Special cases of the Metropolis-Hastings algorithm

Depending on the choice of $Q(x^\star|x_{i-1})$ different special cases result. In particular, three classes are important

- The independence proposal
- The Metropolis algorithm: random walk proposal
- Metropolis-adjusted Langevin algorithm (MALA): Langevin proposal

Independence proposal

- The proposal distribution does not depend on the current value x_{i-1}

$$Q(x|x_{i-1}) = Q(x).$$

- $Q(x)$ is an approximation to $\pi(x)$
  $\Rightarrow$ Acceptance rate should be close to 1.
- The sampler is closer to rejection sampler. However, here if we reject, then we retain the sample.

Experience:

- Performance is either very good or very bad, usually very bad.
- The tails of the proposal distribution should be at least as heavy as the tails of the target distribution.

The Metropolis algorithm

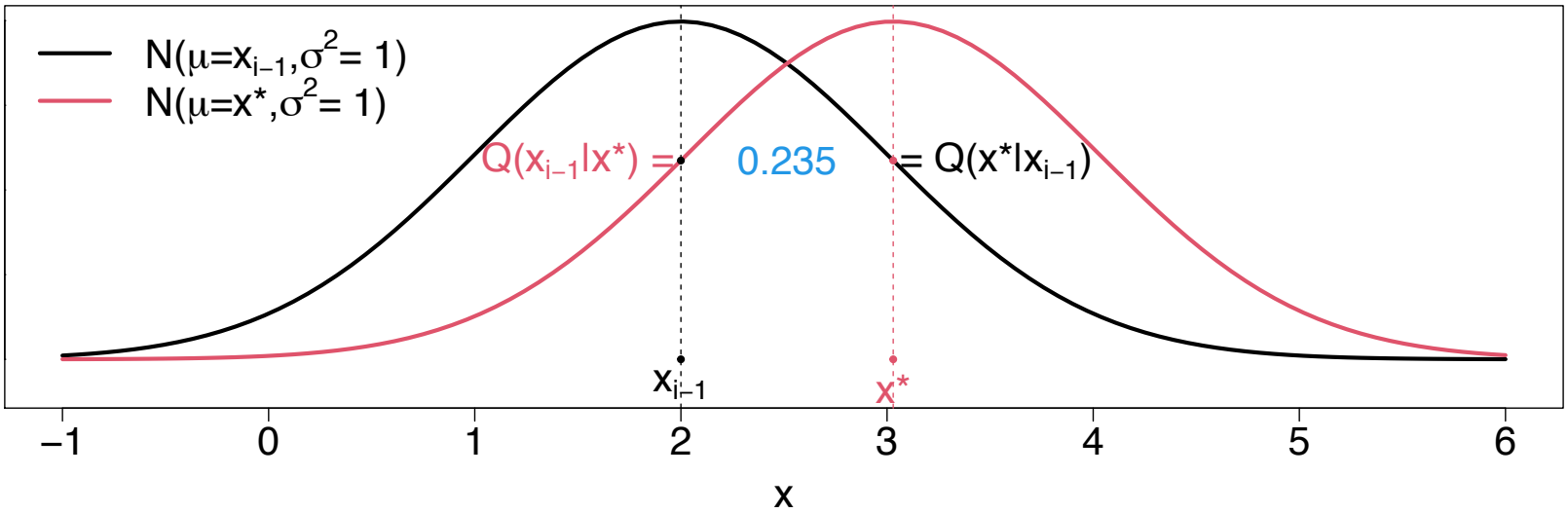
The proposal density is symmetric around the current value, that means

$$Q(x_{i-1}|x^\star) = Q(x^\star|x_{i-1}).$$

Hence,

$$\alpha = \min\left(1, \frac{\pi(x^\star)}{\pi(x_{i-1})} \times \frac{Q(x_{i-1}|x^\star)}{Q(x^\star|x_{i-1})}\right) = \min\left(1, \frac{\pi(x^\star)}{\pi(x_{i-1})}\right)$$

A particular case is the random walk proposal, defined as the current value x_{i-1} plus a random variate of a 0-centred symmetric distribution.



Examples for random walks proposal

Assume x is scalar.

Then all proposal kernels, which add a random variable generated from a zero-symmetrical distribution to the current value x_{i-1}, are random walk proposals. For example:

$$x^\star \sim \mathcal{N}(x_{i-1}, \sigma^2)$$

$$x^\star \sim t_\nu(x_{i-1}, \sigma^2)$$

$$x^\star \sim \mathcal{U}(x_{i-1} - d, x_{i-1} + d)$$

Efficiency of the Metropolis-Hastings algorithm

The efficiency and performance of the Metropolis-Hastings algorithm depends crucially on the relative frequency of acceptance.

An acceptance rate of one is not always good. Consider the random walk proposal:

- Too large acceptance rate $\Rightarrow$ Slow exploration of the target density.
- Too small acceptance rate $\Rightarrow$ Large moves are proposed, but rarely accepted.

Tuning the acceptance rate:

- For random walk proposals, acceptance rates between 20% and 50% are typically recommended. They can be achieved by changing the variance of the proposal distribution.
- For independence proposals a high acceptance rate is desired, which means that the proposal density is close to the target density.

Example: Random walk proposal

Exploration of a standard Gaussian distribution ($\mathcal{N}(0,1)$) using a random walk Metropolis algorithm. As proposal assume a Gaussian distribution with variance σ^2, where.

- $\sigma = 0.24$
- $\sigma = 2.4$
- $\sigma = 24$

See R-code `demo_mcmcRW.R`.

Random walk MH for $N_2(0, \Sigma)$

For the bivariate Gaussian with mean 0, identity variance, and correlation ρ, the Random walk MH sampler goes as follows:

Algorithm

- Initiate $\boldsymbol{x}_0 = (x_0^1, x_0^2)$, $i = 0$.
- Iterate while $i < B$,
  1. Sample $\boldsymbol{x}^* \sim N(\boldsymbol{x}_i, \sigma^2 \boldsymbol{I})$
  2. Accept $\boldsymbol{x}_{i+1} = \boldsymbol{x}^*$ with probability
     $\alpha = \min\{1, \exp(-1/2 \boldsymbol{x}^{*t} \boldsymbol{\Sigma}^{-1} \boldsymbol{x}^* + 1/2 \boldsymbol{x}_i^t \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i)\}$
     Else set $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i$.
  3. $i = i + 1$

The symmetric proposal density cancels in the acceptance rate.

Metropolis-adjusted Langevin algorithm (MALA)

Langevin proposal: $Q(\boldsymbol{x}^\star \mid \boldsymbol{x}_{i-1}) = N(\boldsymbol{x}_{i-1} + \frac{\sigma^2}{2}\nabla \log \pi(\boldsymbol{x}_{i-1}), \sigma^2 \boldsymbol{I})$.

Only requires tuning of $\sigma > 0$. Asymptotically optimal (under some assumptions) to have acceptance rate about 0.57.

Advantage: Step size is longer that using a random walk proposal $\Rightarrow$ samples are less dependent.

Disadvantages:

- Calculation of gradient at every iteration
- Might get stuck in local maximimum
- Tails might be difficult to explore

Gibbs-Sampling algorithm

Idea: Sequentially sampling from univariate conditional distributions (which are often available in closed form).

1. Select starting values $\boldsymbol{x}_0$ and set $i = 0$.
2. Repeatedly:

$$
\begin{aligned}
&\text{Sample} \quad x_{i+1}^1 | \cdot \sim \pi(x^1 | x_i^2, \ldots, x_i^p) \\
&\text{Sample} \quad x_{i+1}^2 | \cdot \sim \pi(x^2 | x_{i+1}^1, x_i^3, \ldots, x_i^p) \\
&\vdots \\
&\text{Sample} \quad x_{i+1}^{p-1} | \cdot \sim \pi(x^{p-1} | x_{i+1}^1, x_{i+1}^2, \ldots, x_{i+1}^{p-2}, x_i^p) \\
&\text{Sample} \quad x_{i+1}^p | \cdot \sim \pi(x^p | x_{i+1}^1, \ldots, x_{i+1}^{p-1})
\end{aligned}
$$

   where $|\cdot$ denotes conditioning on the most recent updates of all other elements of $\boldsymbol{x}$.

3. Increment i and go to step 2.

Gibbs sampler for $N_2(0, \Sigma)$

For the bivariate Gaussian with mean 0, identity variance, and correlation ρ, the Gibbs sampler goes as follows:

Algorithm

- Initiate $\boldsymbol{x}_0 = (x_0^1, x_0^2)$, $i = 0$.
- Iterate while $i < B$,
  1. Sample $x_{i+1}^1 \sim N(\rho x_i^2, 1 - \rho^2)$
  2. Sample $x_{i+1}^2 \sim N(\rho x_{i+1}^1, 1 - \rho^2)$
  3. $i = i + 1$

Why is the acceptance rate 1?

For ease of notation let x denote the current state and $x^\star$ the proposed new state where we update the $j-$th component of x, so that:

$$x = (x^1, \ldots, x^{j-1}, x^j, x^{j+1}, \ldots, x^p)^\top$$
$$x^\star = (x^1, \ldots, x^{j-1}, x^{\star,j}, x^{j+1}, \ldots, x^p)^\top$$

where $x^{\star,j}$ denotes the propsed value for the $j-$th component. Then

$$\frac{\pi(x^\star)}{\pi(x)} \cdot \frac{Q(x \mid x^\star)}{Q(x^\star \mid x)} = \frac{\pi(x^{\star,j} \mid x^{\star,-j})\pi(x^{\star,-j})}{\pi(x^j \mid x^{-j})\pi(x^{-j})} \cdot \frac{Q(x \mid x^\star)}{Q(x^\star \mid x)}$$
$$= \frac{\pi(x^{\star,j} \mid x^{-j})\pi(x^{-j})}{\pi(x^j \mid x^{-j})\pi(x^{-j})} \cdot \frac{Q(x \mid x^\star)}{Q(x^\star \mid x)}$$
$$= \frac{\pi(x^{\star,j} \mid x^{-j})\pi(x^{-j})}{\pi(x^j \mid x^{-j})\pi(x^{-j})} \cdot \frac{\pi(x^j \mid x^{\star,-j})}{\pi(x^{\star,j} \mid x^{-j})}$$
$$= 1$$

Remarks on Gibbs sampling

- High dimensional updates of $\boldsymbol{x}$ can be boiled down to scalar updates.
- Visiting schedule: Various approaches exist (and can be justified) to ordering the variables in the sampling loop. One approach is random sweeps: variables are chosen at random to resample.
- Gibbs sampling assumes that it is easy to sample from the full-conditional distribution. This is sometimes not so easy. Alternatively, a Metropolis-Hastings proposal can be used for the j-th component, i.e. Metropolis-within-Gibbs $\Rightarrow$ Hybrid Gibbs sampler.

Remarks on Gibbs sampling

- Blocking or grouping is possible, that means not all elements of $\boldsymbol{x}$ are treated individually. Might be useful when elements of $\boldsymbol{x}$ are correlated.
- Care must be taken when improper prior are used, which may lead to an improper posterior distribution. Impropriety implies that there does not exist a joint density to which the full-conditional distributions correspond.

Example: A Bayesian hierarchical model

Example from George et al. (1993) regarding the analysis of 10 power plants.

- y_i number of failures of pump i
- t_i length of operation time of pump i (in kilo hours)

Model:

$$y_i \mid \lambda_i \sim \text{Po}(\lambda_i t_i)$$

Conjugate prior for λ_i:

$$\lambda_i \mid \alpha, \beta \sim \text{G}(\alpha, \beta)$$

Hyper-prior on α and β:

$$\alpha \sim \text{Exp}(1.0) \qquad \beta \sim \text{G}(0.1, 10.0)$$

Example: A Bayesian hierarchical model(II)

The posterior of the 12 parameters $(\alpha, \beta, \lambda_1, \ldots, \lambda_{10})$ given $y_1, \ldots, y_{10}$ is proportional to

$$\pi(\alpha, \beta, \lambda_1, \ldots, \lambda_{10} \mid y_1, \ldots, y_{10}) \propto \pi(\alpha)\pi(\beta) \prod_{i=1}^{10} [\pi(\lambda_i \mid \alpha, \beta)\pi(y_i \mid \lambda_i)]$$

$$\propto e^{-\alpha} \beta^{0.1-1} e^{-10\beta} \left\{ \prod_{i=1}^{10} \exp(-\lambda_i t_i) \lambda_i^{y_i} \right\} \left\{ \prod_{i=1}^{10} \exp(-\beta\lambda_i) \lambda_i^{\alpha-1} \right\} \left[\frac{\beta^\alpha}{\Gamma(\alpha)} \right]^{10}.$$

This posterior is not of closed form.

What are the full conditional distributions?

Implementation and convergence diagnostics

Source: http://i.telegraph.co.uk/multimedia/archive/02365/coding_alamy_2365972b.jpg

Numerical note

How should you compute

$$\alpha = \min\left(1, \frac{\pi(x^\star)}{\pi(x_{i-1})} \times \frac{Q(x_{i-1}|x^\star)}{Q(x^\star|x_{i-1})}\right)$$

Convergence

- If well constructed, the Markov chain is guaranteed to have the posterior as limiting distribution.
- However, this does not tell you how long you have to run the MCMC algorithm til convergence.
  - The initial position may have a big influence.
  - The proposal distribution may lead to low acceptance rates.
  - The chain may get caught in a local maximum of the likelihood surface.
- We say the Markov chain mixes well if it can
  - reach the posterior quickly, and
  - moves quickly around the posterior modes.

Convergence diagnostics

Valid inferences from sequences of MCMC outputs are based on the assumption that the outputs are from the desired target distribution.

- There is no overall minimum number of samples to ensure approximation.
- Consequently methods for testing convergence, known as convergence diagnostics, have to be applied.
- However it has to emphasised that these diagnostics do not guarantee convergence.

Trace plots

An initial possibility for deciding if a MCMC output does not converge to the desired posterior distributions is to look at the sample trace for each variable.

- If our chain is taking a long time to move around the parameter space, then it will take longer to converge.
- If the samples form a homogene band (no wave movements or other rare fluctuations), convergence might be indicated.
- Vastly different values at the beginning of the trace indicate burn-in iterations, which should be discarded.

Autocorrelation

To examine dependencies of successive MCMC samples, the autocorrelation function can be used. Let $x_1, \ldots, x_N$, where N denotes the number of samples, denote our MCMC chain.

The lag k autocorrelation $\rho(k)$ is the correlation between every draw and its k-th lag. For N reasonably large

$$\rho(k) \approx \frac{\sum_{i=1}^{N-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2},$$

where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ is the overall mean.

- With increasing lag k we expect lower autocorrelations.
- If autocorrelation is still relatively high for higher values of k, this indicates high degree of correlation between our draws and slow mixing.

Effective sample size

A useful measure to compare the performance of different MCMC samplers is the effective sample size (ESS) Kass et al. (1998) American Statistician 52, 93–100..

- The ESS is the estimated number of independent samples needed to obtain a parameter estimate with the same precision as the MCMC estimate based on N dependent samples.

$$\text{ESS} = \frac{N}{\tau}, \quad \tau = 1 + 2 \cdot \sum_{k=1}^{\infty} \rho(k),$$

where τ is the autocorrelation time and $\rho(k)$ the autocorrelation at lag k.

Autocorrelation time

- There are different stopping criteria for the sum. Geyer (1992, Statistical Science, page 477)) proposed the initial monotone sequence estimator, where

$$\tau = 1 + 2 \cdot \sum_{k=1}^{2m+1} \rho(k)$$

where m is chosen to be the largest integer such that

$$\Gamma_i = \rho(2i) + \rho(2i+1), \quad i = 1, \ldots, m$$

is positive and the sequence $\Gamma_1, \ldots, \Gamma_m$ is monotone decreasing.