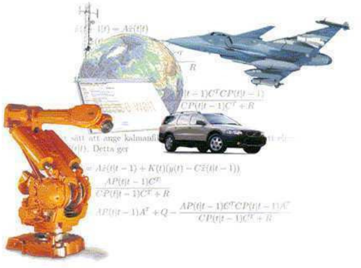
Particle filter-based Gaussian process optimisation for parameter inference

IFAC World Congress 2014, Cape Town, South Africa, August 28, 2014.



Johan Dahlin
johan.dahlin@liu.se

Division of Automatic Control,
Linköping University,
Sweden.

This is collaborative work with

Dr. Fredrik Lindsten (University of Cambridge, United Kingdom)

Summary

Aim

Efficient likelihood parameter inference in nonlinear SSMs.

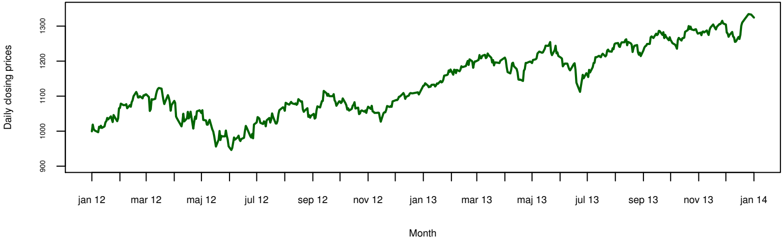
Methods

Gaussian process optimisation.
Sequential Monte Carlo methods.

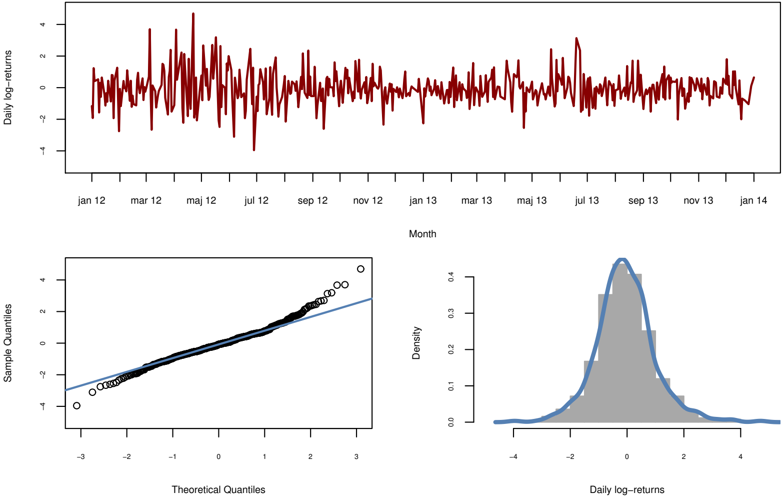
Contributions

Decreased computational cost compared with popular methods.
Interesting method for solving other costly optimisation problems.

Example: Modelling volatility in OMXS30 returns



Example: Modelling volatility in OMXS30 returns



Example: Modelling volatility in OMXS30 returns

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

Task: Estimate $\widehat{\theta}_{\mathsf{ML}} \triangleq \underset{\theta \in \Theta}{\operatorname{argmax}}\, \ell(\theta)$, with $\theta = \{\phi, \boldsymbol{\sigma}, \boldsymbol{\beta}\}$.

Overview of the algorithm

(i) Given iterate θ_k, estimate the log-likelihood $\widehat{\ell}_k \approx \ell(\theta_k)$.

(ii) Given $\{\theta_j, \widehat{\ell}_j\}_{j=0}^{k}$, create a surrogate cost function of $\ell(\theta)$.

(iii) Select a new θ_{k+1} using the surrogate cost function.

Overview of the algorithm

(i) Given iterate θ_k, estimate the log-likelihood $\widehat{\ell}_k \approx \ell(\theta_k)$.

Estimated using a particle filter.

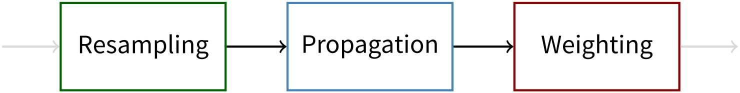
(ii) Given $\{\theta_j, \widehat{\ell}_j\}_{j=0}^{k}$, create a surrogate cost function of $\ell(\theta)$.

The predictive distribution of a Gaussian process.

(iii) Select a new θ_{k+1} using the surrogate cost function.

Acquisition function, a heuristic based on the predictive distribution.

Particle filtering: overview



Given the particle system

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

the filtering density can be approximated by

$$\widehat{p}_\theta(\mathrm{d}x_{1:T} \mid y_{1:T}) = \sum_{i=1}^{N} \left[\frac{w_T^{(i)}}{\sum_{k=1}^{N} w_T^{(k)}} \right] \delta_{x_{1:T}^{(i)}}(\mathrm{d}x_{1:T}).$$

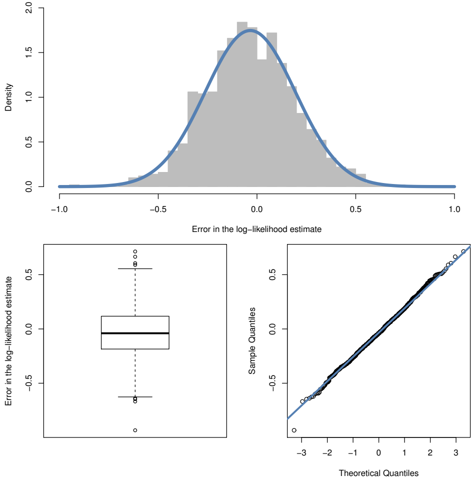
Particle filtering: log-likelihood estimator

Estimator:

$$\widehat{\ell}(\theta) = \sum_{t=1}^{T} \log \left(\sum_{i=1}^{N} w_t^{(i)} \right) - T \log N.$$

Statistical properties (CLT):

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



Gaussian process regression: overview

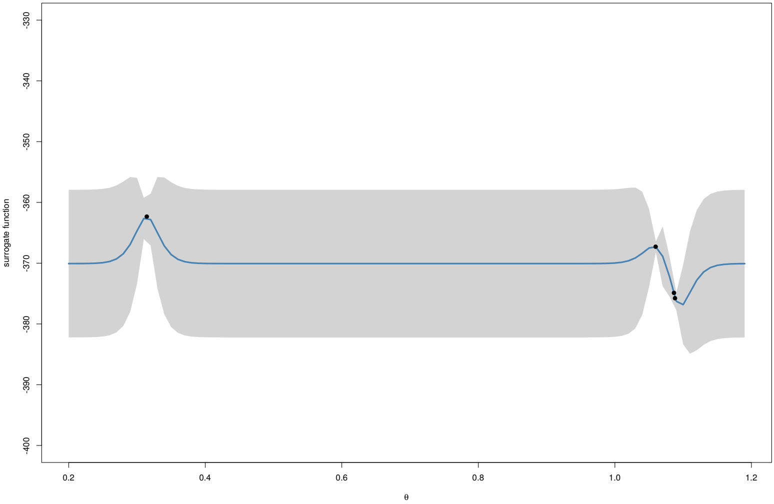
We assume a priori that

$$\ell(\theta) \sim \mathcal{GP}\Big(m(\theta), \kappa(\theta, \theta')\Big),$$

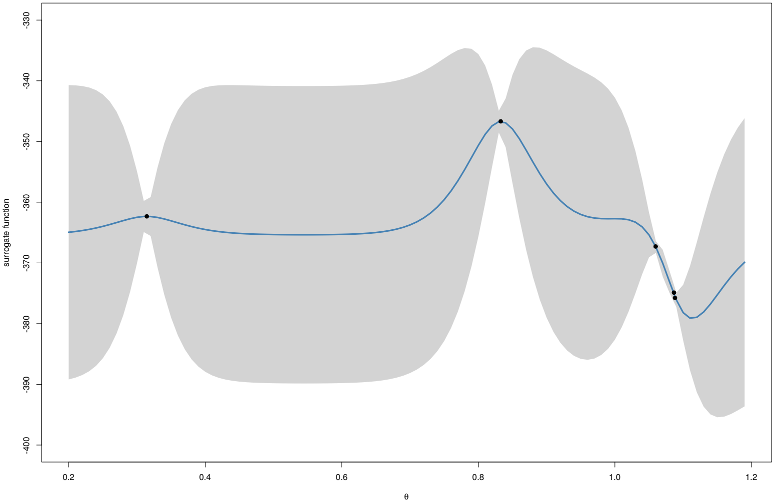
which gives the posterior predictive distribution

$$\ell(\theta_\star)|\mathcal{D}_k \sim \mathcal{N}\Big(\mu(\theta_\star|\mathcal{D}_k), \sigma^2(\theta_\star|\mathcal{D}_k) + \sigma_l^2\Big), \text{ with } \mathcal{D}_k = \{\theta_j, \widehat{\ell}_i\}_{j=1}^{k}.$$

Gaussian process regression: toy example



Gaussian process regression: toy example



Acquisition rule for selecting sampling points

Consider, the 95% upper confidence bound as the acquisition rule

$$\theta_{k+1} = \operatorname*{argmax}_{\theta_\star \in \Theta} \left[\mu(\theta_\star | \mathcal{D}_k) + 1.96 \sqrt{\sigma^2(\theta_\star | \mathcal{D}_k)} \right],$$

to determine the next iterate θ_{k+1}.

Overview of the algorithm

(i) Given iterate θ_k, estimate the log-likelihood $\widehat{\ell}_k \approx \ell(\theta_k)$.

Estimated using a particle filter.

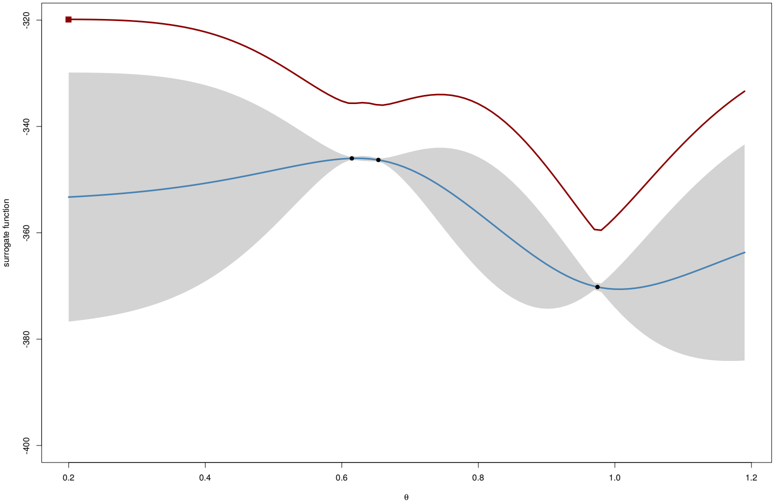
(ii) Given $\{\theta_j, \widehat{\ell}_j\}_{j=0}^{k}$, create a surrogate cost function of $\ell(\theta)$.

The predictive distribution of a Gaussian process.

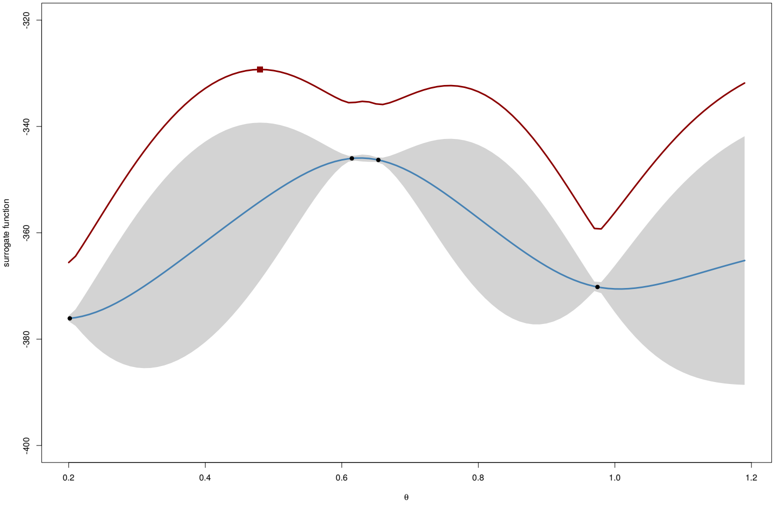
(iii) Select a new θ_{k+1} using the surrogate cost function.

Acquisition function, a heuristic based on the predictive distribution.

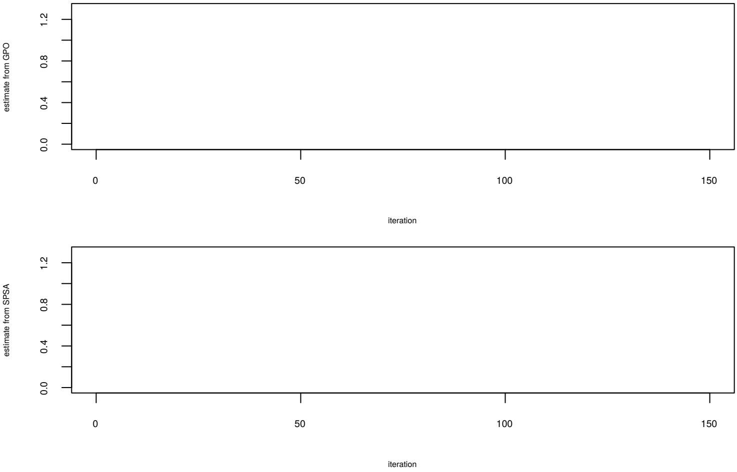
A toy example of the algorithm in action



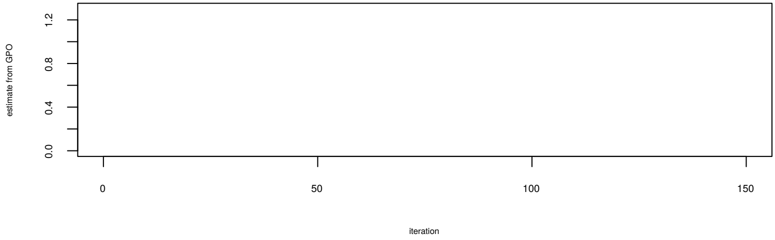
A toy example of the algorithm in action

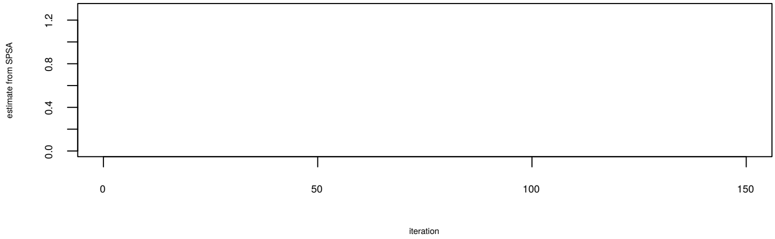


Example: Modelling volatility in OMXS30 returns

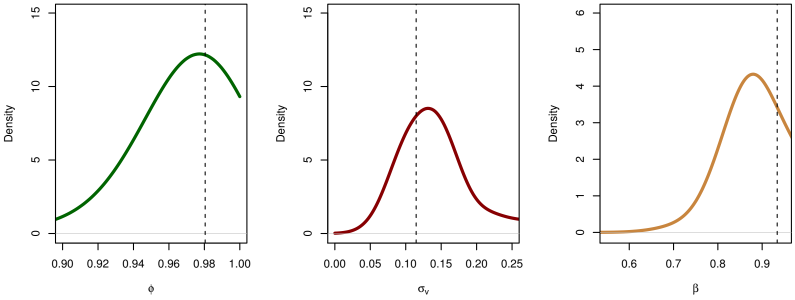


Example: Modelling volatility in OMXS30 returns



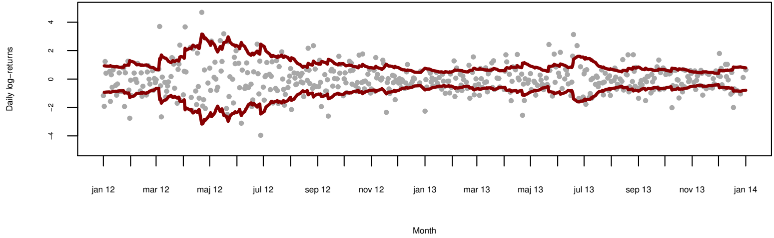


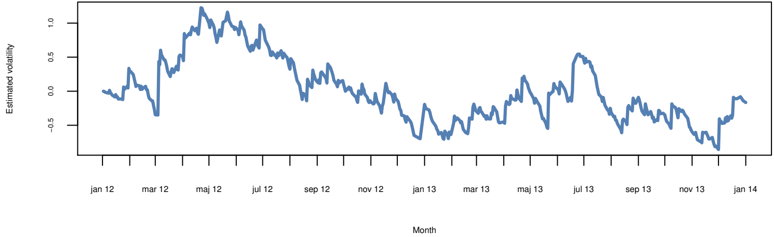
Example: Modelling volatility in OMXS30 returns



Estimator	$\widehat{\phi}$	$\widehat{\sigma}$	$\widehat{\beta}$
Maximum likelihood (GPO)	0.98	0.11	0.93
Bayesian posterior mode	0.98	0.12	0.88
Bayesian posterior mean	0.97	0.14	0.93

Example: Modelling volatility in OMXS30 returns





Conclusions

Methods

Particle filtering for log-likelihood estimation.
CLT for the log-likelihood and Gaussian process modelling.
Acquisition rules.

Contributions

Decreased computational cost compared with popular methods.
Only makes use of *cheap* zero-order information.

Future work

Bias compensation of log-likelihood estimate.
Approximate Bayesian computations. (New paper!)
Input design. (New paper!)

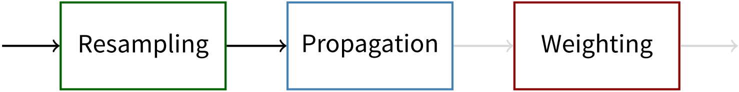
Thank you for your attention!

Questions, comments and suggestions are most welcome.

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

`http://work.johandahlin.com/`.

(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: $w_t^{(i)} = W_\theta\left(x_t^{(i)}, \widetilde{x}_{t-1}^{(i)}\right) = g_\theta(y_t | x_t)$.

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

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

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