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

by

Ming-Gao Gu
Department of Statistics, The Chinese University of Hong Kong
and
Hong-Tu Zhu
Department of Mathematics & Statistics, University of Victoria
AND Department of Mathematics, An Hui Normal University

Acknowledgments: The authors would like to express their gratitude to 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. Gu was supported by a Hong Kong RGC Earmarked grant. The research of Dr. H. Zhu was supported by Pacific Institute of Mathematical Sciences and Natural Science and Engineering Research Council of Canada.

Send correspondence to:

Prof. Ming-Gao Gu
Department of Statistics
The Chinese University of Hong Kong
Shatin, N.T. Hong Kong.
Email: minggao@cuhk.edu.hk

SUMMARY

We propose a two-stage algorithm for computing maximum likelihood estimates for a class of spatial models. The algorithm combines the features of 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 (Polyak and Juditsky, 1992) and adaptive search direction (Gu and Kong, 1998). A new criterion is also build into the algorithm so stopping is automatic once the desired precision is set. Simulation studies and applications to some real data sets have been conducted with three specific spatial models. We compared the proposed algorithm to a direct application of the classical Robbins-Monro algorithm using Wiebe's wheat data (Andrews and Herzberg, 1985) and found that our procedure is at least 15 times faster.

Keywords: Auto-normal model; Ising model; Markov chain Monte Carlo; Off-line average; Spatial models; Stochastic approximation; Very-Soft-Core model.

1. Introduction

Recently, there has been an increasing interest in modeling spatial data with interaction among points. Those include Strauss-type hard-core models (Strauss, 1975; Kelly and Ripley, 1976); inhomogeneous spatial Poisson processes (Baddeley and Turner, 1998); 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 continuous spatial index, problems with lattice index, and spatial point patterns. For a general introduction to statistical methodology for spatial models, see Ripley (1981), Diggle (1983), Stoyan, Kendall and Mecke (1987), Cressie (1993) and Barndorff-Nielsen, Kendall and van Lieshout (1999).

Due to the intractable likelihood function, maximum likelihood estimation has so far rarely been 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). Direct computation of the maximum likelihood estimation by numerical approximation for some pairwise interaction models was developed by Ogata and Tanemura (1984). Other Monte Carlo based methods include the Monte Carlo Newton-Raphson approach (Penttinen, 1984) and the stochastic approximation approach (Younes, 1988, 1989; Moyeed and Baddeley, 1991). Due to the difficulties encountered in directly computing the maximum likelihood estimation, the maximum pseudo-likelihood estimator for spatial models was proposed as an alternative to the maximum likelihood estimation (Besag, 1977; Goulard, Särkkä and Grabarnik, 1996; Baddeley and Turner, 1998). However, the maximum pseudo-likelihood 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 pseudo-likelihood estimator and a one-step Monte Carlo

Newton-Raphson method.

In this paper, we consider computing the maximum likelihood estimation of spatial models via an improved Markov chain Monte Carlo stochastic approximation (MCMC-SA) algorithm. Younes (1988) first proposed to use the Markov chain Monte Carlo stochastic approximation for spatial statistical models. Moyeed and Baddeley (1991) have applied the Robbins and Monro (1951) type algorithms to the Strauss hard-core model for the maximum likelihood estimation. However, the use of stochastic approximation type of algorithms for spatial models was 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 get a starting point for the Monte Carlo likelihood method (Younes, 1989; Geyer, 1999).

Recent developments in stochastic approximation (Kushner and Yin, 1997) and in using it to find the maximum likelihood estimates in general missing data problems (Gu and Kong, 1998; Delyon, Lavielle and Moulines, 1999) shed new light on applying stochastic approximation algorithms for spatial models. The major idea of Gu and Kong (1998) is that stochastic approximation can be used to compute the maximum likelihood estimation and the information matrix simultaneously, and the estimated information matrix at each iteration can be used in updating the estimated maximum likelihood estimation, hence the optimal rate of convergence is achieved. In Delyon, Lavielle and Moulines (1999), two recent breakthroughs in stochastic approximation were introduced: the method of dynamic bounds (Chen, Guo and Gao, 1988), and the method of off-line average (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 the features of Markov chain Monte Carlo methods and these stochastic approximation type methods. Moreover, inspired by the 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 get 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 build into this stage so stopping of the program becomes automatic. Our algorithm has been successfully applied to three examples. At least for moderate data size, our algorithm bears none of the unpleasant features which mark earlier applications of stochastic approximation algorithms 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, some simulation studies and real examples are analyzed to illustrate our methodology. A comparison with the classical stochastic approximation is given in Section 7. A discussion is given in Section 8.

2. The spatial models and the MCMC-SA algorithm

2.1. The spatial models

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

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

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

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

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

$$\ell(\theta;\mathsf{x}_o) = \log f(\mathsf{x}_o|\theta) = -Q(\mathsf{x}_o;\theta) - \log C(\theta). \tag{3}$$

For simplicity of notation, we shall omit x_o in $\ell(\theta;\mathsf{x}_o)$ and $Q(\mathsf{x}_o;\theta)$.

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

2.2 The MCMC-SA algorithm

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

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

$$\bigtriangledown\ell(\theta) = -\bigtriangledown Q(\theta) - \bigtriangledown \log C(\theta) \quad \text{and} \quad \bigtriangledown^2 \ell(\theta) = -\bigtriangledown^2 Q(\theta) - \bigtriangledown^2 \log C(\theta), \tag{4}$$

where $\bigtriangledown$ and $\bigtriangledown^2$ are the first and second derivative operators with respect to θ. Thus, if we can calculate $\bigtriangledown\ell(\theta)$ and $\bigtriangledown^2\ell(\theta)$ at each θ, we can expect to get the maximum likelihood estimate by the Newton-Raphson method. From (4), we need to calculate $\bigtriangledown \log C(\theta)$ and $\bigtriangledown^2 \log C(\theta)$.

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

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

where for vector a, $\mathsf{a}^{\otimes 2} = \mathsf{a}\mathsf{a}^T$. One way to calculate $\bigtriangledown \log C(\theta)$ and $\bigtriangledown^2 \log C(\theta)$ is to use numerical integration in (5). However, the numerical approximation is accurate only for some special cases, which usually gives unstable estimates. Another way is to resort to Monte Carlo integration. So in principle, we may obtain $\bigtriangledown \log C(\theta)$ and $\bigtriangledown^2 \log C(\theta)$ by using the Monte Carlo approximation, if we can simulate $\{\mathsf{X}_\theta(t) : t = 1, \cdots, T\}$ from model (1). In the case of spatial models, the generation of the random samples can be carried out by Markov chain Monte-Carlo methods, e.g. the Gibbs sampler, the Metropolis-Hastings algorithm, birth-and-death process 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 $\bigtriangledown \log C(\theta)$ and $\bigtriangledown^2 \log C(\theta)$ at each θ. The question is how close 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), results from a direct implementation of 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. While this algorithm improves upon the performance of the classical Robbins-Monro algorithm, direct application of this algorithm to the 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 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 estimation.

Another significant development in stochastic approximation is due to Polyak (1990) (see also Polyak and Judiski (1992), and Delyon, Lavielle and Moulines (1999)), who showed that if we run an ordinary Robbins-Monro algorithm with a bigger gain constants sequence ($\gamma_k = k^{-\alpha}$, $1/2 < \alpha < 1$, 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 direct application of this idea to the spatial models discussed in Sections 4-6 is unsatisfactory. The problem here again is the slowness of convergence to the maximum likelihood estimation if the initial value is mildly away. Theoretically, from the proof of Theorem 4 of Delyon, Lavielle and Moulines (1999) and Chapter 11 of Kushner and Yin (1997), we see that the optimal rate of convergence only kicks in when the estimate is sufficiently close to the maximum likelihood estimation.

If the initial value is far away from the maximum likelihood estimation, a large gain constant sequence can be used at first to force the estimates into a small neighborhood of the maximum likelihood estimation. This idea of using larger gain constants when the current estimate is still far away 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 estimation, 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 Judiski (1992). In both stages, the estimated search direction method of Gu and Kong (1998) is employed.

We first introduce two basic steps of the stochastic approximation specialized to our model and notation. Let us keep in mind that θ^k is the current estimate of $\hat{\theta}$, h_k is the current estimate of $E_{\hat{\theta}}[\triangledown Q(\mathsf{X};\hat{\theta})]$ and Γ_k is the current estimate of $-E_{\hat{\theta}}\{\triangledown^2 Q(\mathsf{X};\hat{\theta})\} + E_{\hat{\theta}}\{\triangledown Q(\mathsf{X};\hat{\theta})\}^{\otimes 2}$. We also assume that, for each θ, there exists 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 $f(\mathsf{x}|\theta)$.

Step 1. At the kth iteration, set $\mathsf{X}_{k,0} = \mathsf{X}_{k-1,m}$. For $i = 1, \cdots, m$, generate $\mathsf{X}_{k,i}$ from the transition probability density $\Pi_{\theta^{k-1}}(\mathsf{X}_{k,i-1}, \cdot)$;

Step 2. Update θ^{k-1} to θ^k, h_{k-1} to h_k and Γ_{k-1} to Γ_k by

$$\begin{cases} \mathsf{h}_k = \mathsf{h}_{k-1} + \gamma_k(\overline{H}(\theta^{k-1};\mathsf{X}_k) - \mathsf{h}_{k-1}), \\ \\ \Gamma_k = \Gamma_{k-1} + \gamma_k(\overline{I}(\theta^{k-1};\mathsf{X}_k) - \Gamma_{k-1}), \\ \\ \theta^k = \theta^{k-1} + \gamma_k[\triangledown^2 Q(\theta^{k-1}) + \Gamma_{k-1} - \mathsf{h}_{k-1}^{\otimes 2}]^{-1}[- \triangledown Q(\theta^{k-1}) + \overline{H}(\theta^{k-1};\mathsf{X}_k)], \end{cases} \tag{6}$$

where $\mathsf{X}_k = (\mathsf{X}_{k,1}, \cdots, \mathsf{X}_{k,m})$;

$$\overline{H}(\theta^{k-1};\mathsf{X}_k) = \frac{1}{m}\sum_{i=1}^m \triangledown Q(\mathsf{X}_{k,i};\theta^{k-1}),$$

and

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

Stage I of the proposed algorithm consists of choosing an initial point θ^0, an initial matrix Γ_0, an initial vector h_0, an initial spatial configuration $\mathsf{X}_{0,m}$ and of setting $k = 1$ followed by iterating Steps 1 - 2 with $k = 1, \ldots, K_1$. The gain constants are defined by

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

where $K_1 \geq K_0$ is determined by

$$K_1 = \inf \left\{ K \geq K_0 : \quad \left\| \sum_{k=K-K_0+1}^{K} \mathrm{Sign}(\theta^k - \theta^{k-1})/K_0 \right\| \leq \eta_1 \right\}, \tag{7}$$

where $\mathrm{Sign}(\theta)$ 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 $a_1 \in (0, 1/2)$, η_1 are preassigned.

Stage II starts when Stage I finishes and takes the final values of θ, h, Γ and 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

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

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

$$\tilde{\theta}^k = \tilde{\theta}^{k-1} + (\theta^k - \tilde{\theta}^{k-1})/k, \quad \tilde{\mathsf{h}}_k = \tilde{\mathsf{h}}_{k-1} + (\mathsf{h}_k - \tilde{\mathsf{h}}_{k-1})/k, \quad \tilde{\Gamma}_k = \tilde{\Gamma}_{k-1} + (\Gamma_k - \tilde{\Gamma}_{k-1})/k, \tag{8}$$

assuming $\tilde{\theta}^0 = 0$. After the K_2-th iteration, we use the off-line average $(\tilde{\theta}^{K_2}, \tilde{\mathsf{h}}_{K_2}, \tilde{\Gamma}_{K_2})$ as our final estimate of $(\hat{\theta}, -\bigtriangledown \log C(\hat{\theta}), \bigtriangledown^2 \log C(\hat{\theta}) + [\bigtriangledown \log C(\hat{\theta})]^{\otimes 2})$. This is equivalent to averaging of 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 zero, 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 proposed algorithm will typically move quickly towards the feasible region. In Stage II we use a_2 close to $1/2$, a small integer for b_2, say, $a_2 = 0.6$, $b_2 = 1$. Coupling with the off-line averaging procedure (8), the algorithm will stabilize in the neighborhood of the maximum likelihood estimate.

Exitence of finite value 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 small enough constant, then $(\mathsf{h}_k, \Gamma_k, \theta^k)$, $k = 1, 2, \ldots$ 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 $\tilde{\theta}^k$, $\tilde{\mathsf{h}}_k$ and $\tilde{\Gamma}_k$ in Stage II were given in Chapter 10 and 11 of Kushner and Yin (1997). We also refer the reader to 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 $k \to \infty$,

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

where Σ is the covariance matrix in the central limit theorem

$$\frac{1}{\sqrt{k}} \sum_{j=1}^{k} \left\{\overline{H}(\theta^{j-1}; \mathsf{X}_j) + \bigtriangledown \log C(\theta^{j-1})\right\} \to \mathcal{N}(0, \Sigma), \tag{10}$$

as $k \to \infty$.

The choice of m should not affect the convergence of the proposed procedure. In general, large m reduces the covariance Σ and the correlation between $\overline{H}(\theta^{k-1}; \mathsf{X}_k)$ and $\overline{H}(\theta^k; \mathsf{X}_{k+1})$. Therefore, large m reduces the number of iterations 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 $\bigtriangledown\ell(\theta)$ equals to zero at the maximum likelihood estimate $\hat{\theta}$. It is natural to consider a stopping criterion which is based on small values of $\bigtriangledown\ell(\tilde{\theta}^k)$. At iteration k, define $\Delta_k = (\bigtriangledown\ell(\tilde{\theta}^k))^T\Big[-\bigtriangledown^2\ell(\hat{\theta})\Big]^{-1}(\bigtriangledown\ell(\tilde{\theta}^k))$. Ignoring high order terms, simple algebra shows that Δ_k is asymptotically equivalent to $(\tilde{\theta}^k-\hat{\theta})^T\Big[-\bigtriangledown^2\ell(\hat{\theta})\Big](\tilde{\theta}^k-\hat{\theta})$, which is not effected by the scale since $\hat{\theta}$ has asymptotic variance $\Big[-\bigtriangledown^2\ell(\hat{\theta})\Big]^{-1}$. We consider choosing K_2, the number of iterations of Stage II, such that Δ_{K_2} be small. An estimate of $\bigtriangledown\ell(\tilde{\theta}^k)$ is $\bigtriangledown\tilde{\ell}(\theta^k) = -\bigtriangledown Q(\tilde{\theta}^k) + \tilde{\mathsf{h}}_k$ and an estimate of $-\bigtriangledown^2\ell(\hat{\theta})$ is $-\bigtriangledown^2\tilde{\ell}(\theta^{k-1}) = \bigtriangledown^2 Q(\tilde{\theta}^{k-1}) + \tilde{\Gamma}_{k-1} - \tilde{\mathsf{h}}^{\otimes 2}_{k-1}$. However, if we just use the natural estimate $\bigtriangledown\tilde{\ell}(\theta^k)^T\left[-\bigtriangledown^2\tilde{\ell}(\theta^{k-1})\right]^{-1}\bigtriangledown\tilde{\ell}(\theta^k)$ of Δ_k as our criterion for convergence, then we are ignoring a possibly large Monte Carlo error.

In order to control the Monte Carlo estimation error, we make the following consideration. Expressions (9) and (10) assert that $\sqrt{k}\,\bigtriangledown\tilde{\ell}(\theta^k)$ is asymptotically distributed as $\mathcal{N}(0,\Sigma)$. Therefore, the variance of $\bigtriangledown\tilde{\ell}(\theta^k)^T\left[-\bigtriangledown^2\tilde{\ell}(\theta^{k-1})\right]^{-1}\bigtriangledown\tilde{\ell}(\theta^k)$ is asymptotically $2\,\mathrm{tr}\left\{\left[-\bigtriangledown^2\tilde{\ell}(\theta^{k-1})\right]^{-1}\Sigma\right\}^2/k^2$, where $\mathrm{tr}\{A\}$ denote 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

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

Therefore, we define

$$K_2 = \inf\left\{k: \quad \hat{\Delta}_k \le \eta_2\right\},$$

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

Estimation of Σ can be performed in the following way. If m is large, we may expect the correlations between consecutive $\overline{H}(\theta^{j-1}; \mathsf{X}_j)$s to be small. So a natural estimate of Σ can be constructed via the sample covariance of those values. A more precise estimate of Σ can certainly be used if we treat $\{\overline{H}(\theta^{j-1}; \mathsf{X}_j), j = 1, \ldots, k\}$ 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 estimation, a rough estimate should be enough to serve our purpose. Moreover, in each iteration, $\bigtriangledown^2 Q(\theta^{k-1}) + \Gamma_{k-1} - \mathsf{h}_{k-1}{}^{\otimes 2}$ can be used as a rough estimate of $- \bigtriangledown^2 \ell(\hat{\theta})$, which will save computation time, especially when the dimension of θ is large.

To illustrate the behavior of the proposed Markov chain Monte Carlo stochastic approximation algorithm, two simulation studies and analyses of three real data sets in the literature will be discussed in Sections 4, 5 and 6. All computation were done in the C language on a SUN hpc4500 workstation. In all the examples, the convergence criterion in (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, $\mathsf{X} = \{x(i, j) : (i, j) \in \mathcal{Z}^2_{M_0,N_0}\}$, of the random field, are configurations of pluses and minuses on $\mathcal{Z}^2_{M_0,N_0}$. The statistic that count the excess of like, over unlike, nearest-neighbor points on the lattice, is defined as $V = V(\mathsf{X}) = \sum_{nn} x(i, j)x(u, v)$, where nn means that the summation is over all the pairs (i, j) and (u, v) such that the two points are nearest-neighbors. The potential function is $Q(\mathsf{X}; \theta) = -\theta V(\mathsf{X})$ and the normalizing factor is obtained by summing over all possible configurations X, $C(\theta) = \sum_{\mathsf{X}} e^{\theta V(\mathsf{X})}$. In this model, $V(\mathsf{X})$ is the minimal sufficient statistic for the parameter θ. The sign of θ

determines whether the Ising model is ferromagnetic or anti-ferromagnetic (attractive or repulsive). Let $m(\mathsf{X}) = \sum_{i=1}^{M_0}\sum_{j=1}^{N_0} x(i,j)$ be magnetic moment of configuration X, the spontaneous magnetization is defined by $SM(\theta) = \sum_{\mathsf{X}} m(\mathsf{X})e^{\theta V(\mathsf{X})}/(M_0N_0C(\theta))$. When $|\theta|$ is smaller than the critical temperature near 0.44, $SM(\theta)$ equals to zero (Brémaud, 1998).

In order to check the usefulness of the proposed algorithm, we consider the following simulation study for the Ising model. In this simulation study, the Ising model is set on a 64 × 64 square lattice on the plane: $\{x(l,j) : l, j = 1, \cdots, 64\}$. We assume the periodic boundary for the square lattice, which considers $\{x(i,64), x(i,1)\}$ and $\{x(64,j), x(1,j)\}$ as neighbors 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:

1) if $Q^* \leq Q$, replace $x(l,j)$ and Q by $x(l,j)^*$ and Q^* respectively;

2) if $Q^* > Q$, generate a Uniform(0, 1) random variable U and

   2.1) if $U \leq \exp(Q - Q^*)$, set $x(l,j) = x(l,j)^*$ and $Q = Q^*$;

   2.2) 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 is 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, $SM_T(\theta) = \sum_{t=1}^{T} m(\mathsf{X}^t)/T$ was used to assess the convergence of the Metropolis algorithm, where $\{\mathsf{X}^1, \cdots, \mathsf{X}^T\}(T \geq 320)$ is the the output from the

Metropolis algorithm. When $|SM_T(\theta)|$ is smaller than 0.001, we stoped the algorithm and declared that the equilibrium states was achieved.

Based on the simulated data sets, we applied the Markov chain Monte Carlo stochastic approximation algorithm described in Section 2 to get the maximum likelihood estimate of the unknown parameter. The starting value of θ is 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 $1/(64 \times 64)$ probability, not according to the lexicographical order. For example, if site (1,1) were 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 $\mathsf{X}_{k,i-1}$ to $\mathsf{X}_{k,i}$. The number m was set at $m = 20,000$. Compared with the total number of sites $64 \times 64 = 4096$, $m = 20,000$ is not too large.

To illustrate the performance of the proposed algorithm, 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 CPU time for each estimate are also obtained. The results obtained are summarized in Table 1. It can be seen that the performance of the proposed Markov chain Monte Carlo stochastic approximation algorithm 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 pseudo-likelihood 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 proposed Markov chain Monte Carlo stochastic approximation algorithm is better since the efficiency coefficients are uniformly closer to 1 for the proposed algorithm than those by the DALL optimization subroutine and by the maximum pseudo-likelihood estimate.

For an analysis of real data, we fitted the Ising model on 125 × 12 rectangle lattice to a transfered Wiebe's wheat data (Andrews and Herzberg, 1985). The value "1" denotes "larger than or equal to the mean value" and "-1" stands for "less than the mean value". Figure 1 (a) depicts this transferred Wiebe's wheat data. Inspection of it reveals that there is a strong degree of spatial correlation in the data; that is, whether the wheat yield at a given site is larger or less than the mean value is strongly related to the wheat yields at neighboring sites. We also assume the periodic boundary condition for the data. 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 is 5000 was used to obtain the maximum likelihood estimate $\hat{\theta} = 0.372$ and the standard deviation estimate 0.012. The large value of $\hat{\theta}$ is consistent with the observation in Figure 1 (a). The algorithm was stopped at the 833-th iteration after 6 seconds. The same Metropolis algorithm as in the simulation is used. The likelihood function calculated via the Ogata-Tanemura method and the Onsager formula are presented in Figure 1 (b). The starting value of unknown parameter θ is set at -0.3, which is far from the $\hat{\theta}$. Figure 1 (c) and (d) show the convergence behavior of θ^k, $\tilde{\theta}^k$ and $\hat{\Delta}_k$ at each iteration. Our algorithm shows the robustness to the initial value of unknown parameter θ and can find the true maximum likelihood estimate with high precision.

5. Auto-normal Model

Consider a Gaussian Markov random field $\mathsf{X} = \{x(i,j)\}$ 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 the rest of sites is given by

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

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

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

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 due to the difficulty of evaluating the normalizing constant, since B is a $M_0N_0\times M_0N_0$ 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 auto-normal model. The Gaussian Markov process is set on a 64×64 square lattice on the plane: $\{x(i,j):i,j=1,\cdots,64\}$. 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 neighbor sites of (i,j); and $\beta_{i,j;u,v}=0$ for the other (u,v). In our simulations, β is set to be 0.05, 0.11, 0.159 and 0.233, only processes with $\beta<0.25$ exist (Moran, 1973). The true value of $\sigma=\exp(\sigma^*)$ is set to 1.0; that is, $\sigma^*=0.0$. 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 field is given in Rue (2000). For each site (i,j), selected in the 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 Gelman and Rubin (1992) method and choose to monitor sufficent statistics $s_1(\mathsf{X}) = \sum_{i=1}^{M_0}\sum_{j=1}^{N_0} x(i,j)^2/(M_0N_0)$ and $s_2(\mathsf{X}) = \sum_{i=1}^{M_0}\sum_{j=1}^{N_0} x(i,j)x(i,j)^*/(M_0N_0)$. Starting from four quite different initial state of the process were chosen, four Gibbs algorithm 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 the Gelman and Rubin's (1992) convergence criterions are close to 1, we stopped the Gibbs algorithm and declared that the equilibrium states are achieved.

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

Figure 2 (a) depicts the original Mercer and Hall's wheat yield data on a 20×25 rectangle lattice (Andrews and Herzberg, 1985), which was also analysed in Besag (1974) and Huang and Ogata (1999). We have fitted the first order auto-normal model and subtracted the mean from the data, equivalent to adding a shift parameter to the auto-normal 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 maximum likelihood estimates. It took the algorithm 1504 iterations and 63 seconds CPU time to achieve the maximum likelihood estimation $(\hat{\beta}, \hat{\sigma}^*) = (0.237, -1.025)$ and 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 Figure 2 (b) and (c), respectively. It seems that a large gain constants sequence in Stage I had effectively forced the estimates to a small neighborhood of $(\hat{\beta}, \hat{\sigma}^*)$.

6. Very-Soft-Core Model

A spatial point pattern data is described by the coordinates of points $\mathsf{X} = \{x_i \in A : i = 1, \cdots, n\}$ in a planar region A. The joint density of a pattern X of a pairwise interaction point process is given by (1) with $\theta = \tau$ and the potential function

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

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

$$C(\tau) = \int_{A^n} \exp\left\{ -\sum_{i=1}^{n} \sum_{j>i} \phi(||x_i - x_j||; \tau) \right\} dx_1 \cdots dx_n,$$

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 (VSC) 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 conditional on the observed number of points.

We fitted this Very-Soft-Core model to the Spanish towns data previously analyzed by Ripley (1977) and Ogata and Tanemura (1984). The data set, shown in Figure 3 (a), consists of $n = 69$ points in a 40×40 mile area. Ogata and Tanemura (1984) assumed that the region has a periodic boundary and used the approximate likelihood method to calculate the approximate MLE, getting $\hat{\tau}_{AML} = 0.3036$.

The proposed Markov chain Monte Carlo stochastic approximation algorithm was used to obtain the MLE of the unknown parameter τ. The starting value of τ_0 was set at 1.0. To generate the Markov chain, the Metropolis algorithm as 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:

1) if $Q^* \leq Q$, replace $x(i)$ and Q by $x(i)^*$ and Q^* respectively;

2) if $Q^* > Q$, generate a Uniform(0, 1) random variable U and

   2.1) if $U \leq \exp(Q - Q^*)$, set $x(i) = x(i)^*$ and $Q = Q^*$;

   2.2) otherwise, keep $x(i)$ and Q.

In each iteration of the proposed stochastic approximation algorithm, the same Metropolis-Hasting algorithm was used; however, each site i was selected at random with $1/69$ probability and m is 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 get $\hat{\tau}_{MLE} = 0.167$ and standard error 0.078. It took about 381 seconds for the Markov chain Monte Carlo stochastic approximation algorithm to converge in 2121 iterations. Figure 3 (b) shows the convergence behavior of τ^k and $\tilde{\tau}^k$. If we define an influential range of the Very-Soft-Core model by r_0 such that $\phi(r_0; \hat{\tau}_{MLE}) = 0.1$, 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 $\hat{\tau}_{MLE}$ is quite different from $\hat{\tau}_{AML} = 0.3036$. To justify our approach, we used the Monte Carlo likelihood (Geyer and Thompson, 1992; Geyer, 1999) approach to calculate the log-likelihood ratio $\log f(\mathsf{X}|\hat{\tau}_{AML}) - \log f(\mathsf{X}|\hat{\tau}_{MLE})$. Figure 3 (c) shows the estimates of the log-likelihood ratio basing on N random samples simulated from $f(\mathsf{X}|\hat{\tau}_{MLE})$, in which the number N increases from 1 to 10000. It can be seen that the estimates of the log-likelihood ratios are smaller than zero for large N; that is, $f(\mathsf{X}|\hat{\tau}_{MLE}) > f(\mathsf{X}|\hat{\tau}_{AML})$. 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 space points $\{\tau(s) : s = 1, \cdots, 40\}$ in $(0.0, 0.4)$ and took 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 a process to calculate the log-likelihood function $f(\mathsf{X}|\tau(s))$. 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 Figure 3 (d).

7. Comparison to the Classical Stochastic Approximation

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

$$\theta^k = \theta^{k-1} + \gamma_k[- \bigtriangledown Q(\theta^{k-1}) + \overline{H}(\theta^{k-1}; \mathsf{X}_k)],$$

where we have used $\gamma_k = 1/(1000 + k)$, and X_k is simulated as in Section 4. After about 111 seconds, the stochastic approximation algorithm was stopped at the 14746-th 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 very much and converges gradually to the maximum likelihood estimation. Moreover, $\hat{\Delta}^{(14746)} \doteq 0.206$, which is still not small enough if we use $\eta_2 = 0.001$ in our convergence criterion.

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

8. Discussion

The proposed Markov chain Monte Carlo stochastic approximation algorithm 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 $\ell(\tilde{\theta}^k)$. Each of those has contributed to the improvements of our proposed Markov chain Monte Carlo stochastic approximation algorithm over the classical stochastic approximation algorithm.

Another key idea of our algorithm is to estimate $\hat{\theta}$, $\bigtriangledown \log C(\hat{\theta})$ and $\bigtriangledown^2 \log C(\hat{\theta})$ 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 $\bigtriangledown^2 \log C(\hat{\theta})$ should always be estimated. No extra computation needed to estimate $\bigtriangledown \log C(\hat{\theta})$.

The traditional stopping criterion for an iterative algorithm is based on the relative change of the iterates. For such a stopping criterion in the case of the Monte Carlo Expectation-Maximization algorithm, see Booth and Hobert (1999). We saw in Section 7 that such a criterion does not guarantee convergence. Geyer (1999) has advocated stopping based on the Monte Carlo simulation error. Since our algorithm is iterative and Monte Carlo based, the stopping criterion has to 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 exist simple sufficient statistics for the model (Huffer and Wu, 1998). While the maximum pseudo-likelihood 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 auto-logistic 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 proposed stochastic approximation algorithm should be preferred. If we use the Monte Carlo likelihood method in the later case, we have to store in the computer memory all the simulated configurations $\mathsf{X}_1, \ldots, \mathsf{X}_M$, where M usually depends on the precision desired and is very large. We keep in mind that each configuration X_i is a high-dimensional vector, representing a graph or a spatial point pattern. On the other hand, in the proposed Markov chain Monte Carlo stochastic approximation algorithm, one only have to stored in the computer memory, in addition to the current estimates and their off-line averages, $\mathsf{X}_{k,1}, \ldots, \mathsf{X}_{k,m}$ for iteration k and the number m does not depend on the precision desired. We believe that the proposed Markov chain Monte Carlo stochastic approximation will be useful especially in the latter case.

References

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

Baddeley, A. J. and Turner, R. (1998). Practical maximum pseudo-likelihood for spatial point patterns. To appear in Australian and New Zealand Journal of Statistics.

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

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

Besag, J. E. (1977). Efficiency of pseudo likelihood estimators for simple Gaussian fields. Biometrika, 64, 616-618.

Besag, J. E. and Green, P. (1993). Spatial statistics and Bayesian computation. J. R. Statist. Soc. B, 55, 25-38.

Besag, J. E., Green, P., Higdon, D. and Mengersen, K. (1995). Bayesian computation and stochastic systems. Statistical Science, 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). Maximum 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-Verlag.

Chen, H. F., Guo, L. and Gao, A. J. (1988). Convergence and robustness of the Robbins-Monro algorithm truncated at randomly varying bounds. *Stoch. Process Appl.* *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: John Wiley and Sons.

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*. Academic Press, New York.

Diggle, P. J., Fiksel, T., Grabarnik, P., Ogata, Y., Stoyan, D. and Tanemura, M. (1994). On parameter estimation for pairwise interaction point processess. *International Statistical Review*, *62*, 99-117.

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

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

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.

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

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. Proceedings of National Academic Science, USA, 95, 7270-7274.

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

Hoğmander, 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. Journal of Computational and Graphical Statistics, 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, Computer Science Monographs, 25, The Institute of Statistical Mathematics, Tokyo.

Jensen, J. L. and Møller, J. (1991). Pseudolikelihood for exponential family models of spatial point processes. *Ann. Appl. Prob.* *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. *Annals of Mathematical Statistics*, *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. *Annals of Applied Probability*, *5*, 603-612.

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

Moran, P. A. P. (1973). A Gaussian Markovian process on a square lattice. *Journal of Applied Probability*, *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, Indiana, USA.

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. *Jyväskylä Studies in Computer Science*, Econometrics and Statistics 7.

Pickard, D. K. (1982). Inference for general Ising models. Journal of Applied Probability, 19A, 345-357.

Polyak, B. T. (1990). New stochastic approximation type procedures. Automatica i Telemekh, 98-107. (English translation in Automat. Remote Control. 51).

Polyak, B. T. and Juditski, A. B. (1992). Acceleration of stochastic approximation by averaging. SIAM J. Control Optim. 30, 838-855.

Ripley, B. D. (1977). Modelling spatial patterns. (with Discussion) J. R. Statist. Soc. B, 39, 172-212.

Ripley, B. D. (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-Verlag.

Rue, H. (2000). Fast sampling of Gaussian Markov random fields with application. Technical report. Department of Mathematical Sciences, NTNU, Norway.

Searle, S. R. (1971). Linear Models. New York: John Wiley and Sons.

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

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

Strauss, D. J. (1977). Clustering on coloured lattices. J. Appl. Prob., 14, 135-143.

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

Younes, L. (1989). Parameter estimation for imperfectly observed Gibbsian fields. Probab. Theory and Related Fields., **82**, 625-645.

Table 1: Biases ($\times 10^{-3}$), standard deviations ($\times 10^{-2}$), the root mean square error ($\times 10^{-2}$), and efficiency coefficients of the estimators of the Ising model.

	MLE (SA)					MLE (DALL)				
θ_0	-0.40	-0.20	0.0	0.20	0.40	-0.40	-0.20	0.0	0.20	0.40
Bias	-0.14	0.07	1.01	0.14	1.05	0.94	-0.24	1.47	-0.60	-0.43
SD	0.68	1.00	1.10	1.00	0.67	0.71	0.96	1.08	1.11	0.68
RMS	0.68	1.04	1.15	0.92	0.70	0.78	0.89	1.03	1.35	0.70
EFF	1.00	0.96	0.96	1.08	0.98	0.91	1.08	1.05	0.82	0.97
AVEN	915	322	228	330	936					
AVET	30s	11s	7s	12s	31s					

	MPLE				
θ_0	-0.40	-0.20	0.0	0.20	0.40
Bias	0.01	-0.57	-1.51	-0.11	-0.20
SD	1.22	1.10	1.08	1.19	1.21
RMS	3.93	1.35	1.04	1.70	3.89
EFF	0.31	0.82	1.04	0.70	0.31

In Tables 1 and 2, 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.

Table 2: Bias ($\times 10^{-3}$), RMS ($\times 10^{-2}$), SD ($\times 10^{-2}$), and EFF of the ML estimators of the Auto-normal model.

β					σ^* = 0.0					
true	Bias	RMS	SD	EFF	Bias	RMS	SD	EFF	AVEN	AVET
0.050	-0.560	1.07	1.08	0.99	-0.61	1.07	1.11	1.04	985	179s
0.110	-0.40	0.96	0.95	0.99	0.96	1.07	1.14	1.07	1042	187s
0.159	0.19	0.77	0.77	1.00	0.60	1.14	1.15	1.01	1138	206s
0.233	-0.09	0.29	0.29	1.00	0.84	1.17	1.16	1.00	1688	306s

β	MLE(DALL)			β	MPLE		
true	Bias	SD	EFF	true	Bias	SD	EFF
0.049	1.14	1.16	0.89	0.049	1.11	1.16	0.88
0.110	-0.01	1.00	0.95	0.110	0.54	1.06	0.85
0.159	0.38	0.81	0.95	0.159	0.62	0.86	0.85
0.233	0.01	0.29	1.10	0.233	0.16	0.42	0.53

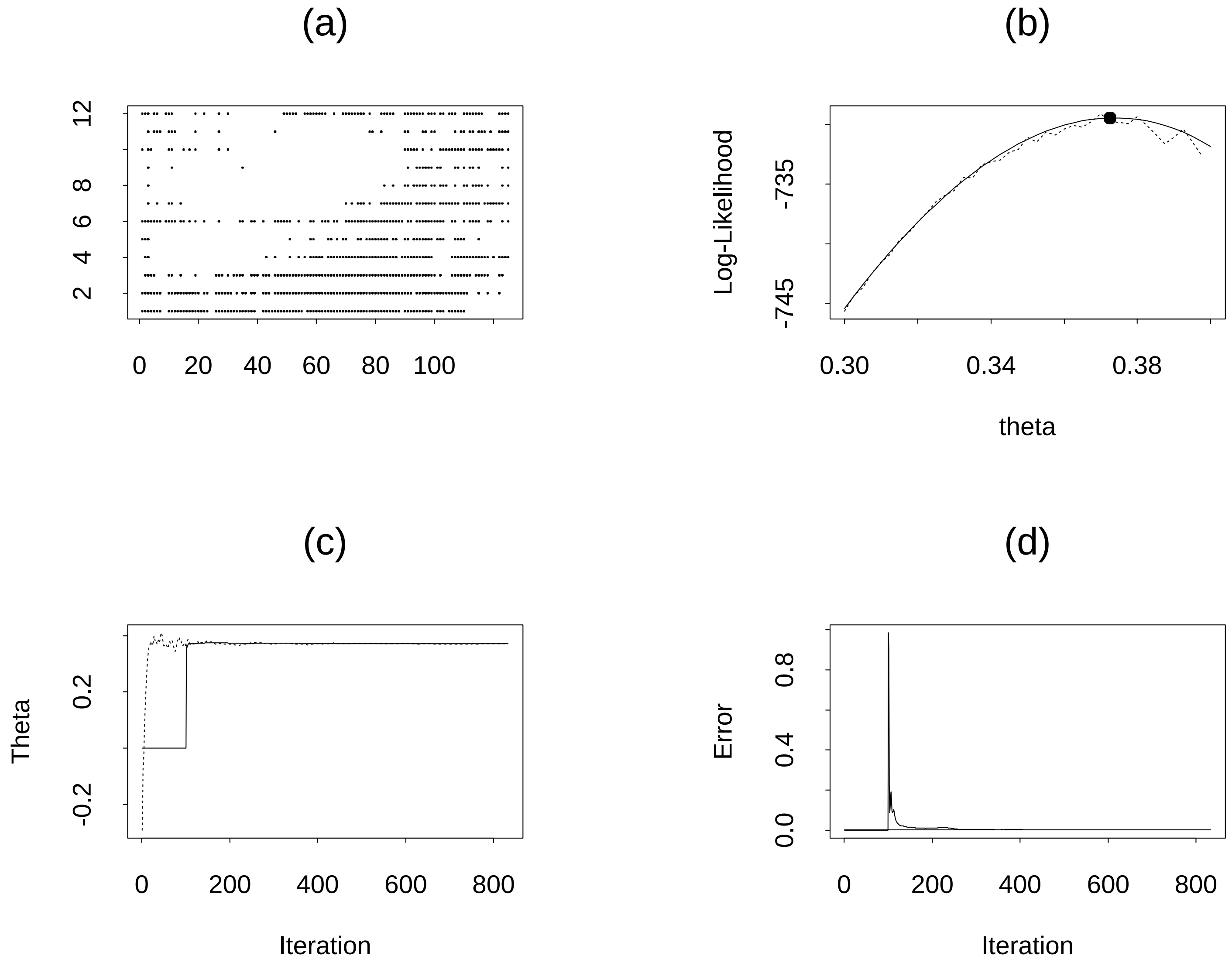

Figure 1: Transferred Wiebe's wheat data: (a) The transferred wiebe's wheat data on a 125×20 rectangle lattice, in which the black circe at a given site denotes "large than or equal to the mean value" and no sign at a given site denotes "less than the mean value"; (b) The curve of the log-likelihood function, in which — denotes the log-likelihood values calculated via the Onsager formula and - - - denotes the log-likelihood values calculated via the Ogata and Tanemura's method. (c) θ^k (- - -) and $\tilde{\theta}^k$ (—–) at each iteration of the MCMC-SA algorithm; (d) $\hat{\Delta}_k$ at each iteration of the MCMC-SA algorithm.

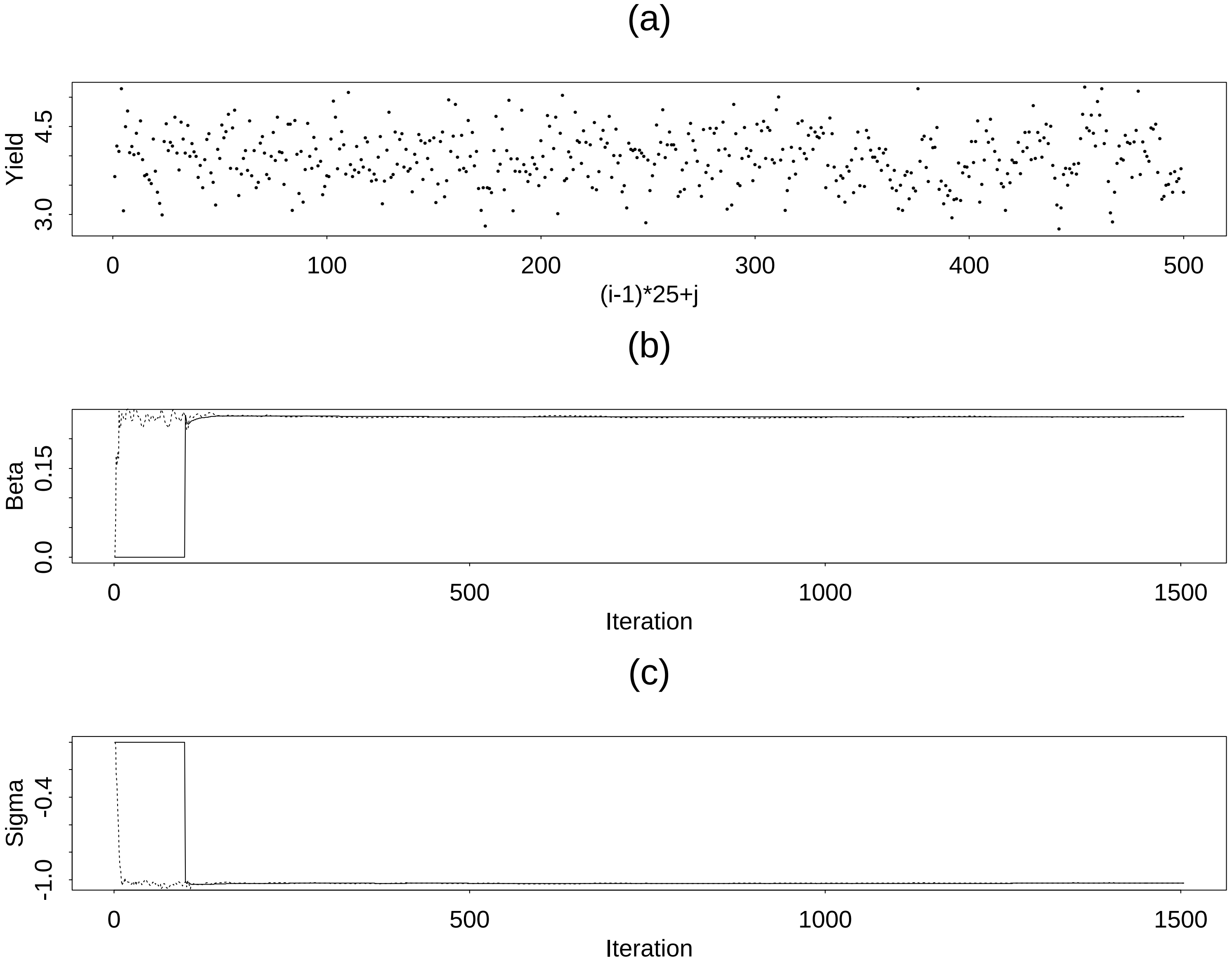

Figure 2: Mercer and Hall's wheat yield data: (a) the wheat yields against $(i-1)\times 25+j$, where (i,j) denotes a site on a 20×25 rectangle lattice; (b) β^k (- - -) and $\tilde{\beta}^k$ (—–) at each iteration of the MCMC-SA algorithm; (c) σ^{*k} (- - -) and $\tilde{\sigma}^{*k}$ (—–) at each iteration of the MCMC-SA algorithm.

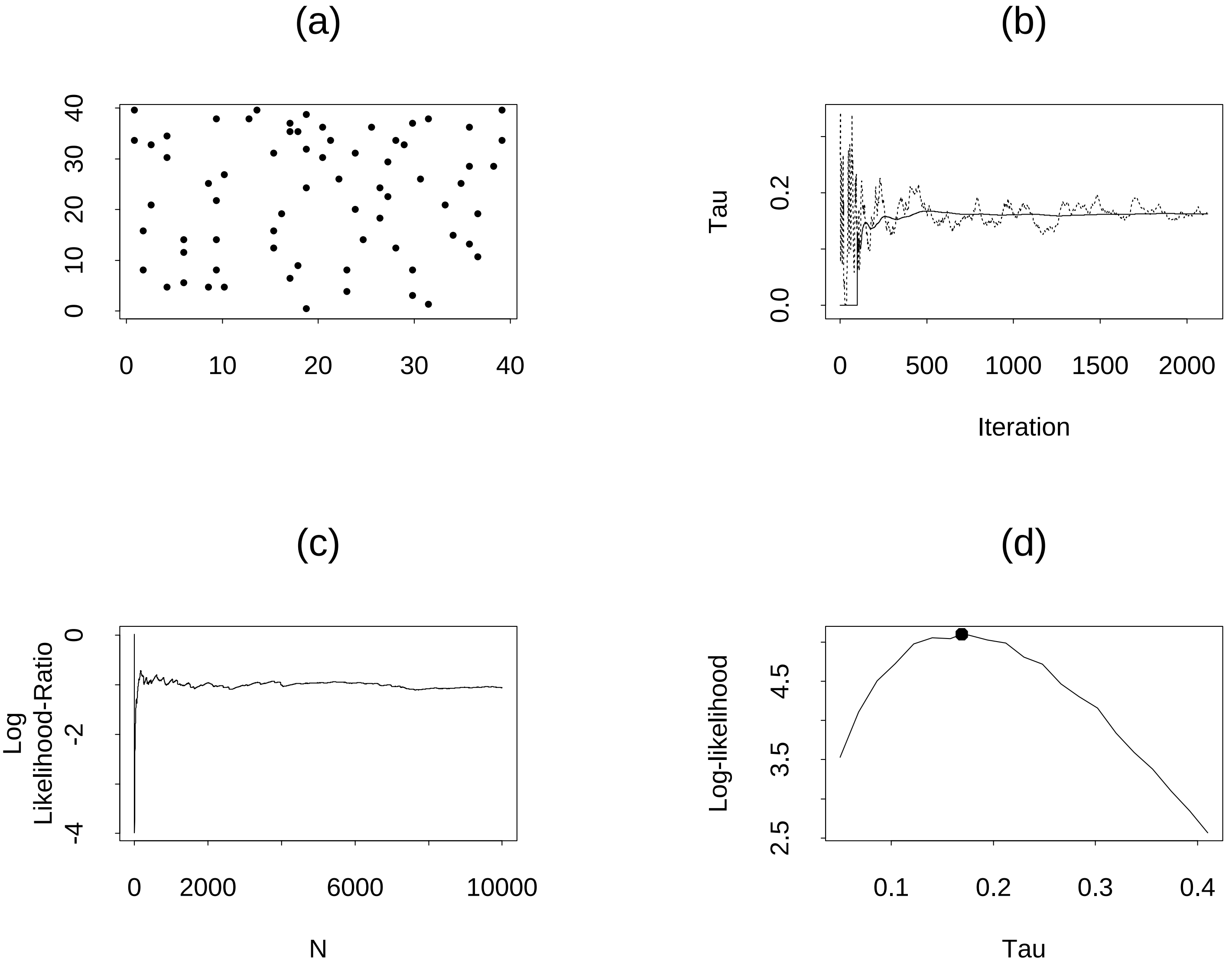

Figure 3: Spanish towns data: (a) locations of 69 Spanish towns in a 40 miles × 40 miles area; (b) τ^k (- - -) and $\tilde{\tau}^k$ (—) at each iteration of the MCMC-SA algorithm; (c) the estimates of $\log f(\mathsf{X}|\hat{\tau}_{AML}) - \log f(\mathsf{X}|\hat{\tau}_{MLE})$ against N, the number of random samples; (d) the estimates of log-likelihood values $\log f(\mathsf{X}|\tau)$ in (0, 0.4).