3.36pt

Advanced Simulation - Lecture 8

George Deligiannidis

February 10th, 2016

Using multiple proposals

- MH with target $\pi\left(x\right)$ where $x \in \mathbb{X}$.
- Can't choose between proposals $q_1\left(\left.x'\right| x\right)$, $q_2\left(\left.x'\right| x\right), ..., q_p\left(\left.x'\right| x\right)$.
- If you build a mixture proposal

$$q\left(\left.x'\right| x\right) = \sum_{j=1}^{p} \beta_j q_j\left(\left.x'\right| x\right),\ \beta_j > 0, \sum_{j=1}^{p} \beta_j = 1,$$

then you have to evaluate $q_j\left(\left.X^{\star}\right| X^{(t-1)}\right)$ for $j = 1, ..., p$.

Composing kernels

- How to use different proposals to sample from π without evaluating all the densities at each step?

- Instead combine Metropolis-Hastings updates K_j using proposal q_j instead? i.e.

$$K_j\left(x,x'\right)=\alpha_j\left(\left.x'\right|x\right)q_j\left(\left.x'\right|x\right)+\left(1-a_j\left(x\right)\right)\delta_x\left(x'\right)$$

where

$$\alpha_j(x'|x)=\min\left(1,\frac{\pi(x')q_j(x|x')}{\pi(x)q_j(x'|x)}\right)$$
$$a_j(x)=\int\alpha_j(x'|x)q_j(x'|x)dx'.$$

Composing kernels

Generally speaking, assume

- p possible updates characterised by kernels $K_j(\cdot, \cdot)$,
- each kernel K_j is π-invariant.

Two ways to combine the p MCMC updates:

- **Cycle**: perform the MCMC updates in a deterministic order.
- **Mixture**: Pick an MCMC update at random.

Cycle of MCMC updates

- Starting with $X^{(1)}$ iterate for $t = 2, 3, ...$
  1. Set $Z^{(t,0)} := X^{(t-1)}$.
  2. For $j = 1, ..., p$, sample $Z^{(t,j)} \sim K_j\left(Z^{(t,j-1)}, \cdot\right)$.
  3. Set $X^{(t)} := Z^{(t,p)}$.
- Full cycle transition kernel is

$$K(x, y) = \int \cdots \int K_1(x, z_1) K_2(z_1, z_2) \cdots K_p(z_{p-1}, y) \, \mathrm{d}z_1 \cdots \mathrm{d}z_p.$$

- K is π-invariant.

Mixture of MCMC updates

- Starting with $X^{(1)}$ iterate for $t = 2, 3, ...$
  1. Sample J from $\{1, ..., p\}$ with $\mathbb{P}(J = k) = \beta_k$.
  2. Sample $X^{(t)} \sim K_J\left(X^{(t-1)}, \cdot\right)$.
- Corresponding transition kernel is

$$K(x, y) = \sum_{j=1}^{p} \beta_j K_j(x, y).$$

- K is π-invariant.
- The algorithm is *different* from using a mixture proposal

$$q(x'|x) = \sum_{j=1}^{p} \beta_j q_j(x'|x).$$

Metropolis-Hastings Design for Multivariate Targets

- If $\dim(\mathbb{X})$ is large, it might be very difficult to design a "good" proposal $q(x'|x)$.
- As in Gibbs sampling, we might want to partition x into $x = (x_1, ..., x_d)$ and denote $x_{-j} := x \setminus \{x_j\}$.
- We propose "local" proposals where only x_j is updated

$$q_j(x'|x) = \underbrace{q_j\left(x'_j \middle| x\right)}_{\text{propose new component } j} \underbrace{\delta_{x_{-j}}\left(x'_{-j}\right)}_{\text{keep other components fixed}}.$$

Metropolis-Hastings Design for Multivariate Targets

- This yields

$$\begin{aligned}
\alpha_j(x,x') &= \min\left(1, \frac{\pi(x'_{-j},x'_j)q_j(x_j|x_{-j},x'_j)}{\pi(x_{-j},x_j)q_j(x'_j|x_{-j},x_j)} \underbrace{\frac{\delta_{x'_{-j}}(x_{-j})}{\delta_{x_{-j}}(x'_{-j})}}_{=1}\right) \\
&= \min\left(1, \frac{\pi(x_{-j},x'_j)q_j(x_j|x_{-j},x'_j)}{\pi(x_{-j},x_j)q_j(x'_j|x_{-j},x_j)}\right) \\
&= \min\left(1, \frac{\pi_{X_j|X_{-j}}(x'_j|x_{-j})q_j(x_j|x_{-j},x'_j)}{\pi_{X_j|X_{-j}}(x_j|x_{-j})q_j(x'_j|x_{-j},x_j)}\right).
\end{aligned}$$

One-at-a-time MH (cycle/systematic scan)

Starting with $X^{(1)}$ iterate for $t = 2, 3, ...$
For $j = 1, ..., d$,

- Sample $X^{\star} \sim q_j(\cdot | X_1^{(t)}, \ldots, X_{j-1}^{(t)}, X_j^{(t-1)}, ..., X_d^{(t-1)})$.
- Compute

$$\alpha_j = \min\left(1, \frac{\pi_{X_j|X_{-j}}\left(X_j^{\star} \mid X_1^{(t)} \ldots X_{j-1}^{(t)}, X_{j+1}^{(t-1)} \ldots X_d^{(t-1)}\right)}{\pi_{X_j|X_{-j}}\left(X_j^{(t-1)} \mid X_1^{(t)} \ldots X_{j-1}^{(t)}, X_{j+1}^{(t-1)} \ldots X_d^{(t-1)}\right)} \times \frac{q_j\left(X_j^{(t-1)} \middle| X_1^{(t)} ... X_{j-1}^{(t)}, X_j^{\star}, X_{j+1}^{(t-1)} ... X_d^{(t-1)}\right)}{q_j\left(X_j^{\star} \middle| X_1^{(t)} ... X_{j-1}^{(t)}, X_j^{(t-1),}, X_{j+1}^{(t-1)} ... X_d^{(t-1)}\right)}\right).$$

- With probability α_j, set $X^{(t)} = X^{\star}$, otherwise set $X^{(t)} = X^{(t-1)}$.

One-at-a-time MH (mixture/random scan)

Starting with $X^{(1)}$ iterate for $t = 2, 3, ...$

- Sample J from $\{1, ..., d\}$ with $\mathbb{P}\left(J = k\right) = \beta_k$.
- Sample $X^\star \sim q_J\left(\cdot \middle| X_1^{(t)}, ..., X_d^{(t-1)}\right)$.
- Compute

$$
\begin{aligned}
\alpha_J \quad = \quad \min & \left(1, \frac{\pi_{X_J|X_{-J}}\left(X_J^\star \mid X_1^{(t-1)} \ldots X_{J-1}^{(t-1)}, X_{J+1}^{(t-1)} \ldots\right)}{\pi_{X_J|X_{-J}}\left(X_J^{(t-1)} \mid X_1^{(t-1)} \ldots X_{J-1}^{(t-1)}, X_{J+1}^{(t-1)} \ldots\right)}\right. \\
& \left. \times \frac{q_J\left(X_J^{(t-1)} \middle| X_1^{(t-1)} ... X_{J-1}^{(t-1)}, X_J^\star, X_{J+1}^{(t-1)} ... X_d^{(t-1)}\right)}{q_J\left(X_J^\star \middle| X_1^{(t-1)} ... X_{J-1}^{(t-1)}, X_J^{(t-1),}, X_{J+1}^{(t-1)} ... X_d^{(t-1)}\right)}\right).
\end{aligned}
$$

- With probability α_J set $X^{(t)} = X^\star$, otherwise $X^{(t)} = X^{(t-1)}$.

Gibbs Sampler as a Metropolis-Hastings algorithm

Proposition

The systematic Gibbs sampler is a cycle of one-at-a time MH whereas the random scan Gibbs sampler is a mixture of one-at-a time MH where

$$q_j\left(x_j'\middle|\, x\right) = \pi_{X_j|X_{-j}}\left(x_j'\middle|\, x_{-j}\right).$$

Proof.

It follows from

$$\frac{\pi\left(x_{-j}, x_j'\right)}{\pi\left(x_{-j}, x_j\right)} \frac{q_j\left(x_j\middle|\, x_{-j}, x_j'\right)}{q_j\left(x_j'\middle|\, x_{-j}, x_j\right)}$$

$$= \frac{\pi\left(x_{-j}\right) \pi_{X_j|X_{-j}}\left(x_j'\middle|\, x_{-j}\right)}{\pi\left(x_{-j}\right) \pi_{X_j|X_{-j}}\left(x_j\middle|\, x_{-j}\right)} \frac{\pi_{X_j|X_{-j}}\left(x_j\middle|\, x_{-j}\right)}{\pi_{X_j|X_{-j}}\left(x_j'\middle|\, x_{-j}\right)} = 1. \qquad \square$$

This is not a Gibbs sampler

Consider a case where $d = 2$. From $X_1^{(t-1)}, X_2^{(t-1)}$ at time $t-1$:

- Sample $X_1^\star \sim \pi(X_1 \mid X_2^{(t-1)})$, then $X_2^\star \sim \pi(X_2 \mid X_1^\star)$. The proposal is then $X^\star = (X_1^\star, X_2^\star)$.
- Compute

$$\alpha_t = \min\left(1, \frac{\pi(X_1^\star, X_2^\star)}{\pi(X_1^{(t-1)}, X_2^{(t-1)})} \frac{q(X^{(t-1)} \mid X^\star)}{q(X^\star \mid X^{(t-1)})}\right)$$

- Accept $X^\star$ or not based on α_t, where here

$$\alpha_t \neq 1$$

!!

Convergence diagnostics

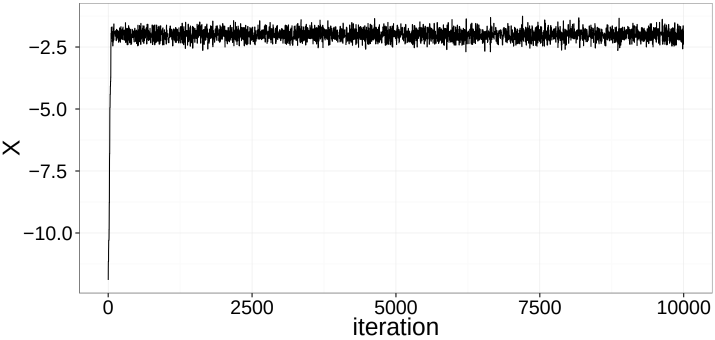
- Goal: assess whether MCMC chains have converged.

- In general, impossible to know for sure that there is no problem.

- But we can sometimes know for sure that there *is* a problem.

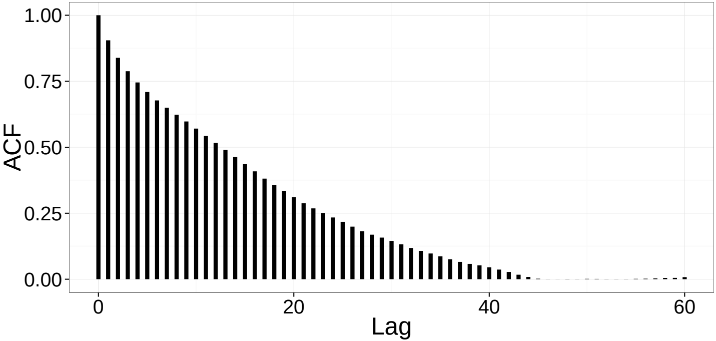
Visual diagnostics: traceplot

Target: $\pi = \mathcal{N}(-2, 0.2^2)$, proposal $q(y \mid x) = \mathcal{N}(y; x, 0.5^2)$.



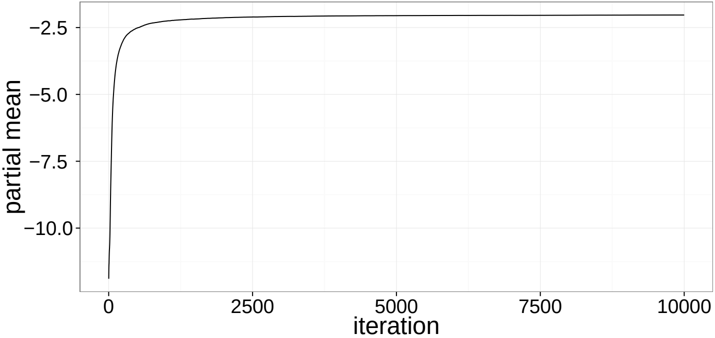
Visual diagnostics: autocorrelogram

Target: $\pi = \mathcal{N}(-2, 0.2^2)$, proposal $q(y \mid x) = \mathcal{N}(y; x, 0.5^2)$.



Visual diagnostics: convergence of estimators

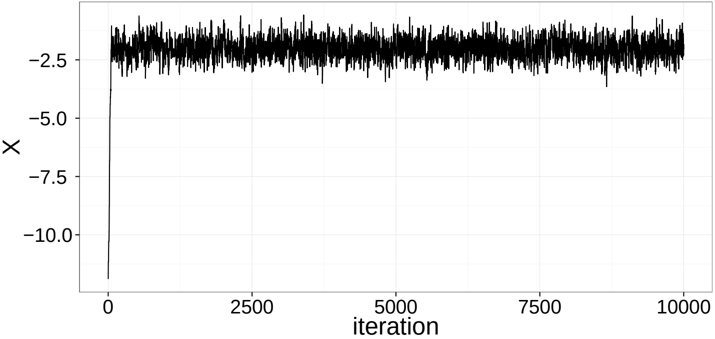
Target: $\pi = \mathcal{N}(-2, 0.2^2)$, proposal $q(y \mid x) = \mathcal{N}(y; x, 0.5^2)$.



Could be also computed on different non-overlapping subsequences, leading to Geweke's diagnostics.

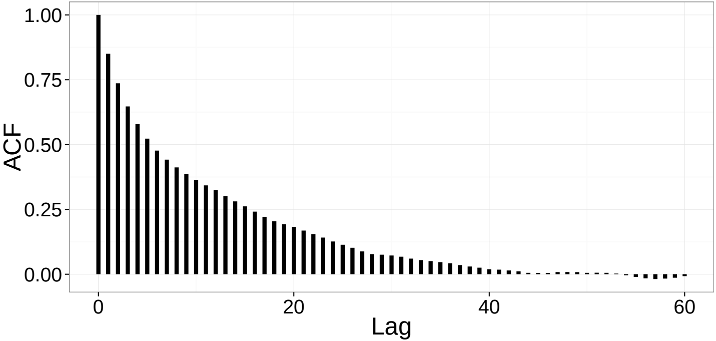
Visual diagnostics: traceplot

Target: $\pi = \frac{1}{2}\mathcal{N}(-2, 0.2^2) + \frac{1}{2}\mathcal{N}(+2, 0.2^2)$, same proposal.



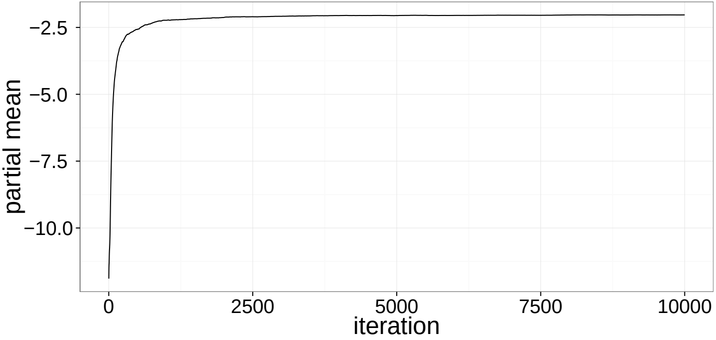
Visual diagnostics: autocorrelogram

Target: $\pi = \frac{1}{2}\mathcal{N}(-2, 0.2^2) + \frac{1}{2}\mathcal{N}(+2, 0.2^2)$, same proposal.



Visual diagnostics: convergence of estimators

Target: $\pi = \frac{1}{2}\mathcal{N}(-2, 0.2^2) + \frac{1}{2}\mathcal{N}(+2, 0.2^2)$, same proposal.



Multiple starting points

- We start *M* chains from various starting points.
- After enough iterations the starting point should not matter and hence we should obtain the *same* results based on each chain.
- We have the classical "sum of squares" decomposition in "intra group" and "inter group" terms:

$$\begin{aligned}\sum_{m=1}^{M}\sum_{t=1}^{T}(X_{m,t}-\bar{X}_{\cdot,\cdot})^2 &= \sum_{m=1}^{M}\sum_{t=1}^{T}(\bar{X}_{m,\cdot}-\bar{X}_{\cdot,\cdot})^2 \quad \text{inter-group}\\ &+\sum_{m=1}^{M}\sum_{t=1}^{T}(X_{m,t}-\bar{X}_{m,\cdot})^2 \quad \text{intra-group}\end{aligned}$$

Multiple starting points

- This leads to considering

$$W = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{T-1} \sum_{t=1}^{T} (X_{m,t} - \bar{X}_{m,\cdot})^2$$

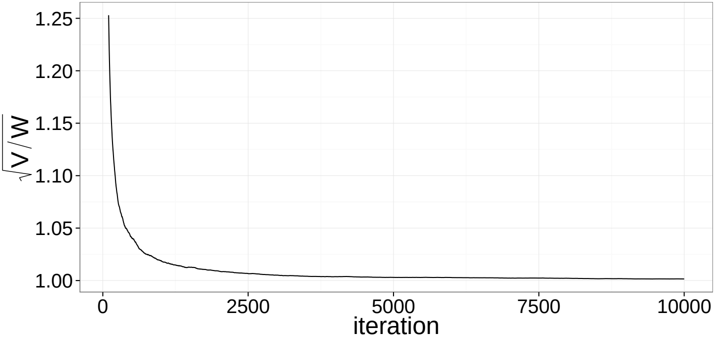
$$B = \frac{1}{M-1} \sum_{m=1}^{M} (\bar{X}_{m,\cdot} - \bar{X}_{\cdot,\cdot})^2$$

$$V = \left(1 - \frac{1}{T}\right) W + B$$

- In principle W and V should both converge to the true variance of the target distribution.
- V would be unbiased if starting points were drawn from the target, whereas W under-estimates the variance.
- We can thus plot $\sqrt{V/W}$ and compare to 1. This is the idea behind Gelman-Rubin diagnostics.

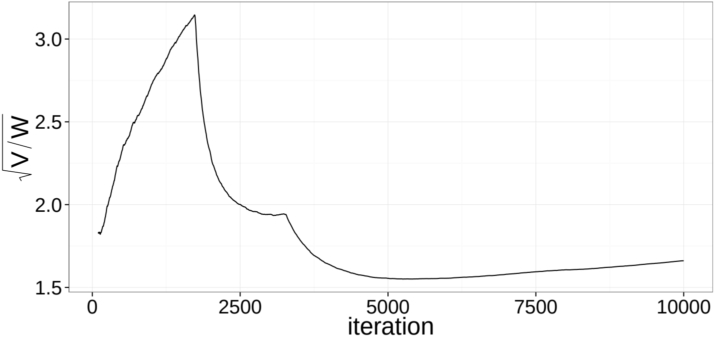
Visual diagnostics: Gelman-Rubin diagnostics

Target: $\pi = \mathcal{N}(-2, 0.2^2)$, $M = 4$ chains.



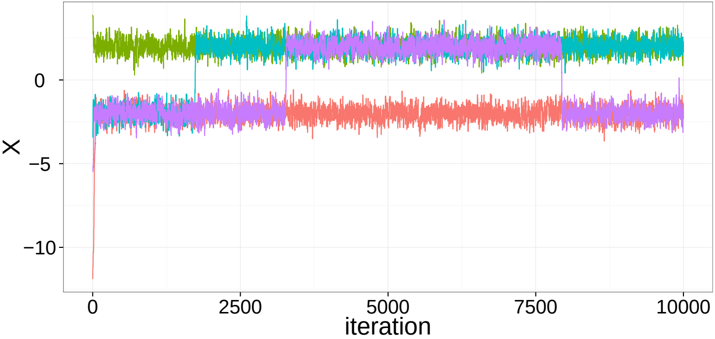
Visual diagnostics: Gelman-Rubin diagnostics

Target: $\pi = \frac{1}{2}\mathcal{N}(-2, 0.2^2) + \frac{1}{2}\mathcal{N}(+2, 0.2^2)$, $M = 4$ chains.



Visual diagnostics: traceplot with M chains

Target: $\pi = \frac{1}{2}\mathcal{N}(-2, 0.2^2) + \frac{1}{2}\mathcal{N}(+2, 0.2^2)$, $M = 4$ chains.



Parallelization

In the past (and in the next?) years, many more parallel cores, but not much more clockspeed.

- Among the methods seen so far, which are parallelizable?
- MCMC methods are by definition iterative methods. Sometimes the likelihood evaluation itself can be parallelized.
- We can run independent MCMC in parallel, as in the Gelman-Rubin diagnostics.
- Should we make the chains interact?

Parallelization of the likelihood evaluation

Consider the evaluation of the likelihood in the normal mixture case: the observations $Y_1, \ldots, Y_n$ come from

$$\forall i \in \{1, \ldots, n\} \quad Y_i \sim \sum_{k=1}^{K} p_k \mathcal{N}(\mu_k, \sigma_k^2).$$

The likelihood can be written

$$\mathcal{L}(\theta; y_1, \ldots, y_n) = \prod_{i=1}^{n} \left(\sum_{k=1}^{K} p_k \varphi(y_i; \mu_k, \sigma_k^2) \right)$$

which can be done by evaluating the n terms in the product in parallel and then taking the product.
Or $n \times K$ terms in parallel, and then partial sums and a product.

Parallelization of the likelihood evaluation

- For i.i.d. data the likelihood evaluation can be parallelized.
- In cases where
  - the likelihood is not so expensive,
  - or the likelihood evaluation cannot be efficiently parallelized.

  then a single-chain Metropolis-Hastings algorithm cannot benefit from multiple processors.
- However we can run multiple chains!

Parallel Tempering

- The idea of parallel tempering is to run N chains targeting different versions of π, of "increasing difficulty".

- Introduce "inverse temperatures":

$$0 < \gamma_1 < \gamma_2 < \ldots < \gamma_N = 1.$$

- Introduce "tempered" distributions π^{γ_n} for $n = 1, \ldots, N$ and N chains one for each π^{γ_k}.

- For $\gamma \approx 0$, π^{γ} is considered easier to sample because the variations of π are smaller.

Parallel Tempering

The "joint chain" is targeting

$$\pi^{\gamma_1} \otimes \pi^{\gamma_2} \otimes \ldots \otimes \pi^{\gamma_N}.$$

- We occasionally perform a swap move:
  - Sample indices k_1, k_2 uniformly in $\{1, \ldots, N\}$.
  - With acceptance probability

$$\min\left(1, \frac{\pi^{\gamma_{k_1}}(x_{k_2})\pi^{\gamma_{k_2}}(x_{k_1})}{\pi^{\gamma_{k_1}}(x_{k_1})\pi^{\gamma_{k_2}}(x_{k_2})}\right).$$

    exchange the value of x_{k_1} and x_{k_2}.
- **FACT:** The swap moves preserve detailed balance.
- This doesn't change the joint target distribution $\pi^{\gamma_1} \otimes \pi^{\gamma_2} \otimes \ldots \otimes \pi^{\gamma_N}$.
- In particular the N-th chain still targets $\pi^{\gamma_N} = \pi$.

Parallel Tempering

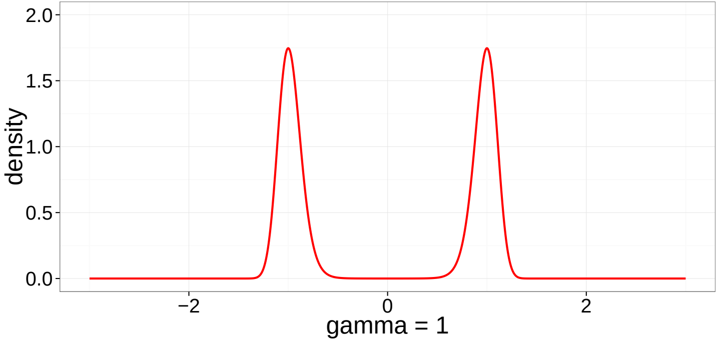


Figure: Target density function.

Parallel Tempering

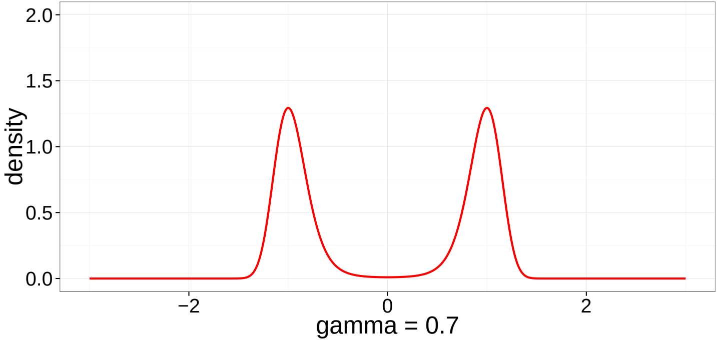


Figure: Target density function.

Parallel Tempering

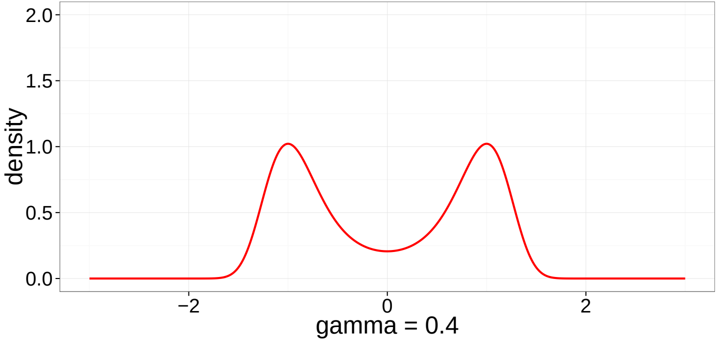


Figure: Target density function.

Parallel Tempering

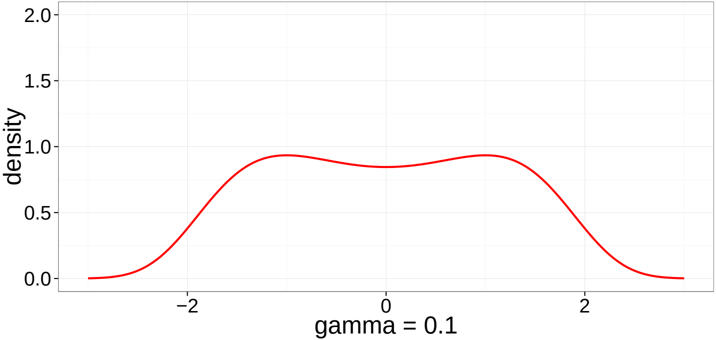


Figure: Target density function.

Parallel Tempering

Let's use $N = 10$ chains and $\gamma_1 = 0.1, \gamma_2 = 0.2, \ldots, \gamma_{10} = 1$. No swapping.

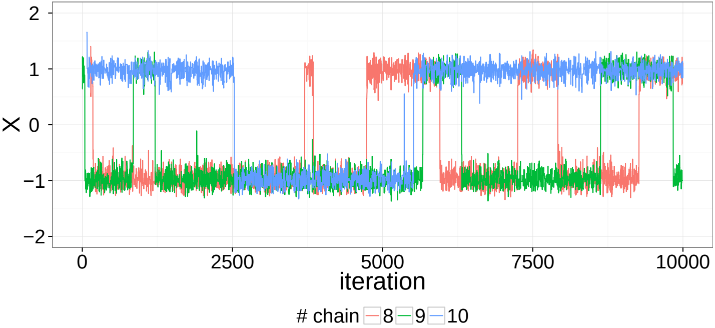


Figure: Trace plot of the "low temperature chains".

Parallel Tempering

Let's use $N = 10$ chains and $\gamma_1 = 0.1, \gamma_2 = 0.2, \ldots, \gamma_{10} = 1$.

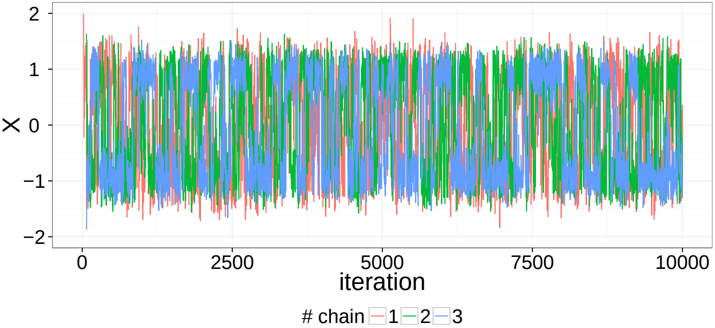


Figure: Trace plot of the "high temperature chains".

Parallel Tempering

- If we want to find the modes of π, we might just use the high temperature chains and forget about sampling directly from π.

- If we want to sample from π, can we use the “high temperature” chains to improve the mixing of the chain targeting π?

- Parallel tempering works by proposing moves where chains of different temperatures are swapped.

Parallel Tempering

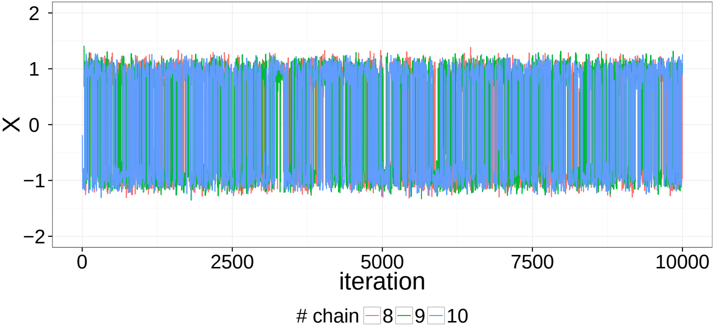


Figure: Trace plot of the “low temperature chains” using swap moves.

Parallel Tempering

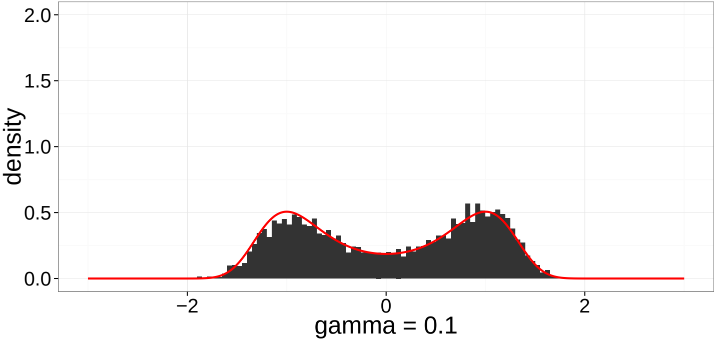


Figure: Histogram of the chain targeting π^{γ_1}.

Parallel Tempering

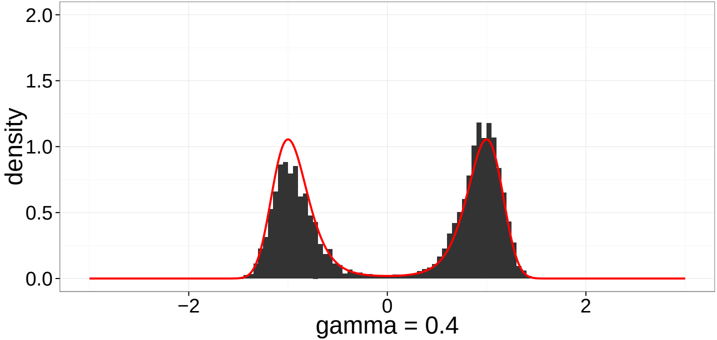


Figure: Histogram of the chain targeting π^{γ_4}.

Parallel Tempering

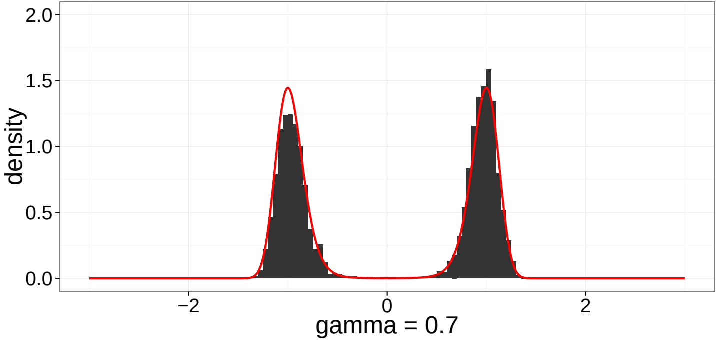


Figure: Histogram of the chain targeting π^{γ_7}.

Parallel Tempering

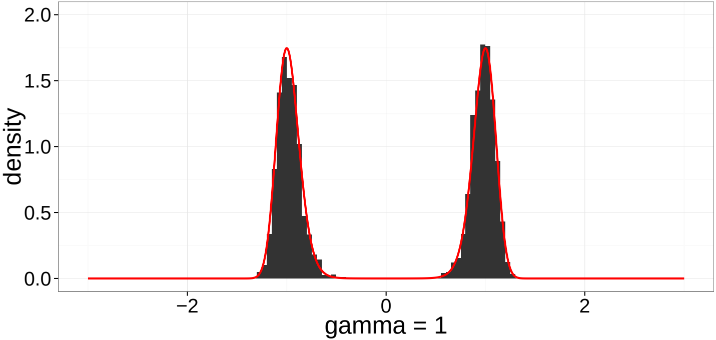


Figure: Histogram of the chain targeting $\pi^{\gamma_{10}}$.

Swap moves improve the mixing of chains with high values of γ.