

Theory and Applications of Stochastic Approximation

Monte Carlo

Faming Liang

Department of Statistics
Texas A&M University

-
1. Liang, F., Liu, C. and Carroll, R.J. (2007) Stochastic approximation in Monte Carlo computation. *JASA*, **102**, 305-320.
 2. Liang, F. (2007) Annealing stochastic approximation Monte Carlo for neural network training. *Machine Learning*, **68**, 201-233.
 3. Liang, F. (2007) Continuous contour Monte Carlo for marginal density estimation with an application to a spatial statistical model, *JCGS*, **16**(3), 608-632.
 4. Liang, F. (2006) Improving SAMC using smoothing methods: theory and applications to Bayesian model selection problems. Revised for *Annals of Statistics*.
 5. Liang, F. (2007) Some asymptotics of SAMC algorithms. Manuscript.

Two motivation examples

Example 1: Consider the problem of sampling from the following mixture distribution,

$$f(x) = \frac{1}{3}N_2(\mu_1, \Sigma_1) + \frac{1}{3}N_2(\mu_2, \Sigma_2) + \frac{1}{3}N_2(\mu_3, \Sigma_3),$$

where

$$\mu_1 = \begin{pmatrix} -8 \\ -8 \end{pmatrix} \quad \Sigma_1 = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix} \quad \mu_2 = \begin{pmatrix} 6 \\ 6 \end{pmatrix} \quad \Sigma_2 = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}$$

$$\mu_3 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \Sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Example 2: Consider minimizing the following function on $[-1.1, 1.1]^2$

$$U(x, y) = -(x \sin(20y) + y \sin(20x))^2 \cosh(\sin(10x)x) \\ - (x \cos(10y) - y \sin(10x))^2 \cosh(\cos(20y)y),$$

whose global minimum is -8.12465, attained at $(x, y) = (-1.0445, -1.0084)$ and $(1.0445, -1.0084)$.

This problem can be solved by sampling from

$$f(x, y) = \begin{cases} \frac{1}{Z} \exp(-U(x, y)/t), & (x, y) \in [-1.1, 1.1]^2 \\ 0 & \text{otherwise} \end{cases}$$

where $t > 0$ is close to 0 and Z is the unknown normalizing constant.

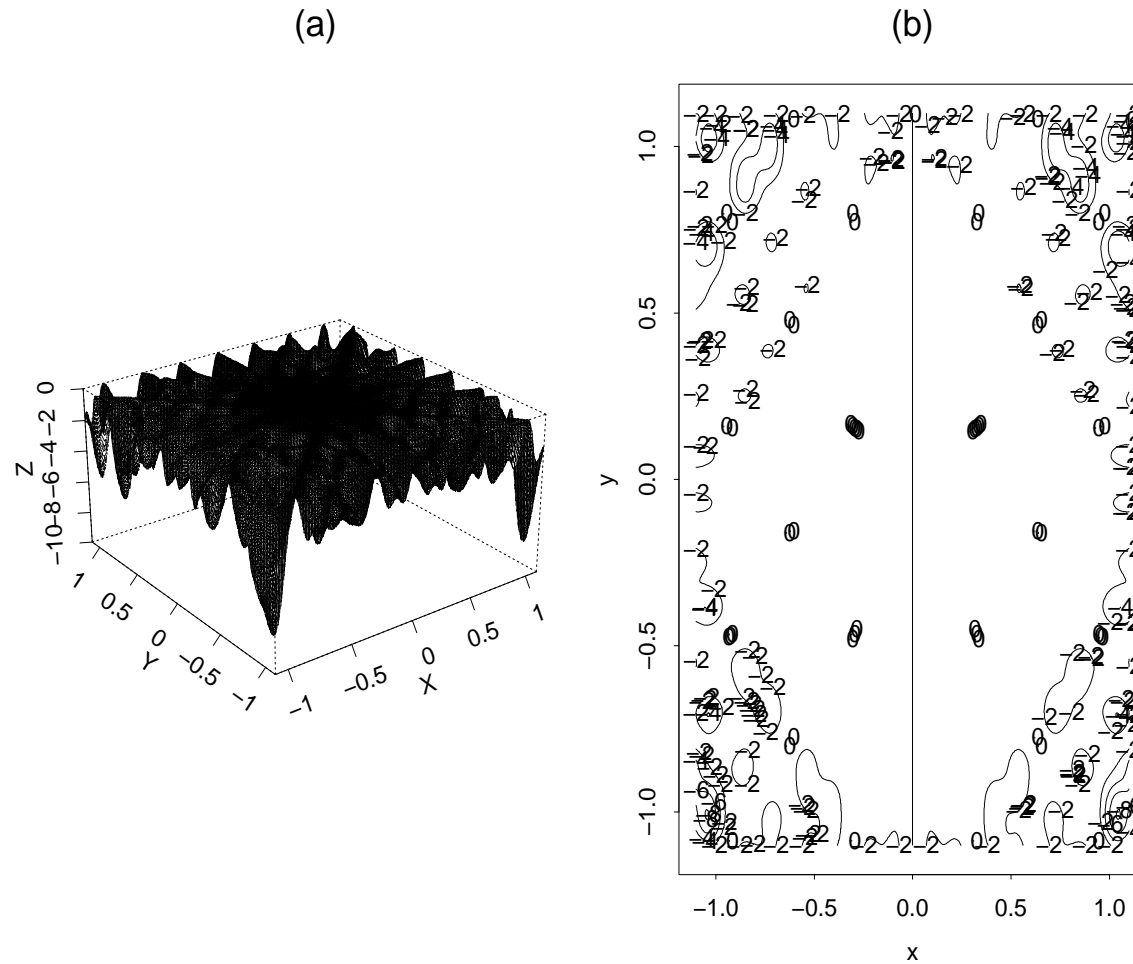


Figure 1: Grid and contour representation of a function defined on $[-1.1, 1.1]^2$.

The above examples can be formulated to simulate from a Boltzmann distribution,

$$f(x) = c\psi(x), \quad x \in \mathcal{X}, \quad (1)$$

where c is a constant, $\psi(x) = \exp(-U(x)/t)$, t is called the temperature, and $U(x)$ is called the energy function.

Two basic MCMC algorithms

- (1) Metropolis-Hastings algorithm (Metropolis et al, 1953; Hastings, 1970)
- (2) The Gibbs sampler. (Geman and Geman, 1984).

Metropolis-Hastings Algorithm

- (a) Propose a new state y from a proposal distribution $T(x_k \rightarrow y)$, where x_k denotes the state of the Markov chain at iteration k .
- (b) Accept y with the probability

$$\min\left\{\frac{\psi(y)T(y \rightarrow x_k)}{\psi(x_k)T(x_k \rightarrow y)}, 1\right\}.$$

If it is accepted, set $x_{k+1} = y$, otherwise, set $x_{k+1} = x_k$.

Some difficult scientific computing Problems

1. Neural Networks.
2. Combinatorial Optimization, e.g., traveling salesman problem.
3. Spin-glasses simulations.
4. Protein folding.

Difficulty

On the energy landscape of these systems, there are a multitude of local minima separated by high energy barriers. The conventional MCMC samplers tend to get trapped in one of local energy minima indefinitely, rendering the simulation ineffective.

Strategies for improving MCMC

1. Use of auxiliary variables:

- Swendsen-Wang algorithm (Swendsen and Wang, 1987)
- Parallel tempering (Geyer, 1991)
- Simulated tempering (Marinari and Parisi, 1992)
- Evolutionary Monte Carlo (Liang and Wong, 2001)

Strength and weakness: The temperature is typically treated as an auxiliary variable. Simulations at high temperatures broaden the space of sampling, and thus are able to help the sampler to escape from local energy minima.

2. Use of past samples:

- Multicanonical sampling (Berg and Neuhaus, 1991)
- $1/k$ -ensemble algorithm (Hesselbo and Stinchcombe, 1995)
- Wang-Landau (WL) Algorithm (Wang and Landau, 2001)
- Generalized Wang-Landau (GWL) Algorithm (Liang, 2005)
- Dynamic weighting (Wong and Liang, 1997)
- Dynamically weighted importance sampling (Liang, 2002)

Strength and weakness:

- **Dynamic weighting**: The variability of the weights is too high.
- **Multicanonical and related algorithms**: They are usually used for discrete systems. In the multicanonical algorithm, the trial distribution is defined as:

$$p^*(x) = \frac{1}{\#\{y : U(y) = U(x)\}},$$

where x and y take values on a discrete set.

There is no rigorous theory to support their convergence.

Basic Idea

- Partition the sample space into different subregions: E_1, \dots, E_m , $\bigcup_{i=1}^m E_i = \mathcal{X}$, and $E_i \cap E_j = \emptyset$ for $i \neq j$.
- Set $g_i = \int_{E_i} \psi(x) dx$, and choose $\pi = (\pi_1, \dots, \pi_m)$, $\pi_i \geq 0$, and $\sum_i \pi_i = 1$.
- Sampling from the distribution

$$p_\theta(x) \propto \sum_{i=1}^m \frac{\pi_i \psi(x)}{e^{\theta^{(i)}}} I(x \in E_i).$$

If $\theta^{(i)} = \log(g_i)$ for all i , sampling from $p_\theta(x)$ will result in a random walk in the space of subregions with each subregion being sampled with probability π_i (viewing each subregion as a “single point”). Therefore, sampling from $p_\theta(x)$ can avoid the local trap problem suffered by conventional MCMC samplers.

Algorithm Setting

- *Condition* (A_1): The sequence $\{a_k\}_{k=0}^{\infty}$ is non-increasing, positive and satisfies the conditions:

$$\sum_{k=1}^{\infty} a_k = \infty, \quad \lim_{k \rightarrow \infty} (ka_k) = \infty, \quad \lim_{k \rightarrow \infty} (a_{k+1}^{-1} - a_k^{-1}) = 0, \quad (2)$$

and for some $\tau \in (0, 1)$

$$\sum_{k=1}^{\infty} \frac{a_k^{(1+\tau)/2}}{\sqrt{k}} < \infty \quad (3)$$

It is clear that $a_k = 1/k^\eta$, $\forall \eta \in (1/2, 1)$, satisfies (2). Then (3) holds for any $\tau > 1/\eta - 1$.

- Let $\pi = (\pi_1, \dots, \pi_m)$ be an m -vector with $0 < \pi_i < 1$ and $\sum \pi_i = 1$. The π is called the desired sampling distribution.
- Define $H(\theta_k, x_{k+1}) = e_{x_{k+1}} - \pi$, where $e_{x_{k+1}} = (I(x_{k+1} \in E_1), \dots, I(x_{k+1} \in E_m))$.

Algorithm

1. (Sampling) Draw sample x_{k+1} by a MH iteration with the invariant distribution

$$\hat{p}_k(x) = \frac{1}{Z_k} \sum_{i=1}^m \frac{\psi(x)}{e^{\theta_k^{(i)}}} I(x \in E_i).$$

2. (Weight updating) Set

$$\theta_{k+\frac{1}{2}} = \theta_k + a_{k+1} H(\theta_k, x_{k+1}).$$

3. (Varying truncation) If $\theta_{k+\frac{1}{2}} \in \Theta$, set $\theta_{k+1} = \theta_{k+\frac{1}{2}}$. Otherwise, set $\theta_{k+1} = \theta_{k+\frac{1}{2}} + c^*$, where c^* is chosen such that $\theta_{k+\frac{1}{2}} + c^* \in \Theta$.

Lyapunov condition on $h(\theta)$

Let $\langle x, y \rangle$ denote the inner product of the vectors x and y .

(A₂) *The function $h : \Theta \rightarrow \mathbb{R}^d$ is continuous, and there exists a continuously differentiable function $v : \Theta \rightarrow [0, \infty)$ such that*

(i) *For any integer $M > 0$, the level set $\mathcal{V}_M = \{\theta \in \Theta, v(\theta) \leq M\} \subset \Theta$ is compact.*

(ii) *There exists $M_0 > 0$ such that*

$$\tilde{\Theta} = \{\theta \in \Theta, \langle \nabla v(\theta), h(\theta) \rangle = 0\} \subset \text{int}(\mathcal{V}_{M_0}),$$

and $\langle \nabla v(\theta), h(\theta) \rangle < 0$ for any $\theta \in \Theta \setminus \mathcal{V}_{M_0}$, where $\text{int}(A)$ denotes the interior of set A .

(iii) *For all $\theta \in \Theta$, $\langle \nabla v(\theta), h(\theta) \rangle \leq 0$, and $\text{int}(v(\tilde{\Theta})) = \emptyset$.*

Stability condition on $h(\theta)$

(A_3) *The mean field function $h(\theta)$ is measurable and locally bounded. There exist a stable matrix F (i.e., all eigenvalues of F are with negative real parts), $\gamma > 0$, and $\rho \in (\tau, 1]$ such that*

$$\|h(\theta) - F(\theta - \theta^0)\| \leq c_1 \|\theta - \theta^0\|^{1+\rho}, \quad \forall \theta \in \{\theta : \|\theta - \theta^0\| \leq \gamma\},$$

where c_1 is a constant.

Drift condition

Condition (A₄): For any $\theta \in \Theta$, the transition kernel P_θ is irreducible and aperiodic. In addition, there exists a function $V : \mathcal{X}^\kappa \rightarrow [1, \infty)$ and constants $\alpha \geq 2$ and $\beta \in (0, 1]$ such that for any compact subset $\mathcal{K} \subset \Theta$,

(i) There exist a set $\mathbf{C} \subset \mathcal{X}$, an integer l , constants $0 < \lambda < 1$, $b, \varsigma, \delta > 0$ and a probability measure ν such that

$$\bullet \quad \sup_{\theta \in \mathcal{K}} P_\theta^l V^\alpha(x) \leq \lambda V^\alpha(x) + bI(x \in \mathbf{C}), \quad \forall x \in \mathcal{X} \quad (4)$$

$$\bullet \quad \sup_{\theta \in \mathcal{K}} P_\theta V^\alpha(x) \leq \varsigma V^\alpha(x), \quad \forall x \in \mathcal{X}. \quad (5)$$

$$\bullet \quad \sup_{\theta \in \mathcal{K}} P_\theta^l(x, A) \geq \delta \nu(A), \quad \forall x \in \mathbf{C}, \forall A \in \mathcal{B}. \quad (6)$$

(ii) *There exists a constant c such that for all $x \in \mathcal{X}$,*

- $\sup_{\theta \in \mathcal{K}} \|H(\theta, x)\| \leq cV(x). \quad (7)$
- $\sup_{(\theta, \theta') \in \mathcal{K}} \|H(\theta, x) - H(\theta', x)\| \leq cV(x)\|\theta - \theta'\|^\beta \quad (8)$

(iii) *There exists a constant c such that for all $(\theta, \theta') \in \mathcal{K} \times \mathcal{K}$,*

- $\|P_\theta g - P_{\theta'} g\|_V \leq c_2 \|g\|_V |\theta - \theta'|^\beta, \quad \forall g \in \mathcal{L}_V. \quad (9)$
- $\|P_\theta g - P_{\theta'} g\|_{V^\alpha} \leq c_2 \|g\|_{V^\alpha} |\theta - \theta'|^\beta, \quad \forall g \in \mathcal{L}(V^\alpha) \quad (10)$

Theoretical Results

Lemma 1 *Assume the drift condition (A_4) and $\sup_{x \in \mathcal{X}} V(x) < \infty$. Let $\epsilon_k = H(\theta_k, x_{k+1}) - h(\theta_k)$. There exist \mathbb{R}^d -valued random processes $\{e_k\}_{k \geq 1}$, $\{\nu_k\}_{k \geq 1}$, and $\{s_k\}_{k \geq 1}$ defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ such that*

(i) $\epsilon_k = e_k + \nu_k + s_k$.

(ii) $\{e_k\}$ is a martingale difference sequence, and $\frac{1}{\sqrt{n}} \sum_{k=1}^n e_k \longrightarrow N(0, Q)$ in distribution, where $Q = \lim_{k \rightarrow \infty} E(e_k e_k')$.

(iii) $E\|\nu_k\| = O(a_k^{(1+\tau)/2})$, where τ is given in condition (A_1) .

(iv) $\|\sum_{k=0}^n a_k s_k\| = O(a_n)$.

THEOREM 1 (*Convergence*)

Let α_σ denote the number of iterations for which the σ -th truncation occurs in the SAMC simulation. Assume the conditions (A_1) and (A_2) hold, and there exists a drift function $V(x)$ such that $\sup_{x \in \mathcal{X}} V(x) < \infty$ and the drift condition (A_4) holds. Then there exists a number σ such that $\alpha_\sigma < \infty$ a.s., $\alpha_{\sigma+1} = \infty$ a.s., and $\{\theta_k\}$ given by the SAMC algorithm has no truncation for $k \geq \alpha_\sigma$, i.e.,

$$\theta_{k+1} = \theta_k + a_k H(\theta_k, x_{k+1}), \quad \forall k \geq \alpha_\sigma,$$

and

$$\theta_k^{(i)} \rightarrow \begin{cases} c + \log(\int_{E_i} \psi(x) dx) - \log(\pi_i + \nu), & \text{if } E_i \neq \emptyset, \\ -\infty. & \text{if } E_i = \emptyset, \end{cases} \quad (11)$$

where $\nu = \sum_{j \in \{i: E_i = \emptyset\}} \pi_j / (m - m_0)$ and m_0 is the number of empty subregions. The constant c can be determined by imposing an extra constraint on θ_k , e.g., $\theta_k^{(m)} = 0$ for all $k \geq 0$.

THEOREM 2 (*Normality*)

Suppose that all subregions are nonempty, and a constraint, e.g., $\theta_k^{(m)} = 0$ for all $k \geq 0$, has been imposed on θ_k such that the constant c in Theorem 1 is uniquely determined. Assume the conditions (A_1) , (A_2) , (A_3) and (A_4) hold. Then

$$\frac{\theta_k - \theta^0}{\sqrt{a_k}} \rightarrow N(0, S) \quad \text{as } k \rightarrow \infty,$$

where S is defined as

$$S = \lim_{k \rightarrow \infty} \sum_{j=1}^k E(v_{k,j+1} v'_{k,j+1} | \mathcal{G}_{k,j}),$$

where $v_{k,j+1} = \left[\prod_{i=j+1}^k (I + a_i F) \right] a_j e_{j+1} / \sqrt{a_{k+1}}$.

THEOREM 3 (*Averaging Normality*)

Under the conditions of Theorem 2, we have

$$\bar{\theta}_k = \frac{1}{k} \sum_{i=1}^k \theta_i$$

is asymptotically efficient; that is,

$$\sqrt{k}(\bar{\theta}_k - \theta^0) \longrightarrow N(\mathbf{0}, S) \quad \text{as } k \rightarrow \infty,$$

where $S = F^{-1}Q(F^{-1})^T$, *and* Q *is as defined in Lemma 1.*

Implementation Issues

1. **Sample space partition.** *It can be made according to our goal and the complexity of the problem. Here are some examples:*

(a) *Importance sampling: Energy function, maximum energy difference ≤ 2 .*

(b) *Model selection: Model index.*

2. **Desired sampling distribution.**

(a) *Set π to be uniform if we aim to estimate g_i 's.*

(b) *Set π to bias the sampling to low energy regions if we aim to minimize the energy function.*

3. **Choices of η , t_0 and the number of iterations.** *The diagnostic statistic:*

$$\epsilon_f(E_i) = \begin{cases} \frac{\hat{\pi}_i - (\pi_i + \nu)}{\pi_i + \nu} \times 100\%, & \text{if } E_i \neq \emptyset, \\ 0, & \text{if } E_i = \emptyset, \end{cases} \quad (12)$$

for $i = 1, \dots, m$. If $\max_{i=1}^m |\epsilon_f(E_i)|$ is large, say, greater than 10%, the convergence of the run should be questioned. In this case, SAMC should be re-run with more iterations, a larger value of t_0 , or a smaller value of η .

x	1	2	3	4	5	6	7	8	9	10
$\pi(x)$	0.05	0.05	0.05	0.05	0.1	0.1	0.1	0.15	0.15	0.2

Table 1: The mass function of the 10-state distribution.

$$T = \begin{pmatrix} .379 & .009 & .059 & .225 & .015 & .078 & .038 & .059 & .060 & .078 \\ .302 & .122 & .109 & .067 & .161 & .004 & .034 & .055 & .067 & .079 \\ .016 & .114 & .147 & .030 & .026 & .092 & .129 & .217 & .121 & .108 \\ .053 & .088 & .175 & .035 & .105 & .123 & .105 & .088 & .140 & .088 \\ .046 & .024 & .029 & .009 & .026 & .080 & .125 & .067 & .322 & .272 \\ .107 & .088 & .104 & .130 & .077 & .104 & .102 & .120 & .065 & .103 \\ .143 & .143 & .143 & .000 & .071 & .143 & .000 & .143 & .143 & .071 \\ .060 & .075 & .086 & .055 & .123 & .092 & .142 & .099 & .115 & .153 \\ .173 & .085 & .005 & .183 & .007 & .081 & .040 & .155 & .163 & .108 \\ .068 & .058 & .125 & .096 & .115 & .135 & .096 & .096 & .096 & .115 \end{pmatrix}$$

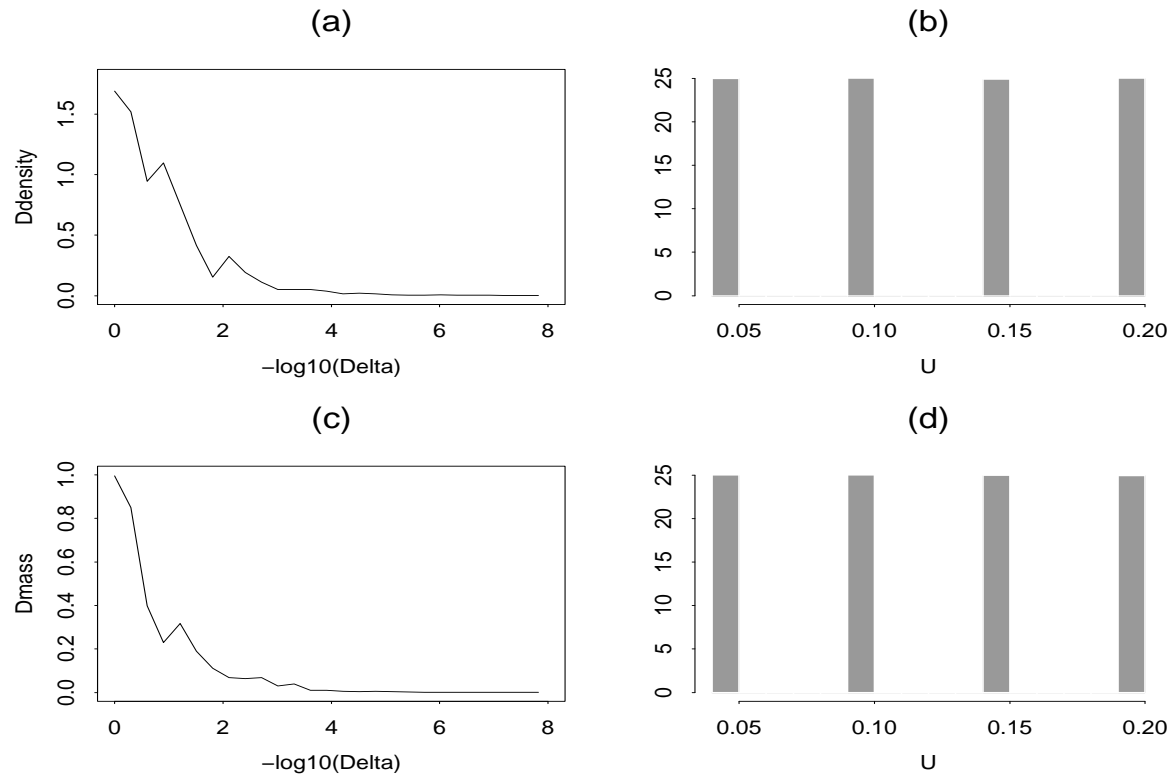


Figure 2: Plots (a) & (b) are for the runs with $\psi(\mathbf{x}) = 1$. (a) Convergence of \hat{g} , where $D_{\text{density}} = \sqrt{\sum_{i=1}^4 (\hat{g}_i - g_i)^2 / g_i}$ with $g = (4, 3, 2, 1)$; (b) Histogram of the samples in the space of subregions. Plots (c) & (d) are for the runs with $\psi(\mathbf{x}) = \pi(\mathbf{x})$. (c) Convergence of \hat{g} , where $D_{\text{mass}} = \sqrt{\sum_{i=1}^4 (\hat{g}_i - g_i)^2 / g_i}$ with $g = (0.2, 0.3, 0.3, 0.2)$. (d) Histogram of the samples in the space of subregions.

The problem *The target distribution $f(x)$ has a rugged energy landscape and is difficult to simulate from using conventional Monte Carlo algorithms.*

Let x_1, \dots, x_n denote samples drawn from a trial density $p^(x)$, and let w_1, \dots, w_n denote the associated importance weights, where $w_i = f(x_i)/p^*(x_i)$ for $i = 1, \dots, n$. The quantity $E_f h(x)$ can then be estimated by*

$$\widehat{E_f h(x)} = \frac{\sum_{i=1}^n h(x_i) w_i}{\sum_{i=1}^n w_i}.$$

Conditions for a good trial density function

- (a) *The resulting importance weights are bounded, that is, there exist a number M such that $f(x)/p^*(x) < M$ for all $x \in \mathcal{X}$.*
- (b) *$p^*(x)$ can be easily simulated from with conventional Monte Carlo algorithms, say, MH.*

The defensive mixture method (Hesterberg, 1995) suggests the following trial density

$$p^*(x) = \lambda f(x) + (1 - \lambda)p_0(x), \quad (13)$$

where $0 < \lambda < 1$ and $p_0(x)$ is another trial density.

SAMC trial density

$$\hat{p}_\infty(x) = \frac{1}{Z_\infty} \sum_{i=1}^m \frac{\exp\{-U(x)/\tau\}}{\hat{g}_i} I(x \in E_i), \quad (14)$$

where the sample space is partitioned according to the energy function, and the maximum energy difference in each subregion is bounded by a reasonable number, say, 2.

Advantages

- *The importance weights are bounded.*

$$\max \hat{g}_i < \int_{\mathcal{X}} \exp\{-U(x)/\tau\} dx < \infty.$$

- *Sampling from $\hat{p}_{\infty}(x)$ will lead to a “random walk” in the space of subregions (if each subregion is regarded as a “point”) with each subregion being sampled with a desired frequency.*
- *Sampling on-line. Draw a sample from $\hat{p}_{\infty}(x)$ and retain it with probability $[\sum_{j=1}^m \hat{g}_j I(x \in E_j)] / [\max_{j=1}^m \hat{g}_j]$.*

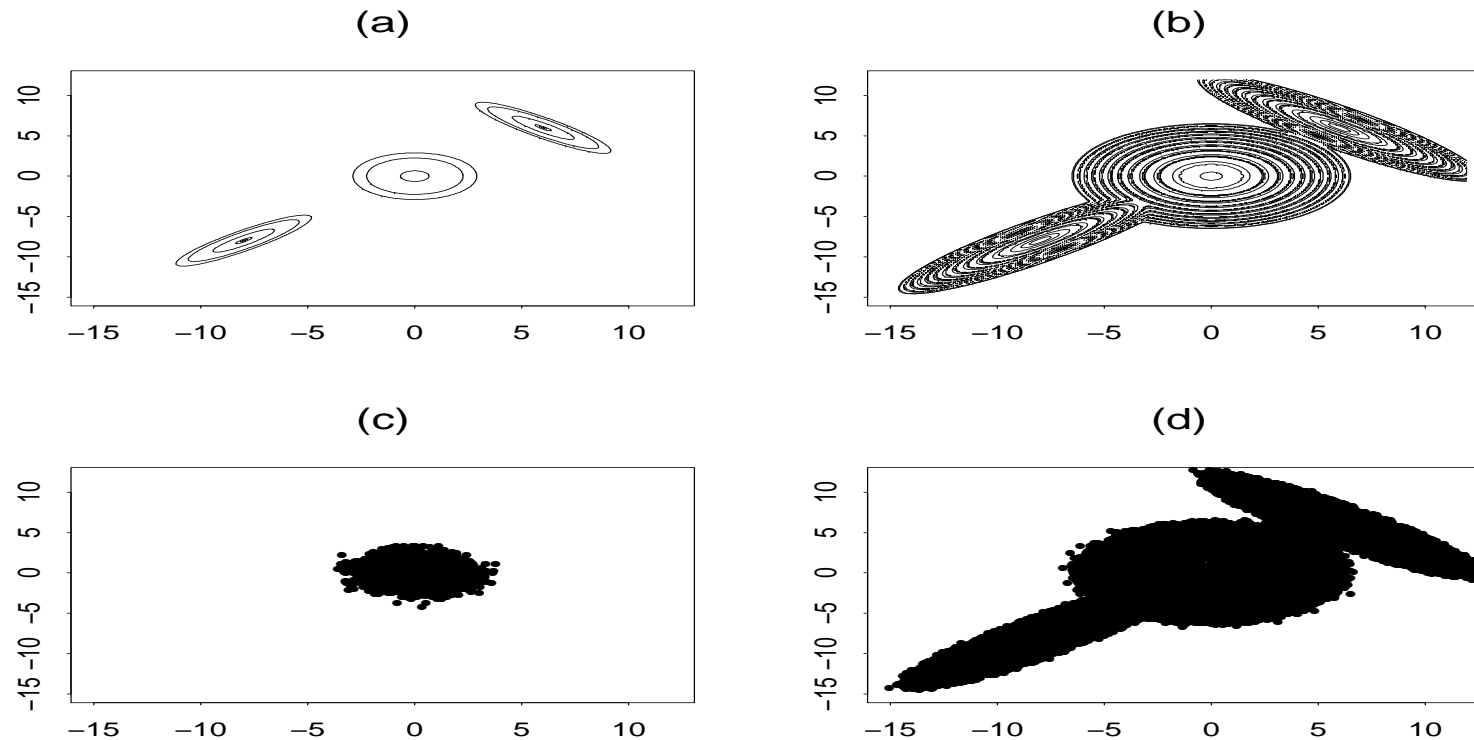


Figure 3: Computational results for the mixture Gaussian example. (a) and (b) show the contour plots of the true and trial densities, respectively. The contour lines correspond to 99%, 95%, 50%, 5% and 1% of the total mass. (c) and (d) show the scatter plots of the samples drawn by the MH algorithm from the true and trial densities.

Let $\mathbf{s} = \{s_i : i \in D\}$ denote the observed binary data, where s_i is called a spin and D is the set of indices of the spins. Let $|D|$ denote the total number of spins in D , and $N(i)$ denote a set of “neighbors” of spin i . The likelihood function of the model is

$$f(\mathbf{s}|\alpha, \beta) = \frac{1}{\varphi(\alpha, \beta)} \exp \left\{ \alpha \sum_{i \in D} s_i + \frac{\beta}{2} \sum_{i \in D} s_i \left(\sum_{j \in N(i)} s_j \right) \right\}, \quad (15)$$

where $(\alpha, \beta) \in \Omega$, and

$$\varphi(\alpha, \beta) = \sum_{\text{for all possible } \mathbf{s}} \exp \left\{ \alpha \sum_{j \in D} s_j + \frac{\beta}{2} \sum_{i \in D} s_i \left(\sum_{j \in N(i)} s_j \right) \right\}.$$

When β is large, say, 0.5, the configuration \mathbf{s} tends to have large clusters of the same orientation, which fluctuate very slowly.

Methods to resolve the difficulty in normalizing constant evaluation:

- Working on a pseudo-likelihood function (Besag, 1975):

$$PL(\alpha, \beta | \mathbf{s}) = \prod_{i \in D} \frac{e^{s_i(\alpha + \beta \sum_{j \in N(i)} s_j)}}{e^{\alpha + \beta \sum_{j \in N(i)} s_j} + e^{-\alpha - \beta \sum_{j \in N(i)} s_j}}. \quad (16)$$

The resulting estimate is called MPLE.

- Working on a Monte Carlo log-likelihood (up to a constant)(Geyer and Thompson, 1992):

$$L_n(\alpha, \beta | \mathbf{s}) = \alpha \sum_{i \in D} s_i + \frac{\beta}{2} \sum_{i \in D} s_i \left(\sum_{j \in N(i)} s_j \right) - \log \left[\frac{1}{n} \sum_{k=1}^n \frac{\psi(\alpha, \beta, \mathbf{s}^{(k)})}{\psi(\alpha^*, \beta^*, \mathbf{s}^{(k)})} \right]. \quad (17)$$

The resulting estimate is called MCMLE.

A natural choice for the trial distribution is a mixture distribution of the form

$$p_{mix}^*(\mathbf{s}) = \frac{1}{m^*} \sum_{j=1}^{m^*} p(\mathbf{s} | \alpha_j, \beta_j), \quad (18)$$

where the values of the parameters $(\alpha_1, \beta_1), \dots, (\alpha_{m^}, \beta_{m^*})$ are pre-specified. To complete this idea, the key is to estimate $\varphi(\alpha_j, \beta_j), \dots, \varphi(\alpha_{m^*}, \beta_{m^*})$ (up to a common multiplicative constant).*

Estimate	single-MCMLE	SAMC
$\text{RMSE}(T_1^{sim})$	59.51	2.90
$\text{RMSE}(T_2^{sim})$	114.91	4.61

Table 2: Comparison of the accuracy of the SAMC and single-MCMLEs for the US cancer data. $T_1 = \sum_i s_i$, $T_2 = \sum_i s_i (\sum_j s_j) / 2$, $\text{RMSE}(T_i^{sim})$ is calculated as $\sqrt{\sum_{k=1}^5 (T_i^{sim,k} - T_i^{obs})^2 / 5}$, where $i = 1, 2$, and $T_i^{sim,k}$ denotes the value of T_i^{sim} calculated based on the k^{th} estimate of (α, β) .

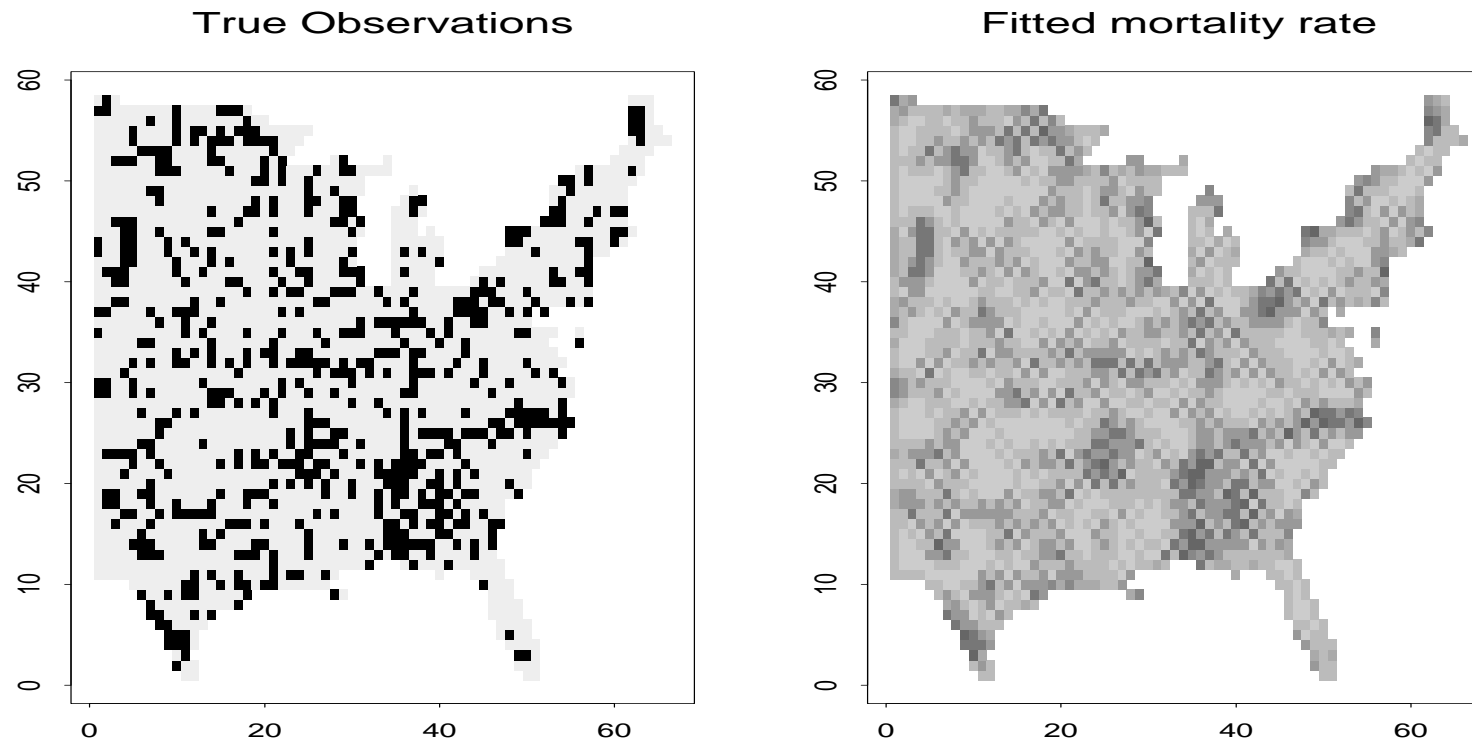


Figure 4: The U.S. cancer mortality rate data. (a) The mortality map of liver and gallbladder cancer (including bile ducts) for white males during the decade 1950-1959. The black squares denote the counties of high cancer mortality rate, and the white squares denote the counties of low cancer mortality rate. (b) Fitted cancer mortality rates. The cancer mortality rate of each county is represented by the gray level of the corresponding square.

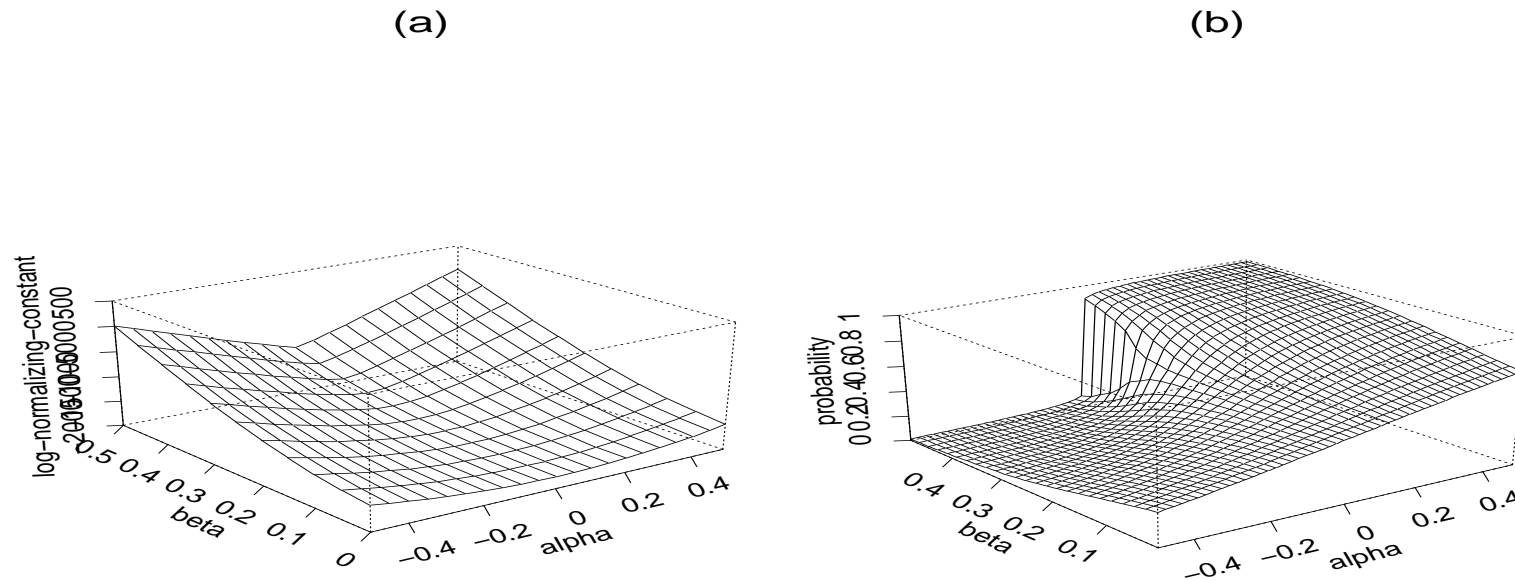


Figure 5: Computational results of SAMC. (a) Estimate of $\log \varphi(\alpha, \beta)$ on a 21×11 lattice with $\alpha \in \{-0.5, -0.45, \dots, 0.5\}$ and $\beta \in \{0, 0.05, \dots, 0.5\}$. (b) Estimate of $P(s_i = +1 | \alpha, \beta)$ on a 50×25 lattice with $\alpha \in \{-0.49, -0.47, \dots, 0.49\}$ and $\beta \in \{0.01, 0.03, \dots, 0.49\}$.

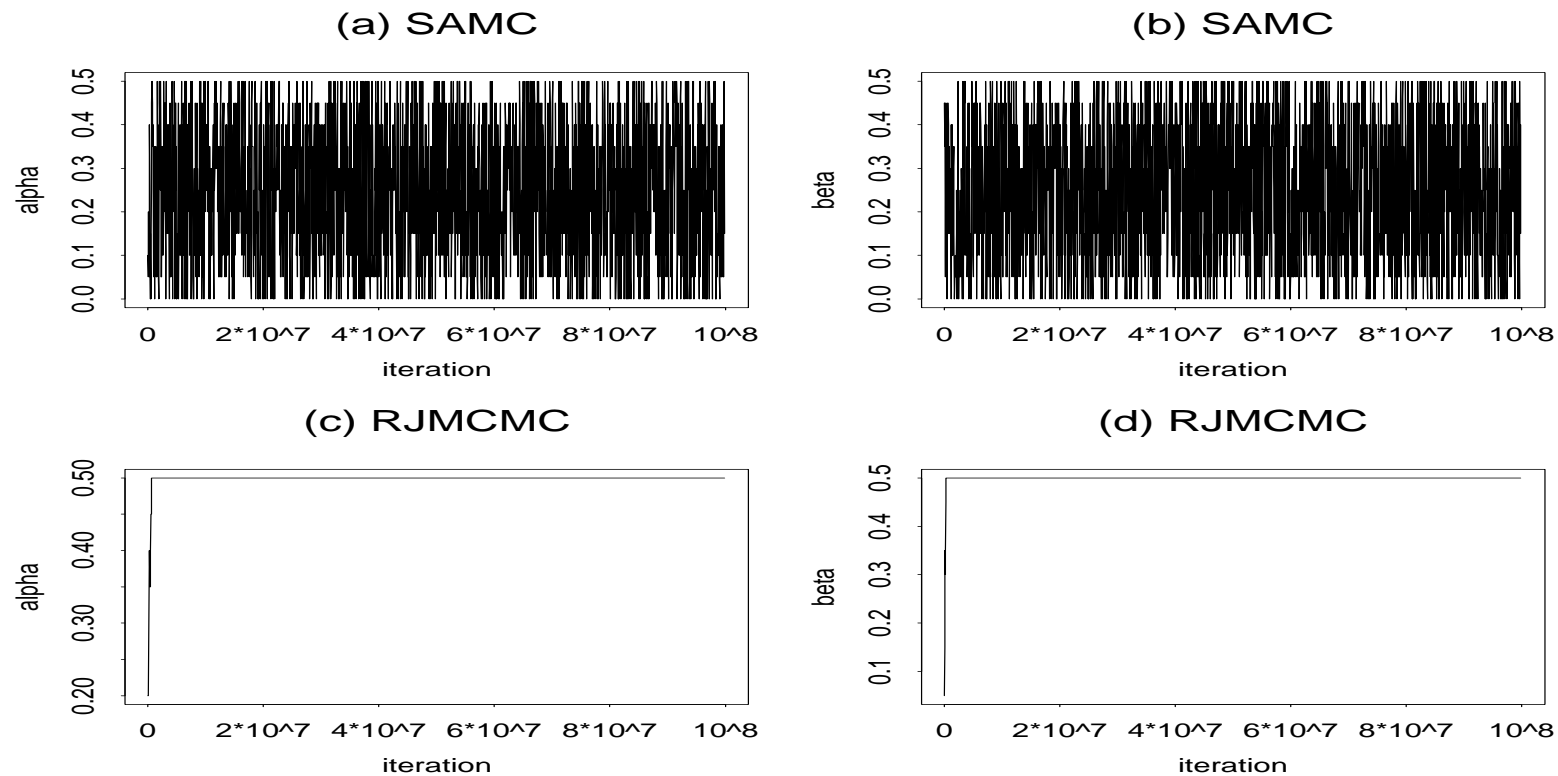


Figure 6: Comparison of SAMC and RJMCMC. Plots (a) and (b) show, respectively, the sample paths of α and β in a run of SAMC. Plots (c) and (d) show, respectively, the sample paths of α and β in a run of RJMCMC.

Motivation for Smoothing SAMC

Intuitively, x_t may contain some information on its neighboring subregions, so the visiting to its neighboring subregions should also be penalized to some extent in the next iteration.

The efficiency of SAMC can be improved by including at each iteration a smoothing step, which distributes the information of each sample to its neighboring subregions. The new algorithm is thus called smoothing-SAMC or SSAMC for simplicity.

Motivation Examples

We note that for many problems, E_1, \dots, E_m can be regarded as a sequence of naturally ordered categories. Here are some examples.

- *Model selection: The model space \mathcal{X} can be partitioned according to the index of models, and the subregions can be naturally ordered according to the number of parameters contained in each model.*
- *Function optimization: The solution space \mathcal{X} can be partitioned according to the objective function, and the subregions can also be naturally ordered according to the objective function.*

Suppose that $x_k^{(1)}, \dots, x_k^{(\kappa)}$ are samples generated using a MH kernel with the invariant distribution $p_{\theta_k}(x)$. Since κ is usually a small number, say, 10 to 20, the samples form a sparse frequency vector $\mathbf{e}_{x_k} = (e_k^{(i)}, \dots, e_k^{(m)})$ with $e_k^{(i)} = \sum_{l=1}^{\kappa} I(x_k^{(l)} \in E_i)$.

The frequency estimate can be improved by a smoothing method. The Nadaraya-Watson kernel estimator works as follows:

$$\hat{p}_k^{(i)} = \frac{\sum_{j=1}^m W\left(\frac{\Lambda(i-j)}{mh_k}\right) \frac{e_k^{(j)}}{\kappa}}{\sum_{j=1}^m W\left(\frac{\Lambda(i-j)}{mh_k}\right)}, \quad (19)$$

where $W(z)$ is a kernel function with bandwidth h_k , and Λ is a rough estimate of the range of $\lambda(x)$, $x \in \mathcal{X}$.

By assuming that $W(z)$ has a bounded support, we can show

$$\hat{p}_k^{(i)} - e_k^{(i)} / \kappa = O(h_k).$$

- (a) **(Sampling)** Simulate samples $x_k^{(1)}, \dots, x_k^{(\kappa)}$ using the MH algorithm with the proposal distribution $q(x_{k^{(i)}}^{(i)}, \cdot)$ and the invariant distribution $p_{\theta_k}(x)$, where $x_k^{(0)} = x_{k-1}^{(\kappa)}$.
- (b) **(Smoothing)** Calculate $\hat{p}_k = (\hat{p}_k^{(i)}, \dots, \hat{p}_k^{(m)})$ using a kernel smoothing method.
- (c) **(Weight updating)** Set

$$\theta_{k+\frac{1}{2}} = \theta_k + a_{k+1}(\hat{p}_k - \pi). \quad (20)$$

If $\theta_{k+\frac{1}{2}} \in \Theta$, set $\theta_{k+1} = \theta_{k+\frac{1}{2}}$; otherwise, set $\theta_{k+1} = \theta_{k+\frac{1}{2}} + c^*$, where c^* is chosen such that $\theta_{k+\frac{1}{2}} + c^* \in \Theta$.

Notations

- Let $Z = (z_1, z_2, \dots, z_n)$ denote a sequence of independent observations.
- Let $\vartheta^{(k)}$ denote a configuration of ϑ with k ones, which represents a model of k change points.
- Let $\eta^{(k)} = (\vartheta^{(k)}, \mu_1, \sigma_1^2, \dots, \mu_{k+1}, \sigma_{k+1}^2)$.
- Let \mathcal{X}_k denote the space of models with k change points, $\vartheta^{(k)} \in \mathcal{X}_k$, and $\mathcal{X} = \cup_{k=0}^n \mathcal{X}_k$.

By assuming appropriate prior distributions, integrating out the parameters $\mu_1, \sigma_1^2, \dots, \mu_{k+1}, \sigma_{k+1}^2$ from the full posterior distribution, and taking a logarithm, we have

$$\begin{aligned} \log P(\boldsymbol{\vartheta}^{(k)} | Z) &= a_k + \frac{k+1}{2} \log 2\pi \\ &- \sum_{i=1}^{k+1} \left\{ \frac{1}{2} \log(c_i - c_{i-1}) - \log \Gamma\left(\frac{c_i - c_{i-1} - 1}{2} + \alpha\right) \right. \\ &\left. + \left(\frac{c_i - c_{i-1} - 1}{2} + \alpha\right) \log \left[\beta + \frac{1}{2} \sum_{j=c_{i-1}+1}^{c_i} z_j^2 - \frac{\left(\sum_{j=c_{i-1}+1}^{c_i} z_j\right)^2}{2(c_i - c_{i-1})} \right] \right\}. \end{aligned} \tag{21}$$

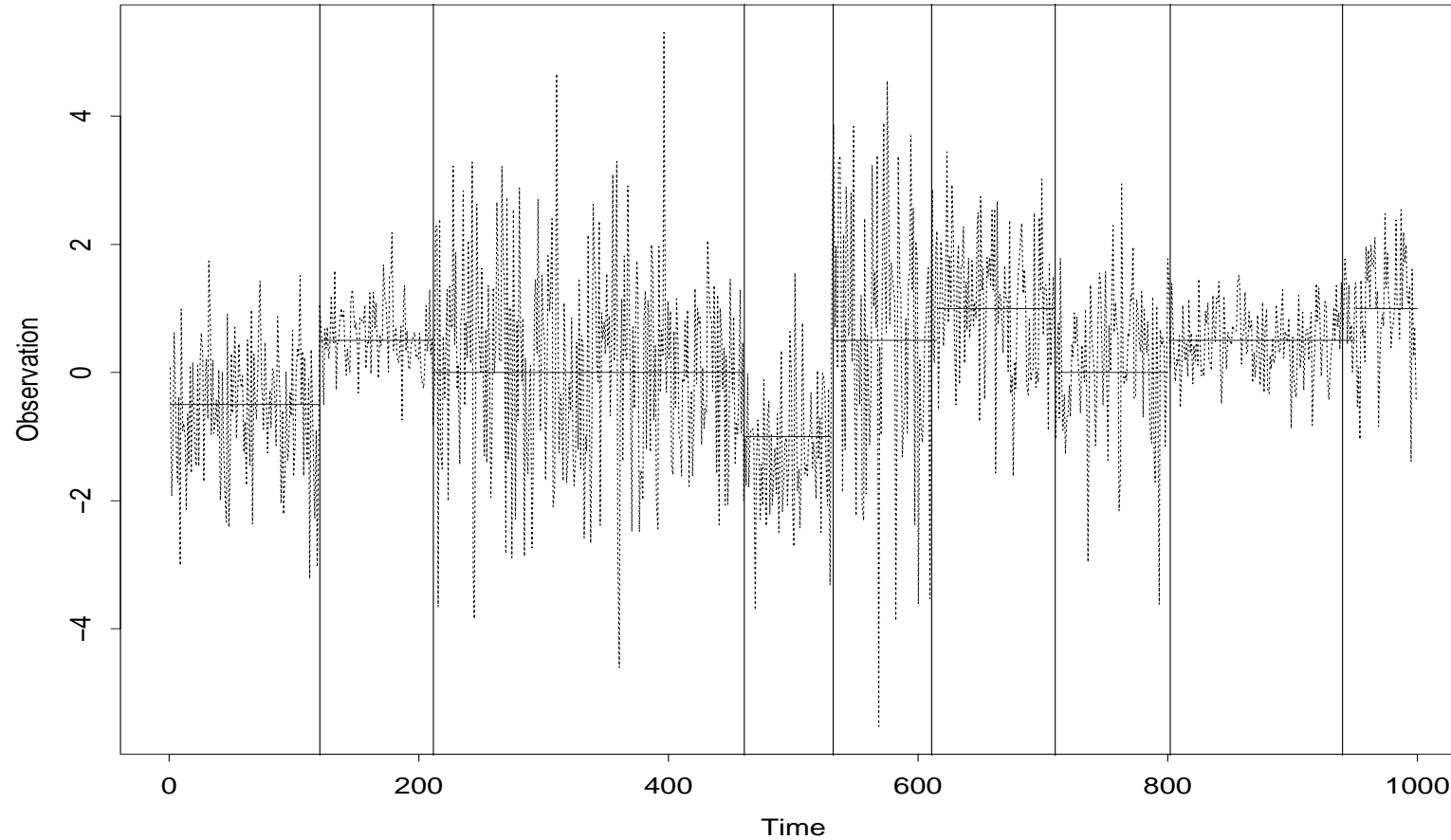


Figure 7: Comparison of the true change-point pattern (horizontal lines) and its MAP estimate (vertical lines).

k	SSAMC		SAMC		RJMCMC	
	prob(%)	SD	prob(%)	SD	prob(%)	SD
7	0.1010	0.0023	0.0944	0.0029	0.0907	0.0046
8	55.4666	0.2470	55.3928	0.6112	55.5726	0.3451
9	33.3744	0.1659	33.3728	0.3573	33.2117	0.2052
10	9.2982	0.1026	9.3647	0.2788	9.3537	0.1441
11	1.5655	0.0287	1.5785	0.0685	1.5694	0.0400
12	0.1768	0.0042	0.1803	0.0097	0.1845	0.0097
13	0.0157	0.0005	0.0154	0.0009	0.0165	0.0011
14	0.0018	0.0001	0.0011	0.0001	0.0009	0.0002

Table 3: The estimated posterior distribution $P(\mathcal{X}_k|Z)$ for the change-point identification example. SD: standard deviation of the estimates.

Given a group of connection weights (α, β) , the MLP approximator can be written as

$$\hat{f}(x_k|\alpha, \beta) = \varphi'(\alpha_0 + \sum_{i=1}^M \alpha_i \varphi(\beta_{i0} + \sum_{j=1}^p \beta_{ij} x_{kj})), \quad (22)$$

where α_i 's and β_{ij} 's are the connection weights from the hidden units to the output unit and from the input units to the hidden units, respectively.

To force \hat{f} to converge to the target function, it is usually to minimize the following objective function

$$U(\alpha, \beta) = \sum_{k=1}^N (\hat{f}(x_k|\alpha, \beta) - y_k)^2, \quad (23)$$

where y_k denotes the target output corresponding to the input pattern x_k .

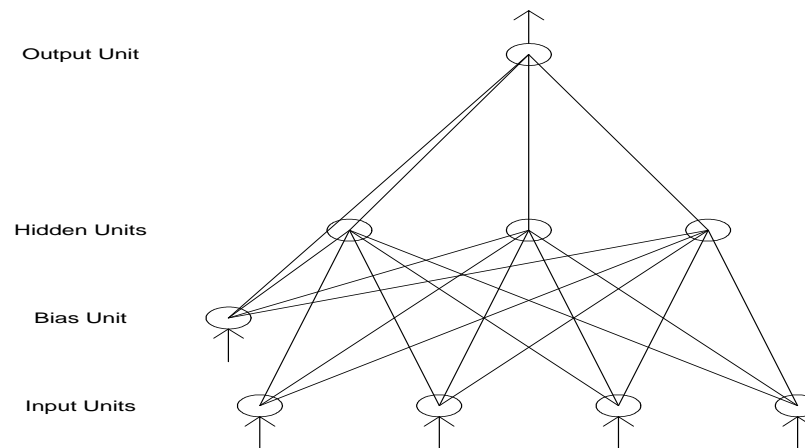


Figure 8: Structure of a MLP with four input units, three hidden units and one output unit. The arrows show the direction of data feeding, where each unit independently processes the values fed to it by the units in the preceding layer and then presents its output to the units in the next layer for further processing.

Difficulty in neural network training

1. *High dimensionality.*
2. *High nonlinearity.*
3. *Multiple local minima.*

MLP training algorithms

- (a) *Back-propagation.*
- (b) *BFGS algorithm.*
- (c) *Simulated annealing.*
- (d) *Genetic algorithms.*

Annealing SAMC

The algorithm initiates the search in the entire sample space $\mathcal{X}_0 = \bigcup_{i=1}^m E_i$, and then iteratively searches in the set

$$\mathcal{X}_k = \bigcup_{i=1}^{\varpi(U_{\min}^{(k)} + \aleph)} E_i, \quad t = 1, 2, \dots, \quad (24)$$

where $\varpi(u)$ denotes the index of the subregion that a sample x with energy u belongs to, $U_{\min}^{(k)}$ is the best function value obtained until iteration k , and $\aleph > 0$ is a user specified parameter which determines the broadness of the sample space at each iteration.

Since the sample space shrinks iteration by iteration, the algorithm is called annealing SAMC.

Algorithm	Mean	S.D.	Minimum	Maximum	Succ	Iter($\times 10^6$)	Time
ASAMC	0.620	0.191	0.187	3.23	15	7.07	94m
SAMC	2.727	0.208	1.092	4.089	0	10.0	132m
SA-1	17.485	0.706	9.02	22.06	0	10.0	123m
SA-2	6.433	0.450	3.03	11.02	0	10.0	123m
BFGS	15.50	0.899	10.00	24.00	0	—	3s

Table 4: Comparison of the ASAMC and SA algorithms for the two-spiral example. “Succ” denotes the number of runs (out of 20) found a solution with energy less than 0.2.

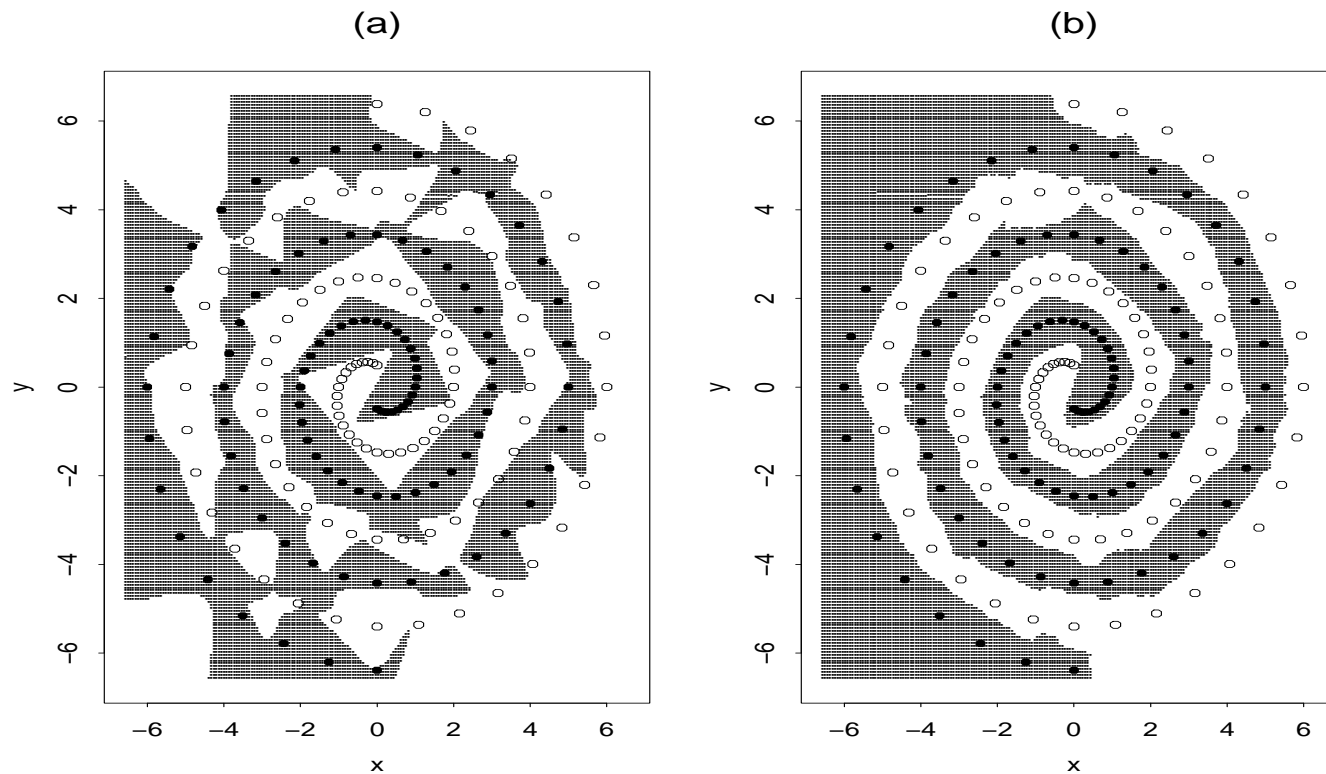


Figure 9: Two-spiral problem: Classification maps learned by a MLP of 30 hidden units. The black and white points show the training data for two different spirals. (a) The classification map learned in one run. (b) The classification map learned in 20 runs.

Other Applications

- *Marginal density estimation (Liang, 2007, JCGS)*
- *Normalizing constant estimation (Liang, 2007, Encyclopedia of Artificial Intelligence)*
- *Protein folding simulation (Liang, 2004, J. Chem. Phys)*
- *Phylogenetic tree reconstruction (Cheon and Liang, 2007, BioSystems)*
- *Elicitation of Bayesian hyperparameters (Liang, Chen and Ibrahim, 2007)*