to calculate Gelman and Rubin's (1992) statistic. As Gelman and Rubin's (1992) convergence criteria are close to 1, we stopped the Gibbs algorithm and declared that equilibrium had been achieved.

For each (β, σ^*), 500 data sets were simulated. On the basis of these, we applied the Markov chain Monte Carlo stochastic approximation algorithm proposed to obtain maximum likelihood estimates of the unknown parameters. The starting value of (β, σ^*) was taken to be (0.025, 0.0). In each iteration, we followed the above Gibbs sampler scheme except that each site (i, j) was selected at random with probability $1/(64 \times 64)$, and then updated $x(i, j) = \beta\, x(i, j)^* + \epsilon(i, j)$. We set $m = 10000$. To illustrate the performance of the algorithm proposed, we also calculated the bias, the mean of the standard deviation estimates and the root-mean-square error based on the 500 estimates, as in the Ising model case. The results obtained are given in Table 2, which also includes the mean number of iterations and the average CPU time for each estimate. The performance of the stochastic approximation algorithm is almost perfect. The ratios of the mean of the standard deviation estimates and the root-mean-square error are all around 1.0 even for the strong interaction case ($\beta_0 = 0.233$). For comparison, the maximum likelihood estimates of β obtained via the DALL optimization subroutine and the maximum pseudolikelihood estimates obtained based on 100 estimates, presented in Huang and Ogata (1999), are included in Table 2.

Fig. 2(a) depicts the original Mercer and Hall wheat yield data on a 20×25 rectangular lattice (Andrews and Herzberg, 1985), which was also analysed in Besag (1974) and Huang and Ogata (1999). We have fitted the first-order autonormal model and subtracted the mean from the data, which is equivalent to adding a shift parameter to the autonormal model. Under the periodic boundary assumption, we used the Markov chain Monte Carlo stochastic approximation algorithm with $(a_1, b_1; a_2, b_2) = (0.3, 2; 0.8, 2)$ and $m = 2000$ to find the

Table 2. Biases, root-mean-square errors, standard deviations and efficiency coefficients of the maximum likelihood estimators of the autonormal model†

True β	*Results for* β				*Results for* $\sigma^* = 0.0$				*AVEN*	*AVET (s)*
	Bias $(\times 10^{-3})$	*RMS* $(\times 10^{-2})$	*SD* $(\times 10^{-2})$	*EFF*	*Bias* $(\times 10^{-3})$	*RMS* $(\times 10^{-2})$	*SD* $(\times 10^{-2})$	*EFF*		
Maximum likelihood estimates (stochastic approximation)										
0.050	−0.560	1.07	1.08	0.99	−0.61	1.07	1.11	1.04	985	179
0.110	−0.40	0.96	0.95	0.99	0.96	1.07	1.14	1.07	1042	187
0.159	0.19	0.77	0.77	1.00	0.60	1.14	1.15	1.01	1138	206
0.233	−0.09	0.29	0.29	1.00	0.84	1.17	1.16	1.00	1688	306
Maximum likelihood estimates (DALL)										
0.049	1.14		1.16	0.89						
0.110	−0.01		1.00	0.95						
0.159	0.38		0.81	0.95						
0.233	0.01		0.29	1.10						
Maximum pseudolikelihood estimates										
0.049	1.11		1.16	0.88						
0.110	0.54		1.06	0.85						
0.159	0.62		0.86	0.85						
0.233	0.16		0.42	0.53						

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

maximum likelihood estimates. It took 1504 iterations and 63 s of CPU time to find the maximum likelihood estimates $(\hat{\beta}, \hat{\sigma}^*) = (0.237, -1.025)$ with standard errors (0.007, 0.034). Starting from $(\beta^0, \sigma^0) = (0.0, 0.0)$, the estimates $(\beta^k, \tilde{\beta}^k)$ and $(\sigma^{*k}, \tilde{\sigma}^{*k})$ at each iteration are shown in Figs 2(b) and 2(c). A large gain constants sequence in stage I effectively forced the estimates to a small neighbourhood of $(\hat{\beta}, \hat{\sigma}^*)$.

6. Very-soft-core model

Spatial point pattern data are described by the co-ordinates of points $\mathbf{X} = \{x_i \in A\colon i = 1, \ldots, n\}$ in a planar region A. The joint density of a pattern $\mathbf{X}$ of a pairwise interaction point process is given by equation (1) with $\theta = \tau$ and the potential function

$$Q(\mathbf{X}; \tau) = \sum_{i=1}^{n} \sum_{j>i} \phi(\|x_i - x_j\|; \tau)$$

where $\phi(\cdot; \tau)$ is a pairwise potential function. The normalizing constant is

$$C(\tau) = \int_{A^n} \exp\left\{ -\sum_{i=1}^{n} \sum_{j>i} \phi(\|x_i - x_j\|; \tau) \right\} \mathrm{d}x_1 \ldots \mathrm{d}x_n,$$

which cannot be computed analytically in general. For example,

$$\phi(t; \tau) = -\log\{1 - \exp(-t^2 \rho/\tau)\}$$

is called the very-soft-core potential function where $\rho = n/|A|$ and $|A|$ denotes the area of A. The function $\phi(t; \tau)$ increases from 0 to 1 when t increases from 0 to ∞. The analysis is performed conditionally on the observed number of points.

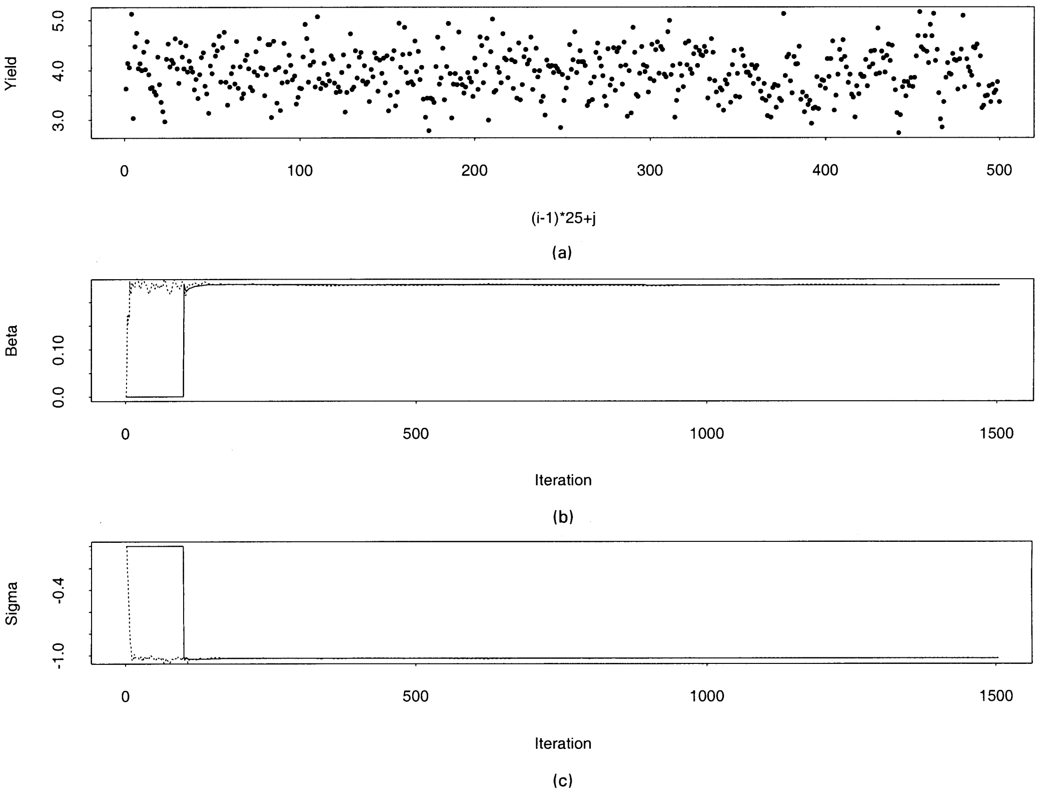
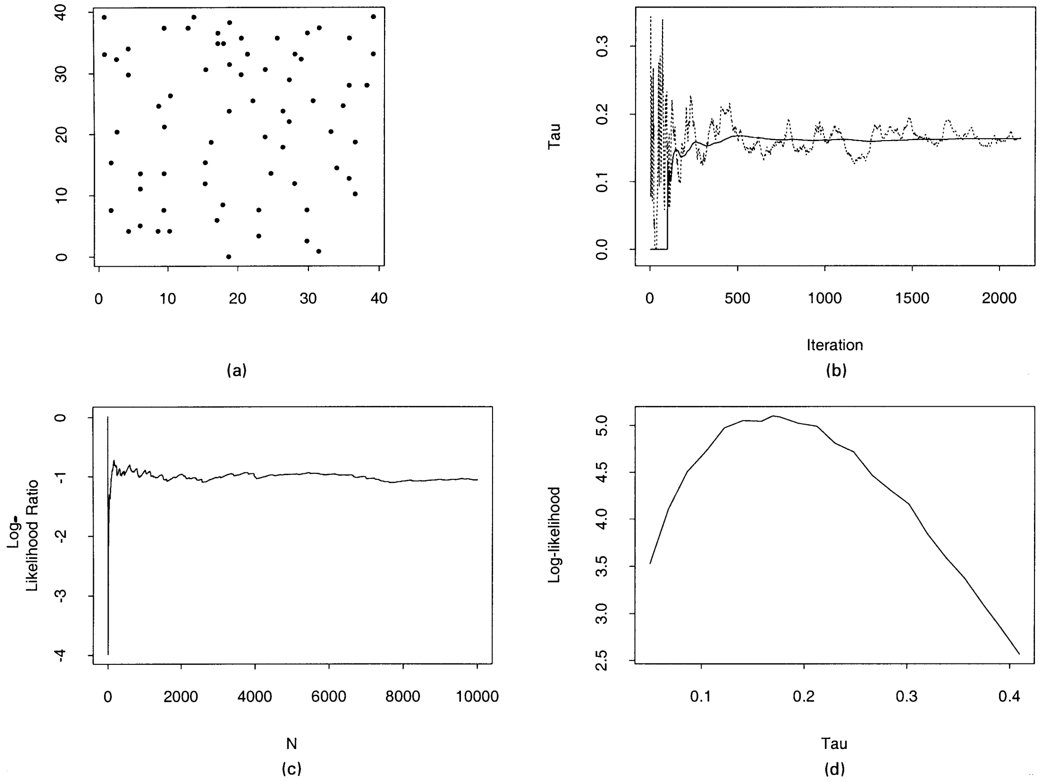


Fig. 2. Mercer and Hall's wheat yield data: (a) yields against $(i-1)\times 25+j$, where (i, j) denotes a site on a 20 × 25 rectangular lattice; (b) β^k (··········) and $\tilde{\beta}^k$ (———) at each iteration of the Markov chain Monte Carlo stochastic approximation algorithm; (c) σ^{*k} (··········) and $\tilde{\sigma}^{*k}$ (———) at each iteration of the Markov chain Monte Carlo stochastic approximation algorithm

We fitted this very-soft-core model to the Spanish towns data analysed by Ripley (1977) and Ogata and Tanemura (1984). The data set, shown in Fig. 3(a), consists of $n = 69$ points in an area 40 miles × 40 miles. Ogata and Tanemura (1984) assumed that the region has a periodic boundary and used the approximate likelihood method to calculate the approximate maximum likelihood estimate, obtaining $\hat{\tau}_{\mathrm{AML}} = 0.3036$.

The Markov chain Monte Carlo stochastic approximation algorithm proposed was used to obtain the maximum likelihood estimate of the unknown parameter τ. The starting value of τ_0 was set at 1.0. To generate the Markov chain, the Metropolis algorithm described in Diggle *et al.* (1994) was used. Let the current value of the process be $X = (x(1), \ldots, x(69))$ and the current total potential value be Q. A trial value $x(i)^*$ at the ith site leads to the potential value Q^*, where $x(i)^*$ is randomly chosen in some square with vertices at the points $(x(i)_1 \pm \delta, x(i)_2 \pm \delta)$ (modulo boundary) and $\delta > 0$ is a preassigned parameter. Then, the Metropolis procedure at the present site (i) continues as follows:

(a) if $Q^* \leqslant Q$, replace $x(i)$ and Q by $x(i)^*$ and Q^* respectively;
(b) if $Q^* > Q$, generate a uniform(0, 1) random variable U and
(c) if $U \leqslant \exp(Q - Q^*)$, set $x(i) = x(i)^*$ and $Q = Q^*$;
(d) otherwise, keep $x(i)$ and Q.



(a) (b) (c) (d)

Fig. 3. Spanish towns data: (a) locations of 69 Spanish towns in an area of 40 miles × 40 miles; (b) τ^k (⋯⋯⋯) and $\bar{\tau}^k$ (———) at each iteration of the Markov chain Monte Carlo stochastic approximation algorithm; (c) estimates of $\log\{f(\mathbf{X}|\hat{\tau}_{\text{AML}})\} - \log\{f(\mathbf{X}|\hat{\tau}_{\text{MLE}})\}$ against N, the number of random samples; (d) estimates of the log-likelihood values $\log\{f(\mathbf{X}|\tau)\}$ in (0, 0.4)

At each iteration of the stochastic approximation algorithm proposed, the same Metropolis–Hasting algorithm was used; however, each site i was selected at random with probability $1/69$ and m was set to 500. The two-stage Markov chain Monte Carlo stochastic approximation algorithm with $(a_1, b_1; a_2, b_2) = (0.3, 2; 0.6, 1)$ and $\tau^0 = 0.10$ was run to obtain $\hat{\tau}_{\text{MLE}} = 0.167$ and standard error 0.078. It took about 381 s for the Markov chain Monte Carlo stochastic approximation algorithm to converge in 2121 iterations. Fig. 3(b) shows the convergence behaviour of τ^k and $\bar{\tau}^k$. We define an influential range of the very-soft-core model by r_0 such that $\phi(r_0; \hat{\tau}_{\text{MLE}}) = 0.1$, which gives $r_0 = 3.02$ miles. This is moderately consistent with the observation of Ripley (1977).

Compared with Ogata and Tanemura's (1984) result, our estimate $\hat{\tau}_{\text{MLE}}$ is quite different from $\hat{\tau}_{\text{AML}} = 0.3036$. To justify our approach, we used the Monte Carlo likelihood approach (Geyer and Thompson, 1992; Geyer, 1999) to calculate the log-likelihood ratio $\log\{f(\mathbf{X}|\hat{\tau}_{\text{AML}})\} - \log\{f(\mathbf{X}|\hat{\tau}_{\text{MLE}})\}$. Fig. 3(c) shows estimates of the log-likelihood ratio based on N random samples simulated from $f(\mathbf{X}|\hat{\tau}_{\text{MLE}})$, in which the number N increases from 1 to 10000. The estimated log-likelihood ratios are negative for large N, i.e. $f(\mathbf{X}|\hat{\tau}_{\text{MLE}}) > f(\mathbf{X}|\hat{\tau}_{\text{AML}})$. To justify our results, we also used the Ogata–Tanemura method to calculate the log-likelihood function values in (0.0, 0.4). We took 40 equally spaced points $\{\tau(s)\colon s = 1, \ldots, 40\}$ in (0.0, 0.4) and 200 equally spaced points in $(0, \tau(s))$ for each s. At each such 200 space points, 20000 random samples were used to calculate the first derivative of the partition function, with a burn-in phase of 4000 iterations. This completed the process to calculate the log-likelihood function $f\{\mathbf{X}|\tau(s)\}$. We repeated this process 20 times and took their means as the final estimates of the log-likelihood function. The results obtained are shown in Fig. 3(d).

7. Comparison with the classical stochastic approximation

To illustrate the advantage of the proposed algorithm over the classical (Robbins and Monro, 1951) stochastic approximation algorithm for computing the maximum likelihood estimate for a spatial model (Younes, 1988, 1989; Moyeed and Baddeley, 1991), we also applied the classical algorithm to Wiebe's data. In the classical algorithm, θ^k is updated according to

$$\theta^k = \theta^{k-1} + \gamma_k\{-\nabla Q(\theta^{k-1}) + \bar{H}(\theta^{k-1}; \mathbf{X}_k)\},$$

where we have used $\gamma_k = 1/(1000 + k)$, and $\mathbf{X}_k$ is simulated as in Section 4. After about 111 s, the stochastic approximation algorithm was stopped at the 14746th iteration with $|\theta^{14746} - \theta^{14745}| < 10^{-6}$. To make a comparison with our Markov chain Monte Carlo stochastic approximation algorithm, we calculate $\hat{\Delta}^{(k)}$ for the classical stochastic approximation algorithm (not reported in this paper). We find that θ^k oscillates greatly and converges gradually to the maximum likelihood estimate. Moreover, $\hat{\Delta}^{(14746)} \doteq 0.206$, which is still not sufficiently small if we use $\eta_2 = 0.001$ in our convergence criterion.

Geyer (1999) concluded that the direct application of the Robbins–Monro method is not suitable for computing the maximum likelihood estimate for even moderate precision and may be only used to obtain a starting-point for the other methods. Our simulation confirms this conclusion. However, comparing the results in Fig. 1 and above, we see that there is a drastic improvement of our algorithm over the classical stochastic approximation algorithm. Moreover, our proposal provides a standard error automatically and the precision of the computation can be controlled by adjusting η_2.

8. Discussion

The Markov chain Monte Carlo stochastic approximation algorithm proposed contains four new distinctive features: the use of large gain constants in stage I, the use of adaptive search directions, the use of off-line averages and the use of a stopping criterion based on $l(\tilde{\theta}^k)$. Each has contributed to the improvements of our Markov chain Monte Carlo stochastic approximation algorithm over the classical stochastic approximation algorithm.

Another key idea of our algorithm is to estimate $\hat{\theta}$, $\nabla[\log\{C(\hat{\theta})\}]$ and $\nabla^2[\log\{C(\hat{\theta})\}]$ simultaneously. This seems inefficient at first but in fact very little extra work is required. Since we always need to estimate the information matrix, the quantity $\nabla^2[\log\{C(\hat{\theta})\}]$ should always be estimated. No extra computation is needed to estimate $\nabla[\log\{C(\hat{\theta})\}]$.

The traditional stopping criterion for an iterative algorithm is based on the relative change in the iterates. For such a stopping criterion in the case of the Monte Carlo EM algorithm, see Booth and Hobert (1999). We saw in Section 7 that such a criterion does not guarantee convergence. Geyer (1999) advocated stopping based on the Monte Carlo simulation error. Since our algorithm is iterative and Monte Carlo based, the stopping criterion must depend on both. The criterion in Section 3 does so.

There are two crucial requirements to implement the Monte Carlo likelihood method

(Geyer and Thompson, 1992; Geyer, 1999). The first is the requirement of a starting value which is sufficiently close to the maximum likelihood estimate (Geyer, 1999). The second is that there are simple sufficient statistics for the model (Huffer and Wu, 1998). Although the maximum pseudolikelihood estimation and stochastic approximation methods can be used to find a good starting value, the second requirement is not easy to overcome. In models like the Ising model, the Strauss hard-core model and the autologistic regression model, simple sufficient statistics do exist and the Monte Carlo likelihood method can be applied. However, when simple sufficient statistics do not exist, such as for the very-soft-core model discussed in Section 6 and for other more complicated spatial point pattern models (see, for example, Högmander and Särkkä (1999)), the stochastic approximation algorithm proposed should be preferred. If we use the Monte Carlo likelihood method in this case, we must store in the computer memory all the simulated configurations $\mathbf{X}_1, \ldots, \mathbf{X}_M$, where M usually depends on the precision that is desired and is very large. Each configuration $\mathbf{X}_i$ is a high dimensional vector, representing a graph or a spatial point pattern. In contrast, in the Markov chain Monte Carlo stochastic approximation algorithm proposed, we must only store in the computer memory, in addition to the current estimates and their off-line averages, $\mathbf{X}_{k,1}, \ldots, \mathbf{X}_{k,m}$ for iteration k and the number m does not depend on the precision that is desired. We believe that the Markov chain Monte Carlo stochastic approximation proposed will be useful especially for complicated spatial models.

Acknowledgements

The authors would like to express their gratitude to the Joint Editor and three referees for their very helpful comments that led to a much improved presentation, and Dr F. Huang, Dr F. M. Liang and Professor Y. Ogata for their generous help. The research of Dr M. G. Gu was supported by a Hong Kong Research Grants Council earmarked grant. The research of Dr H. Zhu was supported by the Pacific Institute of Mathematical Sciences and Natural Science and Engineering Research Council of Canada.

References

Andrews, D. F. and Herzberg, A. M. (1985) *Data*. New York: Springer.

Baddeley, A. J. and Turner, R. (2000) Practical maximum pseudo-likelihood for spatial point patterns (with discussion). *Aust. New Z. J. Statist.*, **42**, 283–322.

Barndorff-Nielsen, O. E., Kendall, W. S. and van Lieshout, M. C. (1999) *Stochastic Geometry: Likelihood and Computation*. London: Chapman and Hall.

Besag, J. E. (1974) Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. R. Statist. Soc.* B, **36**, 192–236.

——— (1977) Efficiency of pseudo likelihood estimators for simple Gaussian fields. *Biometrika*, **64**, 616–618.

Besag, J. and Green, P. J. (1993) Spatial statistics and Bayesian computation. *J. R. Statist. Soc.* B, **55**, 25–37.

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

Besag, J. E. and Moran, P. A. P. (1975) On the estimation and testing of spatial interaction in Gaussian lattice processes. *Biometrika*, **62**, 555–562.

Booth, J. G. and Hobert, J. P. (1999) Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm. *J. R. Statist. Soc.* B, **61**, 265–285.

Brémaud, P. (1998) *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. New York: Springer.

Chen, H. F., Guo, L. and Gao, A. J. (1988) Convergence and robustness of the Robbins-Monro algorithm truncated at randomly varying bounds. *Stoch. Processes Applic.*, **27**, 217–231.

Comets, F. (1992) On consistency of a class of estimators for exponential families of Markov random fields on the lattice. *Ann. Statist.*, **20**, 455–468.

Cressie, N. A. C. (1993) *Statistics for Spatial Data*. New York: Wiley.

Delyon, B., Lavielle, E. and Moulines, E. (1999) Convergence of a stochastic approximation version of the EM algorithm. *Ann. Statist.*, **27**, 94–128.

Diggle, P. J. (1983) *Statistical Analysis of Spatial Point Patterns*. New York: Academic Press.

Diggle, P. J., Fiksel, T., Grabarnik, P., Ogata, Y., Stoyan, D. and Tanemura, M. (1994) On parameter estimation for pairwise interaction point processess. *Int. Statist. Rev.*, **62**, 99–117.

Duflo, M. (1997) *Random Iterative Models*. New York: Springer.

Gelman, A. and Rubin, D. B. (1992) Inference from iterative simulation using multiple sequences (with discussion). *Statist. Sci.*, **7**, 457–511.

Geyer, C. J. (1999) Likelihood inference for spatial point processes. In *Stochastic Geometry: Likelihood and Computation* (eds O. E. Barndorff-Nielsen, W. S. Kendall and M. C. van Lieshout). London: Chapman and Hall.

Geyer, C. J. and Thompson, E. A. (1992) Constrained Monte Carlo maximum likelihood for dependent data (with discussion). *J. R. Statist. Soc.* B, **54**, 657–699.

Goulard, M., Särkkä, A. and Grabarnik, P. (1996) Parameter estimation for marked Gibbs point processes through the maximum pseudo likelihood method. *Scand. J. Statist.*, **23**, 365–379.

Gu, M. G. and Kong, F. H. (1998) A stochastic approximation algorithm with Markov chain Monte Carlo method for incomplete data estimation problems. *Proc. Natn. Acad. Sci. USA*, **95**, 7270–7274.

Guyon, X. (1982) Parameter estimation for a stationary process on a *d*-dimensional lattice. *Biometrika*, **69**, 95–105.

Högmander, H. and Särkkä, A. (1999) Multitype spatial point patterns with hierarchical interactions. *Biometrics*, **55**, 1051–1058.

Huang, F. and Ogata, Y. (1999) Improvements of the maximum pseudo-likelihood estimators in various spatial statistical models. *J. Comput. Graph. Statist.*, **8**, 510–530.

Huffer, F. W. and Wu, H. L. (1998) Markov chain Monte Carlo for auto-logistic regression models with application to the distribution of plant species. *Biometrics*, **54**, 509–524.

Ishiguro, M. and Akaike, H. (1989) DALL: Davidon's algorithm for log likelihood maximization—a FORTRAN subroutine for statistical model builders. *Comput. Sci. Monogr.*, **25**.

Jensen, J. L. and Møller, J. (1991) Pseudolikelihood for exponential family models of spatial point processes. *Ann. Appl. Probab.*, **1**, 445–461.

Kelly, F. P. and Ripley, B. D. (1976) A note on Strauss's model for clustering. *Biometrika*, **63**, 357–360.

Kesten, H. (1957) Accelerated stochastic approximation. *Ann. Math. Statist.*, **28**, 41–59.

Kushner, H. J. and Yin, G. G. (1997) *Stochastic Approximation Algorithms and Applications*. New York: Springer.

Mase, S. (1995) Consistency of the maximum pseudo-likelihood estimator of continuous state space Gibbsian processes. *Ann. Appl. Probab.*, **5**, 603–612.

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

Moran, P. A. P. (1973) A Gaussian Markovian process on a square lattice. *J. Appl. Probab.*, **10**, 54–62.

Moyeed, R. A. and Baddeley, A. J. (1991) Stochastic approximation of the MLE for a spatial point pattern. *Scand. J. Statist.*, **18**, 39–50.

Müller, P. (1991) A generic approach to posterior integration and Gibbs sampling. *Technical Report*. Purdue University, West Lafayette.

Ogata, Y. and Tanemura, M. (1984) Likelihood analysis of spatial point patterns. *J. R. Statist. Soc.* B, **46**, 496–518.

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

Pickard, D. K. (1982) Inference for general Ising models. *J. Appl. Probab.* A, **19**, 345–357.

Polyak, B. T. (1990) New stochastic approximation type procedures. *Autom. Telem.*, 98–107. (Engl. transl. *Autom. Remote Contr.*, **51**, 937–946.)

Polyak, B. T. and Juditski, A. B. (1992) Acceleration of stochastic approximation by averaging. *SIAM J. Contr. Optimzn*, **30**, 838–855.

Ripley, B. D. (1977) Modelling spatial patterns (with discussion). *J. R. Statist. Soc.* B, **39**, 172–212.

——— (1981) *Spatial Statistics*. New York: Wiley.

Robbins, H. and Monro, S. (1951) A stochastic approximation method. *Ann. Math. Statist.*, **22**, 400–407.

Robert, C. P. and Casella, G. (1999) *Monte Carlo Statistical Methods*. New York: Springer.

Rue, H. (2001) Fast sampling of Gaussian Markov random fields. *J. R. Statist. Soc.* B, **63**, 325–338.

Searle, S. R. (1971) *Linear Models*. New York: Wiley.

Stoyan, D., Kendall, W. S. and Mecke, J. (1987) *Stochastic Geometry and Its Applications*. New York: Wiley.

Strauss, D. J. (1975) A model for clustering. *Biometrika*, **62**, 467–475.

——— (1977) Clustering on coloured lattices. *J. Appl. Probab.*, **14**, 135–143.

Younes, L. (1988) Estimation and annealing of Gibbsian fields. *Ann. Inst. H. Poincaré*, **24**, 269–294.

——— (1989) Parameter estimation for imperfectly observed Gibbsian fields. *Probab. Theory Reltd Flds*, **82**, 625–645.