Parameter estimation in non-linear SSMs using Newton optimisation

Johan Dahlin, Manon Kok, Thomas B. Schön and Adrian Wills

Main ideas and contribution

- The likelihood and its gradient and Hessian are intractable for a general SSM.
- Investigate two approximations to estimate the these quantities based on linearisation and sampling.
- Promising results for two synthetic test problems.

Maximum likelihood inference

The parameter estimate for a state space model (SSM),

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

can be computed by

$$\hat{\theta} = \operatorname*{argmax}_{\theta \in \Theta} \ell(\theta),$$

where $\ell(\theta) = \log p(y_{1:T}|\theta)$ denotes the log-likelihood. This can be implemented by the Newton update given by

$$\theta_k = \theta_{k-1} - \varepsilon_k \mathcal{H}^{-1}\left(\theta_{k-1}\right) \mathcal{G}\left(\theta_{k-1}\right), \qquad (1)$$

at iteration k and where $\mathcal{G}(\theta)$ and $\mathcal{H}(\theta)$ denote the gradient and Hessian of the log-likelihood, respectively.

Computing gradients and Hessians

We can compute $\mathcal{G}(\theta)$ using the **Fisher identity**

$$\mathcal{G}(\theta_k) = \sum_{t=1}^{T} \underbrace{\int \xi_\theta(x_{t+1:t}) p_\theta(x_{t+1:t}|y_{1:T})\, \mathrm{d}x_{t+1:t}}_{\triangleq \mathcal{G}_t(\theta)},$$

$$\xi_\theta(x_{t+1:t}) \triangleq \nabla \log f_\theta(x_{t+1}|x_t) + \nabla \log g_\theta(y_t|x_t).$$

We can estimate $\mathcal{H}(\theta)$ by the **Segal-Weinstein estimator**

$$\widehat{\mathcal{H}}(\theta) = \frac{1}{T}\Big[\mathcal{G}(\theta)\Big]\Big[\mathcal{G}(\theta)\Big]^\top - \sum_{t=1}^{T}\Big[\mathcal{G}_t(\theta)\Big]\Big[\mathcal{G}_t(\theta)\Big]^\top,$$

using the gradient for each time step.

The intractability of the two-step smoothing distribution $p_\theta(x_{t+1:t}|y_{1:T})$ is handled by two different kinds of approximations.

Algorithm 1: Newton method for ML parameter estimation

INPUTS: Initial parameter θ_0, maximum no. iterations K.
OUTPUTS: ML parameter estimate $\hat{\theta}$.

1: Set $k = 0$
2: **while** *exit condition is not satisfied* **do**
  a: Run an algorithm to estimate $\widehat{\ell}(\theta_k)$, $\widehat{\mathcal{G}}(\theta_k)$ and $\widehat{\mathcal{H}}(\theta_k)$.
  b: Determine ε_k using line search or a stochastic schedule.
  c: Apply the Newton update (1) to obtain θ_{k+1}.
  d: Set $k = k+1$.
  end while
3: Set $\hat{\theta}_{\text{ML}} = \theta_k$.

Linearisation approximation

We approximate $p_\theta(x_{t+1:t}|y_{1:T})$ by

$$p_\theta(x_{t+1:t}|y_{1:T}) = \mathcal{N}\left(x_{t+1:t}; \hat{x}_{t+1:t|T}, P_{t+1:t|T}\right),$$

where the smoothed state estimate is given by

$$\hat{x}_{t+1:t|T} = \operatorname*{argmax}_{x_{t+1:t}} \log p_\theta(x_{1:T}, y_{1:T}),$$

using a standard **Gauss-Newton solver** which also computes $P_{t+1:t|T}$. The complete data log-likelihood,

$$\log p_\theta(x_{1:T}, y_{1:T}) = \log p_\theta(x_1) + \sum_{t=1}^{T-1} \log f_\theta(x_t|x_{t-1}) + \sum_{t=1}^{T} \log g_\theta(y_t|x_t)$$

is approximated using an **extended Kalman filter**.

Sampling approximation

We approximate $p_\theta(x_{t+1:t}|y_{1:T})$ by the empirical distribution

$$p_\theta(x_{t+1:t}|y_{1:T}) = \sum_{i=1}^{N} w_{t+1:t}^{(i)} \delta_{x_{t+1:t}^{(i)}}(\mathrm{d}x_{t+1:t}),$$

where $x_{t+1:t}^{(i)}$ and $w_{t+1:t}^{(i)}$ denote the particles and their weights generated by the a **particle smoother**, respectively.

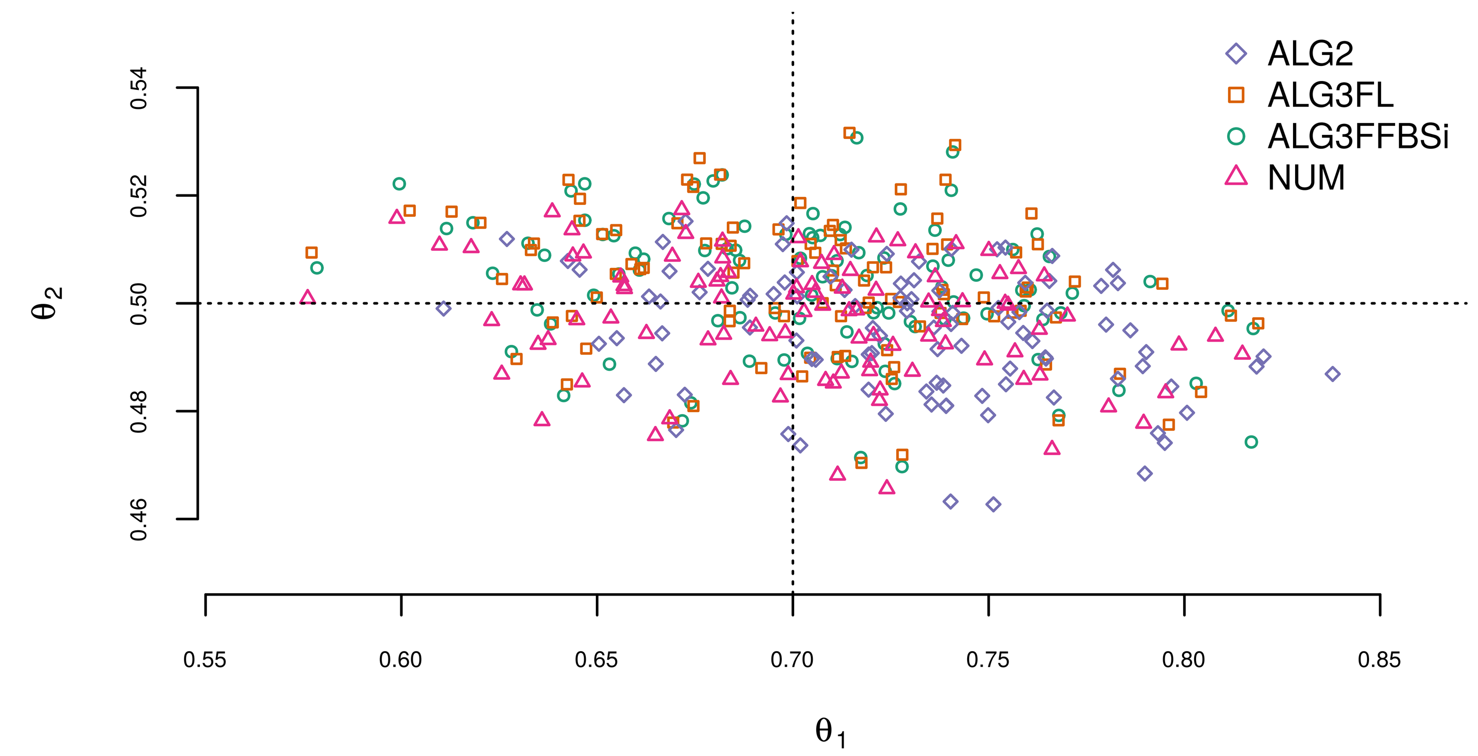
We employ and compare two different particle smoothers: (i) the fast fixed-lag (FL) and (ii) the more accurate forward-filtering backward-simulator (FFBSi).

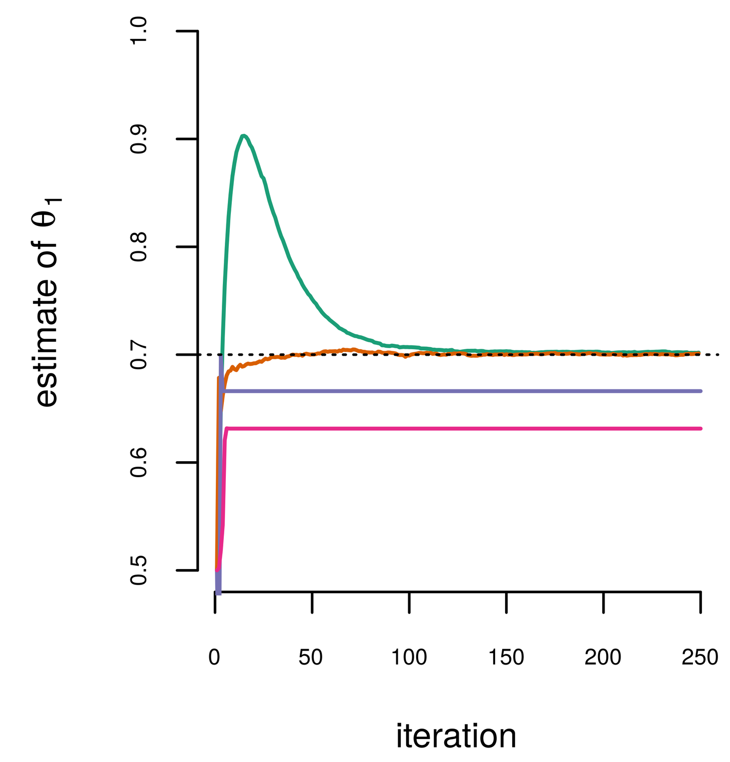
Numerical illustration

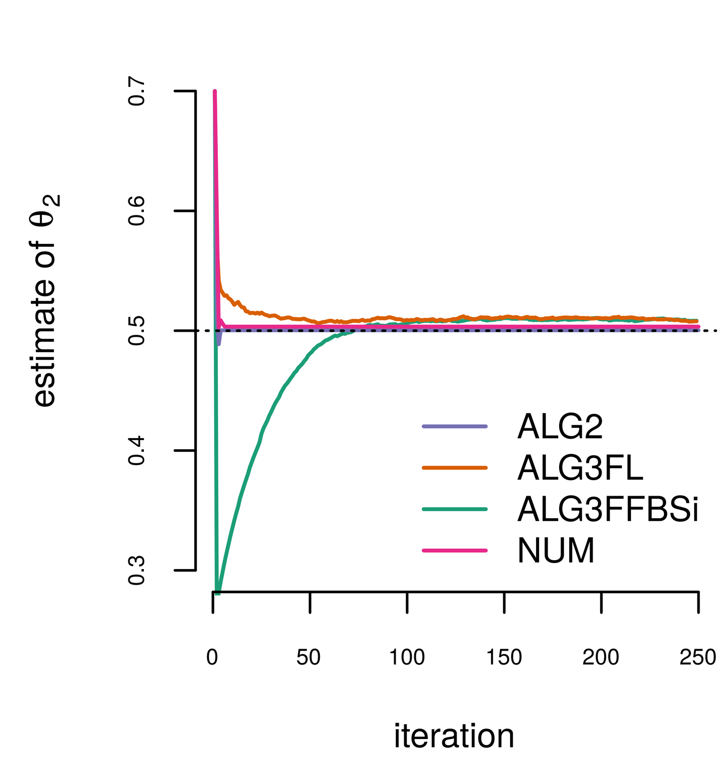
We simulate $T = 1{,}000$ observations with $\theta = \{\theta_1, \theta_2\} = \{0.7, 0.5\}$ from the non-linear SSM given by

$$x_t | x_{t-1} \sim \mathcal{N}\Big(x_t; \theta_1 \arctan(x_t), 1\Big), \quad y_t | x_t \sim \mathcal{N}\Big(y_t; \theta_2 x_t, 0.1^2\Big)$$

and estimate θ using Algorithm 1.







Algorithm	Bias ($\cdot 10^{-4}$) θ_1	Bias θ_2	MSE ($\cdot 10^{-4}$) θ_1	MSE θ_2	Run time (sec/iter)
Linearisation (ALG2)	284	-55	28	2	2.73
FL sampling (ALG3FL)	53	35	24	2	5
FFBSi sampling (ALG3FFBSi)	55	**31**	24	2	22
Numerical differentiation (NUM)	**31**	-27	**23**	**1**	0.19

Paper and source code

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

LINKÖPING UNIVERSITY
UPPSALA UNIVERSITET
THE UNIVERSITY OF NEWCASTLE AUSTRALIA
LINKÖPING UNIVERSITY
Department of Electrical Engineering