



Particle Metropolis-Hastings

Johan Dahlin

`work@johandahlin.com`

Department of Information Technology,
Uppsala University, Sweden.

September 29, 2016



This is collaborative work together with

Dr. Fredrik Lindsten (Uppsala University, Sweden).

Prof. Thomas Schön (Uppsala University, Sweden).

Andreas Svensson (Uppsala University, Sweden).



What are we going to do?

- Give a (hopefully) gentle introduction to (P)MCMC.
- Develop some intuition for PMH and its pros/cons.

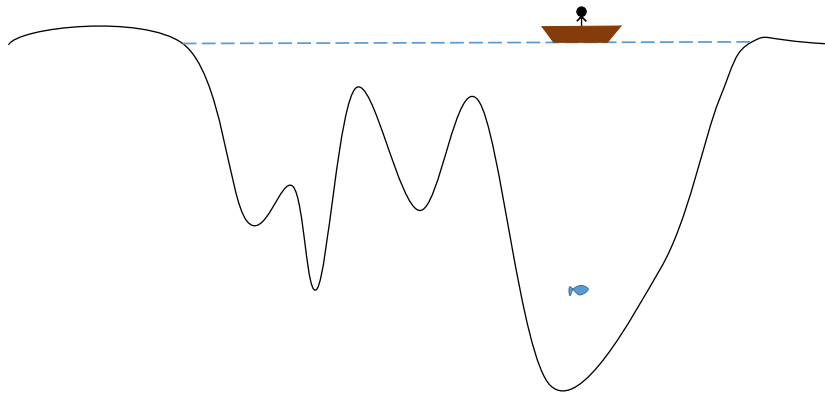
Why are we doing this?

- PMH is general algorithm for Bayesian inference.
- Relatively simple to implement and tune.

How will we do this?

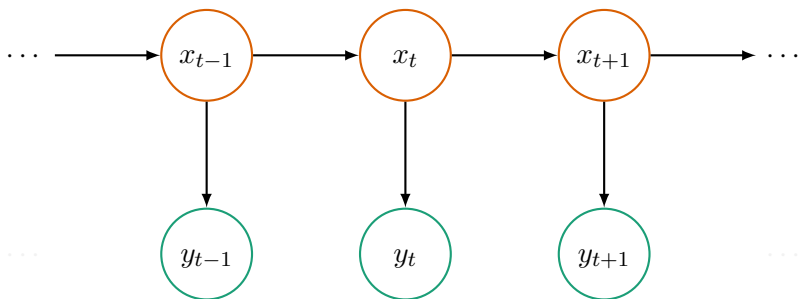
- Employ intuition and analogues with optimisation.
- Investigate PMH using simulations and not maths.
- Illustrate PMH by water tank benchmark.
- By asking questions.

Mapping a lake



State space models

Markov chain $[X_{0:T}, Y_{1:T}]$ with $X_t \in \mathcal{X} = \mathbb{R}, Y_t \in \mathcal{Y} = \mathbb{R}, t \in \mathbb{N}$.



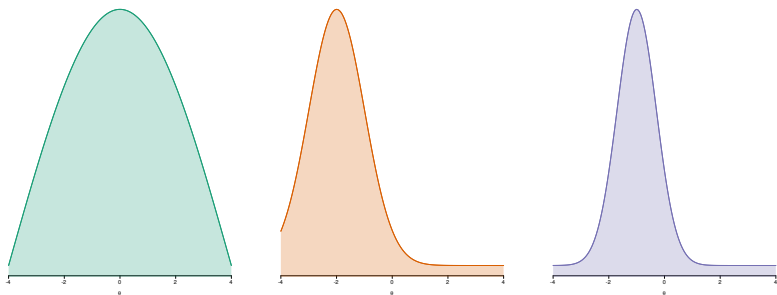
$$x_0 \sim \mu_\theta(x_0) \quad x_{t+1}|x_t \sim f_\theta(x_{t+1}|x_t), \quad y_t|x_t \sim g_\theta(y_t|x_t).$$

Example: linear Gaussian SSM ($\theta = [\mu, \phi, \sigma_v, \sigma_e]$):

$$x_{t+1}|x_t \sim \mathcal{N}(x_{t+1}; \mu + \phi(x_t - \mu), \sigma_v^2), \quad y_t|x_t \sim \mathcal{N}(y_t; x_t, \sigma_e^2).$$



Bayesian parameter inference



$$\pi(\theta) = p(\theta|y) \propto p(y|\theta)p(\theta), \quad \pi[\varphi] = \mathbb{E}_\pi[\varphi(\theta)] = \int \varphi(\theta')\pi(\theta') d\theta'.$$

Exploring posteriors by Markov chains

Markov chains: basic properties

A **sequence of random variables** $\{X_k\}_{k=0}^K$ with the property

$$\mathbb{P}[X_k \in A | x_{0:k-1}] = \mathbb{P}[X_k \in A | x_{k-1}] = \int_A R(x_{k-1}, x_k) dx_k.$$

We will consider **ergodic chains** with the properties:



Reach any point
(irreducible)

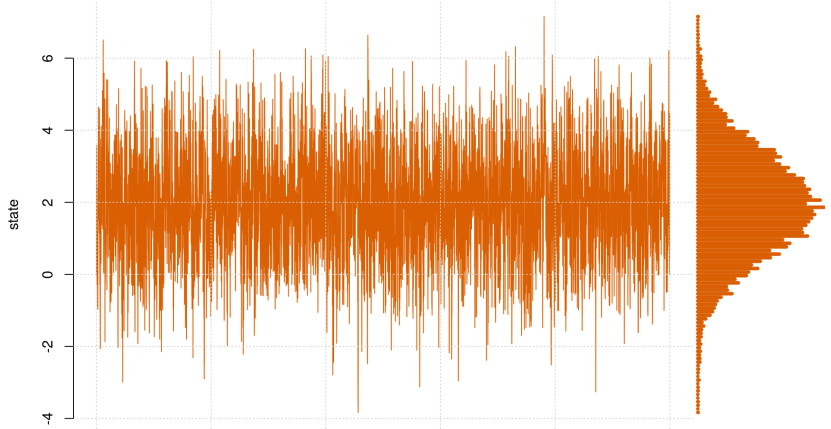


No cycles
(aperiodic)



Does not get stuck
(recurrent)

Markov chains: stationary distribution



$$\theta_k | \theta_{k-1} \sim \mathcal{N}(\theta_k; \mu + \phi(\theta_{k-1} - \mu), \sigma^2).$$

Metropolis-Hastings: algorithm

Initialise in θ_0 and then generate samples $\{\theta_k\}_{k=1}^K$ from $p(\theta|y)$ by

- (i) Sample a **candidate parameter** θ' by

$$\theta' \sim \mathcal{N}(\theta'; \theta_{k-1}, \Sigma).$$

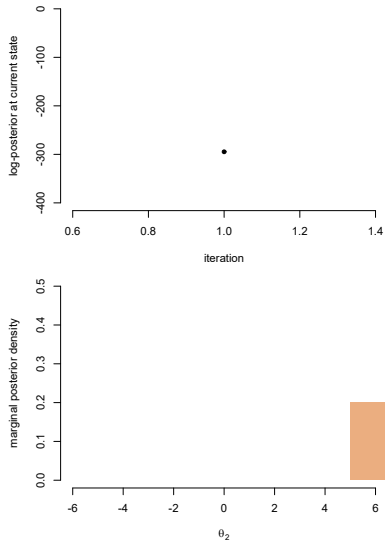
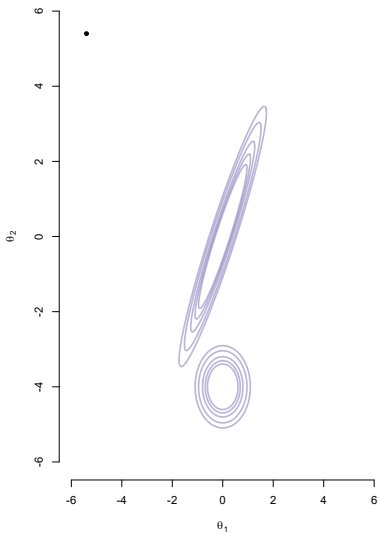
- (ii) **Accept** θ' by setting $\theta_k \leftarrow \theta'$ with probability

$$\min \left\{ 1, \frac{p(\theta'|y)}{p(\theta_{k-1}|y)} \right\} = \min \left\{ 1, \frac{p(\theta')}{p(\theta_{k-1})} \frac{p(y|\theta')}{p(y|\theta_{k-1})} \frac{p(y)}{p(y)} \right\}$$

and otherwise **reject** θ' by setting $\theta_k \leftarrow \theta_{k-1}$.

User choices: K and Σ .

Metropolis-Hastings: toy example



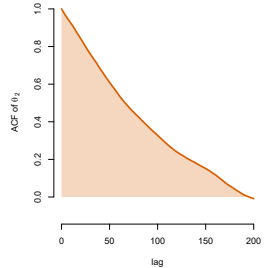
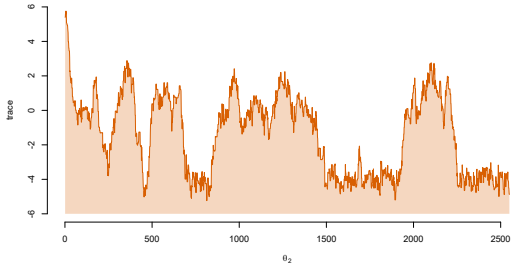
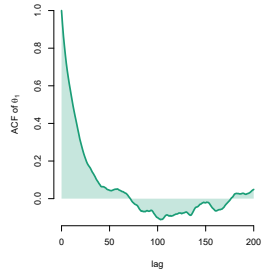
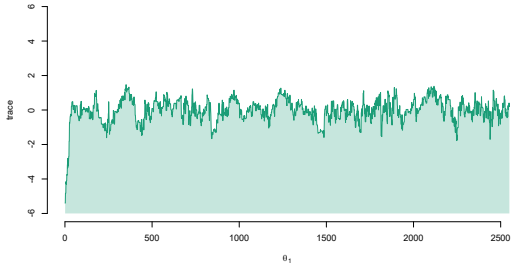


Metropolis-Hastings: toy example

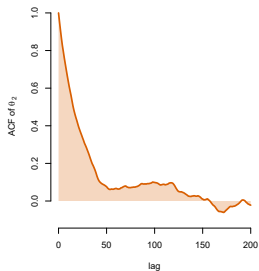
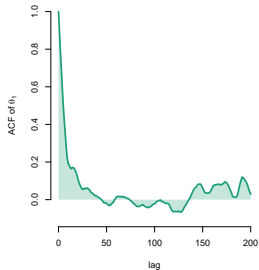
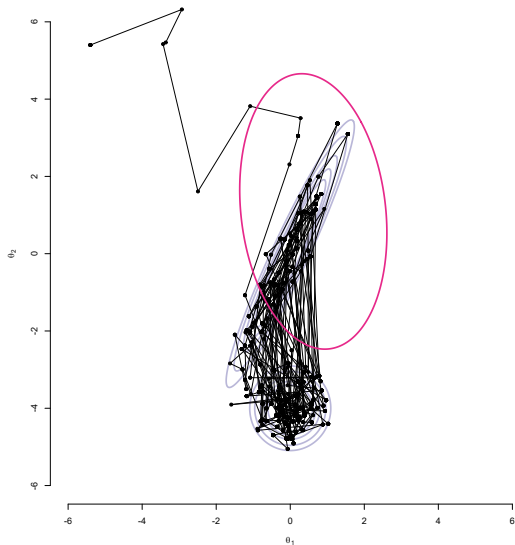


Metropolis-Hastings: toy example

Metropolis-Hastings: proposal and mixing



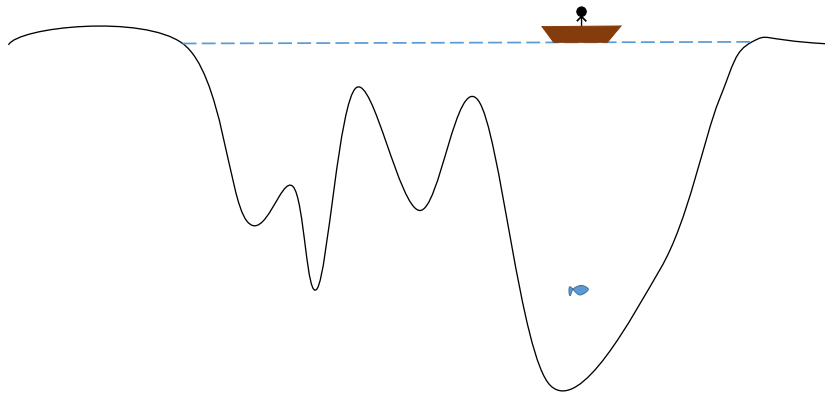
Metropolis-Hastings: proposal and mixing



Approximating the
target by particles



Mapping a stormy lake



Particle Metropolis-Hastings (PMH)

Initialise in θ_0 and then generate samples $\{\theta_k\}_{k=1}^K$ from $p(\theta|y)$ by

- (i) Sample a **candidate parameter** θ' by

$$\theta' \sim \mathcal{N}(\theta'; \theta_{k-1}, \Sigma).$$

- (ii) Run a particle filter with N particles to estimate $\widehat{p}^N(\theta'|y)$.

- (iii) **Accept** θ' by setting $\theta_k \leftarrow \theta'$ with probability

$$\min \left\{ 1, \frac{\widehat{p}^N(\theta'|y)}{\widehat{p}^N(\theta_{k-1}|y)} \right\},$$

and otherwise **reject** θ' by setting $\theta_k \leftarrow \theta_{k-1}$.

User choices: K , Σ and N .



Water tank example: model

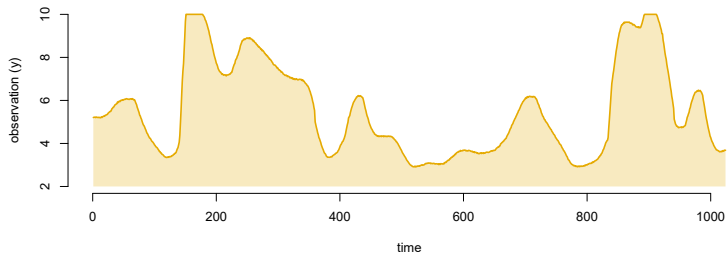
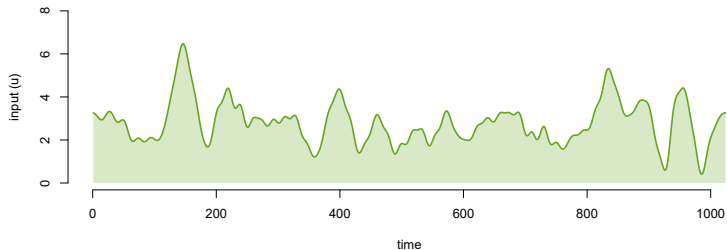
Consider the model

$$\begin{aligned}\dot{x}_1(t) &= -\mathbf{k}_1\sqrt{x_1(t)} + \mathbf{k}_4u(t) + w_1(t), \\ \dot{x}_2(t) &= \mathbf{k}_2\sqrt{x_1(t)} - \mathbf{k}_3\sqrt{x_2(t)} + w_2(t), \\ y(t) &= x_2(t) + e(t),\end{aligned}$$

where $w_1(t), w_2(t), e(t)$ are independent Gaussian and The parameters are $\theta = \{\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4, \dots\}$ with $p(\theta) \propto 1$.

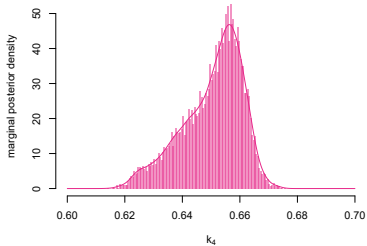
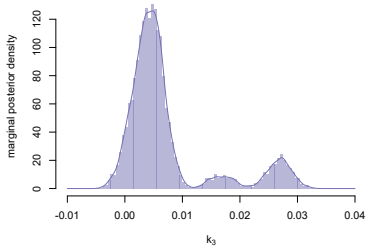
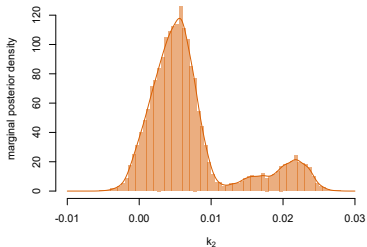
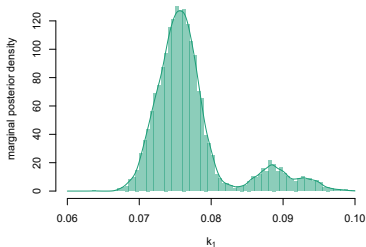


Water tank example: data



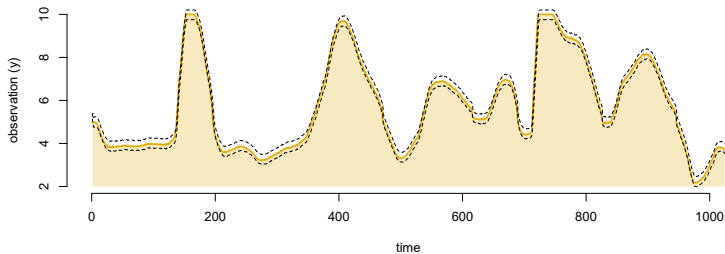
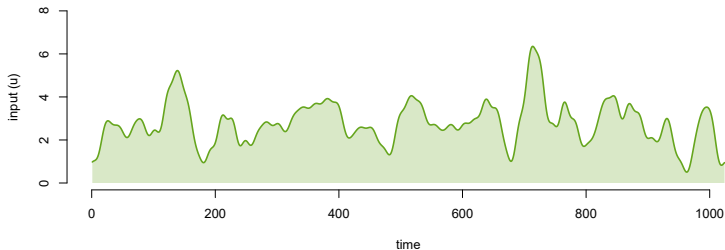


Water tank example: parameter posteriors





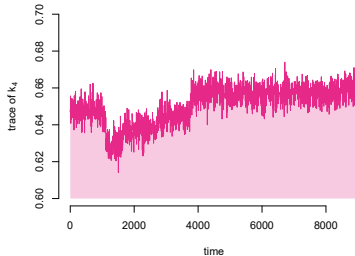
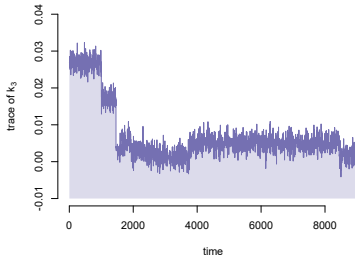
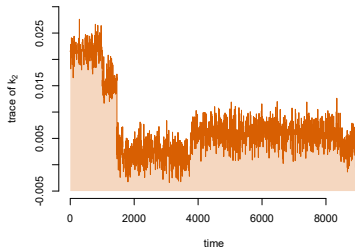
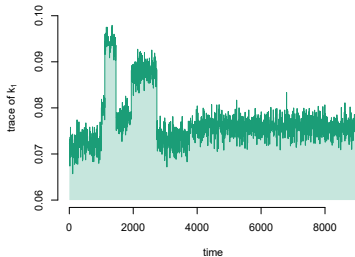
Water tank example: state predictions



Improving the PMH algorithm



Water tank example: trace plots





What are some open questions?

- Decreasing **computational time** when T is large.
Correlating and improving the particle filter.
- Obtaining **better mixing** when $p = |\theta|$ is large (>5).
Add gradient and Hessian information into proposal.
- Devising **better mixing** when $n_x = |x|$ is large (>10).
Improving the particle filter.
- Decrease the **amount of tuning** by the user.
Adaptive algorithms and rules-of-thumb.

What did we do?

- Gave a (hopefully) gentle introduction to (P)MCMC.
- Developed some intuition for PMH and its pros/cons.

Why did we do this?

- PMH is general algorithm for Bayesian inference.
- Relatively simple to implement and tune.

What are you going to do now?

- Remember that the PMH algorithm exist.
- Learning more by reading our tutorial.
- Try to implement the algorithm yourself.



Getting started with particle Metropolis-Hastings for inference in nonlinear dynamical models

Johan Dahlin* and Thomas B. Schön†

April 1, 2016

Abstract

We provide a gentle introduction to the particle Metropolis-Hastings (PMH) algorithm for parameter inference in nonlinear state space models (SSMs) together with a software implementation in the statistical programming language R. Throughout this tutorial, we develop an implementation of the PMH algorithm (and the integrated particle filter), which is distributed as the package **pmhtutorial** available from the CRAN repository. Moreover, we provide the reader with some intuition for how the algorithm operates and discuss some solutions to numerical problems that might occur in

Complete tutorial on PMH is available at [arXiv:1511.01707](https://arxiv.org/abs/1511.01707).

Thank you for listening

Comments, suggestions and/or questions?

Johan Dahlin

`work@johandahlin.com`

`work.johandahlin.com`

Remember: the tutorial is available at [arXiv:1511.01707](https://arxiv.org/abs/1511.01707)

Particle filtering [I/II]

An instance of sequential Monte Carlo (SMC) samplers.

Estimates $\mathbb{E}[\varphi(x_t)|y_{1:t}]$ and $p_\theta(y_{1:T})$.

Computational cost of order $\mathcal{O}(NT)$ (with $N \sim T$).

Well-understood statistical properties.

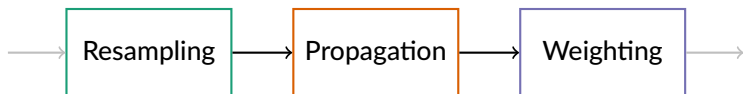
(unbiasedness, large deviation inequalities, CLTs)

References:

A. Doucet and A. Johansen, [A tutorial on particle filtering and smoothing](#). In D. Crisan and B. Rozovsky (editors), The Oxford Handbook of Nonlinear Filtering. Oxford University Press, 2011.

O. Cappé, S.J. Godsill and E. Moulines, [An overview of existing methods and recent advances in sequential Monte Carlo](#). In Proceedings of the IEEE 95(5), 2007.

Particle filtering [II/II]



By iterating:

Resampling: $\mathbb{P}(a_t^{(i)} = j) = \tilde{w}_{t-1}^{(j)}$, for $i, j = 1, \dots, N$.

Propagation: $x_t^{(i)} \sim f_\theta(x_t | x_{t-1}^{(i)})$, for $i = 1, \dots, N$.

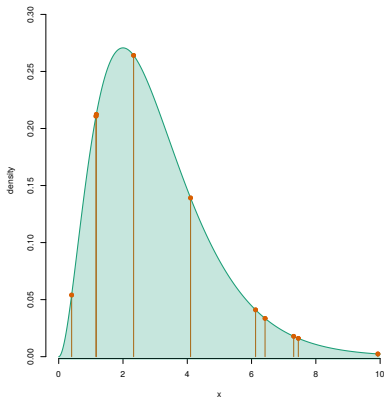
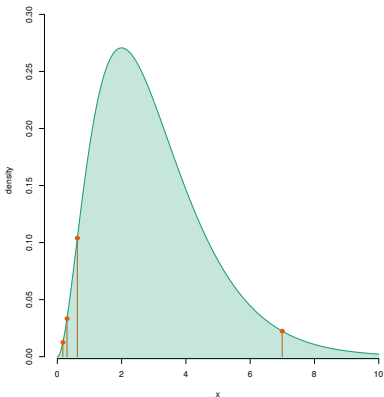
Weighting: $w_t^{(i)} = g_\theta(y_t | x_t^{(i)})$, for $i = 1, \dots, N$.

We obtain the particle system

$$\left[x_{0:T}^{(i)}, w_{0:T}^{(i)} \right]_{i=1}^N.$$



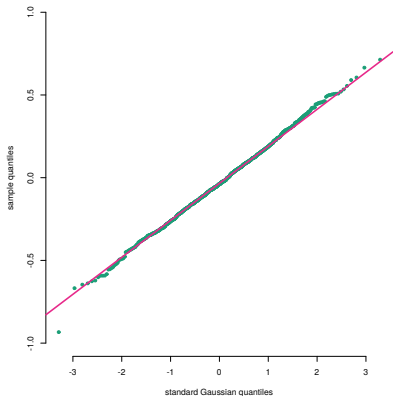
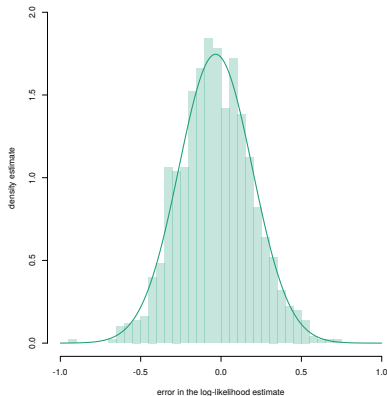
Particle filtering: state estimation



$$\hat{\varphi}_t^N \triangleq \hat{\mathbb{E}}[\varphi(x_t) | y_{1:t}] = \sum_{i=1}^N w_t^{(i)} \varphi(x_t^{(i)}),$$

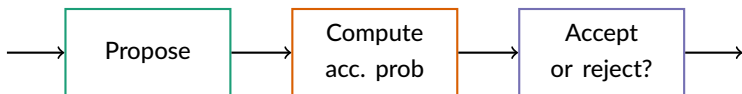
$$\sqrt{N}(\varphi_t - \hat{\varphi}_t^N) \xrightarrow{d} \mathcal{N}(0, \sigma_t^2(\varphi)).$$

Particle filtering: likelihood estimation



$$\underbrace{\log \hat{p}_{\theta}(y_{1:T})}_{\triangleq \hat{\ell}(\theta)} = \sum_{t=1}^T \log \left(\sum_{i=1}^N w_t^{(i)} \right) - T \log N, \quad \sqrt{N} \left(\ell(\theta) - \hat{\ell}(\theta) + \frac{\sigma_{\frac{2}{\pi}}^2}{2N} \right) \xrightarrow{d} \mathcal{N} \left(0, \sigma_{\frac{2}{\pi}}^2 \right).$$

Particle Metropolis-Hastings [I/III]



- Propose: $\theta' \sim q(\theta' | \theta_{k-1}, u')$ and $u' \sim \text{PF}(\theta')$.
- Compute $\widehat{p}_{\theta'}(y_{1:T} | u')$ and the acceptance probability:

$$\alpha(\theta', \theta_{k-1}) = 1 \wedge \frac{p(\theta')}{p(\theta_{k-1})} \frac{\widehat{p}_{\theta'}(y_{1:T} | u')}{\widehat{p}_{\theta_{k-1}}(y_{1:T} | u_{k-1})} \frac{q(\theta_{k-1} | \theta', u')}{q(\theta' | \theta_{k-1}, u_{k-1})}.$$

- Accept or reject? $\theta' \rightarrow \theta_k$ and $u' \rightarrow u_k$ w.p. $\alpha(\theta', \theta_{k-1})$.

Particle Metropolis-Hastings [II/III]

The **target distribution** is given by the parameter proposal

$$\pi(\theta) = \frac{p_{\theta}(y_{1:T})p(\theta)}{p(y_{1:T})}.$$

An **unbiased estimator of the likelihood** is given by

$$\mathbb{E}_m [\hat{p}_{\theta}(y_{1:T}|\mathbf{u})] = \int \hat{p}_{\theta}(y_{1:T}|\mathbf{u})m_{\theta}(\mathbf{u}) \mathbf{d}\mathbf{u} = p_{\theta}(y_{1:T}).$$

An **extended target** is given by

$$\pi(\theta, \mathbf{u}) = \frac{\hat{p}_{\theta}(y_{1:T}|\mathbf{u})m_{\theta}(\mathbf{u})p(\theta)}{p(y_{1:T})} = \frac{\hat{p}_{\theta}(y_{1:T}|\mathbf{u})m_{\theta}(\mathbf{u})\pi(\theta)}{p_{\theta}(y_{1:T})}.$$

Particle Metropolis-Hastings [III/III]

$$\begin{aligned}
 \int \pi(\theta, \mathbf{u}) \, d\mathbf{u} &= \int \frac{\hat{p}_\theta(y_{1:T}|\mathbf{u})m_\theta(\mathbf{u})\pi(\theta)}{p_\theta(y_{1:T})} \, d\mathbf{u} \\
 &= \frac{\pi(\theta)}{p_\theta(y_{1:T})} \underbrace{\int \hat{p}_\theta(y_{1:T}|\mathbf{u})m_\theta(\mathbf{u}) \, d\mathbf{u}}_{=p_\theta(y_{1:T})}, \\
 &= \pi(\theta).
 \end{aligned}$$

That is, the marginal is the desired target distribution and the Markov chain is kept invariant.