The Metropolis-Hastings Algorithm

Prof. Nicholas Zabaras
University of Notre Dame
Notre Dame, IN, USA

Email: nzabaras@gmail.com
URL: https:/www.zabaras.com/

October 4, 2017

Contents

- Christian P. Robert and George Casella, Monte Carlo Statistical Methods, Springer, 2nd edition (Chapters 6, 7, 9 & 10) (Video, Lecture Slides)
- Julian Besag, Markov Chain Monte Carlo for Statistical Inference (2000) (working paper)
- C. Andrieu, et al. , An Introduction to MCMC for Machine Learning (2003)
- S. Chib and E. Greenberg, Understanding the Metropolis-Hastings algorithm, The American Statistician, 1995
- Java applets for the Metropolis Hastings algorithm
- L. Held, Conditional Prior Proposals in Dynamic Models, Scand. J. Statist., 1999
- M.K. Pitt & N. Shephard, Likelihood Analysis of Non-Gaussian Measurement Time Series, Biometrika, 1996

Markov Chain Monte Carlo

- The simplest way to generate a sequence of random variables and be able to say something about asymptotics is using Markov Chains.

- A Markov Chain $\{X_n\}, n = 0,1,2, \dots, \infty$ is fully defined if we know:

  - Initial distribution $p_0(x_0) = \Pr[X_0 = x_0]$ (*this will prove of little significance*)

  - Transition Kernel: $P(x_n, x_{n+1}) = \Pr[X_{n+1} = x_{n+1} \mid X_n = x_n]$.

Autoregressive Model

- We generate a sequence of random variables using Markov Chains.
- A Markov Chain $\{X_n\}$ $n=1, 2,\ldots$ is fully defined if we know:
  - Initial distribution $p_0(x_0) = Pr\,[X_0 = x_0]$ (this will prove of little significance)
  - Transition Kernel: $P(x_n, x_{n+1}) = Pr\,[X_{n+1} = x_{n+1} \,|\, X_n = x_n]$
- An example of a Markov chain is an autoregressive model:

$$X_n = \rho X_{n-1} + Z_n \quad \text{where } X_0, Z_n \sim \mathcal{N}(0, 1) \text{ (i.i.d) with } |\rho|<1$$

- Initial distribution: $X_0 \sim \mathcal{N}(0, 1)$
- Transition Kernel: $X_n \,|\, X_{n-1} \sim \mathcal{N}(\rho\, X_{n-1}, 1), |\rho|<1$

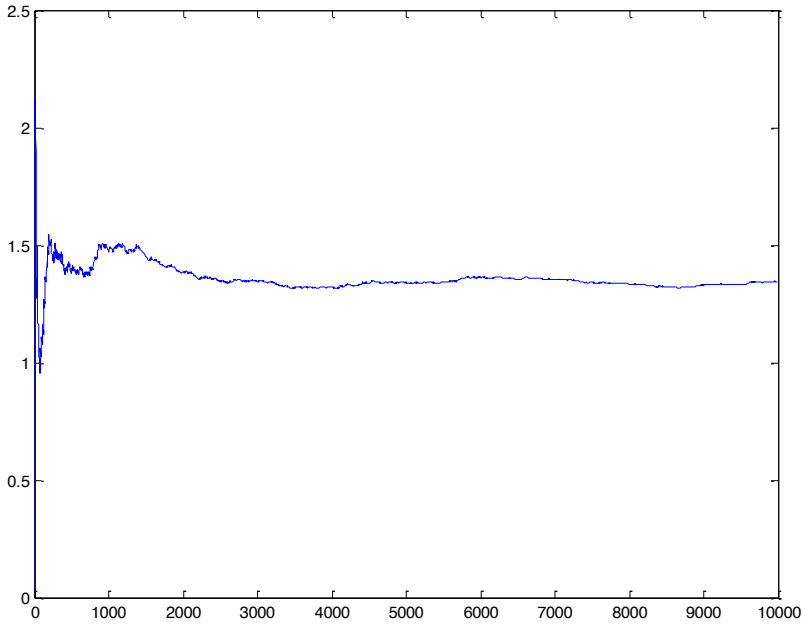
$$\mathbb{E}\{X_n\} = 0, \quad Var\{X_n\} = \rho^{2n} + \frac{1-\rho^{2n}}{1-\rho^2} \to \frac{1}{1-\rho^2}$$

- Asymptotically:

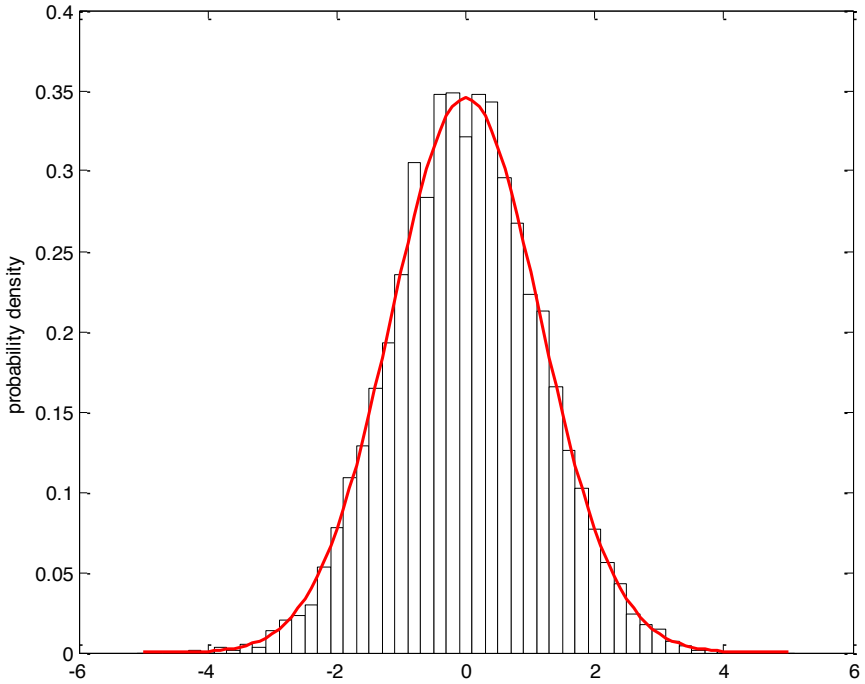
$$X_n \sim \mathcal{N}(0, \frac{1}{1-\rho^2})$$

Autoregressive Model: Example

- Case: ρ=0.5, initial state: $X_0 \sim \mathcal{N}(0, 1)$. Asymptotic variance: 4/3



Variance of the Markov Chain vs. the number of samples



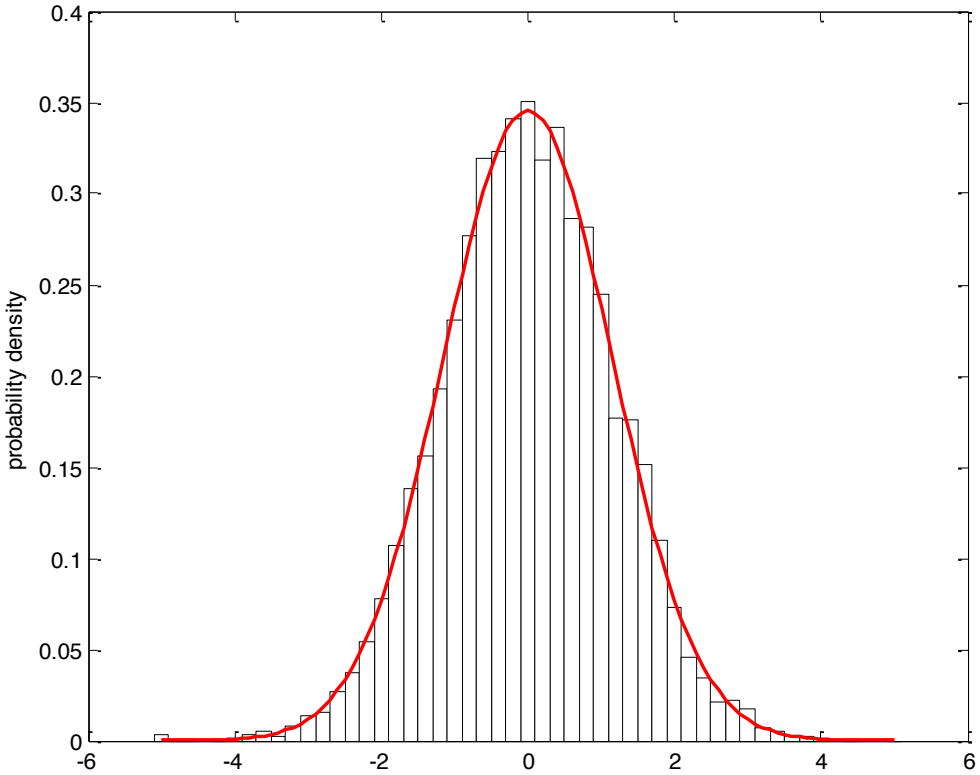
Histogram of the distribution of samples (compared with the exact pdf)

$$\hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \hat{X}_n\right)^2, \; where: \; \hat{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Autoregressive Model: Example

- Case: ρ=0.5, initial state: $X_0 = -1000$ (MatLab implementation)

Since the initial value X_0 here has a significant influence on the estimated "variance" (X_0-μ is much larger than other X_n - μ), the figure of the variance is not presented



Histogram of the distribution of samples (compared with the exact pdf)

Markov Chain Monte Carlo

- We can define a Markov Chain which only requires determining a local rule $P(X_n, X_{n+1})$
- If we make a good selection for the transition kernel, it could asymptotically converge to a target distribution independently of where we started from
- More importantly, we can use the realizations of the Markov Chain in Monte Carlo estimators i.e. we can average across the path.
- However note that even if X_n were exact draws, they are not independent anymore!

Markov Chain Monte Carlo

- Ergodic Markov chain:

$$\hat{I} = \frac{1}{N}\sum_{n=1}^{N} f\left(X_n\right) \rightarrow I = \int f\left(x\right)\pi\left(x\right)dx$$

$\{X_i\}$ form a *Markov Chain* which asymptotically converges to $\pi(x)$ (we haven't discussed yet under which conditions this holds)

- We also care about how fast it converges (particularly when each evaluation of *f* is expensive)

- In standard Monte Carlo using i.i.d. samples we had:

$$Var\left[\hat{I}\right] = \frac{Var_{\pi}[f(x)]}{N}$$

Markov Chain Monte Carlo

$$I = \mathbb{E}_\pi\left[f(x)\right] = \int f(x)\pi(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(X_i) = \hat{I}$$

$$\mathbb{E}\left[\hat{I}\right] = \frac{1}{N}\sum_i \mathbb{E}\left[f(X_i)\right] = \mathbb{E}\left[f\right], \text{ and } \operatorname{var}\left[\hat{I}\right] = \mathbb{E}\left[\hat{I}^2\right] - \left(\mathbb{E}\left[\hat{I}\right]\right)^2$$

$$= \mathbb{E}\left[\left(\frac{1}{N}\sum_{n=1}^{N} f(X_n)\right)\left(\frac{1}{N}\sum_{m=1}^{N} f(X_m)\right)\right] - \left(\mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N} f(X_n)\right]\right)^2 = \frac{1}{N^2}\sum_{n=1}^{N}\sum_{m=1}^{N}\mathbb{E}\left[f(X_n)f(X_m)\right] - \left(\mathbb{E}\left[f\right]\right)^2$$

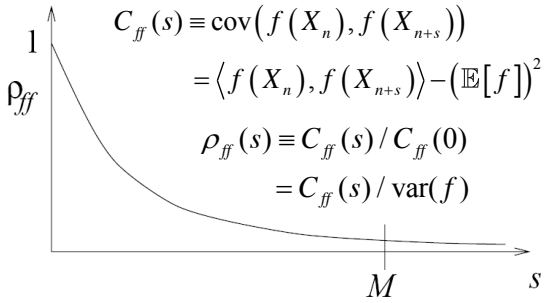
- Let $Z_i = f(X_i) - \mathbb{E}\left[f(X_i)\right]$ and assume it is weakly stationary

$$\operatorname{var}\left[Z_i\right] = \mathbb{E}\left[Z_i^2\right] = \sigma^2, \quad \mathbb{E}\left[Z_i Z_j\right] = \sigma^2 \underbrace{\rho(j-i)}_{\substack{\text{Normalized} \\ \text{auto-covariance} \\ \text{function}}}$$

- Then you can easily show that:

$$\operatorname{var}\left[\hat{I}\right] = \frac{1}{N^2}\sum_{n=1}^{N}\sum_{m=1}^{N}\mathbb{E}\left[Z_n Z_m\right] = \frac{\sigma^2}{N}\left(\underbrace{1 + 2\sum_{j=1}^{N-1}\left(1 - \frac{j}{N}\right)\rho(j)}_{\tau_f \,:\, \text{autocovariance time}}\right) = \frac{\sigma^2 \tau_f}{N}$$

Autocovariance function



- For some M sufficiently large $\rho_{ff}(s) \approx 0$ when $s \geq M$
- For $N \gg M$, the X_0 and X_N samples are totally uncorrelated.

Markov Chain Monte Carlo

- Objective: Given an arbitrary distribution $\pi(\boldsymbol{x})$ we want to construct a Markov Chain that asymptotically *converges* to the target independently of the initial state.

- We want to use the Markov Chain paths in estimators

$$\hat{I} = \frac{1}{N}\sum_{n=1}^{N} f\left(X_n\right) \rightarrow I = \int f\left(x\right)\pi\left(x\right)dx$$

- This requires coming up with a way to produce suitable transition kernels $P(X_n, X_{n+1})$ for any target $\pi(\boldsymbol{x})$

- The first successful attempt was the Metropolis algorithm proposed in 1953 by N. Metropolis, AW Rosenbluth, MN Rosenbluth, AH Teller and E. Teller in " Equations of State calculations by fast computing machines", J. Chem Phys, 21 pp 1087. This paper has been cited 35,438 times since then!

Metropolis Algorithm

- Let $p(\boldsymbol{x})$ the target and $q(\boldsymbol{y}/\boldsymbol{x})$ a **symmetric proposal distribution** such $q(\boldsymbol{y}/\boldsymbol{x}) = q(\boldsymbol{x}/\boldsymbol{y})$. Given state $\boldsymbol{x}_n$ at step n
  - Draw a proposal $\boldsymbol{y}$ from $q(\boldsymbol{y} / \boldsymbol{x}_n)$
  - Calculate the acceptance ratio:

$$\alpha(\boldsymbol{x}_n, \boldsymbol{y}) = \min\left\{1, \frac{\pi(\boldsymbol{y})}{\pi(\boldsymbol{x}_n)}\right\}$$

  - Set

$$\boldsymbol{x}_{n+1} = \begin{cases} \boldsymbol{y} & with\ probability\ \alpha(\boldsymbol{x}_n, \boldsymbol{y}) \\ \boldsymbol{x}_n & with\ probability\ 1-\alpha(\boldsymbol{x}_n, \boldsymbol{y}) \end{cases}$$

N. Metropolis, A W Rosenbluth, M N Resenbluth, A H Teller and E Teller, Equations of State Calculations for Fast Computing Machines, J Chem Physics, Vol 21, pp. 1087 (1953)

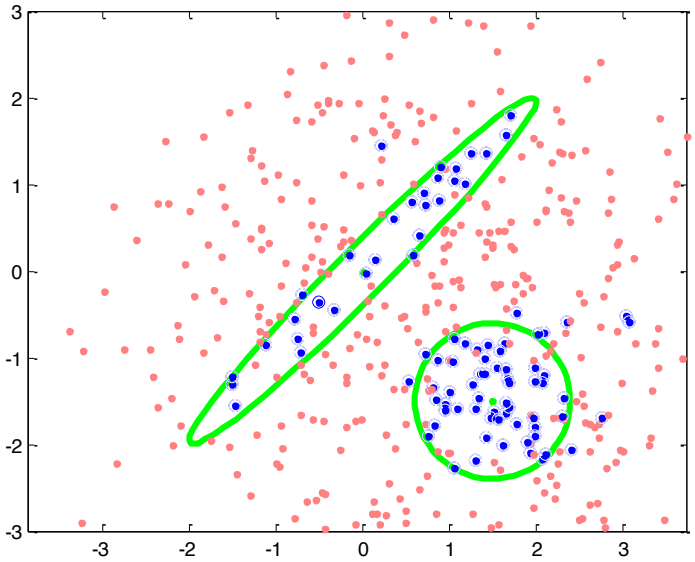
Sampling from a Mixture of Gaussians

- The MatLab demo here shows how you can sample from a probability distribution known up to a normalizing constant using MCMC with random walk proposals.

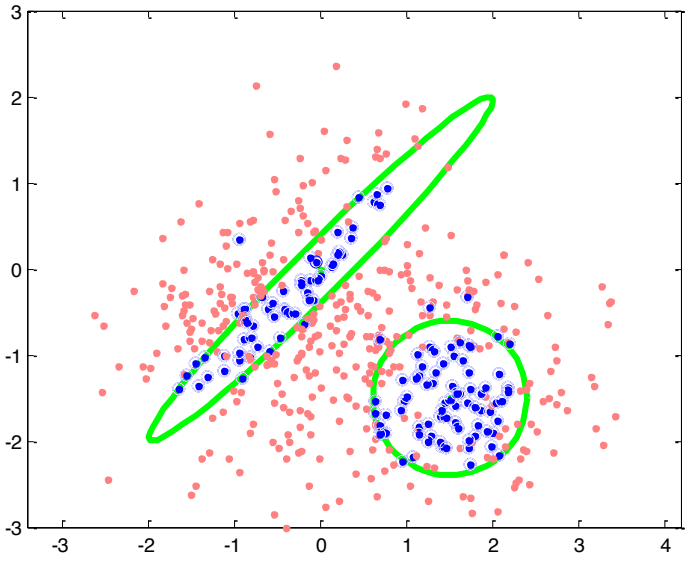
- Suppose that the probability distribution you want to sample from is $p(x)$.
  - 1. Initialize x.
  - 2. Propose a new $x_{new} \sim q(x_{new} \mid x) = \mathcal{N}(x \mid 0, s^2)$. Here $q(x_{new} \mid x)$ leads to a reversible Markov Chain and the classic Metropolis algorithm is used.
  - 3. Draw a random number $u \sim \mathcal{U}[0, 1]$.
  - 4. If $u <= min(1, p(x_{new}) / p(x))$, accept the move, i.e. $x = x_{new}$.
  - 5. Otherwise reject the move.

- The target distribution is a 50%-50% mixture of two Gaussians.

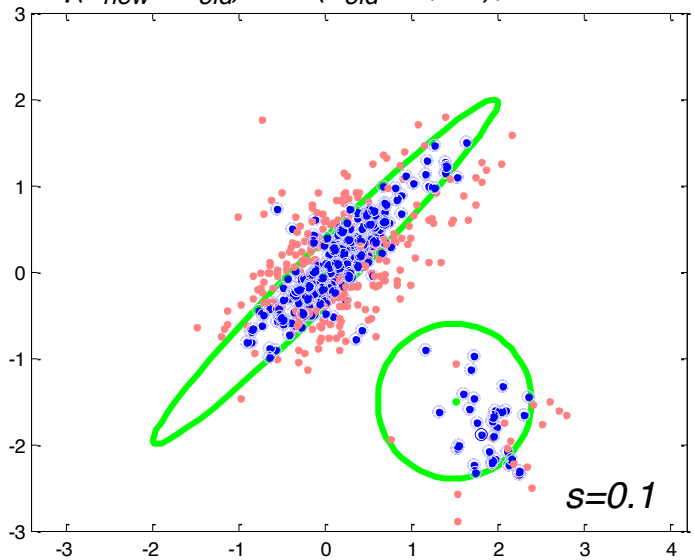
Sampling from a Mixture of Gaussians

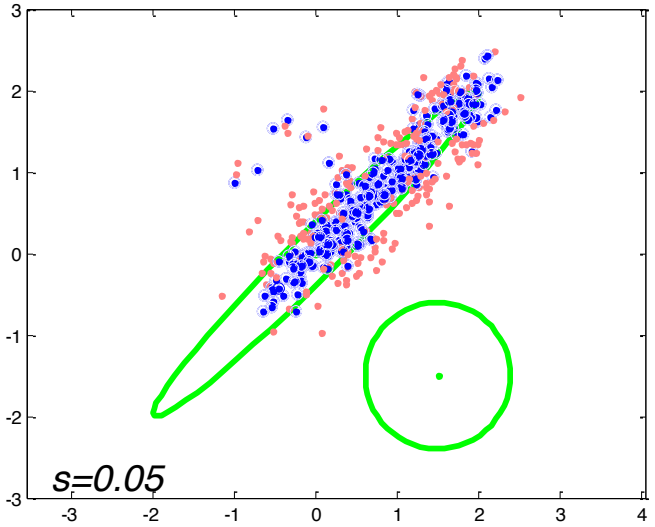


$q(x_{new} \mid x_{old}) = N(x_{old} \mid 0, s^2)$, $s=2.0$



$s=0.5$





Metropolis Algorithm

- To implement the Metropolis scheme we only need to know the target density $\pi(\boldsymbol{x})$ up to a constant!
- If such a scheme is to converge to the target distribution $\pi(\boldsymbol{x})$ then this must be invariant, i.e.

$$\int \pi(\boldsymbol{x}_n) p(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) d\boldsymbol{x}_n = \pi(\boldsymbol{x}_{n+1})$$

  There are many transition kernels $p(\boldsymbol{x}_n, \boldsymbol{x}_{n+1})$ that satisfy this condition.

- Note that the transition kernel $p(\boldsymbol{x}_n, \boldsymbol{x}_{n+1})$ is not the same as the proposal distribution $q(\boldsymbol{y} \mid \boldsymbol{x})$!

$$p(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) = p(\boldsymbol{x}_{n+1} \mid \text{proposal acc.}) \Pr[\text{proposal accepted}] + p(\boldsymbol{x}_{n+1} \mid \text{proposal rejected}) \Pr[\text{proposal rejected}]$$

Detailed Balance vs π invariant

- If $x_{n+1} \neq x_n \qquad P(x_n, x_{n+1}) = q(x_{n+1} \mid x_n) a(x_n, x_{n+1})$
- If $x_{n+1} = x_n \qquad P(x_n, x_{n+1}) = \int 1_{\{y = x_n\}} q(y \mid x_n) dy + \int \big(1 - a(x_n, y)\big) q(y \mid x_n) dy$
- The transition kernel p satisfies the detailed balance condition (reversibility) $\pi(x_n) p(x_n, x_{n+1}) = \pi(x_{n+1}) p(x_{n+1}, x_n)$
- Detailed balance implies that π is invariant. Indeed:

$$\begin{aligned} \int \pi(x_n) p(x_n, x_{n+1}) dx_n &= \int \pi(x_{n+1}) p(x_{n+1}, x_n) dx_n \\ &= \pi(x_{n+1}) \int p(x_{n+1}, x_n) dx_n = \pi(x_{n+1}) \end{aligned}$$

- Detailed balance (reversibility) is thus stronger than invariance. Many more kernels are π-invariant than π-reversible.
- Fortunately, it is easier to construct a transition kernel that is π-reversible than just π-invariant.

aperiodicity

- We already have seen that π–invariance is not enough to guarantee that the chain converges to π.
- In addition we need: aperiodicity and π-irreducibility.
- *Aperiodicity:* Let *M* be an irreducible Markov chain with transition matrix *P* and let ***x*** be a fixed state. Define the set

$$T = \left\{ k : p^k(\boldsymbol{x}, \boldsymbol{x}) > 0, k > 0 \right\}$$

  These are the steps on which it is possible for a chain which starts in state ***x*** to revisit ***x***. The greatest common divisor (g.c.d.) of the integers in *T* is called the period of state ***x***.
- The chain is said to be **periodic** if the period of any of its states is greater than one.
- A state with period one is **aperiodic**, i.e. one does not visit in a periodic way the state-space.

Irreducibility and Ergodicity

- Irreducibility is a measure of the sensitivity of the Markov Chain to initial conditions:
  - $p(x' \mid x)$ is π-irreducible if for any set $\mathrm{A} \subset \Omega$ with $\int_A \pi(x)dx > 0$,

    $\Pr(X_n \in \mathrm{A} \text{ for some finite } n \mid X_0 = x) > 0$, so that the chain can hit any set that has finite probability in π
  - It is satisfied if $\forall\ \boldsymbol{y} : \pi(\boldsymbol{y}) > 0 \Rightarrow q(\boldsymbol{y} \mid \boldsymbol{x}) > 0\ \forall\ \boldsymbol{x}$

- Theorem (Ergodicity from reversibility)
  - Let $\pi(x)$ be a given probability density on Ω. If $p(x' \mid x)$ is π-irreducible and if p is reversible and aperiodic with respect to π, then

    $$\int_A \pi^{(n)}(x)dx \to \int_A \pi(x)dx \quad \text{as} \quad n \to \infty$$

    for any set $A \subset \Omega$ and starting distribution $\pi^{(0)}$

Metropolis -Hastings Algorithm

- Let $\pi(\boldsymbol{x})$ the target and $q(\boldsymbol{y} \mid \boldsymbol{x})$ any (symmetric or not) distribution such $q(\boldsymbol{y} \mid \boldsymbol{x}) = q(\boldsymbol{x} \mid \boldsymbol{y})$. Given state $\boldsymbol{x}_n$ at step n
  - Draw a proposal $\boldsymbol{y}$ *from* $q(\boldsymbol{y} \mid \boldsymbol{x}_n)$
  - Calculate acceptance ratio:

$$\alpha(\boldsymbol{x}_n, \boldsymbol{y}) = \min\left\{1, \frac{\pi(\boldsymbol{y})}{\pi(\boldsymbol{x}_n)} \frac{q(\boldsymbol{x}_n \mid \boldsymbol{y})}{q(\boldsymbol{y} \mid \boldsymbol{x}_n)}\right\}$$

  - Set

$$\boldsymbol{x}_{n+1} = \begin{cases} \boldsymbol{y} & \text{with probability } \alpha(\boldsymbol{x}_n, \boldsymbol{y}) \\ \boldsymbol{x}_n & \text{with probability } 1 - \alpha(\boldsymbol{x}_n, \boldsymbol{y}) \end{cases}$$

W. Hastings, Monte Carlo Sampling Methods using Markov Chains and their Applications, Biometrica, Vol. 57(1), pp. 97-109 (1970).

Metropolis -Hastings Algorithm

- The Metropolis-Hastings algorithm extends the original Metropolis algorithm allowing for an arbitrary proposal.
- The target distribution π needs to be known only up to a constant
- This is useful in Bayesian inference where the target distribution is the posterior (not known normalizing factor)

$$p(\boldsymbol{x}) = p(\boldsymbol{x} \mid data) \propto p(data \mid \boldsymbol{x})\, p(\boldsymbol{x})$$

- Irreducibility & aperiodicity are satisfied under weak conditions
- M-H is a stochastic algorithm, i.e. even if you draw the same proposal ***y****, this is accepted with a certain* probability.

Invariant Distribution of the Metropolis-Hastings

- The transition kernel associated to the MH algorithm can be re-written as

$$K(\theta,\theta') = \alpha(\theta,\theta')q(\theta,\theta') + \underbrace{\left(1-\int \alpha(\theta,u)q(\theta,u)du\right)}_{\text{Rejection Probability}}\delta_\theta(\theta')$$

This is a loose notation for

$$K(\theta,d\theta') = \alpha(\theta,\theta')q(\theta,\theta')d\theta' + \left(1-\int \alpha(\theta,u)q(\theta,u)du\right)\delta_\theta(d\theta')$$

- Clearly we need to satisfy $\int K(\theta,\theta')d\theta' = 1$. Indeed:

$$\int K(\theta,\theta')d\theta' = \int \alpha(\theta,\theta')q(\theta,\theta')d\theta' + \left(1-\int \alpha(\theta,u)q(\theta,u)du\right)\int \delta_\theta(\theta')d\theta' = 1$$

Invariant Distribution of the Metropolis-Hastings

- We want to show that

$$\int \pi(\theta) K(\theta, \theta') d\theta = \pi(\theta')$$

- Note that this condition is satisfied if the reversibility property is satisfied: For all θ, θ'

$$\pi(\theta) K(\theta, \theta') = \pi(\theta') K(\theta', \theta)$$

i.e. the probability of being in A and moving to B is equal to the probability of being in B and moving to A.

- Indeed the reversibility condition implies that:

$$\begin{aligned} \int \pi(\theta) K(\theta, \theta') d\theta &= \int \pi(\theta') K(\theta', \theta) d\theta \\ &= \pi(\theta') \int K(\theta', \theta) d\theta = \pi(\theta') \end{aligned}$$

The MH Kernel is Reversible

- By definition of the kernel we have

$$\pi(\theta)K(\theta,\theta') = \pi(\theta)\alpha(\theta,\theta')q(\theta,\theta') + \left(1 - \int \alpha(\theta,u)q(\theta,u)du\right)\delta_\theta(\theta')\pi(\theta)$$

- Then $\pi(\theta)\alpha(\theta,\theta')q(\theta,\theta') = \pi(\theta)\min\left(1, \frac{\pi(\theta')q(\theta',\theta)}{\pi(\theta)q(\theta,\theta')}\right)q(\theta,\theta')$

$$= \min\left(\pi(\theta)q(\theta,\theta'), \pi(\theta')q(\theta',\theta)\right)$$

$$= \pi(\theta')\min\left(1, \frac{\pi(\theta)q(\theta,\theta')}{\pi(\theta')q(\theta',\theta)}\right)q(\theta',\theta)$$

$$= \pi(\theta')\alpha(\theta',\theta)q(\theta',\theta)$$

- We also have obviously

$$\left(1 - \int \alpha(\theta,u)q(\theta,u)du\right)\delta_\theta(\theta')\pi(\theta) = \left(1 - \int \alpha(\theta',u)q(\theta',u)du\right)\delta_{\theta'}(\theta)\pi(\theta')$$

- It follows that $\pi(\theta)K(\theta,\theta') = \pi(\theta')K(\theta',\theta)$

- Hence, π is the invariant distribution of the transient kernel K.

Irreducibility and Aperiodicity

- To ensure irreducibility, a sufficient but not necessary condition is that

$$\pi(\theta') > 0 \Rightarrow q(\theta, \theta') > 0$$

- Aperiodicity is automatically ensured as there is always a strictly positive probability to reject the candidate.

- Theoretically, the MH algorithm converges under very weak assumptions to the target distribution π.

- The convergence can be very slow.

Selecting the Proposal in Random Walk

- Consider now a random walk move. In this case, there is no clear guideline how to select the proposal distribution.

- When the variance of the random walk increments (if it exists) is very small then the acceptance rate can be expected to be around 0.5-0.7.

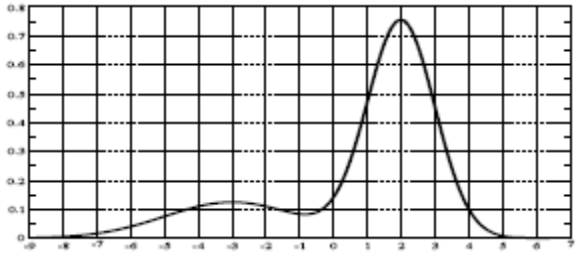
- You would like to scale the random walk moves such that it is possible to move reasonably fast in regions of positive probability masses under π.

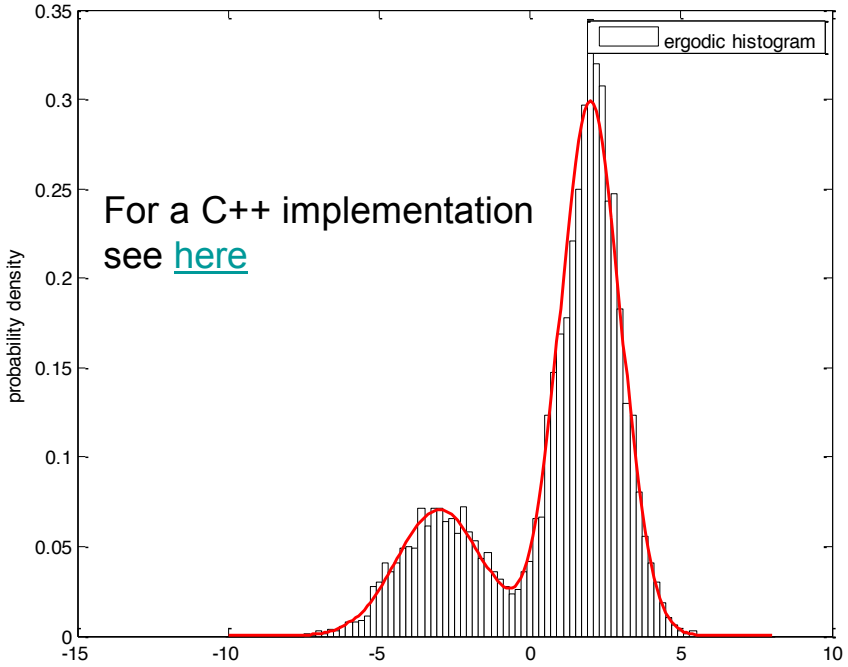
Random Walk Metropolis-Hastings

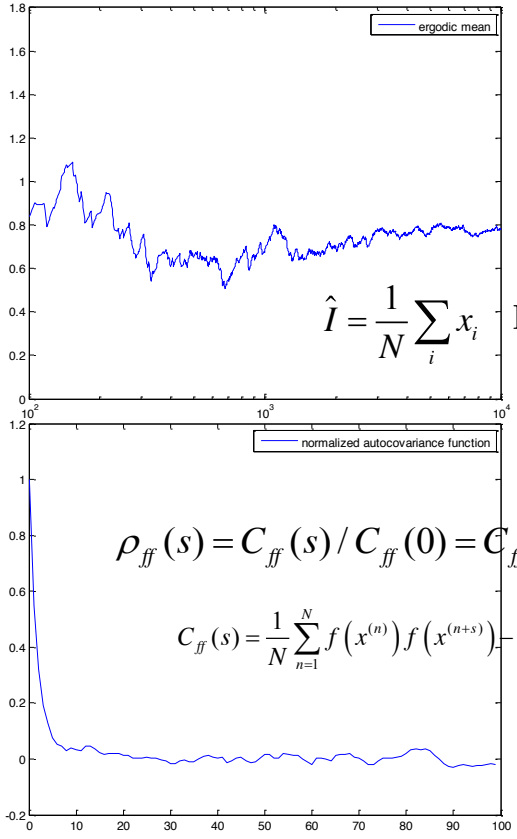
- Target: $\pi(x) = 0.25\mathcal{N}(-3, 2) + 0.75\,\mathcal{N}(2, 1)$
- Random walk proposal: $X_{n+1} = X_n + z_n$

$$p(z_n) = \mathcal{N}(0, \sigma^2) \Rightarrow \quad q(x_{n+1} \mid x_n) = \mathcal{N}(x_n, \sigma^2)$$

- Case: $\sigma = 5$, $x_0 = 0.0$, length of chain = 10000







$$\hat{I} = \frac{1}{N}\sum_i x_i \qquad \mathbb{E}_\pi[X] = 0.75$$

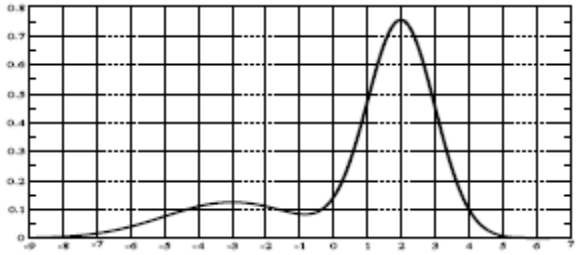
$$\rho_{ff}(s) = C_{ff}(s) / C_{ff}(0) = C_{ff}(s) / \mathrm{var}(f)$$

$$C_{ff}(s) = \frac{1}{N}\sum_{n=1}^{N} f\left(x^{(n)}\right) f\left(x^{(n+s)}\right) - \frac{1}{N^2}\left[\sum_{n=1}^{N} f\left(x^{(n)}\right)\right]^2$$

Acceptance Ratio: 0.38, the best among the three choices considered

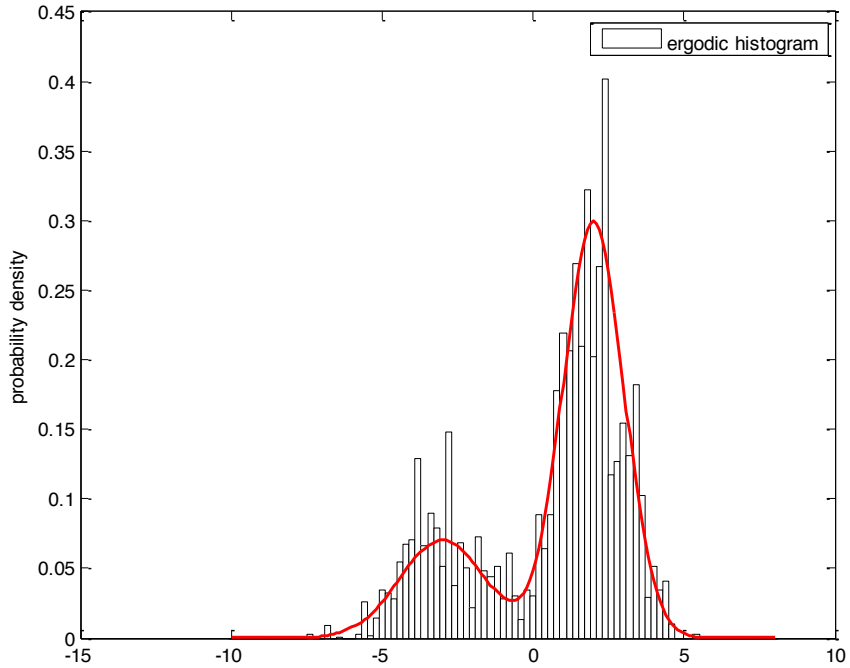
Random Walk Metropolis-Hastings

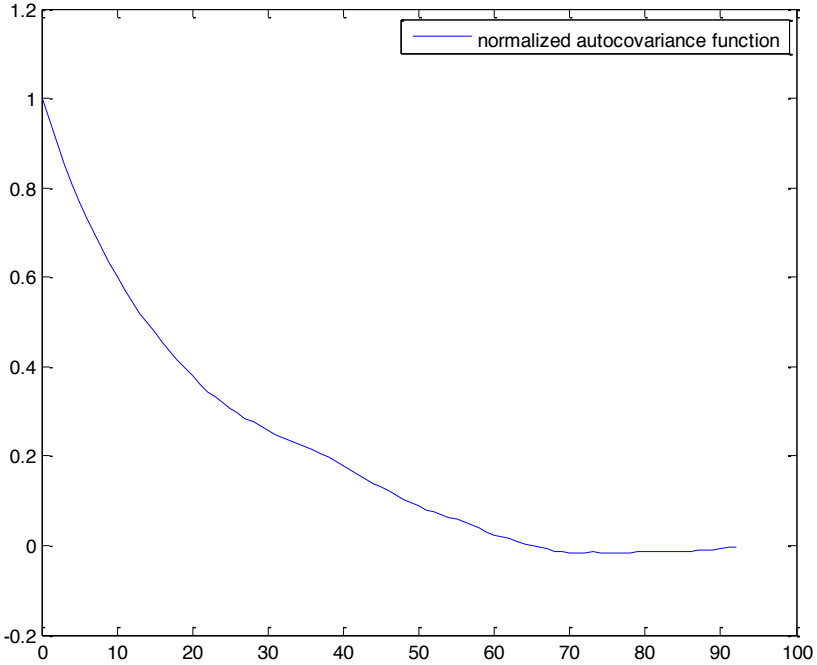
- Target: $\pi(x) = 0.25\mathcal{N}(-3, 2) + 0.75\,\mathcal{N}(2, 1)$
- Random walk proposal: $X_{n+1} = X_n + z_n$



$$p(z_n) = \mathcal{N}(0, \sigma^2) \Rightarrow \quad q(x_{n+1} \mid x_n) = \mathcal{N}(x_n, \sigma^2)$$

- Case: $\sigma = 50$, $x_0 = 0.0$, length of chain = 10000

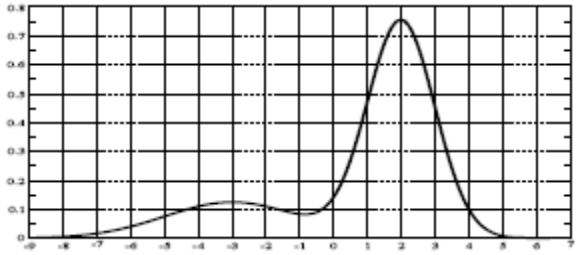




Acceptance Ratio: 0.05, the acceptance rate is very low, the auto-correlation very high and thus the convergence rate very slow

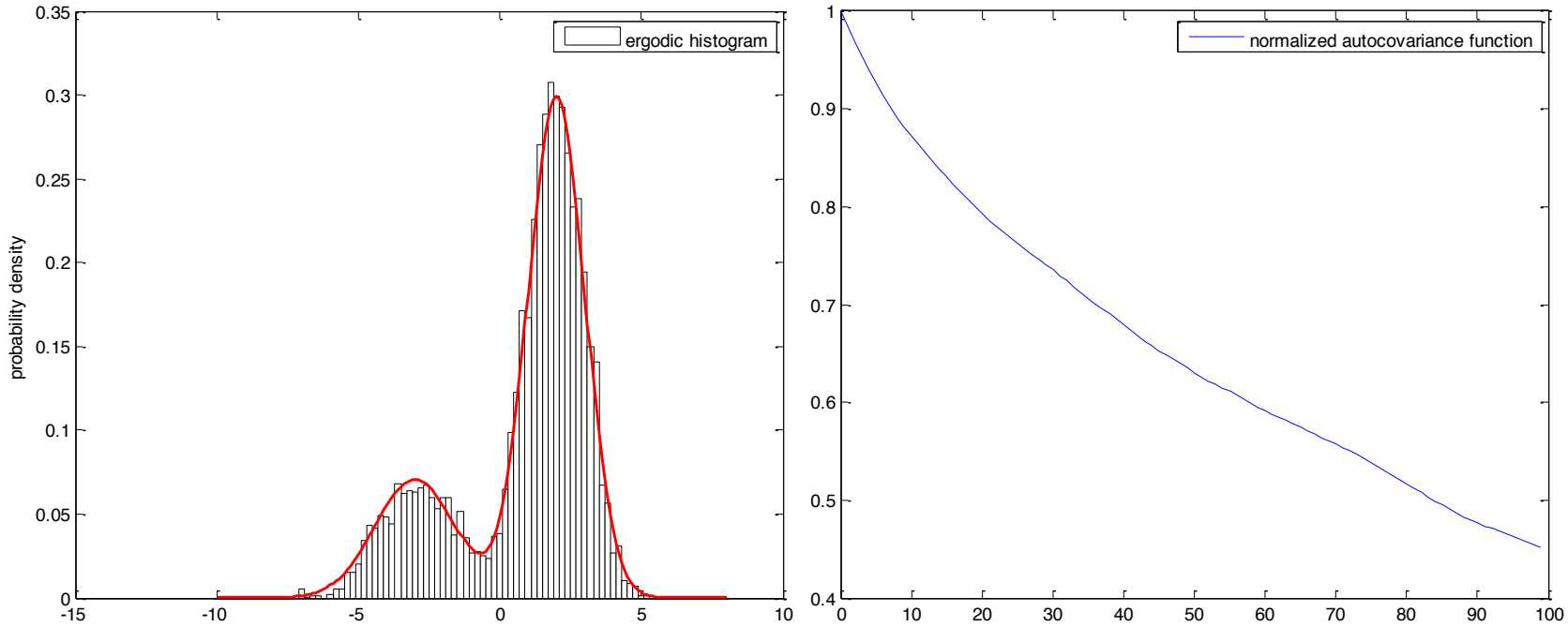
Random Walk Metropolis-Hastings

- Target: $\pi(x) = 0.25\mathcal{N}(-3, 2) + 0.75\,\mathcal{N}(2, 1)$
- Random walk proposal: $X_{n+1} = X_n + z_n$



$$p(z_n) = \mathcal{N}(0, \sigma^2) \Rightarrow \qquad q(x_{n+1} \mid x_n) = \mathcal{N}(x_n, \sigma^2)$$

- Case: $\sigma = 0.5$, $x_0 = 0.0$, length of chain = 10000



Acceptance Ratio: 0.76, the acceptance rate is the highest from the 3 cases considered, the auto-correlation also the highest and thus the convergence rate very slow

Example

- Consider the case where

$$\pi(\theta) \propto e^{-\frac{\theta^2}{2}}$$

- We implement the MH algorithm for

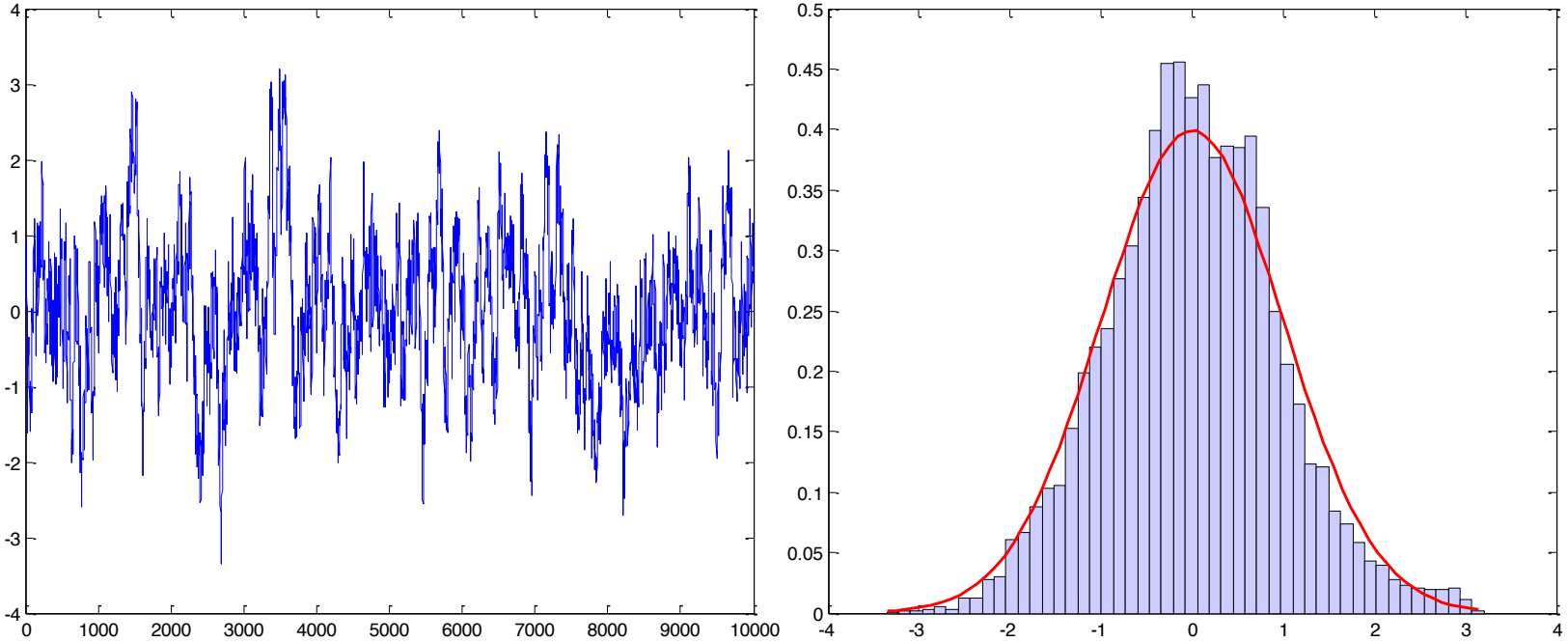
$$q_1(\theta, \theta') \propto e^{-\frac{(\theta-\theta')^2}{2(0.2)^2}}$$

- We implement the MH algorithm for

$$q_2(\theta, \theta') \propto e^{-\frac{(\theta'-\theta)^2}{2(5)^2}}$$

Example

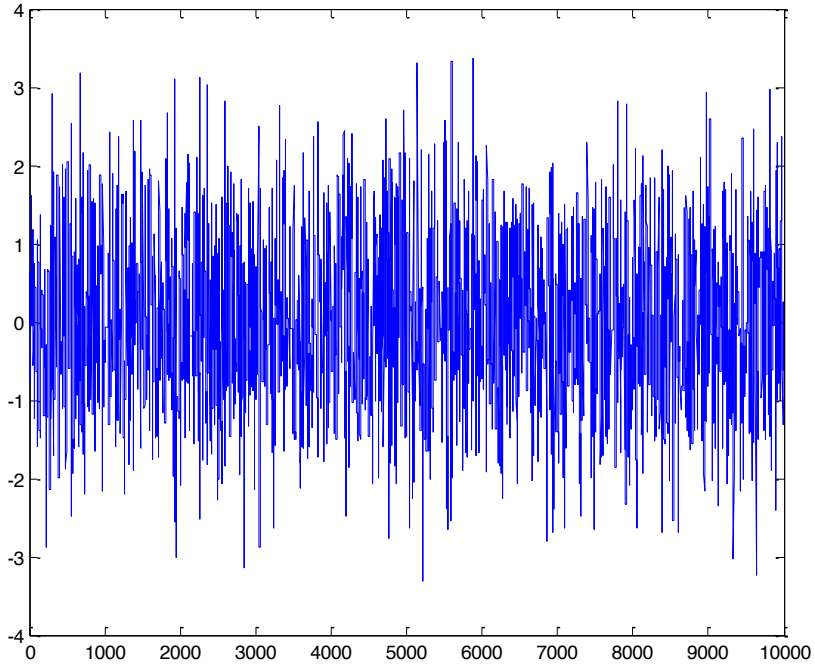
- MCMC output for q_1, we estimate $\mathbb{E}(\theta)= 0.0126$ and $\text{Var}(\theta)=0.9371$.

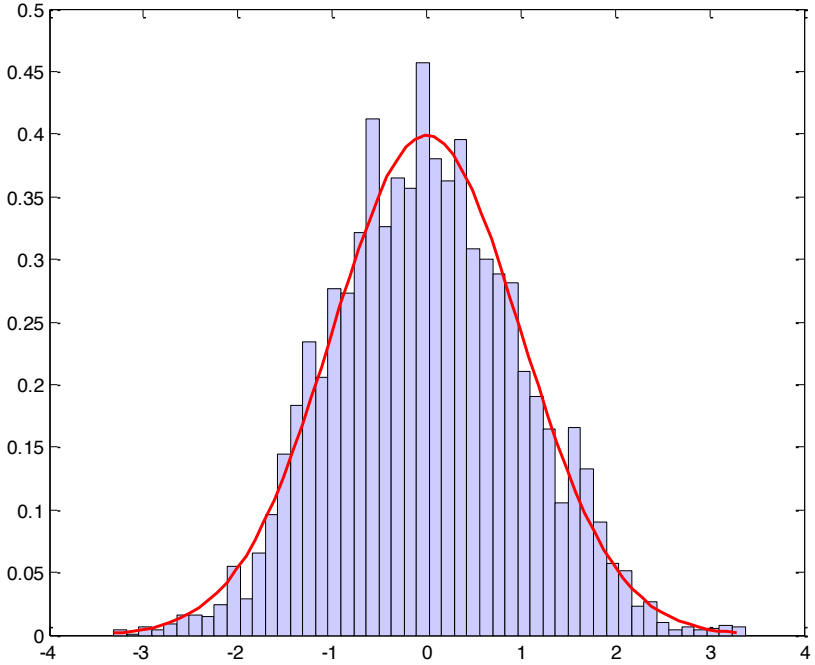


A MatLab implementation is given here

Example

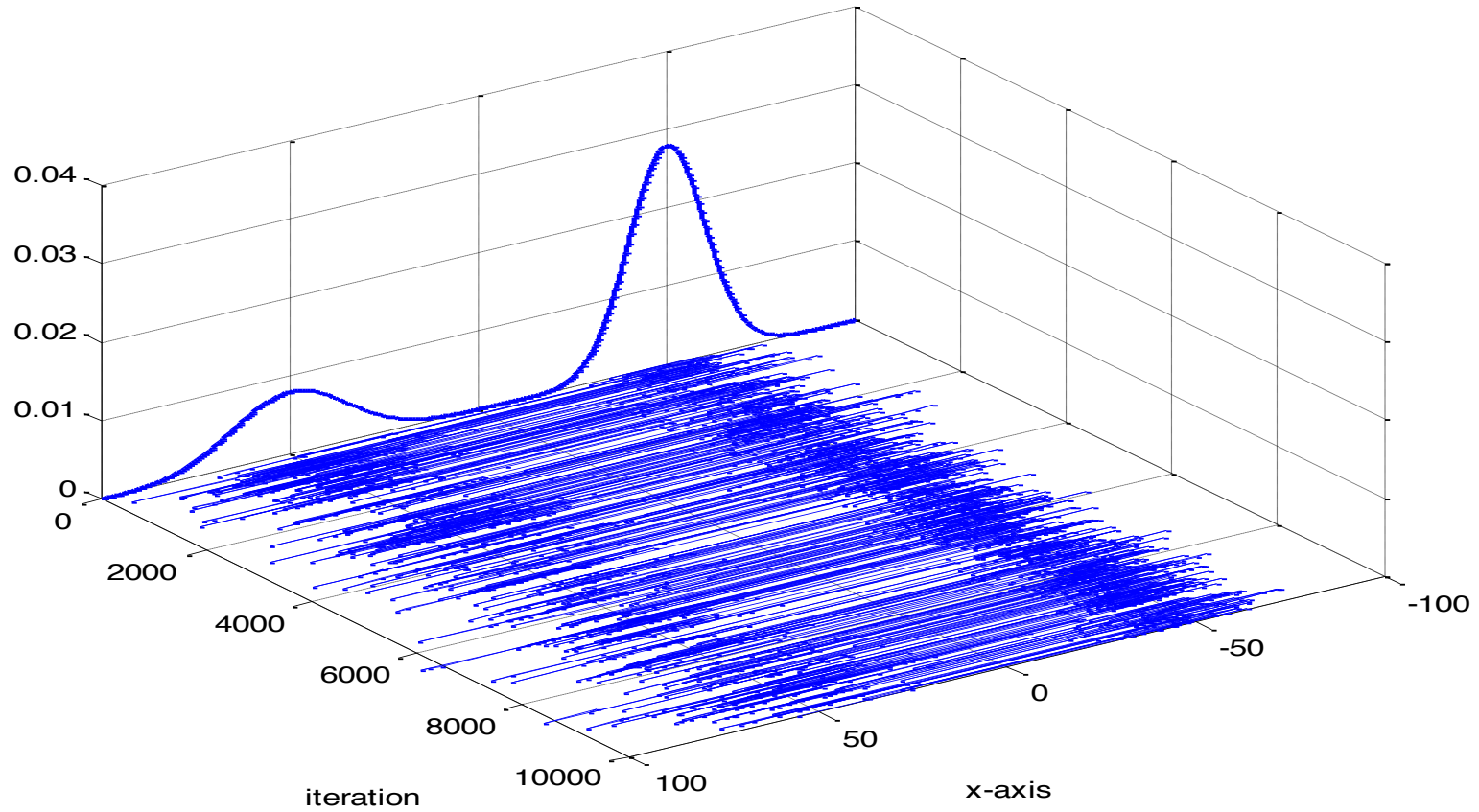
- MCMC output for q_2, we estimate $\mathbb{E}(\theta) = 0.0034$ and $\text{Var}(\theta)=1.0081$.





Example: Bimodal Distribution

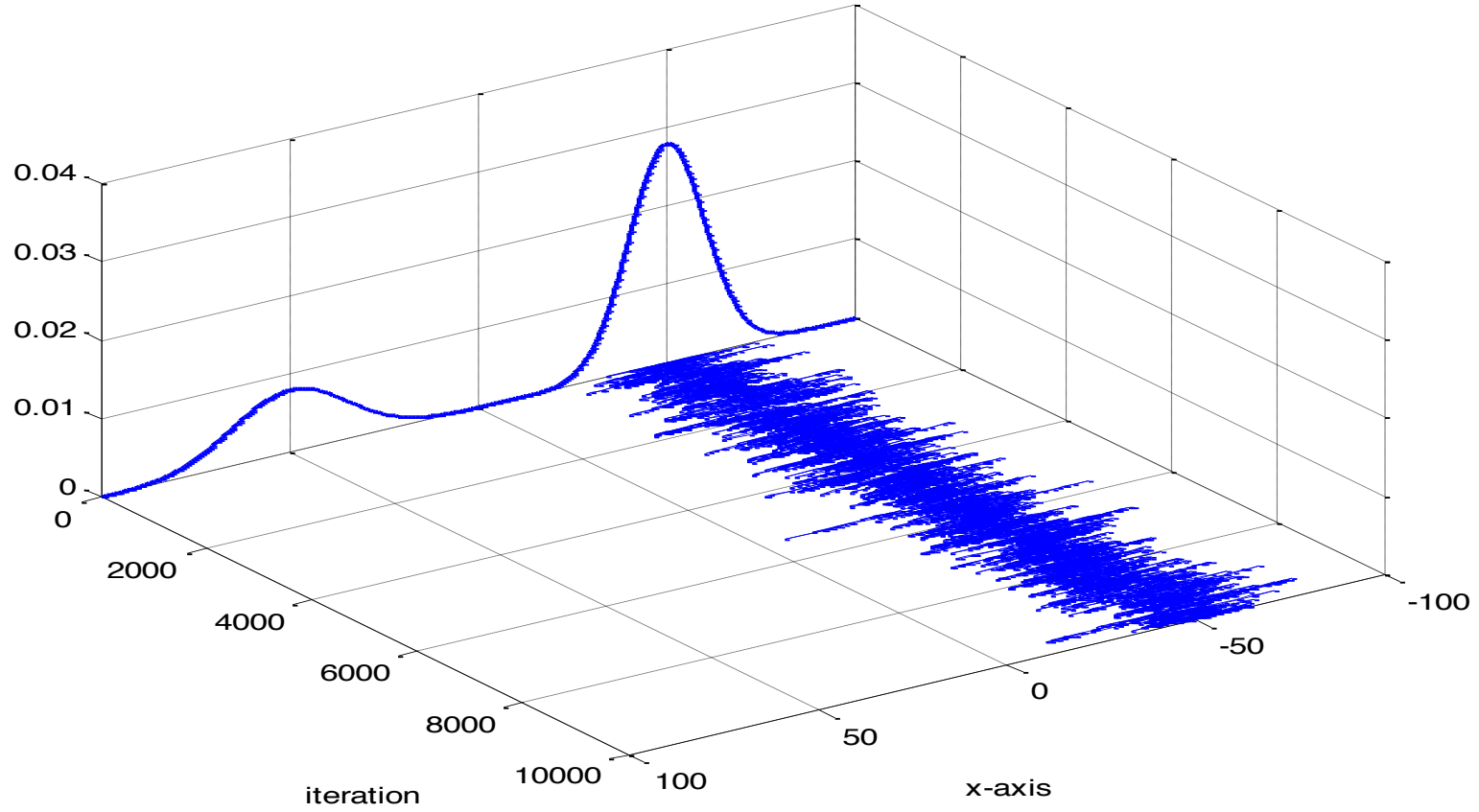
- Exploration of a bimodal distribution using a random walk MH algorithm



A MatLab implementation is given here

Example: Bimodal Distribution

- Bad exploration of a bimodal distribution using a random walk MH algorithm. The variance of the random walk increments is too small.



Random Walk Metropolis-Hastings

- A rule of thump is to have an average acceptance ratio between 0.2 and 0.4.

- You should not adapt σ^2 on the fly in order to achieve an acceptance ratio in that range.

  The chain is not Markov anymore and the desired convergence properties might be lost.

- Heavy tails increments can prevent you from getting trapped in modes.

Independent Metropolis-Hastings

- Independent proposal $q(\theta,\theta') = q(\theta')$ then:

$$\alpha(\theta,\theta') = \min\left(1, \left(\frac{\pi(\theta')}{q(\theta')}\right) \Big/ \left(\frac{\pi(\theta)}{q(\theta)}\right)\right)$$

- If you are using independent proposals then you would like to have $q(\theta) \cong \pi(\theta)$

- Similarly to Rejection sampling or Importance Sampling, you need to ensure that

$$\frac{\pi(\theta)}{q(\theta)} \leq C$$

  to obtain good performance.

- Without the above constraint in the selection of $q(\theta)$, the algorithm might not work at all.

Independent Metropolis-Hastings

- One might argue that since the proposed state does not depend on the previous state, the states of the Markov Chain are independent and therefore the autocorrelation is zero and the achieved convergence rate optimal.

- This is not the case because the proposals are not always accepted!

- In addition, if the proposal focuses on a region of low probability mass, it will spend most of its time there.

Example

- Consider the case where

$$\pi(\theta) \propto e^{-\frac{\theta^2}{2}}$$

- We implement the MH algorithm for

$$q_1(\theta) \propto e^{-\frac{\theta^2}{2(0.2)^2}}$$

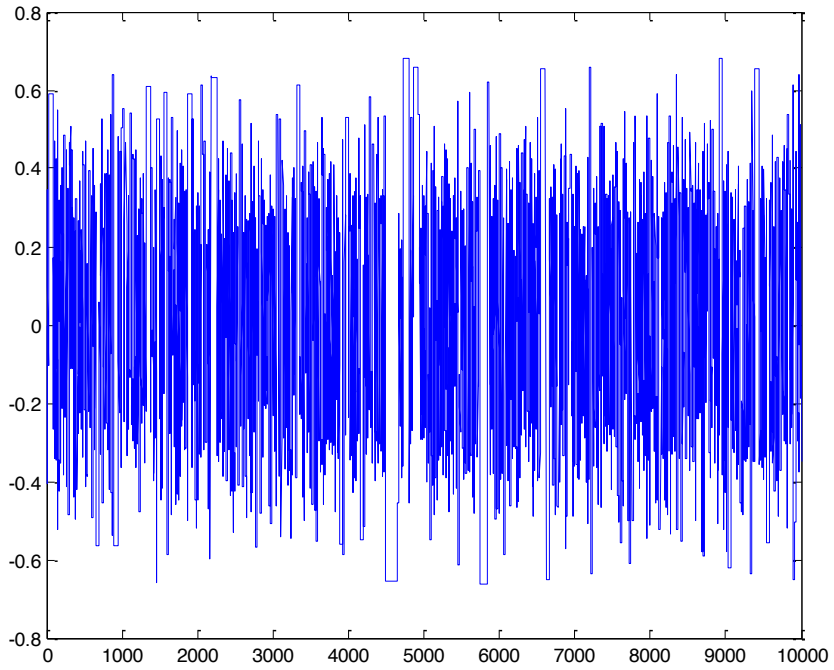
so $\pi(\theta)/q_1(\theta) \to \infty, as\ \theta \to \infty$ and for

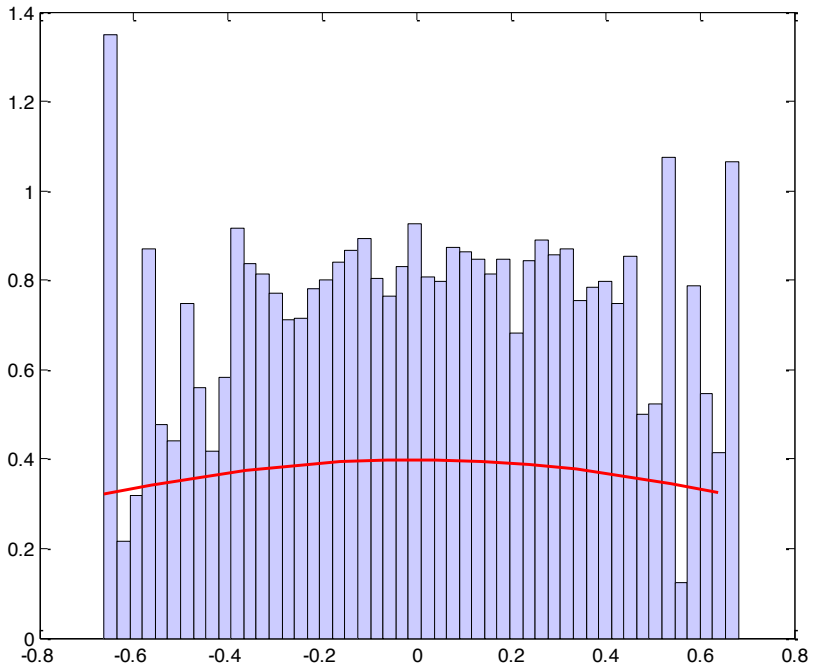
$$q_2(\theta) \propto e^{-\frac{\theta^2}{2(5)^2}}$$

so $\pi(\theta)/q_2(\theta) \le C\ for\ all\ \theta.$

Example

- MCMC output for q_1, we estimate $\mathbb{E}(\theta)=0.0174$ and $\text{Var}(\theta)=0.1374$.

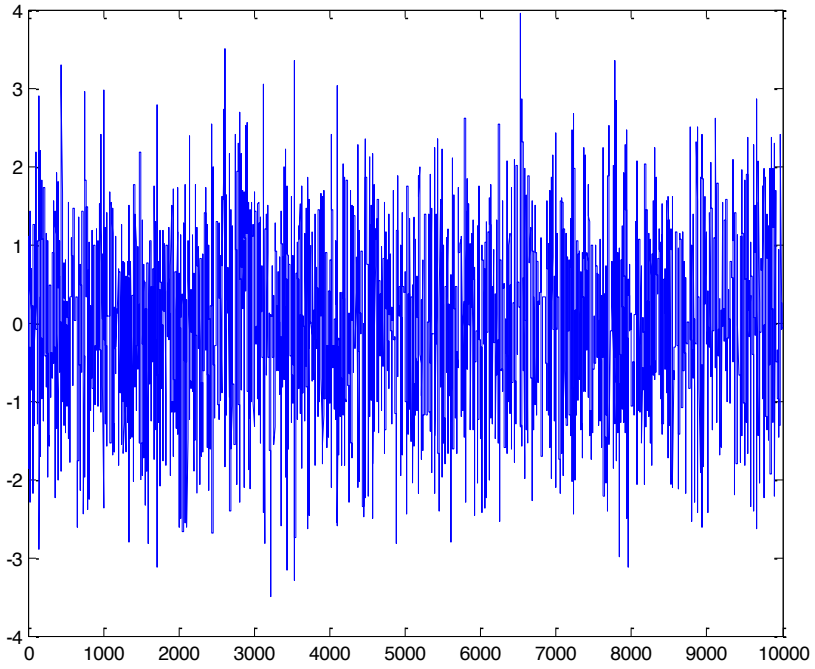


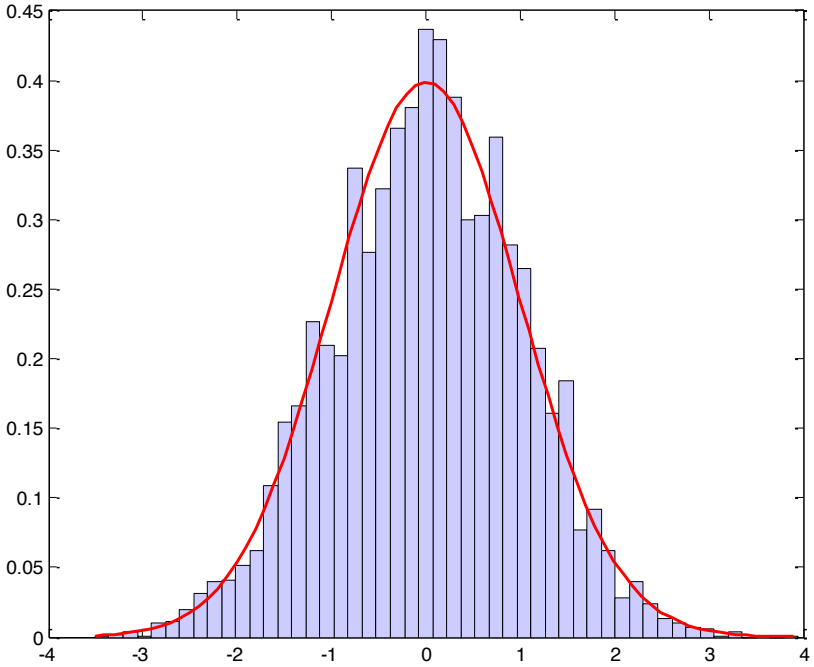


A MatLab implementation is given here

Example

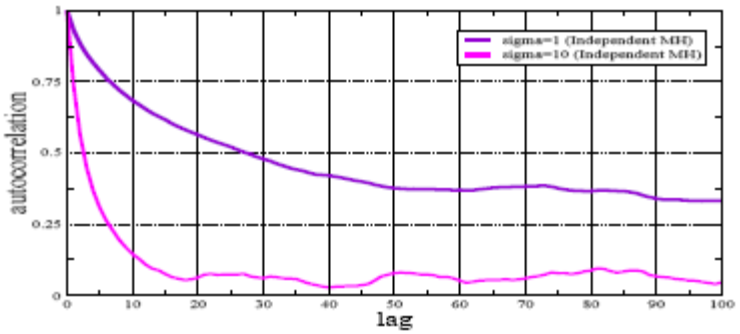
- MCMC output for q_2, we estimate $\mathbb{E}(\theta) = 0.0193$ and $\mathrm{Var}(\theta) = 1.0107$.



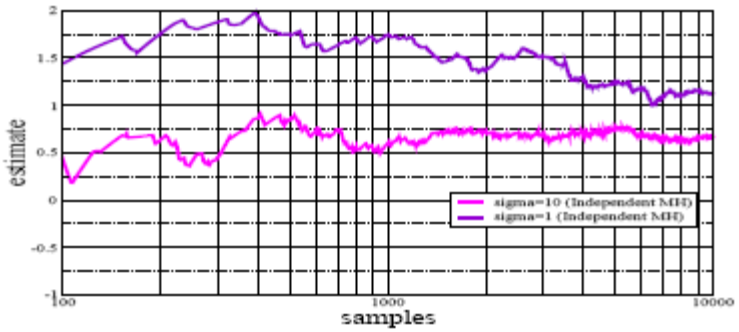


Independent Metropolis-Hastings

- Target: $\pi(x) = 0.25\mathcal{N}(-3, 2) + 0.75\,\mathcal{N}(2, 1)$
- Independent Proposal : $q(x) = \mathcal{N}(0, \sigma^2)$
- Cases shown: $\sigma = 1$, $\sigma = 10$



Autocorrelation for various σ. Acceptance ratio ~ 0.24 for both proposals!



Ergodic mean for the two σ. True value 0.75.

Mixture of Proposals

- In practice, random walk proposals can be used to explore locally the space whereas independent walk proposals can be used to jump into the space.
- A good strategy can be to use a proposal distribution of the following mixture form

$$q(\theta,\theta') = \lambda q_1(\theta') + (1-\lambda) q_2(\theta,\theta')$$

where $0 < \lambda < 1$.

- This algorithm is valid (satisfies all needed properties of transition kernels) as it is a particular case of the MH algorithm.
- Combining random walk (conservative small steps) with independent (large jumps) proposals takes advantage of the merits of both algorithms.

Mixture of MH Kernels

- An alternative is to use a transition kernel

$$K(\theta,\theta') = \lambda K_1(\theta,\theta') + (1-\lambda)K_2(\theta,\theta')$$

where K_1 (respectively, K_2) is an MH algorithm of proposal q_1 (respectively, q_2)

- This algorithm is different from using $q(\theta,\theta') = \lambda q_1(\theta') + (1-\lambda)q_2(\theta,\theta')$.

- It is computationally cheaper and valid as

$$\begin{aligned}\int \pi(\theta)K(\theta,\theta')d\theta &= \lambda\int \pi(\theta)K_1(\theta,\theta')d\theta + (1-\lambda)\int \pi(\theta)K_2(\theta,\theta')d\theta \\ &= \lambda\pi(\theta') + (1-\lambda)\pi(\theta') \\ &= \pi(\theta')\end{aligned}$$

Mixture of MH Kernels

- A sufficient condition to ensure that K is irreducible and aperiodic is to have either K_1 or K_2 irreducible and aperiodic.
- You do NOT need to have both kernels to be irreducible and aperiodic. In the limiting case, you could have $K_2(\theta, \theta') = \delta_\theta(\theta')$ and the total kernel K would still be irreducible and aperiodic if K_1 is irreducible and aperiodic.
- None of the kernels have to be irreducible and aperiodic to ensure that K is irreducible and aperiodic (sufficient but not necessary condition).

Composition of MH Kernels

- Alternatively, we can apply at each iteration of the algorithm first the kernel K_1 then the kernel K_2, i.e. in this case we have at iteration i

$$Z \sim K_1(\theta^{(i-1)}, \cdot) \quad and \quad \theta^{(i)} \sim K_2(Z, \cdot)$$

- The composition of these kernels corresponds to

$$K(\theta, \theta') = \int K_1(\theta, z) K_2(z, \theta') dz$$

If K_1 and K_2 are π-invariant, then the composition is also π-invariant.

- The algorithm admits the right invariant distribution as

$$\int \pi(\theta) K(\theta, \theta') d\theta = \int \left(\int \pi(\theta) K_1(\theta, z) d\theta \right) K_2(z, \theta') dz =$$

$$= \int \pi(z) K_2(z, \theta') dz = \pi(\theta')$$

Composition of MH Kernels

- A sufficient condition to ensure that K is irreducible and aperiodic is to have either K_1 or K_2 irreducible and aperiodic.

- You do NOT need to have both kernels to be irreducible and aperiodic to have K irreducible and aperiodic, e.g. take K_1 irreducible and aperiodic and $K_2(\theta, \theta') = \delta_\theta(\theta')$

- None of the kernels have to be irreducible and aperiodic to ensure that K is irreducible and aperiodic.

Composition of MH Kernels

- The MH algorithm is a simple and very general algorithm to sample from a target distribution $\pi(\theta)$.
- In practice, in the performance of the algorithm, the choice of the proposal distribution is crucial.
- In high dimensional problems, a simple MH algorithm is useless. It will be necessary to use a combination of MH kernels.
- Of course, using mixture and composition of kernels can be a powerful approach.

Applications of Mixture and Composition of MH algorithms

- Consider the target distribution $\pi(\theta_1, \theta_2)$.
- We use two MH kernels to sample from this distribution,
  - the kernel K_1 updates θ_1 and keeps θ_2 fixed whereas
  - the kernel K_2 updates θ_2 and keeps θ_1 fixed.
- We then combine these kernels through mixture or composition.

Description of Transition Kernels

- The proposal $\bar{q}_1(\theta,\theta')$ associated to $K_1(\theta,\theta')$ is given by

$$\bar{q}_1(\theta,\theta') = \bar{q}_1\left((\theta_1,\theta_2),(\theta_1',\theta_2')\right) = q_1\left((\theta_1,\theta_2),\theta_1'\right)\delta_{\theta_2}\left(\theta_2'\right)$$

- The acceptance probability is given by $\alpha_1(\theta,\theta') = \min\left(1, r_1(\theta,\theta')\right)$ where:

$$r_1(\theta,\theta') = \frac{\pi(\theta')\bar{q}_1(\theta',\theta)}{\pi(\theta)\bar{q}_1(\theta,\theta')} = \frac{\pi(\theta_1',\theta_2')q_1\left((\theta_1',\theta_2'),\theta_1\right)\delta_{\theta_2'}(\theta_2)}{\pi(\theta_1,\theta_2)q_1\left((\theta_1,\theta_2),\theta_1'\right)\delta_{\theta_2}(\theta_2')}$$

$$= \frac{\pi(\theta_1',\theta_2)q_1\left((\theta_1',\theta_2),\theta_1\right)}{\pi(\theta_1,\theta_2)q_1\left((\theta_1,\theta_2),\theta_1'\right)}$$

$$= \frac{\pi(\theta_1' \mid \theta_2)q_1\left((\theta_1',\theta_2),\theta_1\right)}{\pi(\theta_1 \mid \theta_2)q_1\left((\theta_1,\theta_2),\theta_1'\right)}$$

- This move is also equivalent to an MH step of invariant $\pi(\theta_1 \mid \theta_2)$

Description of Transition Kernels

- The proposal $\bar{q}_2(\theta,\theta')$ associated to $K_2(\theta,\theta')$ is given by

$$\bar{q}_2(\theta,\theta') = \bar{q}_2\left((\theta_1,\theta_2),(\theta_1',\theta_2')\right) = q_2\left((\theta_1,\theta_2),\theta_2'\right)\delta_{\theta_1}\left(\theta_1'\right)$$

- The acceptance probability is given by $\alpha_2(\theta,\theta') = \min\left(1, r_2(\theta,\theta')\right)$ where:

$$r_2(\theta,\theta') = \frac{\pi(\theta')\bar{q}_2(\theta',\theta)}{\pi(\theta)\bar{q}_2(\theta,\theta')} = \frac{\pi(\theta_1',\theta_2')q_2\left((\theta_1',\theta_2'),\theta_2\right)\delta_{\theta_1'}(\theta_1)}{\pi(\theta_1,\theta_2)q_2\left((\theta_1,\theta_2),\theta_2'\right)\delta_{\theta_1}(\theta_1')} =$$

$$= \frac{\pi(\theta_1,\theta_2')q_2\left((\theta_1,\theta_2'),\theta_2\right)}{\pi(\theta_1,\theta_2)q_2\left((\theta_1,\theta_2),\theta_2'\right)} =$$

$$= \frac{\pi(\theta_2'\,|\,\theta_1)q_2\left((\theta_1,\theta_2'),\theta_2\right)}{\pi(\theta_2\,|\,\theta_1)q_2\left((\theta_1,\theta_2),\theta_2'\right)}$$

- This move is also equivalent to an MH step of invariant $\pi(\theta_2\,|\,\theta_1)$

Composition of MH Kernels

- Assume we use a composition of these kernels, then the resulting algorithm proceeds as follows at iteration i.

- **MH Step to Update Component 1**

  - *Sample* $\theta_1^* \sim q_1\left(\left(\theta_1^{(i-1)}, \theta_2^{(i-1)}\right), \cdot\right)$ *and compute*

$$\alpha_1\left(\left(\theta_1^{(i-1)}, \theta_2^{(i-1)}\right), \left(\theta_1^*, \theta_2^{(i-1)}\right)\right) = \min\left(1, \frac{\pi\left(\theta_1^* \mid \theta_2^{(i-1)}\right) q_1\left(\left(\theta_1^*, \theta_2^{(i-1)}\right), \theta_1^{(i-1)}\right)}{\pi\left(\theta_1^{(i-1)} \mid \theta_2^{(i-1)}\right) q_1\left(\left(\theta_1^{(i-1)}, \theta_2^{(i-1)}\right), \theta_1^*\right)}\right)$$

  - *With probability* $\alpha_1\left(\left(\theta_1^{(i-1)}, \theta_2^{(i-1)}\right), \left(\theta_1^*, \theta_2^{(i-1)}\right)\right)$, *set* $\theta_1^{(i)} = \theta_1^*$;

  otherwise set $\theta_1^{(i)} = \theta_1^{(i-1)}$

Composition of MH Kernels

- Assume we use a composition of these kernels, then the resulting algorithm proceeds as follows at iteration i.

- **MH Step to Update Component 2**

  - *Sample* $\theta_2^* \sim q_2\left(\left(\theta_1^{(i)}, \theta_2^{(i-1)}\right), \cdot\right)$ *and compute*

$$\alpha_2\left(\left(\theta_1^{(i)}, \theta_2^{(i-1)}\right), \left(\theta_1^{(i)}, \theta_2^{*}\right)\right) = \min\left(1, \frac{\pi\left(\theta_2^* \mid \theta_1^{(i)}\right) q_2\left(\left(\theta_1^{(i)}, \theta_2^{*}\right), \theta_2^{(i-1)}\right)}{\pi\left(\theta_2^{(i-1)} \mid \theta_2^{(i)}\right) q_1\left(\left(\theta_1^{(i)}, \theta_2^{(i-1)}\right), \theta_2^{*}\right)}\right)$$

  - *With probability* $\alpha_2\left(\left(\theta_1^{(i)}, \theta_2^{(i-1)}\right), \left(\theta_1^{(i)}, \theta_2^{*}\right)\right)$, *set* $\theta_2^{(i)} = \theta_2^*$;

  otherwise set $\theta_2^{(i)} = \theta_2^{(i-1)}$

Mixture of MH Kernels

- Assume we use an even mixture of these kernels, then the resulting algorithm proceeds as follows at iteration i.

  - Sample the index of the component to update $J \sim \mathcal{U}[1,2]$
  - Sample $\theta_{-J}^{(i)} = \theta_{-J}^{(i-1)}$
  - *Sample* $\theta_J^* \sim q_J\left(\left(\theta_1^{(i-1)}, \theta_2^{(i-1)}\right), \cdot\right)$ *and compute*

$$\alpha_J\left(\left(\theta_1^{(i)}, \theta_2^{(i-1)}\right), \left(\theta_J^{(i)}, \theta_{-J}^*\right)\right) = \min\left(1, \frac{\pi\left(\theta_J^* \mid \theta_{-J}^{(i)}\right) q_J\left(\left(\theta_J^{(i)}, \theta_{-J}^*\right), \theta_J^{(i-1)}\right)}{\pi\left(\theta_J^{(i-1)} \mid \theta_{-J}^{(i)}\right) q_K\left(\left(\theta_J^{(i-1)}, \theta_{-J}^{(i)}\right), \theta_J^*\right)}\right)$$

  - *With probability* $\alpha_J\left(\left(\theta_J^{(i-1)}, \theta_J^{(i-1)}\right), \left(\theta_J^*, \theta_{-J}^{(i)}\right)\right)$ *, set* $\theta_J^{(i)} = \theta_J^*$ *;*

otherwise set $\theta_J^{(i)} = \theta_J^{(i-1)}$

Properties

- It is clear that in such cases both K_1 and K_2 are NOT irreducible and aperiodic.

  $\Rightarrow$ Each of them only updates one component!!!!

- However, the composition and mixture of these kernels can be irreducible and aperiodic because then all the components are updated.

Using Full Conditionals Leads to Gibbs Sampler

- Consider now the case where

$$q_1\left((\theta_1,\theta_2),\theta_1'\right)=\pi\left(\theta_1'\mid\theta_2\right)$$

then

$$r_1(\theta,\theta')=\frac{\pi\left(\theta_1'\mid\theta_2\right)q_1\left(\left(\theta_1',\theta_2\right),\theta_1\right)}{\pi\left(\theta_1\mid\theta_2\right)q_1\left(\left(\theta_1,\theta_2\right),\theta_1'\right)}=\frac{\pi\left(\theta_1'\mid\theta_2\right)\pi\left(\theta_1\mid\theta_2\right)}{\pi\left(\theta_1\mid\theta_2\right)\pi\left(\theta_1'\mid\theta_2\right)}=1$$

- Similarly if $q_2\left((\theta_1,\theta_2),\theta_2'\right)=\pi\left(\theta_2'\mid\theta_2\right)$, then $r_2(\theta,\theta')=1$

- If you take for proposal distributions in the MH kernels the full conditional distributions then you have the Gibbs sampler!

General Hybrid Algorithm

- Generally speaking, to sample from $\pi(\theta)$ where $\theta = (\theta_1, \theta_2, ..., \theta_p)$, we can use the following algorithm at iteration i.

- Iteration i, $i \geq 1$
  - For k=1:p
    - Sample $\theta_k^{(i)}$ using an MH step of proposal distribution

$$q_k\left((\theta_{-k}^{(i)}, \theta_k^{(i-1)}), \theta_k^{'}\right) \text{ and target } \pi(\theta_k \mid \theta_{-k}^{(i)})$$

where $\theta_{-k}^{(i)} = (\theta_1^{(i)}, ..., \theta_{k-1}^{(i)}, \theta_{k+1}^{(i-1)}, ..., \theta_p^{(i-1)})$

General Hybrid Algorithm

- If we have $q_k\left(\theta_{1:p}, \theta_k^{'}\right) = \pi\left(\theta_k^{'} \mid \theta_{-k}\right)$ then we are back to the Gibbs sampler.

- We can update some parameters according to $\pi\left(\theta_k^{'} \mid \theta_{-k}\right)$ (and the move is automatically accepted) and others according to different proposals.

- Example: Assume we have $\pi(\theta_1, \theta_2)$ where it is easy to sample from $\pi(\theta_1 \mid \theta_2)$ and then use an MH step of invariant distribution $\pi(\theta_2 \mid \theta_1)$.

General Hybrid Algorithm

- At iteration $i, i \geq 1$
  - Sample $\theta_1^{(i)} \sim \pi\left(\theta_1 \mid \theta_2^{(i-1)}\right)$
  - Sample $\theta_2^{(i)}$ using an MH step of proposal distribution

$$q_2\left((\theta_1^{(i)}, \theta_2^{(i-1)}), \theta_2\right) \text{ and target } \pi\left(\theta_2 \mid \theta_1^{(i)}\right)$$

- Remark: There is no need to run the MH Algorithm multiple steps to ensure that $\theta_2^{(i)} \sim \pi(\theta_2 \mid \theta_2^{(i-1)})$

Metropolis-Hastings with the Prior as the Proposal

- Considering sampling from the posterior

$$p(\boldsymbol{x} \mid data) \sim p(data \mid \boldsymbol{x}) p(\boldsymbol{x})$$

- Let us use independent Metropolis-Hastings with the prior as the proposal distribution

$$\textit{acceptance ratio } a(\boldsymbol{x} \mid \boldsymbol{y}) = \min\left\{1, \frac{p(data \mid \boldsymbol{y}) p(\boldsymbol{y})}{p(data \mid \boldsymbol{x}) p(\boldsymbol{x})} \frac{p(\boldsymbol{x})}{p(\boldsymbol{y})}\right\} =$$

$$= \min\left\{1, \frac{p(data \mid \boldsymbol{y})}{p(data \mid \boldsymbol{x})}\right\}$$

- This works if the effect of the data is not significant – i.e. the prior is close to the likelihood.

Using Gradient Information to Build the Proposal

- We usually want to sample candidates in regions of high probability

- We can use

$$\theta' = \theta + \frac{\sigma^2}{2}\nabla \log \pi(\theta) + \sigma V, V \sim \mathcal{N}(0,1)$$

where σ^2 is selected such that the acceptance ratio is approximately 0.57. Why 0.57

- The motivation is that, we know that in continuous-time

$$d\theta_t = \frac{1}{2}\nabla \log \pi(\theta) + \sigma dW_t$$

admits π as an invariant distribution.

Alternative Acceptance Probabilities

- The standard MH algorithm uses the acceptance probability:

$$\alpha(\theta,\theta') = \min\left(1, \frac{\pi(\theta')q(\theta',\theta)}{\pi(\theta)q(\theta,\theta')}\right)$$

- This is not necessary and one can use any function

$$\alpha(\theta,\theta') = \frac{\delta(\theta',\theta)}{\pi(\theta)q(\theta,\theta')}$$

which is such that $\delta(\theta',\theta) = \delta(\theta,\theta') \; and \; 0 \le \alpha(\theta,\theta') \le 1$

- For example (Baker, 1965)

$$\alpha(\theta,\theta') = \frac{\pi(\theta')q(\theta',\theta)}{\pi(\theta')q(\theta',\theta)+\pi(\theta)q(\theta,\theta')}$$

Alternative Acceptance Probabilities

- Indeed, one can check that

$$K(\theta,\theta')=\alpha(\theta,\theta')q(\theta,\theta')+\left(1-\int\alpha(\theta,u)q(\theta,u)du\right)\delta_\theta(\theta')$$

  is π-reversible.

- We have:

$$\begin{aligned}\pi(\theta)\alpha(\theta,\theta')q(\theta,\theta')&=\pi(\theta)\frac{\delta(\theta',\theta)}{\pi(\theta)q(\theta,\theta')}q(\theta,\theta')\\&=\delta(\theta,\theta')\\&=\delta(\theta',\theta)\\&=\pi(\theta')\alpha(\theta',\theta)q(\theta',\theta)\end{aligned}$$

- The MH acceptance is favored as it increases the acceptance probability.

Alternative Acceptance Probabilities

- The MH algorithm is a simple and very general algorithm to sample from a target distribution $\pi(\theta)$.

- In practice, the choice of the proposal distribution is absolutely crucial on the performance of the algorithm.

- In high dimensional problems, a simple MH algorithm may be useless. It will be necessary to use a combination of MH kernels.

Discussion

- In practice, we divide the parameter space $\theta = \left(\theta_1, ..., \theta_p\right)$
- We update each parameter θ_k according to an MH step of proposal distribution $q_k(\theta_{1:p}, \theta_k^{'}) = q_k\left((\theta_{-k}, \theta_k), \theta_k^{'}\right)$ and invariant distribution $\pi(\theta_k \mid \theta_{-k})$

Hybrid Metropolis Proposal

- Hybrid MC is essentially Metropolis with a special choice of a proposal.
- Assume that you want to sample $x \sim \pi(x)$, where $\pi(x)$ is known up to a proportionality constant.
- Consider that x represents the position of some real particles. Then write:

$$\pi(x) = \exp(\log(\pi(x)) = \exp(-V(x)) \quad (1)$$

where we have defined: $V(x) = -\log(\pi(x))$.

- Look at Eq. (1) and notice its similarity with the Boltzmann distribution at inverse temperature equal to one (statistical mechanics).
- Think of $V(x) = -\log(\pi(x))$ as the POTENTIAL of the system at x.
- To complete the picture, introduce the momenta p (of the same dimension as x) and write a probability distribution in the extended space:

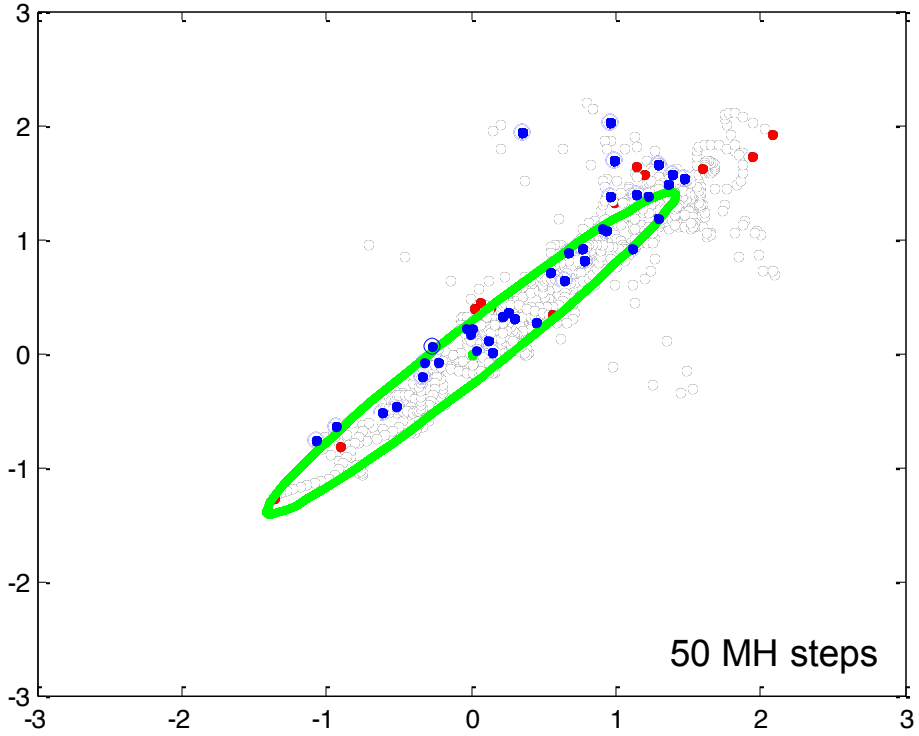
$$x, p \sim \pi(x, p) = q(x) \times \mathcal{N}(p|0,1) = \exp(-V(x) - p^2 / 2)$$

Example: Hybrid Metropolis Proposal

$$x, p \sim \pi(x, p) = \pi(x) \times \mathcal{N}(p|0,1) = \exp(-V(x) - p^2/2)$$

- Write $H(x,p) = V(x) + p^2/2$ for the Hamiltonian of the system.
- To construct the proposal, we bring into the picture the ergodicity the dynamics described by the Hamiltonian.
- If you integrate the equations of motion described by $H(x,p)$ starting at any initial condition for a long time, you will get a sample from the Boltzmann probability distribution $\pi(x,p)$.
- Based on this our hybrid Metropolis proposal is constructed as:
  - 1. Sample an initial p from a Gaussian (in $\pi(x,p), x$ and p are decoupled and the probability distribution of p is $\mathcal{N}(p|0,1)$)
  - 2. Evolve the equations of motion for a finite amount of time using a finite time step.
  - 3. Use the x at the final step as the proposed move.
- Notice that the proposal built this way is reversible if the integration scheme is reversible. To guarantee this, we use a the Leapfrog integration scheme for the integration of motion which has the property of preserving the value of the Hamiltonian.

Example: Hybrid Metropolis Proposal



MatLab Implementation with animation of the dynamics