Stan for the people

Two day introductory workshop
on Bayesian modeling

McGill University
January 25th 2019

mc-stan.org

II

Algorithms and computational considerations

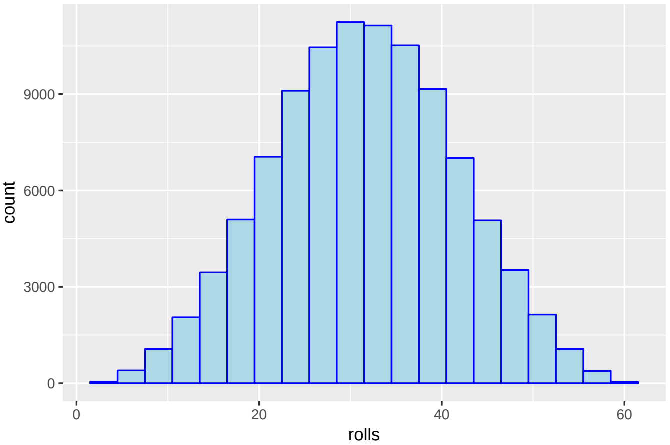
Monte Carlo method

Often, it is difficult to find an exact or analytical expression for a distribution.

Eg: Rolling dices

- How will the outcome of rolling two regular dices be distributed?
- What if I roll three dices with 20 sides each?

Empirical distributions obtained with 10000 rolls



Eg: Sequential data generating process

- Posterior distribution of a Poisson process with a Gamma prior
- Posterior distribution of a Poisson process with a lognormal prior

In general, we would like to sample from:

$$P(\theta|Z) = \frac{P(Z|\theta)P(\theta)}{P(Z)}$$

where $P(Z)$ is unknown.

- A popular method is Markov chain Monte Carlo (MCMC).

Example: Metropolis-Hasting algorithm

- Start at an initial point in the *parameter space*.
- Until we have generated *n* samples, do
  - Take a random step in the parameter space, from $\theta^{(i)}$ to $\theta^{(i+1)}$ to propose a new sample.
  - Accept the proposal with probability

$$\mathrm{Pr} = \min\left(\frac{P(Z|\theta^{(i+1)})P(\theta^{(i+1)})}{P(Z|\theta^{(i)})P(\theta^{(i)})}, 1\right)$$

.

Example: Metropolis-Hasting algorithm

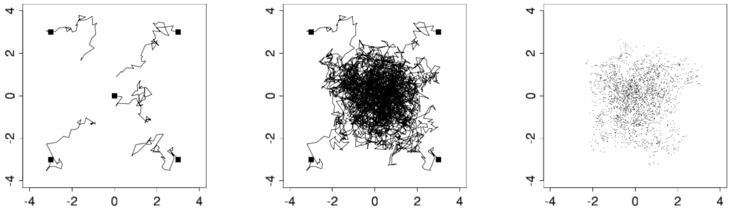


Figure from *Bayesian Data Analysis* [Gelman et al., 2013].

Benefits:

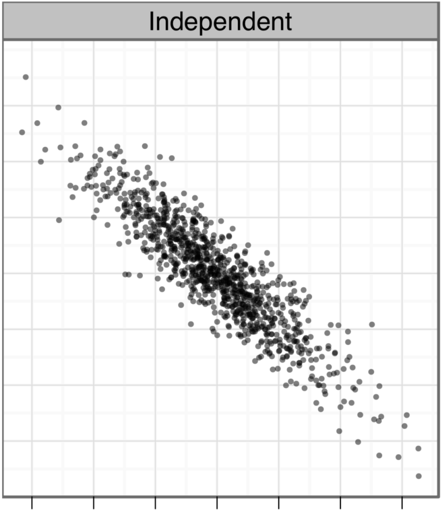
- the algorithm only requires $p(\theta|Z)$ and $p(\theta)$.
- in the asymptotic limit, the algorithm samples from to the true distribution.

Drawbacks:

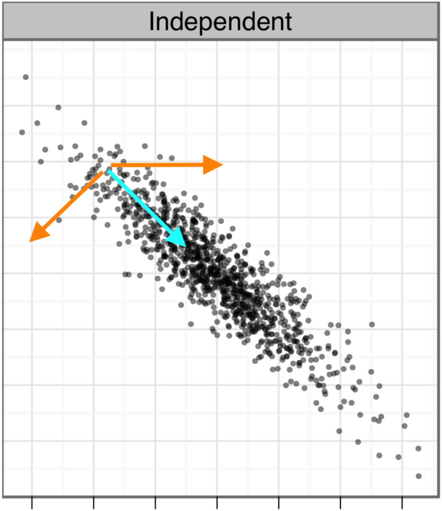
- The samples are not independent.
- We do not know for sure whether the chains have converged to the target distribution.

- Metropolis-Hasting is one example of a random walk algorithm.
- Another one is Gibbs sampling.

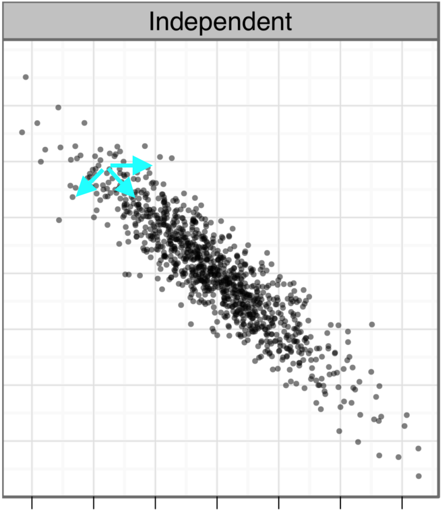
Geometric structure in the distribution



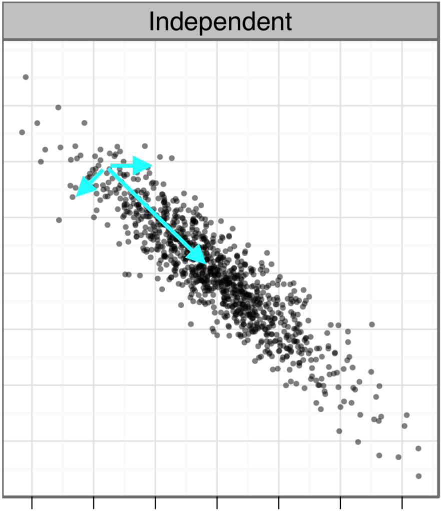
Geometric structure in the distribution



Geometric structure in the distribution



Geometric structure in the distribution

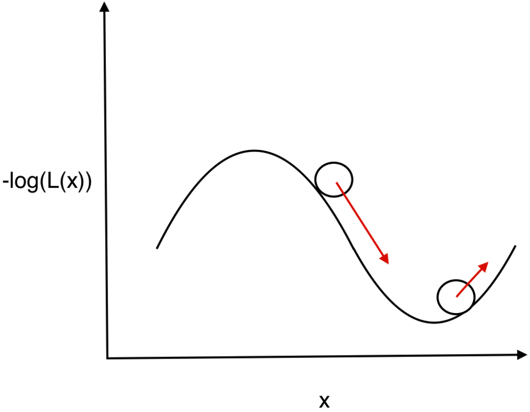


Hamiltonian Monte Carlo

Idea:

- Treat the Markov chain like a physical *particle*.
- Treat the negative log density like a physical *potential*.
- Give the particle a random shove instead of a random step.
- Simulate a the laws of classical mechanics.

Hamiltonian Monte Carlo



Comparison between sampling methods

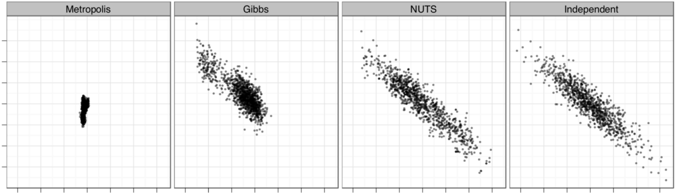


Figure from [Hoffman and Gelman, 2014].

Practical concerns

- To simulate physical trajectories, we need the *gradient* of the log posterior.
- HMC has many tuning parameters.

Practical concerns

- To simulate physical trajectories, we need the *gradient* of the log posterior. $\rightarrow$ Automatic differentiation
- HMC has many tuning parameters. $\rightarrow$ No-U Turn Sampler

There is more to the story

For a thorough treatment of Hamiltonian Monte Carlo, see *A Conceptual introduction to HMC* [Betancourt, 2017].

The original paper on the No-U Turn Sampler (NUTS) is [Hoffman and Gelman, 2014].

For details on Automatic Differentiation, see [Margossian, 2019] and [Carpenter et al., 2015].

References I

[Betancourt, 2017] Betancourt, M. (2017).
A conceptual introduction to hamiltonian monte carlo.
arXiv:1701.02434v1.

[Carpenter et al., 2015] Carpenter, B., Hoffman, M. D., Brubaker, M. A., Lee, D., Li, P., and Betancourt, M. J. (2015).
The stan math library: Reverse-mode automatic differentiation in c++.
arXiv 1509.07164.

[Gelman et al., 2013] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013).
Bayesian Data Analysis.
Chapman & Hall.

[Hoffman and Gelman, 2014] Hoffman, M. D. and Gelman, A. (2014).
The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo.
Journal of Machine Learning Research, pages 1593–1623.

[Margossian, 2019] Margossian, C. C. (2019).
A review of automatic differentiation and its efficient implementation.
WIREs Data Mining and Knowledge.