Getting started with particle Metropolis-Hastings

Johan Dahlin
johan.dahlin@liu.se / work.johandahlin.com

Division of Automatic Control,
Linköping University, Sweden.

LINKÖPING UNIVERSITY

Download MATLAB skeleton

`http://goo.gl/omfFVY`

Why are we doing this?

Teach you how to implement the PMH algorithm!
Make my PhD defense a bit more understandable.

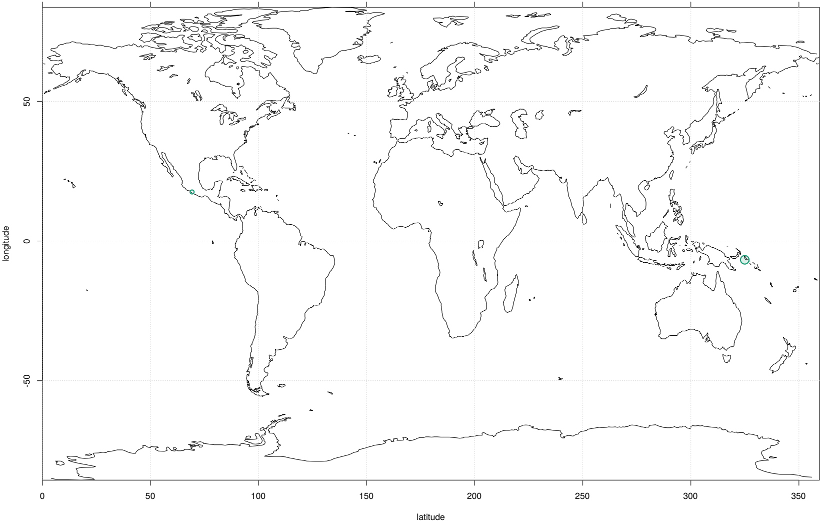
How will we do this?

Interactive session!
Some theory on slides and you code it up yourself.

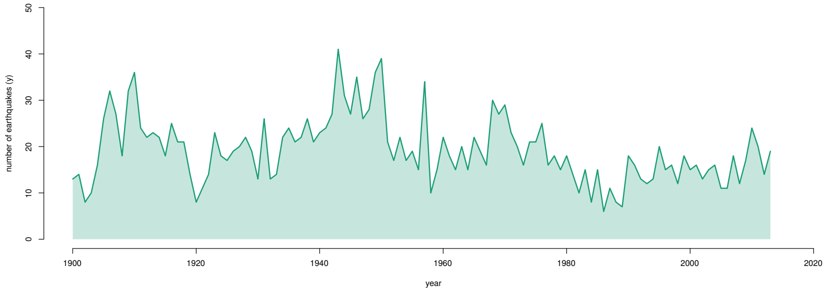
What are we going to do?

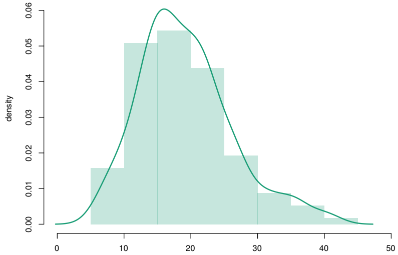
Discuss Markov chain theory.
Implement PMH in MATLAB using code skeleton.
Construct a model of Earthquake counts.
Discuss some open problems with PMH.

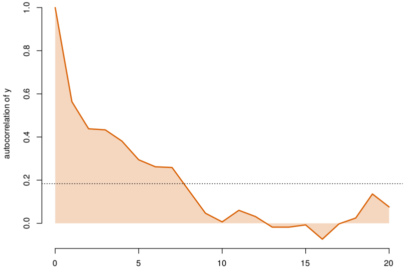
Problem: modelling earthquake counts [I/III]



Problem: modelling earthquake counts [II/III]







Problem: modelling earthquake counts [III/III]

We consider the model

$$x_{t+1}|x_t \sim \mathcal{N}\Big(x_{t+1}; \phi x_t, \boldsymbol{\sigma}^2\Big),$$
$$y_t|x_t \sim \mathcal{P}\Big(y_t; \beta \exp(x_t)\Big),$$

where the parameters describe:

ϕ: persistence of intensity.

$\boldsymbol{\sigma}$: standard deviation of innovation in intensity.

β: *nominal* number of annual earthquakes.

Task: Estimate $\boldsymbol{\theta} = \{\phi, \boldsymbol{\sigma}, \beta\}$ and $x_{0:T}$ given $y_{1:T}$.

Particle filtering

Three steps to approximate the state and the likelihood:

(i) Resample the particle $x_{t-1}^{(i)}$ using $\{w_{t-1}^{(i)}\}_{i=1}^{N}$ to obtain $\tilde{x}_{t-1}^{(i)}$.

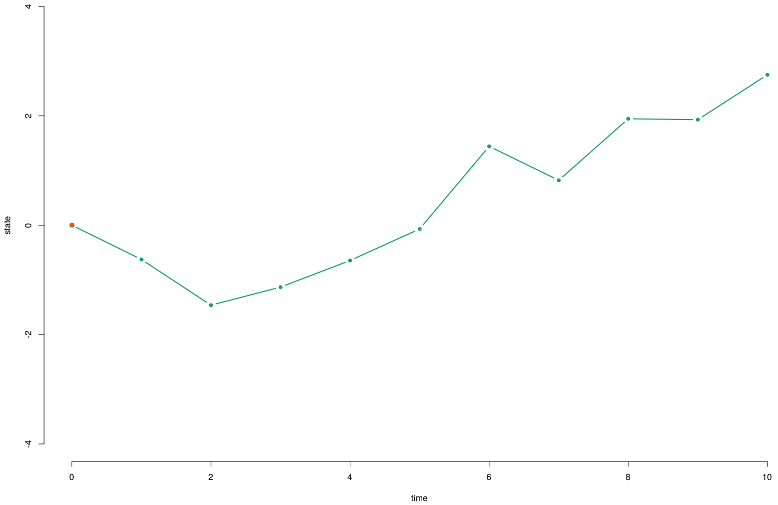
(ii) Propagate the particle by

$$x_t^{(i)} \sim R_\theta\left(x_t^{(i)} | \tilde{x}_{t-1}^{(i)}\right).$$

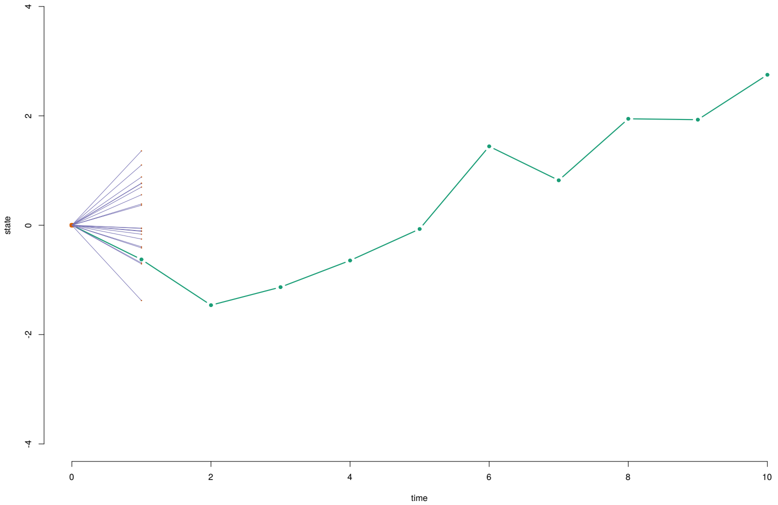
(iii) Compute the weight for the particle by

$$\tilde{w}_t^{(i)} = W\left(x_t^{(i)}\right), \quad w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}}.$$

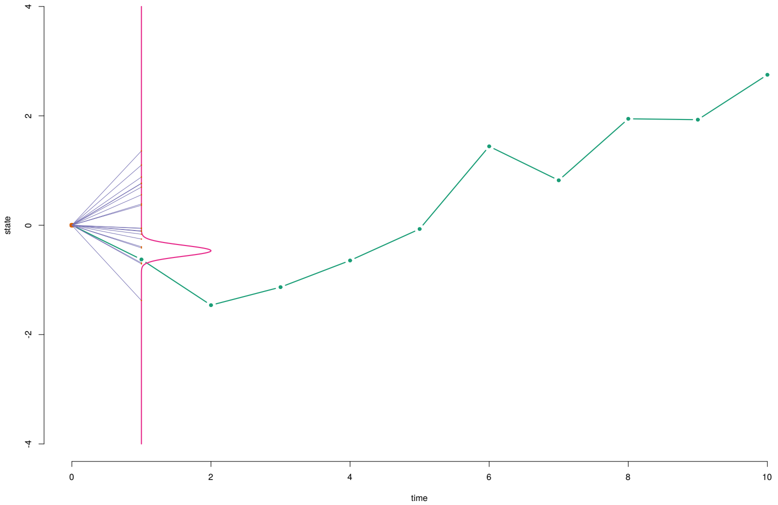
Particle filtering: toy example



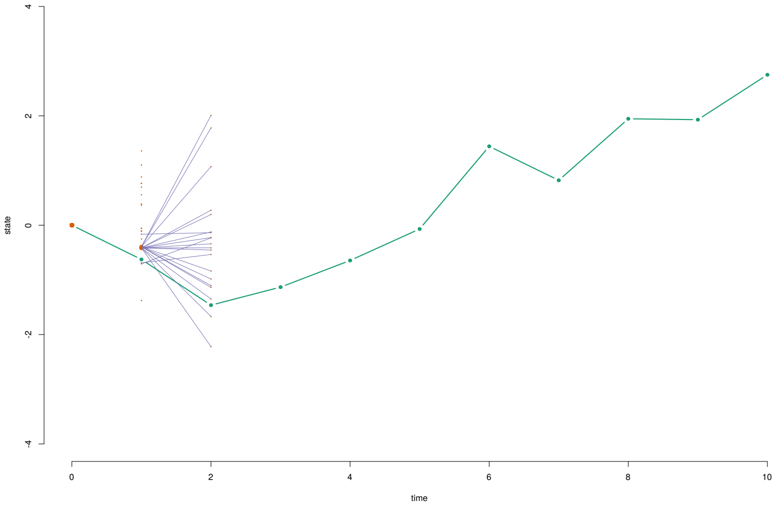
Particle filtering: toy example



Particle filtering: toy example



Particle filtering: toy example



LINKÖPING UNIVERSITY

Bootstrap particle filtering for the Earthquake model

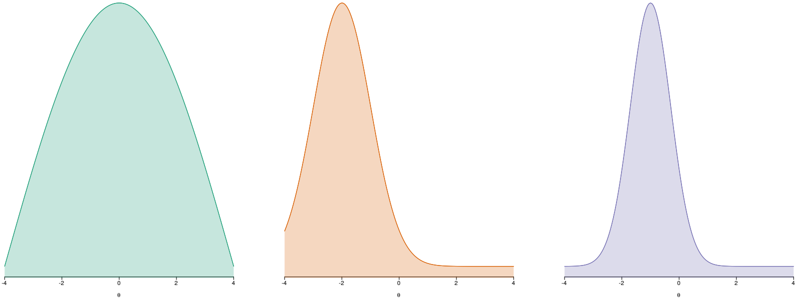
Lets have a look at

`example2_earthquake_state.m`

and

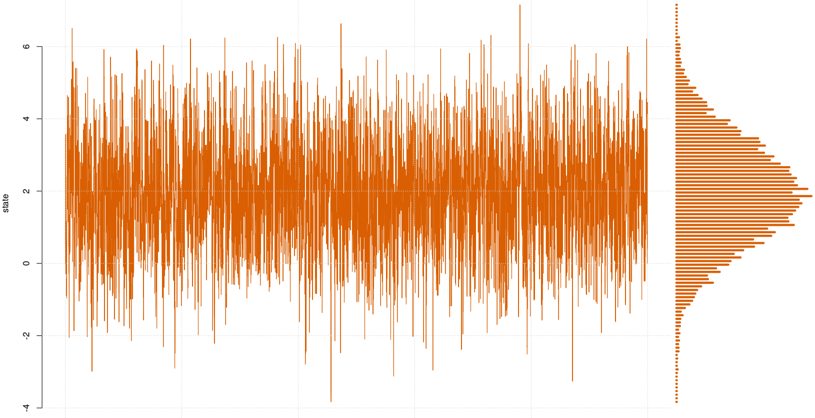
`sm_earthquake.m`.

Bayesian parameter inference



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

Stationary distribution: Autoregressive process



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

Reversibility

Given an ergodic chain, a sufficient condition the existence of a stationary distribution π is

$$\pi(\theta_{k-1})R(\theta_{k-1},\theta_k) = \pi(\theta_k)R(\theta_k,\theta_{k-1}), \quad \text{for any } \theta_{k-1},\theta_k \in \Theta.$$

Lets use this to construct a chain such that $\pi(\theta) = p(\theta|y)$.

Metropolis-Hastings

Consists of two steps to generate a Markov chain $\{\theta_k\}_{k=1}^{K}$:

(i) Sample a candidate parameter θ' from a proposal distribution.

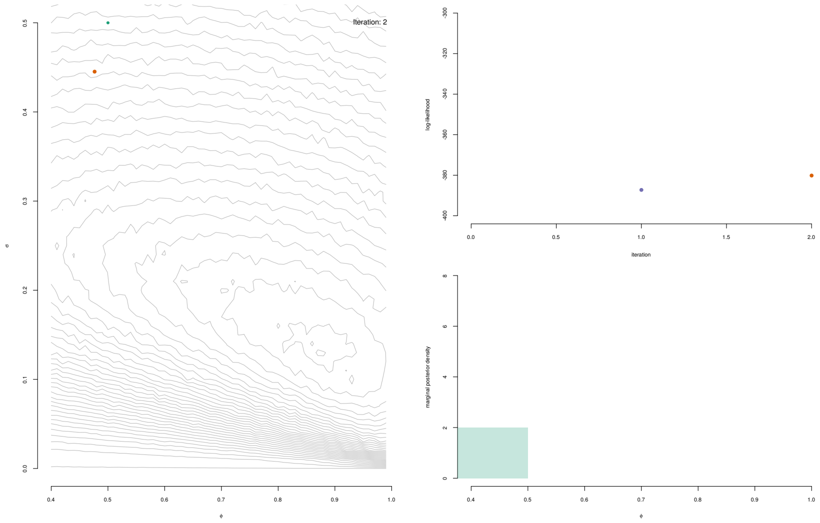
$$\theta' \sim q(\theta' | \theta_{k-1}).$$

(ii) Accept θ' by setting $\theta_k = \theta'$ with probability

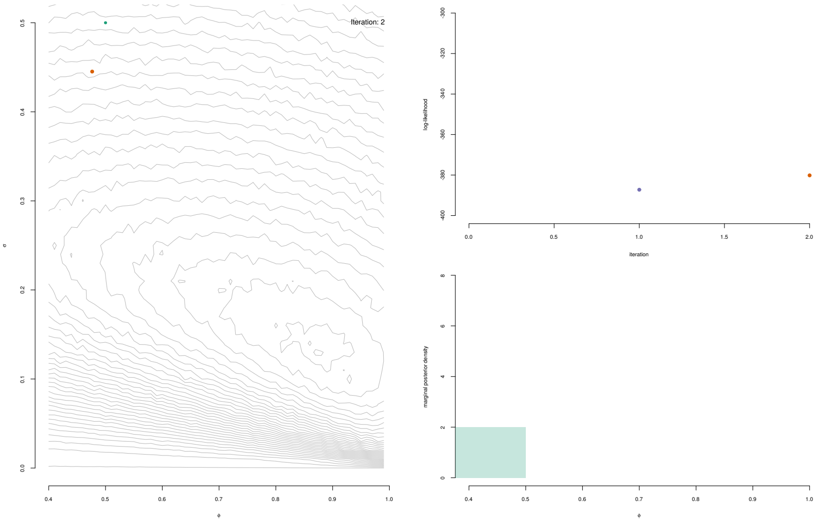
$$\min\left\{1, \alpha(\theta_{k-1}, \theta')\right\}, \quad \alpha(\theta_{k-1}, \theta') = \frac{\pi(\theta')}{\pi(\theta_{k-1})} \frac{q(\theta_{k-1} | \theta')}{q(\theta' | \theta_{k-1})}$$

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

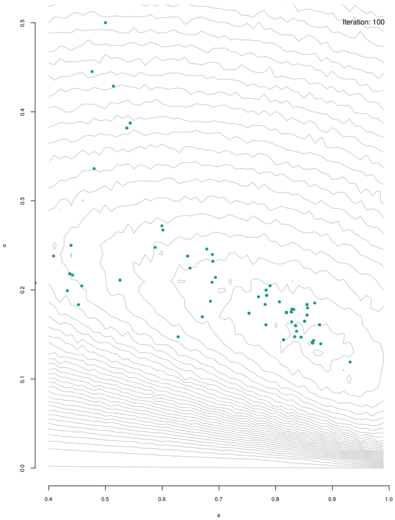
Metropolis-Hastings: toy example

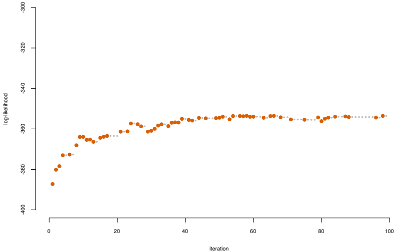


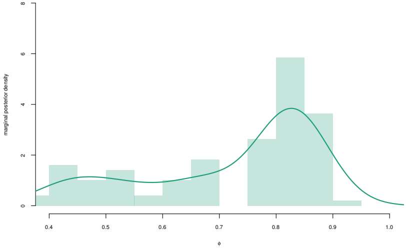
Metropolis-Hastings: toy example



Metropolis-Hastings: toy example







Metropolis-Hastings for inference in SSMs

We select the parameter posterior as the target distribution

$$\pi(\theta) = \frac{p(y|\theta)p(\theta)}{p(y)},$$

and a Gaussian random walk for the proposal distribution

$$q(\theta'|\theta_{k-1}) = \mathcal{N}\left(\theta'; \theta_{k-1}, \epsilon^2 \Sigma\right).$$

Hence, we obtain the acceptance probability

$$\alpha(\theta_{k-1}, \theta') = \frac{p(y|\theta')}{p(y|\theta_{k-1})} \frac{p(\theta')}{p(\theta_{k-1})} \frac{p(y)}{p(y)} \frac{q(\theta'|\theta_{k-1})}{q(\theta'|\theta_{k-1})} = \frac{p(y|\theta')}{p(y|\theta_{k-1})} \frac{p(\theta')}{p(\theta_{k-1})}.$$

where the likelihood $p(y|\theta)$ can be estimated using a particle filter.

PMH for the Earthquake model [I/II]

We have $\theta = \{\phi, \sigma, \beta\}$ and use $p(\theta) \propto 1$ in the target distribution

$$\pi(\theta) \propto p(y|\theta)p(\theta),$$

and a Gaussian random walk for the proposal distribution

$$q(\theta'|\theta_{k-1}) = \mathcal{N}\left(\theta'; \theta_{k-1}, 0.8 \begin{bmatrix} 0.07^2 & 0 & 0 \\ 0 & 0.03^2 & 0 \\ 0 & 0 & 2^2 \end{bmatrix}\right).$$

Hence, we obtain the acceptance probability

$$\alpha(\theta_{k-1}, \theta') = \frac{\widehat{p}^N(y|\theta')}{\widehat{p}^N(y|\theta_{k-1})} = \exp\left[\log \widehat{p}^N(y|\theta') - \log \widehat{p}^N(y|\theta_{k-1})\right].$$

PMH for the Earthquake model [II/II]

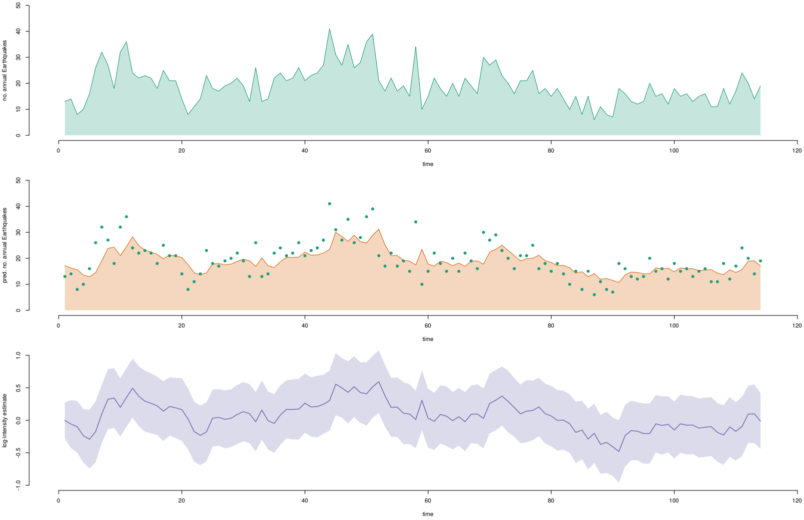
Lets have a look at

`example3_earthquake_parameters.m`

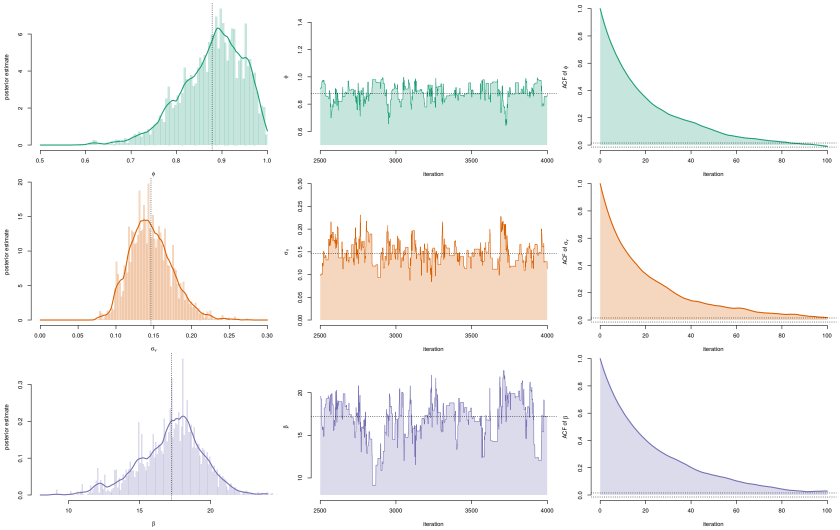
and

`pmh_earthquake.m`.

PMH for the Earthquake model: state estimate



PMH for the Earthquake model: parameter estimate



Some challenges

- How do we choose N, R_θ and W_θ in the particle filter?
- How do we choose K and q in the PMH algorithm?
- Scalability in T and p?

- Rule-of-thumbs (does not always work):
  - Choose N such that

$$\mathbb{V}\left[\log \widehat{p}^N(y|\theta)\right] \approx 1.4.$$

  - Choose q (if target is close to a Gaussian) to be

$$q(\theta'|\theta_{k-1}) = \mathcal{N}\left(\theta'; \theta_{k-1}, \frac{2.562^2}{p}\widehat{\Sigma}\right),$$

  or use adaptive algorithms.

Some of the improvements proposed in my thesis

- [Papers B and C] Tailor $q(\theta'|\theta_{k-1})$ to better fit $\pi(\theta)$.
  Result: we can reduce K and simplify tuning.

- [Paper D] Introduce a positive correlation in $\widehat{p}^N(y|\theta)$.
  Result: we can reduce N and K.

Why did we do this?

- Teach you how to implement the PMH algorithm!
- Make my PhD defense a bit more understandable.

How did we do it?

- Interactive session!
- Some theory on slides and you code it up yourself.

What did we achieve?

- A fairly general implementation of PMH.
- Can be used for inference in (any) scalar SSM.
- Simple to tailor to your own specific problem.

Thank you for listening

Comments, suggestions and/or questions?

Johan Dahlin

johan.dahlin@liu.se

work.johandahlin.com

Complete tutorial is found at arXiv:1511.01707

LINKÖPING UNIVERSITY