Markov chain Monte Carlo idea

- Situation:
  - Given a target distribution $f(x)$
  - Want to estimate

$$\mu = \mathrm{E}_f[g(X)] = \int g(x)f(x)\mathrm{d}x$$

  - Want to generate samples from $f(x)$

- Idea:
  - construct a Markov chain $\{X_i\}_{i=1}^{\infty}$ so that

$$\lim_{i\to\infty} P(X_i = x) = f(x)$$

  - simulate the Markov chain for many iterations
  - for m large enough $x_m, x_{m+1}, \ldots$ are (essentially) from $f(x)$
  - estimate μ by

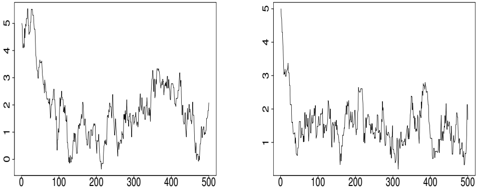
$$\widetilde{\mu} = \frac{1}{n}\sum_{i=m}^{m+n-1} g(x_i)$$

Metropolis–Hastings algorithm

- We have discussed:
  - how to construct the Markov chain
  - different proposal strategies
  - how to combine proposal strategies
  - how to evaluate the convergence/burn-in based on simulation output
- Remains to discuss:
  - how to evaluate the convergence/burn-in based on simulation output
  - how to compare algorithms
  - variance estimation from simulation output
  - typical MCMC problems

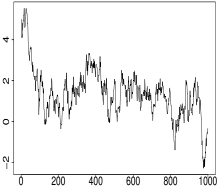
Convergence diagnostics

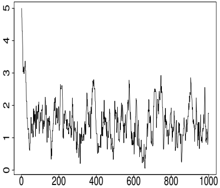
- ⋆ Formal convergence diagnostics exists
  - some based on a single Markov chain run
  - some based on several Markov chain runs
- ⋆ To see when a chain has convergence, we need to simulate much longer than to convergence
- ⋆ If some properties of the target distribution is known: use it to check convergence!
- ⋆ All convergence diagnostics can (and do) fail
  - has this bivariate chain converged?



Convergence diagnostics

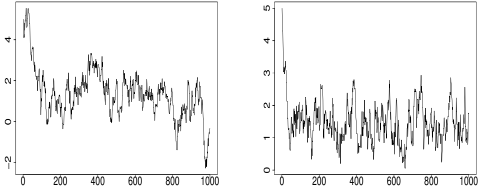
⋆ Has it converged now?



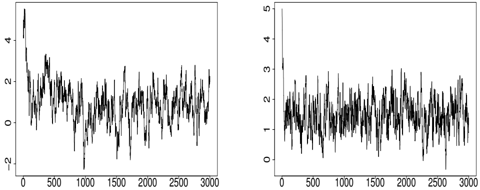


Convergence diagnostics

⋆ Has it converged now?

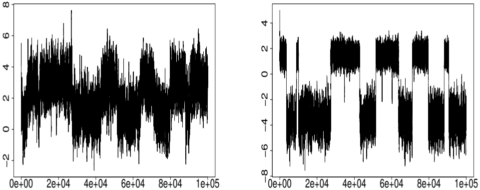


⋆ And now?



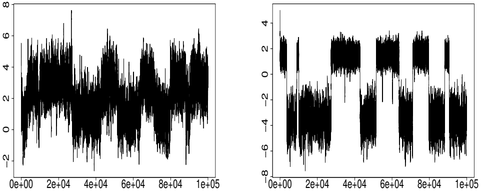
Convergence diagnostics

⋆ And now?

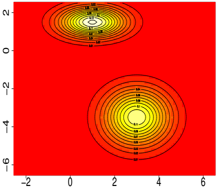


Convergence diagnostics

- ⋆ And now?



- ⋆ This is how the distribution looks like



  – used random walk proposals $y|x \sim \mathrm{N}_2(0, 0.3^2 \cdot I)$

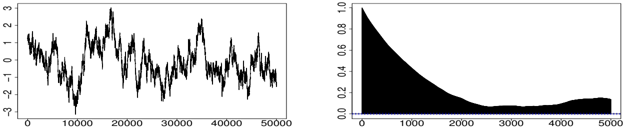
Compare algorithms

- ⋆ Assume: have two (or more) Markov chains with limiting distribution $f(x)$
- ⋆ Which one should we prefer?
- ⋆ Estimate and compare autocorrelation functions
  - ignore burn-in periods!
  - assume stationary time series
  - must again consider scalar functions $g(x)$

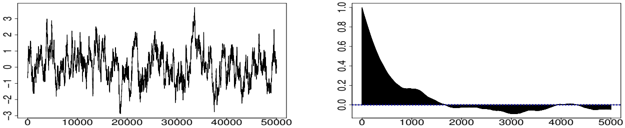
Compare algorithms: Toy example

- ⋆ Random walk proposal example, choice of tuning parameter

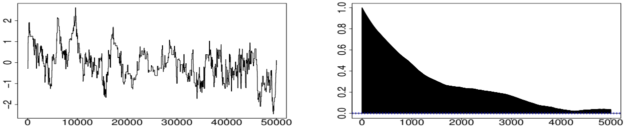
$\sigma = 0.05$, acceptance rate $= 0.69$



$\sigma = 0.20$, acceptance rate $= 0.11$



$\sigma = 0.30$, acceptance rate $= 0.018$

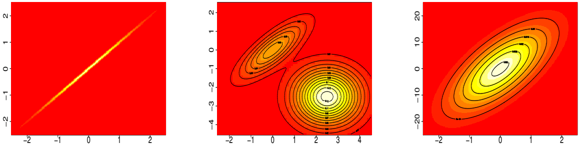


Typical MCMC problems

⋆ Note: If you knows the solution, it is easy to solve a problem!

⋆ Properties of $f(x)$ that may make MCMC difficult

- strong dependency between variables
- several modes
- different scales on different variables



⋆ In toy examples: this is not a problem

- we know how $f(x)$ looks like

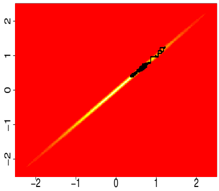
⋆ In real problems: this may be difficult

- we have a formula for $f(x)$
- we don't know how $f(x)$ looks like

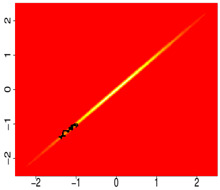
▶ Need to iterate

Strong dependencies

- ⋆ Gibbs sampling doesn't work



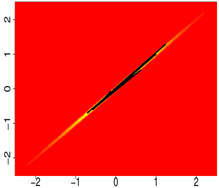
- ⋆ Changing one variable at a time doesn't work



Strong dependencies

- Blocking may solve the problem
  - $x = x(x^1, x^2, \ldots, x^n)$
  - x^1 and x^2 are highly correlated
  - propose joint updates for x^1 and x^2
    - block Gibbs: $(y^1, y^2)|x \sim f(y^1, y^2|x^{-\{1,2\}})$
    - random walk Metropolis–Hastings:

$$(y^1, y^2)|x \sim \mathrm{N}_2\left(\left[\begin{array}{c} x^1 \\ x^2 \end{array}\right], R\right)$$



    - in toy example: target correlation 0.999, proposal correlation 0.90

Strong dependencies

- Reparameterisation may solve the problem
  - $x = (x^1, x^2, \ldots, x^n)$
  - x^1 and x^2 are highly correlated
  - define

$$\begin{bmatrix} \tilde{x}^1 \\ \tilde{x}^2 \end{bmatrix} = A \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}$$

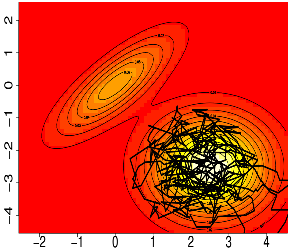
    and

$$\tilde{x}^i = x^i \quad \text{for } i = 3, \ldots, n$$

  - with suitable choice of matrix A, the correlation between $\tilde{x}^1$ and $\tilde{x}^2$ in $f(\tilde{x})$ will be much lower

Multimodal target distribution

⋆ Random walk proposals doesn't work



⋆ To come from one mode to another: needs to visit low probability area — happens very seldomly

Multimodal target distributions

⋆ If you know (approximately) the modes

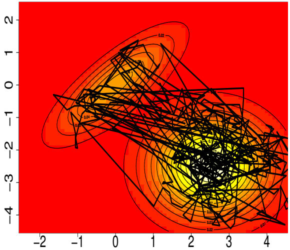
- can combine
  - independent proposals

$$y|x \sim \frac{1}{2}g_1(y) + \frac{1}{2}g_2(y)$$

  - random walk proposals

$$y|x \sim \mathrm{N}(x, R)$$

- randomly or systematically

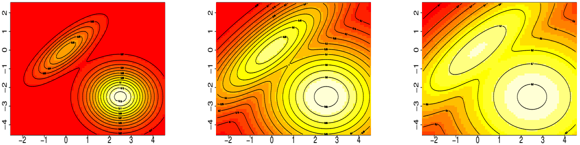


Multimodal target distributions

- Simulated tempering
  - let $f(x) = c \exp\{-U(x)\}$
  - introduce an extra variable, $k \in \{1, 2, \ldots, K\}$
  - define K temperatures: $1 = T_1 < T_2 < \ldots < T_K$
  - define K distributions and constants $c_1, \ldots, c_K$

$$f_k(x) = c_k \exp\left\{-\frac{1}{T_k} U(x)\right\}$$

    * note: $f_0(x) = f(x)$



  - define joint distribution: $f(x, k) \propto f_k(x)$
  - simulate from $f(x, k)$ with Metropolis–Hastings
  - keep simulated x's that corresponds to $k = 1$
- Note: the T_k's and c_k's must be chosen carefully

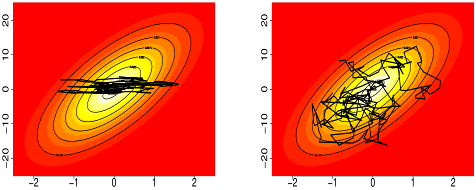
Multimodal target distributions

- Other solutions has been proposed
  - MCMCMC: Metropolis coupled MCMC
    - simulate one x_k for each temperate T_k
    - simulate each x_k by standard Metropolis-Hastings
    - occasionally propose to swap "neighbour" states x_k and x_{k+1}
    - accept/reject according to MH acceptance probability
  - mode-jumping
    - in a Metropolis–Hastings algorithm: use local optimisation to locate a local maximum, propose a new value from that mode

Different scales

- ⋆ With Gibbs: different scales are not a problem
  - Gibbs finds the appropriate scale
- ⋆ If Gibbs not possible: have to tune to find appropriate scales

equal scale in proposals | tuned scales in proposals



- ⋆ Tempting to tune the proposal scales automatically based on the history of the Markov chain
  - careful!! it is no longer Markov
  - more difficult to get the required limiting distribution
  - some *adaptive MCMC* algorithms exist