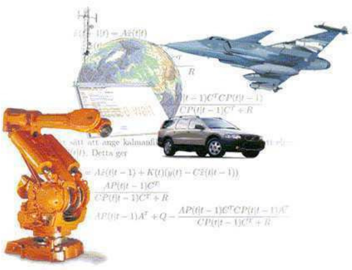
Sequential Monte Carlo for inference in nonlinear state space models

Licentiate's degree seminar, Linköping, May 28, 2014.



Johan Dahlin
johan.dahlin@liu.se

Division of Automatic Control,
Linköping University,
Sweden.

This is collaborative work with:

Dr. Fredrik Lindsten (Uni. of Cambridge / Linköping University)
Prof. Thomas B. Schön (Uppsala University)
Dr. Cristian Rojas (Royal Institute of Technology, KTH)
Patricio E. Valenzuela (Royal Institute of Technology, KTH)
Prof. Mattias Villani (Linköping University)

Summary

Aim

To build models of dynamical systems using domain knowledge and recorded (input-)output data.

Methods

Dynamical statistical models (state space models).
Bayesian and maximum likelihood inference.
Computer intensive statistical simulation methods.

Contributions

Novel parameter inference algorithms.
Improved efficiency of existing methods.
Generalised existing methods to new model classes.

Motivating example: simulating the Swedish economy

Dynamic stochastic general equilibrium.

circa 40 states.
circa 60 parameters.

12 outputs.

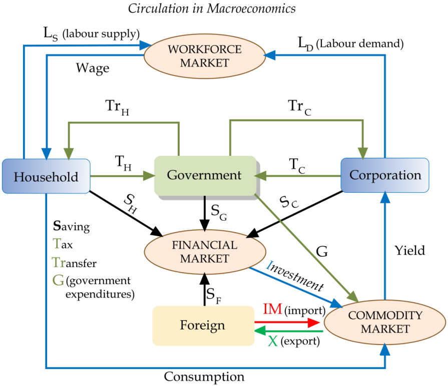
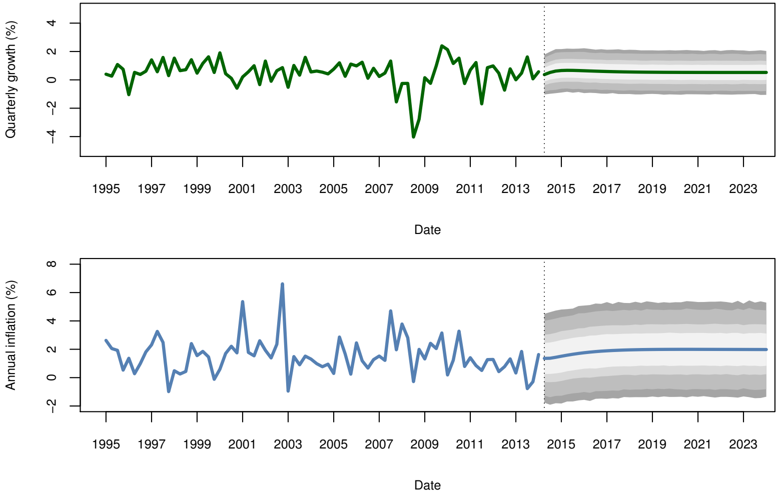


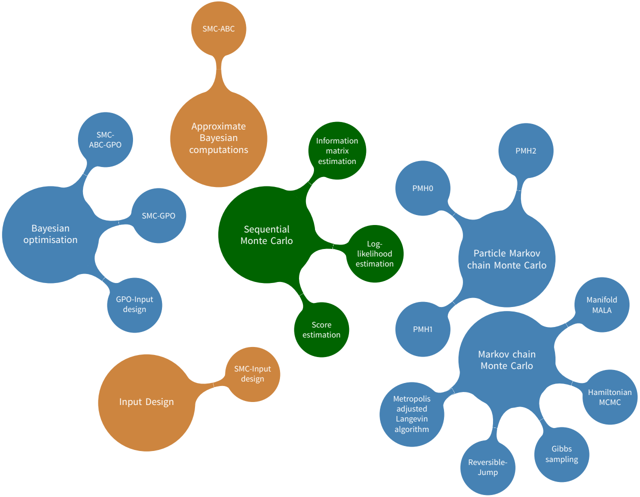
Image used with courtesy of LadyofHats@Wikipedia.

Motivating example: simulating the Swedish economy

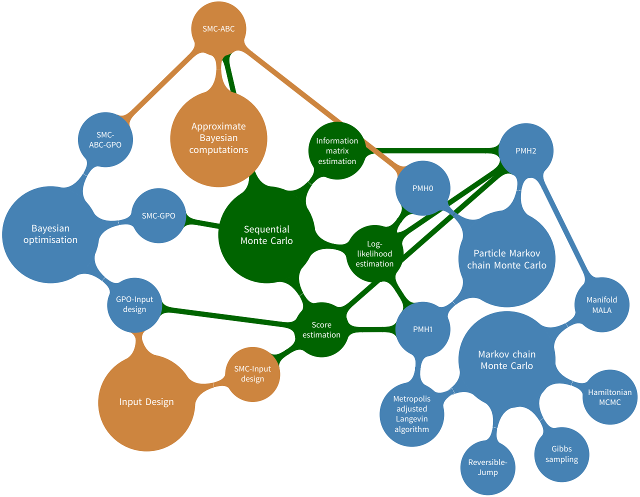
Motivating example: simulating the Swedish economy



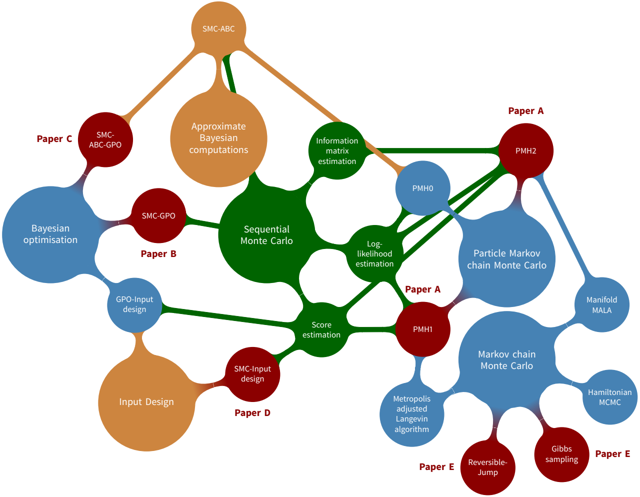
Overview of the thesis



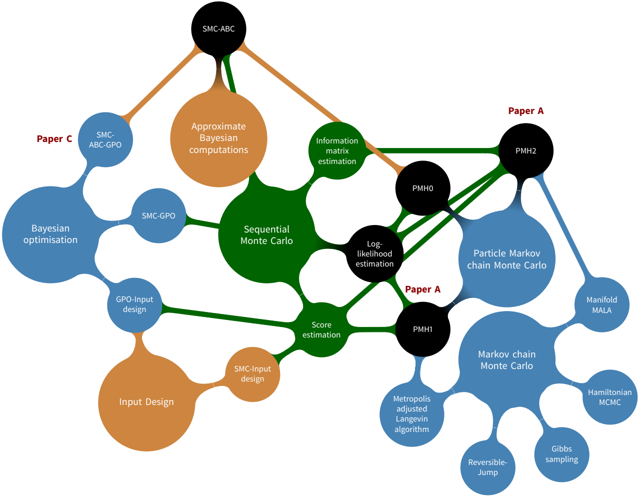
Overview of the thesis



Overview of the thesis



Overview of the thesis



Nonlinear state space model

Consider a *nonlinear state space model*

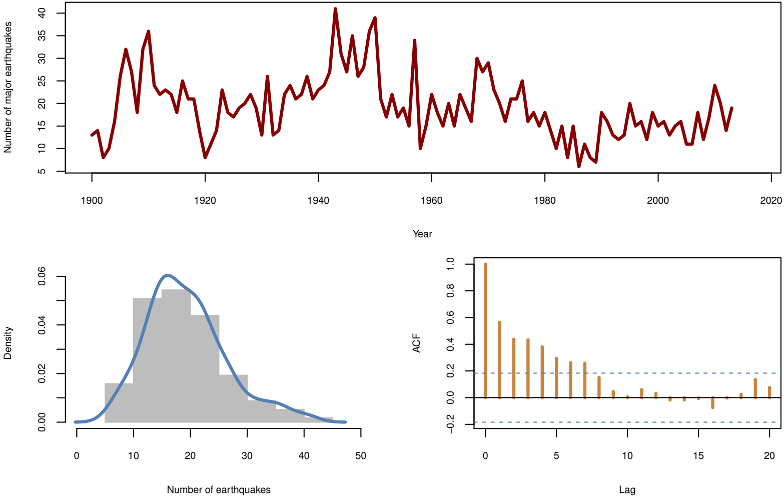
$$\begin{aligned} x_0 &\sim \mu(x_0), \\ x_{t+1} | x_t &\sim f_{\boldsymbol{\theta}}(x_{t+1} | x_t), \\ y_t | x_t &\sim g_{\boldsymbol{\theta}}(y_t | x_t), \end{aligned}$$

where $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^d$, $x_t \subset \mathbb{R}^n$ and $y_t \subset \mathbb{R}^m$.

Inference: compute estimates of $x_{0:T}$ and θ given $y_{1:T}$.

Example: Earthquakes between 1900 and 2013

Example: Earthquakes between 1900 and 2013



Example: A simple model of annual earthquake counts

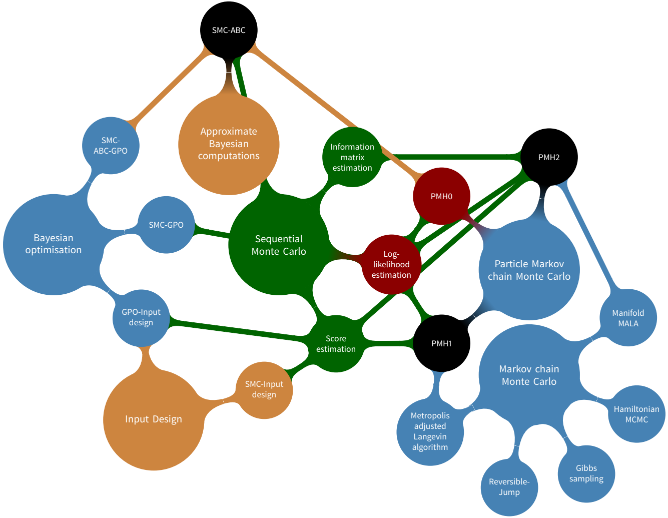
$$x_{t+1}|x_t \sim \mathcal{N}\Big(x_{t+1}; \phi x_t, \boldsymbol{\sigma}^2\Big),$$
$$y_t|x_t \sim \mathcal{P}\Big(y_t; \boldsymbol{\beta} \exp(x_t)\Big),$$

where the parameters describe:

- ϕ: persistence of intensity.
- $\boldsymbol{\sigma}$: standard deviation of innovation in intensity.
- $\boldsymbol{\beta}$: *nominal* number of annual earthquakes.

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

Overview of the thesis



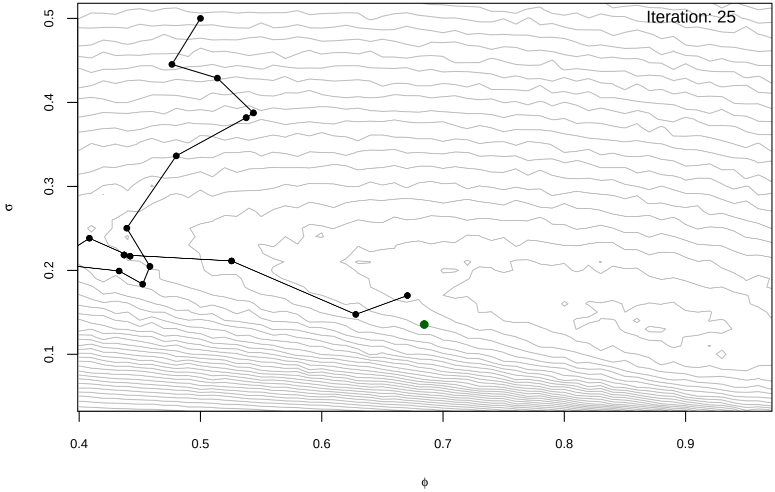
Bayesian parameter inference

Consider the *parameter posterior*

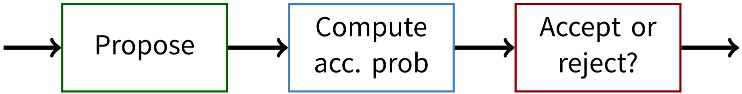
$$\pi(\theta) = p(\theta|y_{1:T}) = \frac{p_\theta(y_{1:T})p(\theta)}{p(y_{1:T})} \propto p_\theta(y_{1:T})p(\theta).$$

posterior = *prior* + *information in data* + *model*.

Metropolis-Hastings algorithm



Metropolis-Hastings algorithm



- Propose: $\theta' \sim q(\theta' | \theta_k)$.
- Compute acceptance probability:

$$\alpha(\theta', \theta_k) = \min\left\{1, \frac{\pi(\theta')}{\pi(\theta_k)} \frac{q(\theta_k | \theta')}{q(\theta' | \theta_k)}\right\}$$

- Accept or reject? $\theta' \to \theta_k$ w.p. $\alpha(\theta', \theta_k)$.

Particle Metropolis-Hastings algorithm

Problem

We cannot compute $p_\theta(y_{1:T})$ in closed form.

Idea

Replace the likelihood with an unbiased estimate $\widehat{p}_\theta(y_{1:T}|u)$.

Implementation

Run a particle filter to estimate the likelihood and $\alpha(\theta'', \theta')$.

Exact approximations

Keeps the Markov chain invariant.
The marginal of the stationary distribution is $\pi(\theta)$.

Particle Metropolis-Hastings algorithm (cont.)

Propose $\rightarrow$ **Compute acc. prob** $\rightarrow$ Accept or reject?

- Propose: $\theta' \sim q(\theta' | \theta_k, u_k)$.
- Compute $\widehat{p}_{\theta'}(y_{1:T} | u_k)$ and the acceptance probability:

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

- Accept or reject? $\theta' \rightarrow \theta_k$ w.p. $\alpha(\theta', \theta_k)$.

Particle filtering

Resampling → Propagation → Weighting

Given the particle system (the random variables u)

$$u = \left\{ a_t^{(i)}, \widetilde{x}_t^{(i)}, w_t^{(i)} \right\}_{i=1}^{N},$$

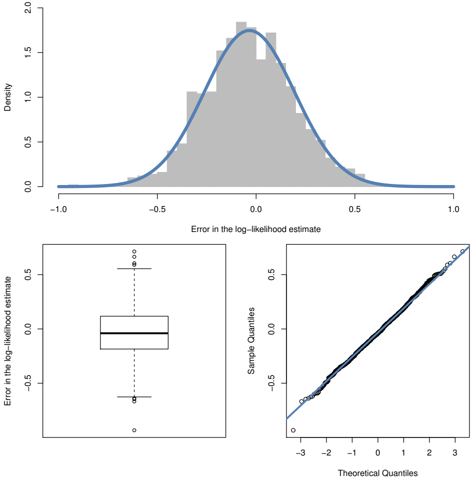
the filtering density is approximated by an *empirical distribution*

$$\widehat{p}_\theta(\mathrm{d}x_t | y_{1:t}) = \sum_{i=1}^{N} \left[\frac{w_t^{(i)}}{\sum_{k=1}^{N} w_t^{(k)}} \right] \delta_{x_t^{(i)}}(\mathrm{d}x_t).$$

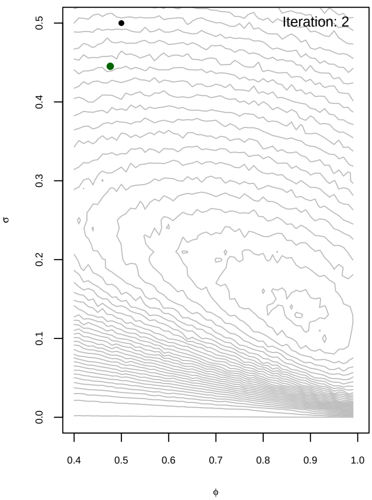
Likelihood estimator

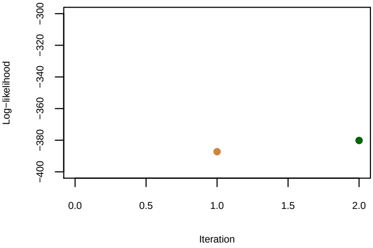
$$\widehat{\mathcal{L}}(\theta) = \widehat{p}_\theta(y_{1:T} | u) = \frac{1}{N^T} \prod_{t=1}^{T} \sum_{i=1}^{N} w_t^{(i)}.$$

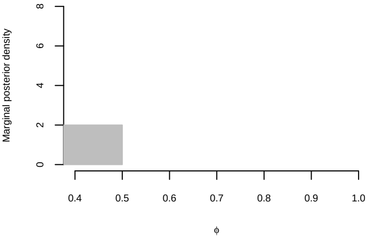
$$\sqrt{N}\Big(\mathcal{L}(\theta) - \widehat{\mathcal{L}}(\theta)\Big) \xrightarrow{d} \mathcal{N}\left(0, \sigma_l^2\right).$$



Example: Parameter inference in earthquake model



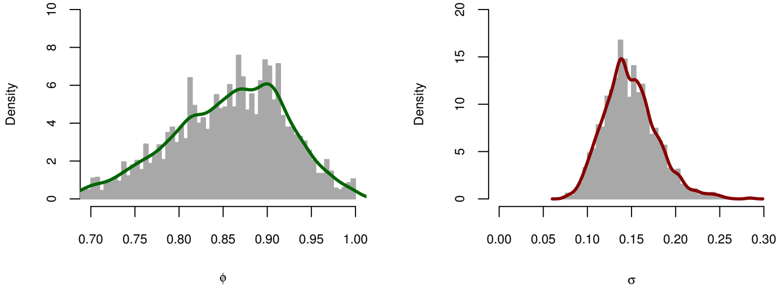




Example: Parameter inference in earthquake model

Example: Parameter inference in earthquake model

Example: Parameter inference in earthquake model



	ϕ	σ
Posterior mean	0.86	0.15
Posterior median	0.86	0.15
Posterior mode	0.90	0.14

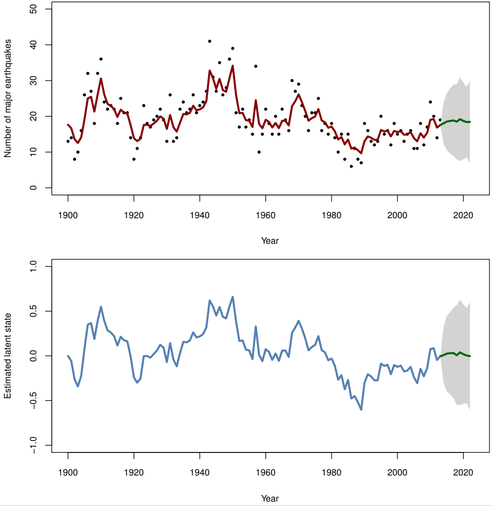
State inference in the earthquake model

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

$$y_t | x_t \sim \mathcal{P}\Big(y_t; \beta \exp(x_t)\Big),$$

with parameters:

$\phi = 0.88$ (persistence.)
$\sigma = 0.15$ (sd. of innovation.)
$\beta = 17.65$ (*nominal* number.)

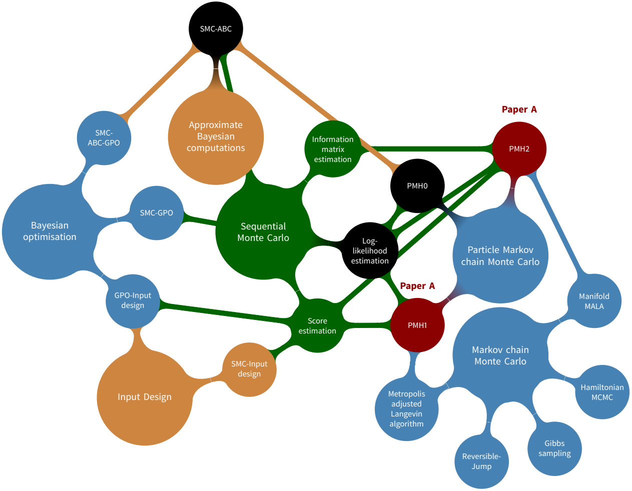


Recap

Previous

Metropolis-Hastings.
Particle Metropolis-Hastings.
Likelihood estimation.
Inclusion of u' in the proposal $q(\theta''|\theta', u')$. (Paper A)

Overview of the thesis



The proposal distribution

A typical Gaussian random walk proposal is given by

$$q(\theta''|\theta') = \mathcal{N}\left(\theta''; \theta', \epsilon^2 I_d\right),$$

where ϵ denotes the standard deviation of the random walk.

Designing the first order proposal

The target distribution:

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

A noisy gradient-based ascent algorithm:

$$\theta'' = \theta' + \frac{\epsilon^2}{2}\nabla_\theta \log \pi(\theta)\big|_{\theta=\theta'} + \epsilon z, \quad z \sim \mathcal{N}(z; 0, 1).$$

The first order proposal

$$q(\theta''|\theta', u') = \mathcal{N}\left(\theta''; \theta' + \frac{\epsilon^2}{2}\widehat{\mathcal{S}}(\theta'|u'), \epsilon^2 I_d\right),$$

$$\mathcal{S}(\theta') = \nabla_\theta \log \pi(\theta)\big|_{\theta=\theta'}.$$

Designing the second order proposal

The target distribution:

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

A noisy Newton algorithm:

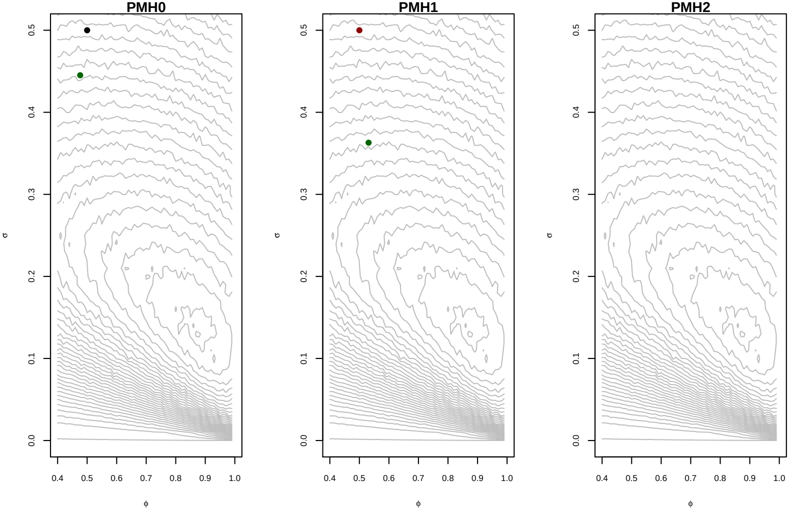
$$\theta'' = \theta' + \frac{\epsilon^2}{2}\left[-\nabla_\theta^2 \log \pi(\theta)\Big|_{\theta=\theta'}\right]^{-1} \mathcal{S}(\theta') + \epsilon\left[-\nabla_\theta^2 \log \pi(\theta)\Big|_{\theta=\theta'}\right]^{-1/2} z.$$

This leads to the second order proposal

$$q(\theta''|\theta', u') = \mathcal{N}\left(\theta''; \theta' + \frac{\epsilon^2}{2}\widehat{\mathcal{S}}(\theta'|u')\left[\widehat{\mathcal{J}}(\theta'|u')\right]^{-1}, \epsilon^2\left[\widehat{\mathcal{J}}(\theta'|u')\right]^{-1}\right),$$

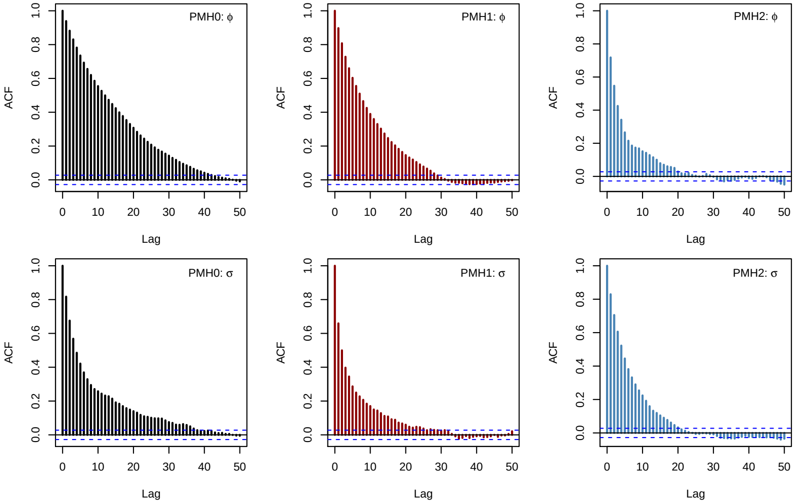
$$\mathcal{J}(\theta') = -\nabla_\theta^2 \log p_\theta(y_{1:T})\Big|_{\theta=\theta'}.$$

Example: Parameter inference in earthquake model



Example: Parameter inference in earthquake model

Integrated autocorrelation time



Integrated autocorrelation time (cont.)

IACT: the number of iterations between two uncorrelated samples (lower is better).

	Acceptance rate	IACT(ϕ)	IACT(σ)
PMH0	0.47	31.82	18.94
PMH1	0.38	21.38	**12.06**
PMH2	0.54	**10.89**	14.15

Summary of Paper A

Contributions

Improved particle Metropolis-Hastings algorithm (Paper A).
Fast estimation of first and second order information (Paper A).

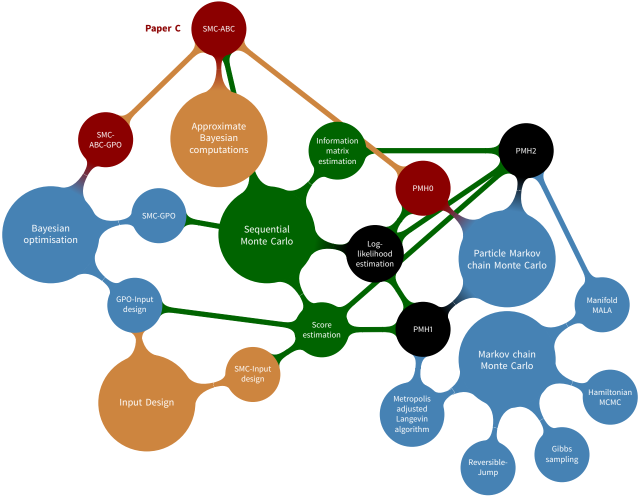
Methods

Include u into the proposal.
Particle fixed-lag smoothing.
Laplace approximation / Random walk on a Riemann manifold.

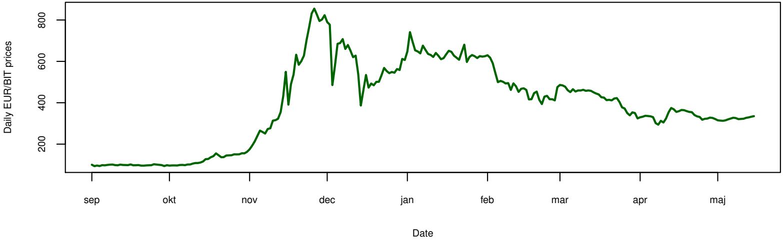
Results

Linear computational cost.
Increased efficiency (by a factor of 3).
Lower computational cost.
Simplified tuning by the Hessian information.

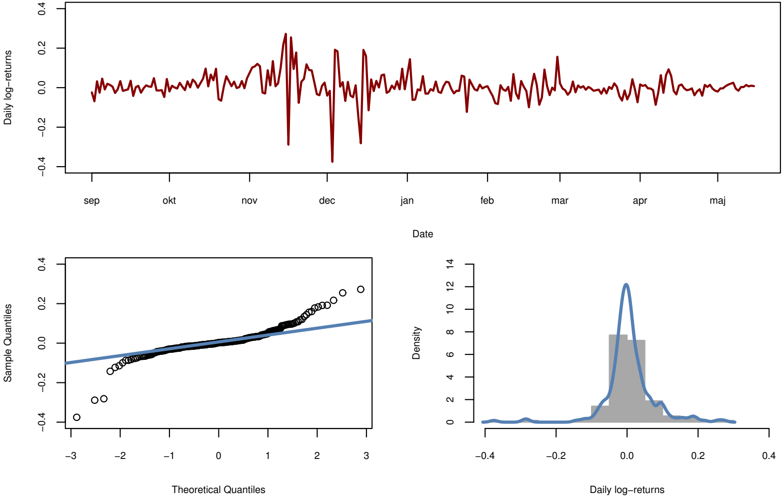
Overview of the thesis



Modelling volatility in Bitcoin returns



Modelling volatility in Bitcoin returns



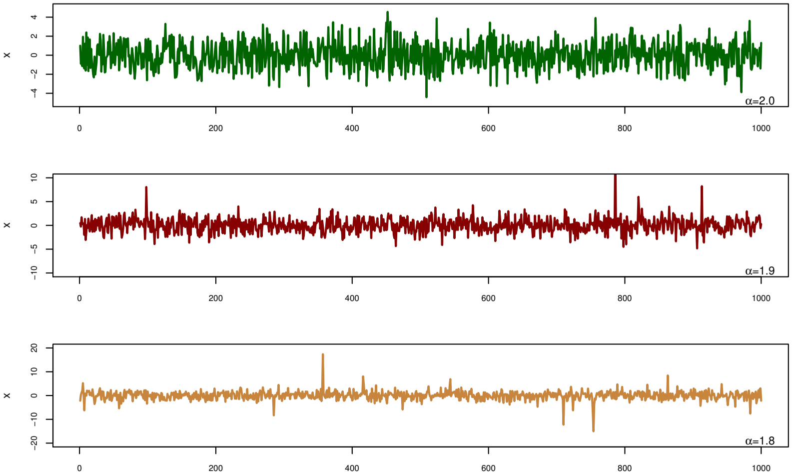
α-stable distributions

$$x \sim \mathcal{A}(x; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\eta}),$$

where the parameters describe:

- $\boldsymbol{\alpha} \in [0, 2]$: stability.
- $\boldsymbol{\beta} \in [-1, 1]$: skewness.
- $\boldsymbol{\gamma} \in \mathbb{R}_+$: scale (spread).
- $\boldsymbol{\eta} \in \mathbb{R}$: location.

α-stable distributions



Stochastic volatility with α-stable returns

$$x_{t+1} | x_t \sim \mathcal{N}\Big(x_{t+1}; \phi x_t, \boldsymbol{\sigma}^2\Big),$$
$$y_t | x_t \sim \mathcal{A}\Big(y_t; \boldsymbol{\alpha}, 0, \exp(x_t), 0\Big).$$

where the parameters describe:

ϕ: persistence of volatility.

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

α: stability.

Task: Estimate $x_{0:T}$ given $y_{1:T}$ (requires $\boldsymbol{\theta} = \{\phi, \boldsymbol{\sigma}, \alpha\}$).

Approximate Bayesian computations

Problem

We cannot evaluate $g_\theta(y_t|x_t)$.

Idea

Data generated from $s_t \sim g_\theta(\cdot|x_t)$ should be similar to y_t.

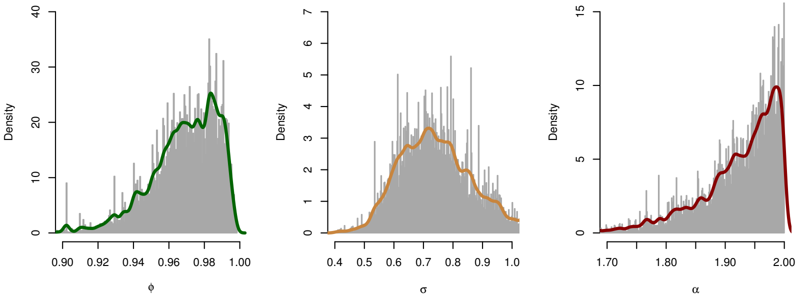
Implementation

Replace the $g_\theta(y_t|x_t)$ with $\mathcal{K}_\epsilon(s_t, y_t)$.

Gives unbiased estimate of the likelihood

We can make use of particle filtering and the particle Metropolis-Hastings.

Modelling volatility in Bitcoin returns



	Bitcoin			OMXS30		
	ϕ	σ	α	ϕ	σ	α
Posterior mean	0.97	**0.75**	1.92	0.96	**0.31**	1.93
Posterior median	0.97	**0.74**	1.94	0.96	**0.30**	1.94
Posterior mode	0.98	**0.72**	1.99	0.97	**0.22**	1.94

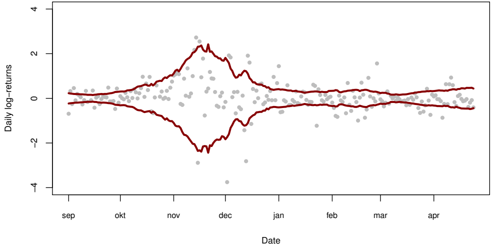
Modelling volatility in Bitcoin returns

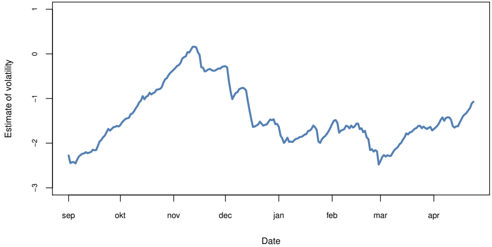
$$x_{t+1} | x_t \sim \mathcal{N}\Big(x_{t+1}; \phi x_t, \sigma^2\Big),$$

$$y_t | x_t \sim \mathcal{A}\Big(y_t; \alpha, 0, \exp(x_t), 0\Big),$$

with parameters:

$\phi = 0.965$ (persistence.)
$\sigma = 0.740$ (sd. of innovation.)
$\alpha = 1.925$ (stability.)





Summary of Paper C

Contribution

Novel parameter inference method for models with intractable likelihoods. (Paper C)

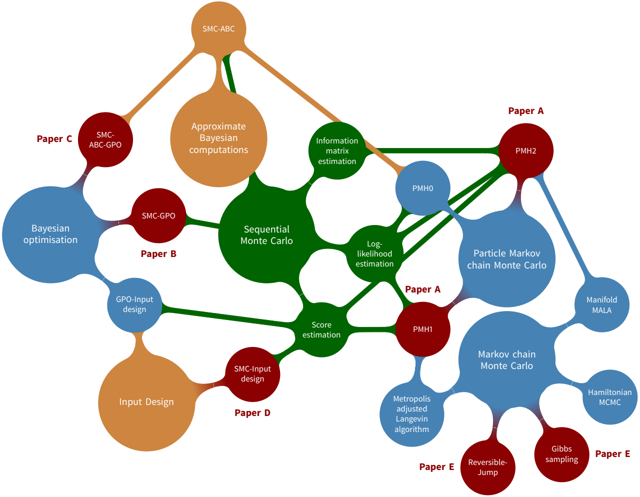
Methods

Bayesian optimisation with sequential Monte Carlo. (Paper B)
Sequential Monte Carlo with approximate Bayesian computations.

Results

Computational time 1.5 hours vs. 240 hours (parallel PMH).

Overview of the thesis



Conclusions

Particle Metropolis-Hastings

Metropolis-Hastings with unbiased estimator of the likelihood.

Sequential Monte Carlo

Estimation of log-likelihood, gradients and Hessians.

Approximate Bayesian computations

For inference in models with intractable likelihoods.

Applications

Dynamic stochastic general equilibrium, earthquake counts and Bitcoin volatility.

Future work

Particle Metropolis-Hastings and Hamiltonian Monte Carlo

Theory, parallel implementations, online methods and applications.

Bayesian optimisation

Method development, online methods and applications.

Approximate Bayesian computations

Combine with particle Metropolis-Hastings and Bayesian optimisation for applications.

Thank you for your attention!

Questions, comments and suggestions are most welcome.

The thesis and code to replicate the results within it are found at:

`http://users.isy.liu.se/en/rt/johda87/`.

Particle Metropolis-Hastings algorithm

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

$$\mathbb{E}_m\left[\widehat{p}_\theta(y_{1:T}|u)\right] = \int \widehat{p}_\theta(y_{1:T}|u)m_\theta(u)\,\mathrm{d}u = p_\theta(y_{1:T}).$$

An *extended target* is given by

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

Particle Metropolis-Hastings algorithm (cont.)

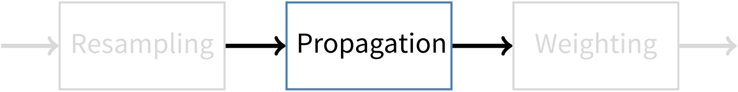
$$\begin{aligned}\int \pi(\theta, u)\,\mathrm{d}u &= \int \frac{\widehat{p}_\theta(y_{1:T}|u)m_\theta(u)\pi(\theta)}{p_\theta(y_{1:T})}\,\mathrm{d}u \\ &= \frac{\pi(\theta)}{p_\theta(y_{1:T})}\underbrace{\int \widehat{p}_\theta(y_{1:T}|u)m_\theta(u)\,\mathrm{d}u}_{=p_\theta(y_{1:T})}, \\ &= \pi(\theta.\end{aligned}$$

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

(bootstrap) Particle filtering

- Resampling: $\mathbb{P}(a_t^{(i)} = j) = \widetilde{w}_{t-1}^{(j)}$ and set $\widetilde{x}_{t-1}^{(i)} = x_{t-1}^{a_t^{(i)}}$.
- Propagation: $x_t^{(i)} \sim R_\theta\left(x_t | \widetilde{x}_{t-1}^{(i)}\right) = f_\theta(x_t | \widetilde{x}_{t-1}^{(i)})$.
- Weighting: $w_t^{(i)} = W_\theta\left(x_t^{(i)}, \widetilde{x}_{t-1}^{(i)}\right) = g_\theta(y_t | x_t)$.

(bootstrap) Particle filtering



- Resampling: $\mathbb{P}(a_t^{(i)} = j) = \widetilde{w}_{t-1}^{(j)}$ and set $\widetilde{x}_{t-1}^{(i)} = x_{t-1}^{a_t^{(i)}}$.
- Propagation: $x_t^{(i)} \sim R_\theta\left(x_t | \widetilde{x}_{t-1}^{(i)}\right) = f_\theta(x_t | \widetilde{x}_{t-1}^{(i)})$.
- Weighting: $w_t^{(i)} = W_\theta\left(x_t^{(i)}, \widetilde{x}_{t-1}^{(i)}\right) = g_\theta(y_t | x_t)$.

(bootstrap) Particle filtering

$\rightarrow$ Resampling $\rightarrow$ Propagation $\rightarrow$ **Weighting** $\rightarrow$

- Resampling: $\mathbb{P}(a_t^{(i)} = j) = \widetilde{w}_{t-1}^{(j)}$ and set $\widetilde{x}_{t-1}^{(i)} = x_{t-1}^{a_t^{(i)}}$.
- Propagation: $x_t^{(i)} \sim R_\theta\left(x_t | \widetilde{x}_{t-1}^{(i)}\right) = f_\theta(x_t | \widetilde{x}_{t-1}^{(i)})$.
- Weighting: $w_t^{(i)} = W_\theta\left(x_t^{(i)}, \widetilde{x}_{t-1}^{(i)}\right) = g_\theta(y_t | x_t)$.

Likelihood estimation using the APF

The likelihood for an SSM can be decomposed by

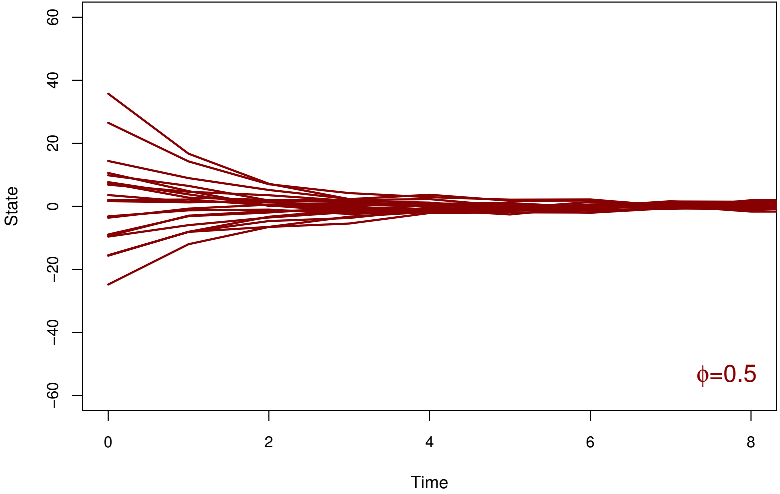
$$\mathcal{L}(\theta) = p_\theta(y_{1:T}) = p_\theta(y_1) \prod_{t=2}^{T} p_\theta(y_t | y_{1:t-1}),$$

where the *one-step ahead predictor* can be computed by

$$\begin{aligned} p_\theta(y_t|y_{1:t-1}) &= \int f_\theta(x_t|x_{t-1}) g_\theta(y_t|x_t) p_\theta(x_{t-1}|y_{1:t-1}) \, \mathrm{d}x_t \\ &= \int W_\theta(x_t|x_{t-1}) R_\theta(x_t|x_{t-1}) p_\theta(x_{t-1}|y_{1:t-1}) \, \mathrm{d}x_t. \end{aligned}$$

$$p_\theta(y_t|y_{1:t-1}) \approx \frac{1}{N} \sum_{i=1}^{N} \int W_\theta(x_t|x_{t-1}) \delta_{x_t^{(i)}, \widetilde{x}_{t-1}^{(i)}} \, \mathrm{d}x_t = \frac{1}{N} \sum_{i=1}^{N} w_t^{(i)}.$$

Fixed-lag particle smoothing



Fixed-lag particle smoothing (cont.)

Assume that

$$p_\theta(x_t|y_{1:T}) \approx p_\theta(x_t|y_{1:\kappa_t}), \qquad \kappa_t = \min\{T, t+\Delta\},$$

for some $0 \leq \Delta \leq T$. It follows that

$$\widehat{p}_\theta(x_{t-1:t}|y_{1:T}) = \sum_{i=1}^{N} \widetilde{w}_{\kappa_t}^{(i)} \delta_{\widetilde{x}_{t-1:t,\kappa_t}^{(i)}}(\mathrm{d}x_{t-1:t})$$

which can be used to estimate the gradient and Hessian information about the log-target.

Score estimation using the FL smoother

The score can be estimated using *Fisher's identity* given by

$$\begin{aligned}\nabla_\theta \log p_\theta(y_{1:T})\big|_{\theta=\theta'} &= \int \nabla_\theta \log p_\theta(x_{1:T}, y_{1:T}) p_{\theta'}(x_{1:T} | y_{1:T}) \mathrm{d}x_{1:T} \\ &\approx \int \nabla_\theta \log p_\theta(x_{1:T}, y_{1:T}) \widehat{p}_{\theta'}(x_{1:T} | y_{1:T}) \mathrm{d}x_{1:T}\end{aligned}$$

We also know that

$$\nabla_\theta \log p_\theta(x_{1:T}, y_{1:T}) = \sum_{t=1}^{T} \underbrace{\Big[\nabla_\theta \log f_\theta(x_t | x_{t-1}) + \nabla_\theta \log g_\theta(y_t | x_t)\Big]}_{\triangleq \eta(x_t, x_{t-1})},$$

which gives

$$\nabla_\theta \log p_\theta(y_{1:T})\big|_{\theta=\theta'} \approx \sum_{t=1}^{T} \sum_{i=1}^{N} \widetilde{w}_{\kappa_t}^{(i)} \eta(\widetilde{x}_{t-1,\kappa_t}^{(i)}, \widetilde{x}_{t,\kappa_t}^{(i)}).$$

Mixing

Let $\varphi(\theta)$ denote a *test function*, then

$$\sqrt{M}\left[\widehat{\varphi}_{\mathsf{MH}} - \mathbb{E}[\varphi(\theta)]\right] \xrightarrow{d} \mathcal{N}(0, \sigma^2_\varphi).$$

Here, σ^2_φ depends on the *integrated autocorrelation time* (IACT)

$$\mathsf{IACT}(\theta_{1:M}) = 1 + 2\sum_{k=1}^{\infty} \rho_k(\theta_{1:M}).$$