Sampling Methods for Bayesian Inference

Groupe de lecture – Session 5

Nadège Polette

CEA, DAM, DIF, F-91297 Arpajon / Mines Paris PSL

Table of contents

Bayesian Framework

Guiding example:

$$\boldsymbol{y} = f(\boldsymbol{x}) + \varepsilon, \text{ with } \varepsilon \sim \mathcal{N}(0, \theta^{-1}\mathrm{I}_M) \quad (1)$$

Bayes' law:

$$p_{\text{post}}(\boldsymbol{x}, \theta|\boldsymbol{y}) = \frac{\mathcal{L}(\boldsymbol{y}|\boldsymbol{x}, \theta)p(\boldsymbol{x}, \theta)}{p(\boldsymbol{y})}, \quad (2)$$

where $\boldsymbol{x} \in \mathbb{R}^N$ are *unknown parameters to estimate*

$\boldsymbol{y} \in \mathbb{R}^M$ are *the observations*,

$\theta > 0$ is the error precision,

$\mathcal{L}(\boldsymbol{y}|\boldsymbol{x}, \theta) \propto \theta^{M/2} \exp\left(\frac{-\theta}{2} \|f(\boldsymbol{x}) - \boldsymbol{y}\|_2^2\right)$ is the *likelihood*,

$p(\boldsymbol{x}, \theta)$ is the *prior* and

$p(\boldsymbol{y}) = \int_{x \in \mathbb{R}^N, \theta > 0} \mathcal{L}(\boldsymbol{y}|\boldsymbol{x}, \theta)p(\boldsymbol{x}, \theta) d\boldsymbol{x} d\theta$ is the *evidence*.

Objective

Bayes' law:

$$p_{\text{post}}(\boldsymbol{x}, \theta | \boldsymbol{y}) = \frac{\mathcal{L}(\boldsymbol{y}|\boldsymbol{x}, \theta) p(\boldsymbol{x}, \theta)}{p(\boldsymbol{y})}, \tag{2}$$

Objective: We want to *sample from* $p_{\text{post}}(\boldsymbol{x}, \theta | \boldsymbol{y})$ and/or *estimate statistics* of this posterior distribution.

Direct simulation:

- Monte–Carlo sampling: draw $(\boldsymbol{x}, \theta) \sim p_{\text{post}}(\cdot|\boldsymbol{y})$
- Approximation with an usual law $\rightarrow$ guess for the search space and for the points to evaluate
- Inverse CDF method (1D)
- Rejection sampling
- Importance sampling: typically used for rare event estimation
- ...

These methods produce *independent samples*.

Objective

Bayes' law:

$$p_{\text{post}}(\boldsymbol{x}, \theta | \boldsymbol{y}) = \frac{\mathcal{L}(\boldsymbol{y}|\boldsymbol{x}, \theta) p(\boldsymbol{x}, \theta)}{p(\boldsymbol{y})}, \qquad (2)$$

Objective: We want to *sample from* $p_{\text{post}}(\boldsymbol{x}, \theta | \boldsymbol{y})$ and/or *estimate statistics* of this posterior distribution.

Problem: In general, we only know p_{post} *up to a multiplicative factor: the evidence*

- We can compare two propositions: $p_{\text{post}}(\boldsymbol{x}_1, \theta_1 | \boldsymbol{y}) > p_{\text{post}}(\boldsymbol{x}_2, \theta_2 | \boldsymbol{y})$?
- But we cannot assign a density probability

Numerical integration

Principle: we want to estimate $\int_\Omega q(x)dx$

- deterministic: quadrature rules low error, difficult in high dimension $\mathcal{O}(e^d)$

$$\int_\Omega q(\boldsymbol{x})d\boldsymbol{x} \simeq \sum_i w_i q(\boldsymbol{x}^{(i)}) \tag{3}$$

- stochastic: Monte–Carlo sampling high variance $\mathcal{O}(\frac{1}{\sqrt{n}})$, OK in high dimension

$$\mathbb{E}\left[f(\boldsymbol{x})\right] = \int_\Omega f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} \simeq \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}^{(i)}) \text{ with } \boldsymbol{x}^{(i)} \sim p(\boldsymbol{x})$$

Table of contents

What is a Markov Chain ?

Markov Chain: $(X^{(i)})_{1\leqslant i\leqslant N} \in \mathbb{R}^{d\times N}$ such that

$$p(X^{(k)}|X^{(1\leqslant i\leqslant k-1)}) = p(X^{(k)}|X^{(k-1)}) = p_{\mathrm{tr}}(X^{(k-1)}, X^{(k)}) \text{ and } X^{(0)} \sim \nu$$

$$X^{(1)} \xrightarrow{p_{\mathrm{tr}}(X^{(1)},\cdot)} X^{(2)} \xrightarrow{p_{\mathrm{tr}}(X^{(2)},\cdot)} \dots \xrightarrow{p_{\mathrm{tr}}(X^{(N-1)},\cdot)} X^{(N)}$$

$$p((X^{(i)})_{1\leqslant i\leqslant N}) = \nu(X^{(0)}) \prod_{i=1}^{N} p_{\mathrm{tr}}(X^{(i-1)}, X^{(i)})$$

Detailed balance: $\pi(X)p_{\mathrm{tr}}(X, X') = \pi(X')p_{\mathrm{tr}}(X', X)$

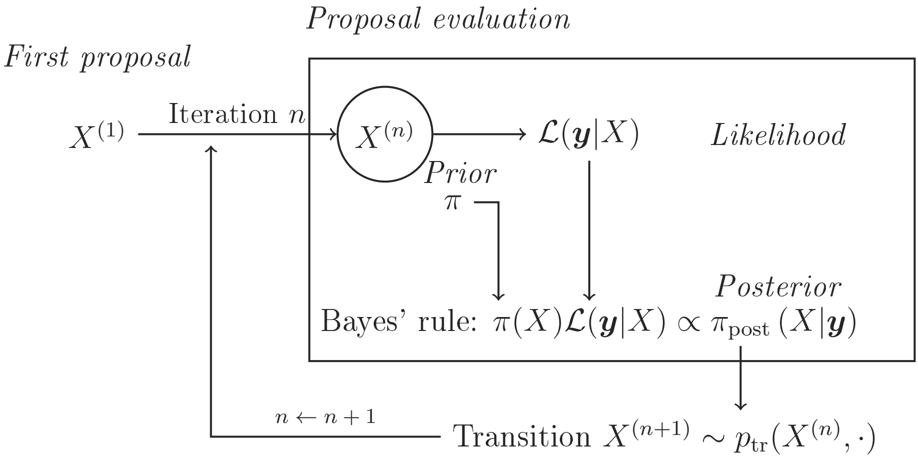
π is the *stationnary distribution* of the Markov Chain.

Utility: define p_{tr} such that $\forall \nu$,

$$p(X^{(N)}) = \nu(X^{(0)})p_{\mathrm{tr}}^{N}(X^{(0)}, \cdot) \xrightarrow[N\to+\infty]{} \pi = p_{\mathrm{post}}(X|\boldsymbol{y})$$

[Note the slight abuse of notation between X random variable and x real]

Markov Chain Monte–Carlo algorithm



Gibbs sampling

Objective: Sample from $\pi = p_{\mathrm{post}}(X|\boldsymbol{y})$

Principle: Knowing $X^{(k)}$, $X^{(k+1)}$ is computed as follows

$$p_{\mathrm{tr}}(X^{(k)}, X^{(k+1)}) = \prod_{i=1}^{d} p_{\mathrm{post}}(X_i | X_{j>i}^{(k)}, X_{j<i}^{(k+1)}, \boldsymbol{y})$$

The elements of X are updated *one by one*, using their *marginal distributions*.

When to use it ? Need to have access to the marginal distributions

$$p_{\mathrm{post}}(X_i | X_{j>i}^{(k)}, X_{j<i}^{(k+1)}, \boldsymbol{y}) = \frac{\mathcal{L}(\boldsymbol{y}|X) p(X_i | X_{j>i}^{(k)}, X_{j<i}^{(k+1)})}{p(\boldsymbol{y} | X_{j>i}^{(k)}, X_{j<i}^{(k+1)})}$$

Example: Conjugate priors

Gibbs example

Guiding example:

$$\boldsymbol{y} = f(\boldsymbol{x}) + \varepsilon = A\boldsymbol{x} + \varepsilon, \text{ with } \varepsilon \sim \mathcal{N}(0, \theta^{-1}\mathrm{I}_M)$$

$$p_{\text{post}}(\boldsymbol{x}, \theta|\boldsymbol{y}) = \frac{\mathcal{L}(\boldsymbol{y}|\boldsymbol{x}, \theta)p(\boldsymbol{x}, \theta)}{p(\boldsymbol{y})},$$

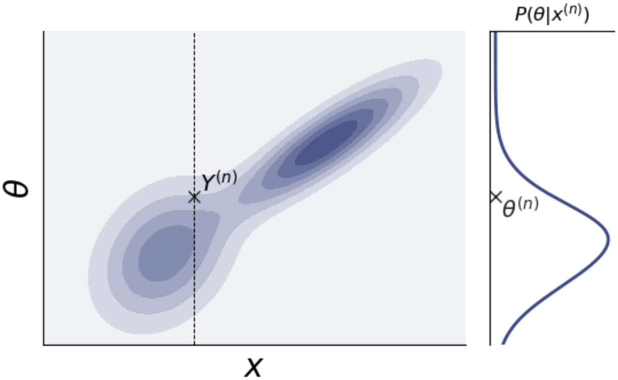
with $p(\boldsymbol{x}, \theta) = p_x(\boldsymbol{x})p_\theta(\theta)$, $\boldsymbol{x} \sim \mathcal{N}(0, \mathrm{I})$ and $\theta \sim \mathrm{Gamma}(\alpha, \beta)$.

Knowing $(\boldsymbol{x}^{(k)}, \theta^{(k)})$, we can draw $(\boldsymbol{x}^{(k+1)}, \theta^{(k+1)})$ as follows

- $\theta^{(k+1)}_{|\boldsymbol{x}^{(k)},\boldsymbol{y}} \sim \mathrm{Gamma}(\frac{M}{2} + \alpha, \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \beta)$
- $\boldsymbol{x}^{(k+1)}_{|\boldsymbol{\theta}^{(k+1)},\boldsymbol{y}} \sim \mathcal{N}(\frac{2A\boldsymbol{y}}{A^\top A + \theta^{-1}}, (\theta A^\top A + 1)^{-1})$

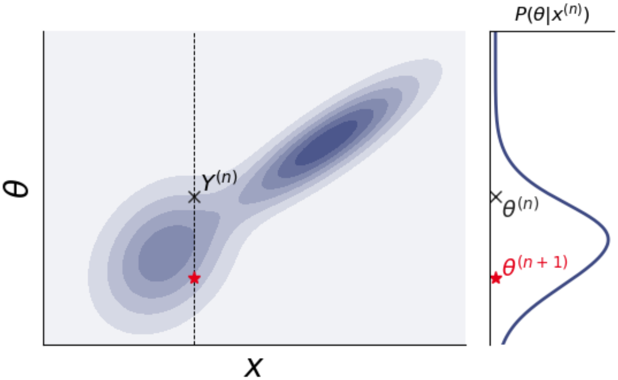
Gibbs example - illustration

- Consider $p(\theta|x^{(n)})$



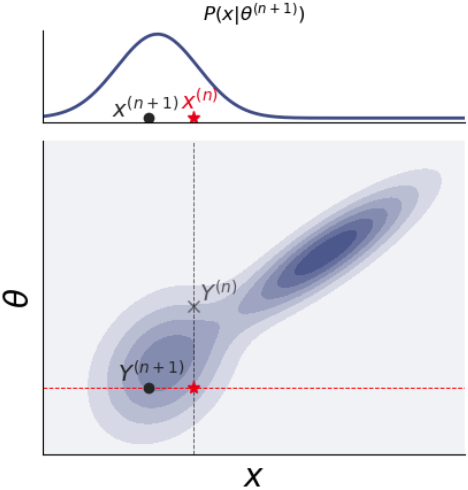
Gibbs example - illustration

- Draw $\theta^{(n+1)} \sim p(\theta|x^{(n)})$



Gibbs example - illustration

- Draw $x^{(n+1)} \sim p(x|\theta^{(n+1)})$



Metropolis–Hastings algorithm

Objective: Sample from $\pi = p_{\mathrm{post}}(X|\boldsymbol{y})$, when marginals not available

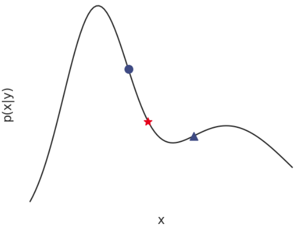
Principle: Knowing $X^{(k)}$, $X^{(k+1)}$ is computed as follows

- Propose X^* (*e.g.* *random walk*: $X^* \sim \mathcal{N}(X^{(k)}, K)$)
- Compute $p_{\mathrm{post}}(X^*|\boldsymbol{y}) \propto \mathcal{L}(\boldsymbol{y}|X^*)p(X^*)$
- Draw a random variable $u \sim \mathcal{U}(0,1)$
- Update X

$$X^{(k+1)} = \begin{cases} X^* \text{ if } u < \min\left(1, r_{\mathrm{MH}}\right) \\ X^{(k)} \text{ else,} \end{cases}$$

r_{MH} is the *Metropolis–Hastings ratio*

$$r_{\mathrm{MH}} = \frac{p(X^*)p_{\mathrm{tr}}(X^*, X^{(k)})}{p(X^{(k)})p_{\mathrm{tr}}(X^{(k)}, X^*)}$$



MH example

Guiding example:

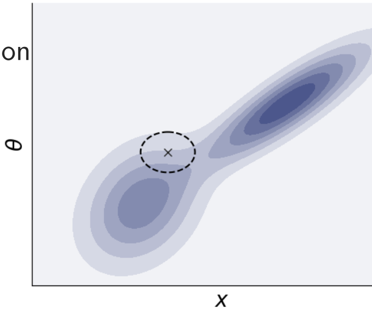
$$\boldsymbol{y} = f(\boldsymbol{x}) + \varepsilon, \text{ with } \varepsilon \sim \mathcal{N}(0, \theta^{-1}\mathrm{I}_M)$$

$$p_{\text{post}}(\boldsymbol{x}, \theta|\boldsymbol{y}) = \frac{\mathcal{L}(\boldsymbol{y}|\boldsymbol{x}, \theta)p(\boldsymbol{x}, \theta)}{p(\boldsymbol{y})},$$

with $p(\boldsymbol{x}, \theta) = p_x(\boldsymbol{x})p_\theta(\theta)$, $\boldsymbol{x} \sim \mathcal{N}(0, \mathrm{I})$ and $\theta \sim \mathrm{Gamma}(\alpha, \beta)$.
Knowing $(\boldsymbol{x}^{(k)}, \theta^{(k)})$, we can draw $(\boldsymbol{x}^{(k+1)}, \theta^{(k+1)})$ as follows

- $(\boldsymbol{x}^*, \theta^*) \sim \mathcal{N}((\boldsymbol{x}^{(k)}, \theta^{(k)}), K)$
- Accept/reject $(\boldsymbol{x}^*, \theta^*)$ with MH criterion

→ *Animation !*



Combined example

- Gibbs is a particular case of MH sampling
- Gibbs and MH are simple *building blocks* for more difficult samplings

Combined example

- Gibbs is a particular case of MH sampling
- Gibbs and MH are simple *building blocks* for more difficult samplings

Guiding example:

$$\boldsymbol{y} = f(\boldsymbol{x}) + \varepsilon, \text{ with } \varepsilon \sim \mathcal{N}(0, \theta^{-1}\mathrm{I}_M)$$

$$p_{\text{post}}(\boldsymbol{x}, \theta|\boldsymbol{y}) = \frac{\mathcal{L}(\boldsymbol{y}|\boldsymbol{x}, \theta)p(\boldsymbol{x}, \theta)}{p(\boldsymbol{y})},$$

with $p(\boldsymbol{x}, \theta) = p_x(\boldsymbol{x})p_\theta(\theta)$, $\boldsymbol{x} \sim \mathcal{N}(0, \mathrm{I})$ and $\theta \sim \mathrm{Gamma}(\alpha, \beta)$. Knowing $(\boldsymbol{x}^{(k)}, \theta^{(k)})$, we can draw $(\boldsymbol{x}^{(k+1)}, \theta^{(k+1)})$ as follows

- $\theta^{(k+1)}_{|\boldsymbol{x}^{(k)}, \boldsymbol{y}} \sim \mathrm{Gamma}\left(\frac{M}{2} + \alpha, \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \beta\right)$
- $\boldsymbol{x}^* \sim \mathcal{N}(\boldsymbol{x}^{(k)}, K)$
- Accept/reject $\boldsymbol{x}^*$ with MH criterion

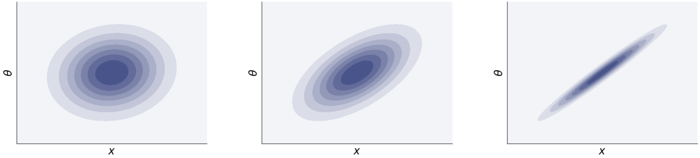
Table of contents

Auxiliary variables

Problem: the conditional sampling can be slow if the parameters are highly dependent

Basic solution: Reparametrization

Example: Sample (x, θ, q) such that $(x, \theta) \sim \mathcal{N}(0, C(q))$ (*hierarchical formulation*)



(a) $q = 10$ (b) $q = 60$ (c) $q = 98$

Solution: sample $Y^{\mathrm{ref}} \sim \mathcal{N}(0, \mathrm{I}_2)$, $q \sim p_q$ and compute $(x, \theta) = C(q)^{1/2} Y^{\mathrm{ref}}$. [Betancourt and Girolami, 2013]

MCMC convergence

- MCMC produces *dependent samples*
  - *Burn-in phase* vs sampling phase: discard the K first iterations from the analysis.
  - In order to explore the whole search space, it is better to use *several medium-length parallel MCMC chains* rather than an only large-length one.

MCMC convergence

- MCMC produces *dependent samples*
  - *Burn-in phase* vs sampling phase: discard the K first iterations from the analysis.
  - In order to explore the whole search space, it is better to use *several medium-length parallel MCMC chains* rather than an only large-length one.
- Way to *monitor the convergence*
  - Effective sample size (ESS) based on Autocorrelation function (ACF) approximation [Vats et al., 2019]
  - Gelman-Rubin diagnostic [Brooks and Gelman, 1998]
  - Acceptance rate (in the case of MH)
  - Visual aids

MCMC proposals

Objective: define K, the proposal covariance of the random walk $X^* \sim \mathcal{N}(X^{(N)}, K)$.

General form: $K = \mathrm{lr} \times \alpha \times \widehat{\mathbb{C}\mathrm{ov}}$

- Gaussian approximation empirical rule: the covariance proposal scaling factor must be close to [Gelman et al., 1996]

$$\alpha = 2.38^2/d$$

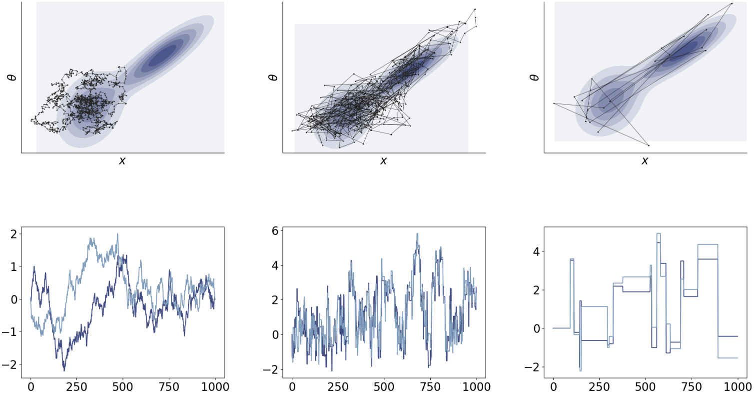
- Burn-in phase: covariance proposal adaptation (*Animation !*). Every m iterations,

$$K \leftarrow \mathrm{lr} \times \alpha \times \widehat{\mathbb{C}\mathrm{ov}}(X^{(1),\dots,(N)})$$ [Haario et al., 2001]

- Burn-in phase: use it to adapt the learning rate. For instance, every m iterations,

$$\mathrm{lr} \leftarrow \begin{cases} 1.2\mathrm{lr} \text{ if } \mathrm{AR} > 0.5, \\ 0.8\mathrm{lr} \text{ if } \mathrm{AR} < 0.15, \\ \mathrm{lr} \text{ else.} \end{cases}$$

Illustrations learning rates



(a) $\mathrm{lr} = 0.1$ (b) $\mathrm{lr} = 1.2$ (c) $\mathrm{lr} = 10$

Table of contents

Transdimensional RJ-MCMC

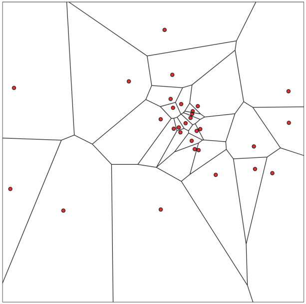
Objective: Explore propositions of different dimensions

Example: Model selection (gaussian mixture with number of gaussian undetermined, Gaussian/Matern kernel, polynomial degree. . .), Voronoï representation. . .

Principle: At each iteration, multiple choices (randomly selected)

- Change a parameter value/position
- Remove a parameter
- Add a parameter

[Piana Agostinetti et al., 2015]

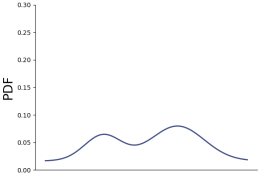


Voronoï cells example [Bodin et al., 2012]

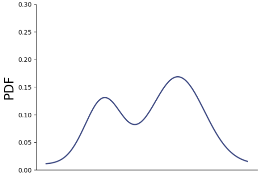
Extensions based on optimization

Example of *simulated/parallel tempering*: $p_{\mathrm{temp}} = p^{(1-k)} p_{\mathrm{post}}^{k}$

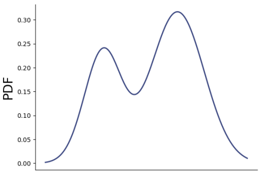
(a) $k = 0$



(b) $k = 0.2$



(c) $k = 0.5$



(d) $k = 1$

Hamiltonian (Hybrid) Monte Carlo

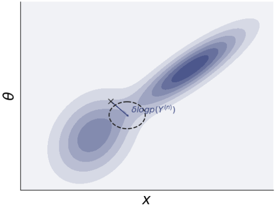
Idea: use gradient information to faster explore the posterior distribution

Physical analogy: with M the mass matrix and ϕ the momentum (auxiliary variable)

$$\underbrace{H(X,\phi)}_{\text{Hamiltonian}} = \underbrace{-\log(p_{\text{post}}(X|\boldsymbol{y}))}_{\text{Potential energy}} + \underbrace{\frac{1}{2}\phi^\top M^{-1}\phi}_{\text{Kinetic energy}}\,. \tag{4}$$

The numerical integrator

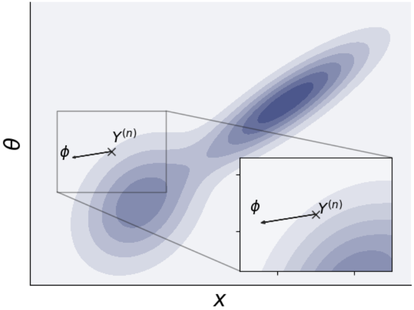
- preserves the total energy (in theory)
- preserves the volume element
- is time-reversible



[Duane et al., 1987], [Betancourt, 2018], [Fichtner et al., 2019]

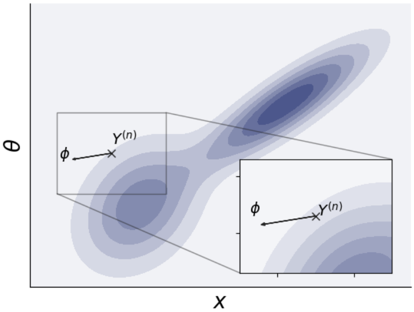
Hamiltonian Monte Carlo algorithm

- For each MCMC iteration:
  - Draw a momentum $\phi \sim \mathcal{N}(0, M)$, initialize $Y^* \leftarrow Y^{(n)}$



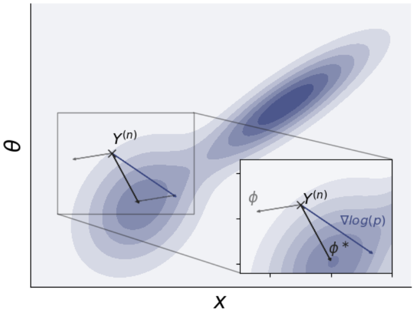
Hamiltonian Monte Carlo algorithm

- For each MCMC iteration:
  - Draw a momentum $\phi \sim \mathcal{N}(0, M)$, initialize $Y^* \leftarrow Y^{(n)}$
  - Leapfrog (Stormer-Verlet) scheme (given L and ε):



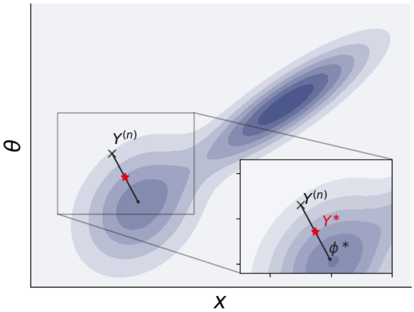
Hamiltonian Monte Carlo algorithm

- For each MCMC iteration:
  - Draw a momentum $\phi \sim \mathcal{N}(0, M)$, initialize $Y^* \leftarrow Y^{(n)}$
  - Leapfrog (Stormer-Verlet) scheme (given L and ε):
    - $\phi \leftarrow \phi + 0.5\varepsilon\nabla \log p_{\text{post}}(Y^*)$



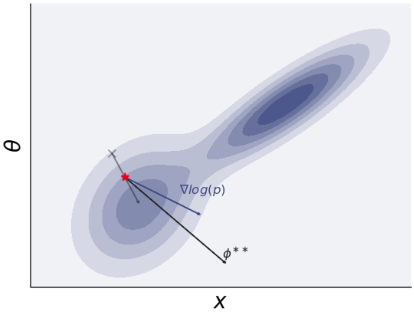
Hamiltonian Monte Carlo algorithm

- For each MCMC iteration:
  - Draw a momentum $\phi \sim \mathcal{N}(0, M)$, initialize $Y^* \leftarrow Y^{(n)}$
  - Leapfrog (Stormer-Verlet) scheme (given L and ε):
    - $\phi \leftarrow \phi + 0.5\varepsilon\nabla \log p_{\mathrm{post}}(Y^*)$
    - $Y^* \leftarrow Y^* + \varepsilon M^{-1}\phi$



Hamiltonian Monte Carlo algorithm

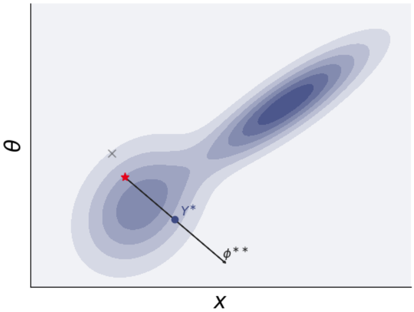
- For each MCMC iteration:
  - Draw a momentum $\phi \sim \mathcal{N}(0, M)$, initialize $Y^* \leftarrow Y^{(n)}$
  - Leapfrog (Stormer-Verlet) scheme (given L and ε):
    - $\phi \leftarrow \phi + 0.5\varepsilon \nabla \log p_{\mathrm{post}}(Y^*)$
    - $Y^* \leftarrow Y^* + \varepsilon M^{-1} \phi$
    - $\phi \leftarrow \phi + 0.5\varepsilon \nabla \log p_{\mathrm{post}}(Y^*)$
    - Repeat



Hamiltonian Monte Carlo algorithm

- For each MCMC iteration:
  - Draw a momentum $\phi \sim \mathcal{N}(0, M)$, initialize $Y^* \leftarrow Y^{(n)}$
  - Leapfrog (Stormer-Verlet) scheme (given L and ε):
    - $\phi \leftarrow \phi + 0.5\varepsilon\nabla \log p_{\mathrm{post}}(Y^*)$
    - $Y^* \leftarrow Y^* + \varepsilon M^{-1}\phi$
    - $\phi \leftarrow \phi + 0.5\varepsilon\nabla \log p_{\mathrm{post}}(Y^*)$
    - Repeat
- Accept with probability $\min\left[1, \exp(-H(Y^\star) + H(Y^{(n)}))\right]$

Animation !



Hamiltonian Monte Carlo properties

- Optimal acceptance rate around 65% ($>$ MH acceptance rate (around 25%))
- Choice of mass matrix M: by default I or Fisher information matrix
- Dealing with restricted areas: refuse, or bounce, or transform
- Tuning parameters ε and L: updated during burn-in phase to improve acceptance rate
- More sophisticated algorithms: no U-Turn (*Animation !*), Riemaniann HMC (*Animation !*)...

[Hoffman and Gelman, 2011], [Girolami and Calderhead, 2011]

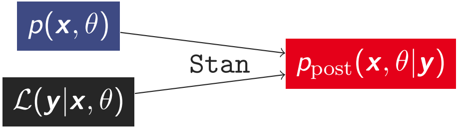
Table of contents

Presentation of Stan

- Software for statistical modeling and high-performance statistical computation
- Available standalone or as a module (R, Python, shell, MATLAB, Julia, Stata)
- Contains
  - full Bayesian statistical inference with MCMC sampling: NUTS-HMC
  - approximate Bayesian inference with variational inference: Pathfinder and ADVI
  - penalized maximum likelihood estimation with optimization

`https://mc-stan.org/`

Presentation of Stan



For visualization: Arviz (Python)

Presentation of Stan

```
import stan

schools_code = """
data {
  int<lower=0> J;         // number of schools
  array[J] real y;              // estimated treatment effects
  array[J] real<lower=0> sigma; // standard error of effect estimates
}
parameters {
  real mu;                // population treatment effect
  real<lower=0> tau;      // standard deviation in treatment effects
  vector[J] eta;          // unscaled deviation from mu by school
}
transformed parameters {
  vector[J] theta = mu + tau * eta;        // school treatment effects
}
model {
  target += normal_lpdf(eta | 0, 1);       // prior log-density
  target += normal_lpdf(y | theta, sigma); // log-likelihood
}
"""

schools_data = {"J": 8,
                "y": [28,  8, -3,  7, -1,  1, 18, 12],
                "sigma": [15, 10, 16, 11,  9, 11, 10, 18]}

posterior = stan.build(schools_code, data=schools_data)
fit = posterior.sample(num_chains=4, num_samples=1000)
eta = fit["eta"]  # array with shape (8, 4000)
df = fit.to_frame()  # pandas `DataFrame, requires pandas
```

Conclusion

- Sampling useful in the case of intractable evidence
- General methods = building blocks
- Existence of other methods (*e.g.* Variational Bayes, see Charlie's presentation !)

Thank you !

nadege.polette@cea.fr

Keywords: Sampling, Bayesian, MCMC

References I

Animations: GitHub chi-feng, mcmc-demo

- M. J. Betancourt and Mark Girolami. *Hamiltonian Monte Carlo for Hierarchical Models*. Dec. 2013. DOI: 10.48550/arXiv.1312.0906. (Visited on 09/08/2023).
- Michael Betancourt. *A Conceptual Introduction to Hamiltonian Monte Carlo*. July 2018. (Visited on 04/10/2024).
- T. Bodin et al. "Transdimensional Tomography with Unknown Data Noise". In: *Geophysical Journal International* 189.3 (June 2012), pp. 1536–1556. ISSN: 0956-540X. DOI: 10.1111/j.1365-246X.2012.05414.x.
- Stephen P. Brooks and Andrew Gelman. "General Methods for Monitoring Convergence of Iterative Simulations". In: *Journal of Computational and Graphical Statistics* 7.4 (Dec. 1998), pp. 434–455. ISSN: 1061-8600. DOI: 10.1080/10618600.1998.10474787. (Visited on 02/09/2024).

References II

Simon Duane et al. "Hybrid Monte Carlo". In: *Physics Letters B* 195.2 (Sept. 1987), pp. 216–222. ISSN: 0370-2693. DOI: 10.1016/0370-2693(87)91197-X. (Visited on 04/10/2024).

Andreas Fichtner, Andrea Zunino, and Lars Gebraad. "Hamiltonian Monte Carlo Solution of Tomographic Inverse Problems". In: *Geophysical Journal International* 216.2 (Feb. 2019), pp. 1344–1363. ISSN: 0956-540X, 1365-246X. DOI: 10.1093/gji/ggy496. (Visited on 07/13/2023).

A. Gelman, G. O. Roberts, and W. R. Gilks. "Efficient Metropolis Jumping Rules". In: *Bayesian Statistics*. Ed. by J. M. Bernardo et al. Oxford University Press, Oxford, 1996, pp. 599–608.

Mark Girolami and Ben Calderhead. "Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.2 (2011), pp. 123–214. ISSN: 1467-9868. DOI: 10.1111/j.1467-9868.2010.00765.x. (Visited on 07/22/2023).

References III

H. Haario, Saksman E, and J. Tamminen. "An Adaptive Metropolis Algorithm". In: *Bernoulli. Official Journal of the Bernoulli Society for Mathematical Statistics and Probability* 7.2 (2001), pp. 223–242. DOI: bj/1080222083.

Matthew D Hoffman and Andrew Gelman. "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". In: (2011).

Nicola Piana Agostinetti, Genny Giacomuzzi, and Alberto Malinverno. "Local Three-Dimensional Earthquake Tomography by Trans-Dimensional Monte Carlo Sampling". In: *Geophysical Journal International* 201.3 (Apr. 2015), pp. 1598–1617. ISSN: 1365-246X, 0956-540X. DOI: 10.1093/gji/ggv084.

D. Vats, J. M. Flegal, and G. L. Jones. "Multivariate output analysis for Markov chain Monte Carlo". In: *Biometrika* 106.2 (June 2019), pp. 321–337. ISSN: 0006-3444. DOI: 10.1093/biomet/asz002.