Newton-based maximum likelihood estimation in nonlinear SSMs

SysId conference 2015, Beijing

M. Kok[1], J. Dahlin[1], T. B. Schön[2] and A. Wills[3]

[1]Linköping University, Sweden
[2]Uppsala University, Sweden
[3]University of Newcastle, Australia

LINKÖPING UNIVERSITY

Introduction: nonlinear state space models

Nonlinear state space models (SSMs)

$$
\begin{aligned}
x_{t+1} &= f_\theta(x_t, v_t), \\
y_t &= g_\theta(x_t, e_t),
\end{aligned}
$$

Introduction: nonlinear state space models

Nonlinear state space models (SSMs)

$$x_{t+1} = f_\theta(x_t, v_t),$$
$$y_t = g_\theta(x_t, e_t),$$

x_t: latent state variables at time t

Introduction: nonlinear state space models

Nonlinear state space models (SSMs)

$$
\begin{aligned}
x_{t+1} &= f_\theta(x_t, v_t),\\
y_t &= g_\theta(x_t, e_t),
\end{aligned}
$$

x_t: latent state variables at time t

y_t: measurements at time t

Introduction: nonlinear state space models

Nonlinear state space models (SSMs)

$$
\begin{aligned}
x_{t+1} &= f_\theta(x_t, v_t), \\
y_t &= g_\theta(x_t, e_t),
\end{aligned}
$$

x_t: latent state variables at time t

y_t: measurements at time t

$f_\theta(\cdot), g_\theta(\cdot)$: (nonlinear) dynamic and measurement model

Introduction: nonlinear state space models

Nonlinear state space models (SSMs)

$$x_{t+1} = f_{\theta}(x_t, v_t),$$
$$y_t = g_{\theta}(x_t, e_t),$$

x_t: latent state variables at time t
y_t: measurements at time t
$f_\theta(\cdot), g_\theta(\cdot)$: (nonlinear) dynamic and measurement model
θ: parameters

Introduction: nonlinear state space models

Nonlinear state space models (SSMs)

$$x_{t+1} = f_\theta(x_t, v_t),$$
$$y_t = g_\theta(x_t, e_t),$$

x_t: latent state variables at time t
y_t: measurements at time t
$f_\theta(\cdot), g_\theta(\cdot)$: (nonlinear) dynamic and measurement model
θ: parameters
v_t, e_t: process and measurement noise

Introduction: ML estimation

Maximum likelihood (ML) estimates of parameters in nonlinear SSMs

$$\widehat{\theta}_{\mathsf{ML}} = \arg\max_{\theta} \ \ell_\theta(y_{1:N})$$

Introduction: ML estimation

Maximum likelihood (ML) estimates of parameters in nonlinear SSMs

$$\widehat{\theta}_{\mathsf{ML}} = \arg\max_{\theta} \ell_{\theta}(y_{1:N})$$

$\widehat{\theta}_{\mathsf{ML}}$: ML estimate of θ

Introduction: ML estimation

Maximum likelihood (ML) estimates of parameters in nonlinear SSMs

$$\widehat{\theta}_{\mathsf{ML}} = \arg\max_{\theta} \ell_\theta(y_{1:N})$$

$\widehat{\theta}_{\mathsf{ML}}$: ML estimate of θ
$\ell_\theta(y_{1:N})$: log-likelihood of the measurements $y_{1:N}$

Introduction: Newton methods

Estimating θ using Newton methods. Iteratively update:

$$\theta_{k+1} = \theta_k - \varepsilon_k \Big[\mathcal{H}(\theta_k)\Big]^{-1} \Big[\mathcal{G}(\theta_k)\Big],$$

Introduction: Newton methods

Estimating θ using Newton methods. Iteratively update:

$$\theta_{k+1} = \theta_k - \varepsilon_k \Big[\mathcal{H}(\theta_k)\Big]^{-1} \Big[\mathcal{G}(\theta_k)\Big],$$

$\mathcal{G}(\theta_k)$: gradient of $\ell_\theta(y_{1:N})$

Introduction: Newton methods

Estimating θ using Newton methods. Iteratively update:

$$\theta_{k+1} = \theta_k - \varepsilon_k \Big[\mathcal{H}(\theta_k)\Big]^{-1} \Big[\mathcal{G}(\theta_k)\Big],$$

$\mathcal{G}(\theta_k)$: gradient of $\ell_\theta(y_{1:N})$
$\mathcal{H}(\theta_k)$: Hessian of $\ell_\theta(y_{1:N})$

Introduction: Newton methods

Estimating θ using Newton methods. Iteratively update:

$$\theta_{k+1} = \theta_k - \varepsilon_k \Big[\mathcal{H}(\theta_k)\Big]^{-1} \Big[\mathcal{G}(\theta_k)\Big],$$

$\mathcal{G}(\theta_k)$: gradient of $\ell_\theta(y_{1:N})$
$\mathcal{H}(\theta_k)$: Hessian of $\ell_\theta(y_{1:N})$
ε_k: Step size at iteration k

Problem description

- In nonlinear SSMs, the log-likelihood $\ell_\theta(y_{1:N})$, the gradient $\mathcal{G}(\theta_k)$ and the Hessian $\mathcal{H}(\theta_k)$ can not be computed analytically.

Problem description

- In nonlinear SSMs, the log-likelihood $\ell_\theta(y_{1:N})$, the gradient $\mathcal{G}(\theta_k)$ and the Hessian $\mathcal{H}(\theta_k)$ can not be computed analytically.

$\Rightarrow$ Approximations are needed.

Contributions

- Use Fisher's identity to estimate the gradient and Hessian / Extend the approach from Segal and Weinstein (1989)[1] to nonlinear SSMs.
- This allows us to do Newton optimization in nonlinear SSMs.

[1] Segal, M. and Weinstein, E. (1989). A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems. IEEE Transactions on Information Theory, 35(3), $682 - 687$.

Contributions

- Use Fisher's identity to estimate the gradient and Hessian / Extend the approach from Segal and Weinstein (1989)[1] to nonlinear SSMs.
- This allows us to do Newton optimization in nonlinear SSMs.
- It requires approximation of the log-likelihood and of the smoothing problem.

[1] Segal, M. and Weinstein, E. (1989). A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems. IEEE Transactions on Information Theory, 35(3), $682 - 687$.

Contributions

- Use Fisher's identity to estimate the gradient and Hessian / Extend the approach from Segal and Weinstein (1989)[1] to nonlinear SSMs.
- This allows us to do Newton optimization in nonlinear SSMs.
- It requires approximation of the log-likelihood and of the smoothing problem.
  - We explore two approximation methods: based on *linearization* and based on *sampling*.

[1] Segal, M. and Weinstein, E. (1989). A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems. IEEE Transactions on Information Theory, 35(3), $682 - 687$.

Contributions

- Use Fisher's identity to estimate the gradient and Hessian / Extend the approach from Segal and Weinstein (1989)[1] to nonlinear SSMs.
- This allows us to do Newton optimization in nonlinear SSMs.
- It requires approximation of the log-likelihood and of the smoothing problem.
    - We explore two approximation methods: based on *linearization* and based on *sampling*.
- Show the workings of the methods on simulated data from two simple nonlinear SSMs.

[1] Segal, M. and Weinstein, E. (1989). A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems. IEEE Transactions on Information Theory, 35(3), $682 - 687$.

Computing $\mathcal{G}(\theta_k)$ and $\mathcal{H}(\theta_k)$

Fisher's identity:

$$\begin{aligned}\mathcal{G}(\theta_k) &= \tfrac{\partial}{\partial\theta}\ell_\theta(y_{1:N})\Big|_{\theta=\theta_k} \\ &= \int \frac{\partial}{\partial\theta}\log p_\theta(x_{1:N}, y_{1:N})\Big|_{\theta=\theta_k} p_\theta(x_{1:N}|y_{1:N})\,\mathrm{d}x_{1:N}\end{aligned}$$

Computing $\mathcal{G}(\theta_k)$ and $\mathcal{H}(\theta_k)$

Fisher's identity:

$$
\begin{aligned}
\mathcal{G}(\theta_k) &= \frac{\partial}{\partial\theta}\ell_\theta(y_{1:N})\Big|_{\theta=\theta_k} \\
&= \int \frac{\partial}{\partial\theta} \log p_\theta(x_{1:N}, y_{1:N})\Big|_{\theta=\theta_k} p_\theta(x_{1:N}|y_{1:N})\, \mathrm{d}x_{1:N}
\end{aligned}
$$

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

Computing $\mathcal{G}(\theta_k)$ and $\mathcal{H}(\theta_k)$

Fisher's identity:

$$\begin{aligned}\mathcal{G}(\theta_k) &= \frac{\partial}{\partial\theta}\ell_\theta(y_{1:N})\Big|_{\theta=\theta_k} \\ &= \sum_{t=1}^{N}\int \xi_{\theta_k}(x_{t+1:t})p_{\theta_k}(x_{t+1:t}|y_{1:N})\,\mathrm{d}x_{t+1:t} \triangleq \sum_{t=1}^{N}\mathcal{G}_t(\theta_k)\end{aligned}$$

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

Computing $\mathcal{G}(\theta_k)$ and $\mathcal{H}(\theta_k)$

Fisher's identity:

$$\begin{aligned}\mathcal{G}(\theta_k) &= \frac{\partial}{\partial\theta}\ell_\theta(y_{1:N})\Big|_{\theta=\theta_k} \\ &= \sum_{t=1}^{N}\int \xi_{\theta_k}(x_{t+1:t})p_{\theta_k}(x_{t+1:t}|y_{1:N})\,\mathrm{d}x_{t+1:t} \triangleq \sum_{t=1}^{N}\mathcal{G}_t(\theta_k)\end{aligned}$$

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

$$\xi_{\theta_k}(x_{t+1:t}) \triangleq \frac{\partial}{\partial\theta}\log f_\theta(x_{t+1}|x_t)\Big|_{\theta=\theta_k} + \frac{\partial}{\partial\theta}\log g_\theta(y_t|x_t)\Big|_{\theta=\theta_k}$$

Computing $\mathcal{G}(\theta_k)$ and $\mathcal{H}(\theta_k)$

Fisher's identity:

$$\begin{aligned}\mathcal{G}(\theta_k) &= \frac{\partial}{\partial\theta}\ell_\theta(y_{1:N})\Big|_{\theta=\theta_k} \\ &= \sum_{t=1}^{N}\int \xi_{\theta_k}(x_{t+1:t})\,{\color{red}p_{\theta_k}(x_{t+1:t}|y_{1:N})}\,\mathrm{d}x_{t+1:t} \triangleq \sum_{t=1}^{N}\mathcal{G}_t(\theta_k)\end{aligned}$$

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

$$\xi_{\theta_k}(x_{t+1:t}) \triangleq \frac{\partial}{\partial\theta}\log f_\theta(x_{t+1}|x_t)\Big|_{\theta=\theta_k} + \frac{\partial}{\partial\theta}\log g_\theta(y_t|x_t)\Big|_{\theta=\theta_k}$$

Estimate the two-step joint smoothing distribution!

Computing $\mathcal{G}(\theta_k)$ and $\mathcal{H}(\theta_k)$

Fisher's identity:

$$
\begin{aligned}
\mathcal{G}(\theta_k) &= \frac{\partial}{\partial\theta}\ell_\theta(y_{1:N})\Big|_{\theta=\theta_k} \\
&= \sum_{t=1}^{N}\int \xi_{\theta_k}(x_{t+1:t})p_{\theta_k}(x_{t+1:t}|y_{1:N})\,\mathrm{d}x_{t+1:t} \triangleq \sum_{t=1}^{N}\mathcal{G}_t(\theta_k)
\end{aligned}
$$

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

Estimate the two-step joint smoothing distribution!

Algorithm

Newton method for ML parameter estimation

1. Set $k = 0$
2. **while** *exit condition is not satisfied* **do**
   1. Run an algorithm to estimate the log-likelihood $\widehat{\ell}(\theta_k)$, its gradient $\widehat{\mathcal{G}}(\theta_k)$ and its Hessian $\widehat{\mathcal{H}}(\theta_k)$.
   2. Determine ε_k.
   3. Apply the Newton update to obtain θ_{k+1}.
   4. Set $k = k + 1$.

   end while
3. Set $\widehat{\theta}_{\mathsf{ML}} = \theta_k$.

Algorithm

Newton method for ML parameter estimation

1. Set $k = 0$
2. **while** *exit condition is not satisfied* **do**
    1. Run an algorithm to estimate the log-likelihood $\widehat{\ell}(\theta_k)$, its gradient $\widehat{\mathcal{G}}(\theta_k)$ and its Hessian $\widehat{\mathcal{H}}(\theta_k)$.
    2. Determine ε_k.
    3. Apply the Newton update to obtain θ_{k+1}.
    4. Set $k = k + 1$.

   end while
3. Set $\widehat{\theta}_{\mathsf{ML}} = \theta_k$.

Linearization approximation

Linearization approximation

- Use extended Kalman filter to estimate $\widehat{\ell}(\theta_k)$.

Linearization approximation

- Use extended Kalman filter to estimate $\widehat{\ell}(\theta_k)$.
- Compute the smoothed state estimates and covariances by solving the optimization problem

$$\widehat{x}_{t|N} = \arg\max_{x_{1:N}} p_{\theta_k}(x_{1:N}, y_{1:N}).$$

using a standard Gauss-Newton solver.

Sampling approximation: particle smoother

Fixed lag smoother	**Forward filter backward simulator**

Sampling approximation: particle smoother

Fixed lag smoother	**Forward filter backward simulator**
Assumes that the SSM forgets about its past within a few time steps	Can be seen as the particle approximation of the Rauch-Tung-Striebel smoother

Sampling approximation: particle smoother

Fixed lag smoother	**Forward filter backward simulator**
Assumes that the SSM forgets about its past within a few time steps	Can be seen as the particle approximation of the Rauch-Tung-Striebel smoother
Fast (for a particle method)	Slow

Sampling approximation: particle smoother

Fixed lag smoother	**Forward filter backward simulator**
Assumes that the SSM forgets about its past within a few time steps	Can be seen as the particle approximation of the Rauch-Tung-Striebel smoother
Fast (for a particle method)	Slow
Less accurate and biased	Accurate

Sampling approximation: particle smoother

Fixed lag smoother	**Forward filter backward simulator**
Assumes that the SSM forgets about its past within a few time steps	Can be seen as the particle approximation of the Rauch-Tung-Striebel smoother
Fast (for a particle method)	Slow
Less accurate and biased	Accurate
More information: Dahlin, J. (2014). Sequential Monte Carlo for inference in nonlinear state space models. Licentiate's thesis no. 1652, Linköping University, Linköping, Sweden.	More information: Lindsten, F. and Schön, T. B. (2013). Backward simulation methods for Monte Carlo statistical inference. In Foundations and Trends in Machine Learning, volume 6, 1–143.

Algorithm

Newton method for ML parameter estimation

1. Set $k = 0$
2. **while** *exit condition is not satisfied* **do**
   1. Run an algorithm to estimate the log-likelihood $\widehat{\ell}(\theta_k)$, its gradient $\widehat{\mathcal{G}}(\theta_k)$ and its Hessian $\widehat{\mathcal{H}}(\theta_k)$.
   2. Determine ε_k.
   3. Apply the Newton update to obtain θ_{k+1}.
   4. Set $k = k + 1$.

   end while
3. Set $\widehat{\theta}_{\mathsf{ML}} = \theta_k$.

Algorithm

Newton method for ML parameter estimation

1. Set $k = 0$
2. **while** *exit condition is not satisfied* **do**
    1. Run an algorithm to estimate the log-likelihood $\widehat{\ell}(\theta_k)$, its gradient $\widehat{\mathcal{G}}(\theta_k)$ and its Hessian $\widehat{\mathcal{H}}(\theta_k)$.
    2. Determine ε_k.
    3. Apply the Newton update to obtain θ_{k+1}.
    4. Set $k = k + 1$.

    end while
3. Set $\widehat{\theta}_{\text{ML}} = \theta_k$.

Linearization approximation	Sampling approximation

Algorithm

Newton method for ML parameter estimation

1. Set $k = 0$
2. **while** *exit condition is not satisfied* **do**
    1. Run an algorithm to estimate the log-likelihood $\widehat{\ell}(\theta_k)$, its gradient $\widehat{\mathcal{G}}(\theta_k)$ and its Hessian $\widehat{\mathcal{H}}(\theta_k)$.
    2. Determine ε_k.
    3. Apply the Newton update to obtain θ_{k+1}.
    4. Set $k = k + 1$.

   end while
3. Set $\widehat{\theta}_{\mathsf{ML}} = \theta_k$.

Linearization approximation	**Sampling approximation**
Estimate $\widehat{\ell}(\theta_k)$ using an EKF.	

Algorithm

Newton method for ML parameter estimation

1. Set $k = 0$
2. **while** *exit condition is not satisfied* **do**
   1. Run an algorithm to estimate the log-likelihood $\widehat{\ell}(\theta_k)$, its gradient $\widehat{\mathcal{G}}(\theta_k)$ and its Hessian $\widehat{\mathcal{H}}(\theta_k)$.
   2. Determine ε_k.
   3. Apply the Newton update to obtain θ_{k+1}.
   4. Set $k = k + 1$.

   end while
3. Set $\widehat{\theta}_{\text{ML}} = \theta_k$.

Linearization approximation	Sampling approximation
Estimate $\widehat{\ell}(\theta_k)$ using an EKF. Estimate $\widehat{\mathcal{G}}(\theta_k), \widehat{\mathcal{H}}(\theta_k)$ using Gauss-Newton optimization.	Estimate $\widehat{\mathcal{G}}(\theta_k), \widehat{\mathcal{H}}(\theta_k)$ using a particle smoother.

Algorithm

Newton method for ML parameter estimation

1. Set $k = 0$
2. **while** *exit condition is not satisfied* **do**
    1. Run an algorithm to estimate the log-likelihood $\widehat{\ell}(\theta_k)$, its gradient $\widehat{\mathcal{G}}(\theta_k)$ and its Hessian $\widehat{\mathcal{H}}(\theta_k)$.
    2. Determine ε_k.
    3. Apply the Newton update to obtain θ_{k+1}.
    4. Set $k = k + 1$.

   end while
3. Set $\widehat{\theta}_{\mathsf{ML}} = \theta_k$.

Linearization approximation	**Sampling approximation**
Estimate $\widehat{\ell}(\theta_k)$ using an EKF.	
Estimate $\widehat{\mathcal{G}}(\theta_k), \widehat{\mathcal{H}}(\theta_k)$ using Gauss-Newton optimization.	Estimate $\widehat{\mathcal{G}}(\theta_k), \widehat{\mathcal{H}}(\theta_k)$ using a particle smoother.
Determine ε_k using a line search.	Use a pre-defined schedule for ε_k.

Simulation results: SSM1

$1\,000$ data points, 100 data sets

$$\begin{aligned} x_{t+1} &= \arctan x_t + v_t, & v_t &\sim \mathcal{N}(0,1), \\ y_t &= \theta_1 x_t + \theta_2 + e_t, & e_t &\sim \mathcal{N}(0,0.1^2). \end{aligned}$$

Simulation results: SSM1

$1\,000$ data points, 100 data sets

$$\begin{aligned} x_{t+1} &= \arctan x_t + v_t, & v_t &\sim \mathcal{N}(0, 1), \\ y_t &= \theta_1 x_t + \theta_2 + e_t, & e_t &\sim \mathcal{N}(0, 0.1^2). \end{aligned}$$

Parameters only in the linear part of the SSM.

$\Rightarrow$ The linearization approach works very well.

Simulation results: SSM1

$1\,000$ data points, 100 data sets

$$
\begin{aligned}
x_{t+1} &= \arctan x_t + v_t, & v_t &\sim \mathcal{N}(0,1), \\
y_t &= \theta_1 x_t + \theta_2 + e_t, & e_t &\sim \mathcal{N}(0,0.1^2).
\end{aligned}
$$

Parameters only in the linear part of the SSM.

$\Rightarrow$ The linearization approach works very well.

Alg.	Bias ($\cdot 10^{-4}$)		MSE ($\cdot 10^{-4}$)	
	θ_1	θ_2	θ_1	θ_2
Linearization	10	10	1	10
Sampling (FL)	38	-214	2	16
Sampling (FFBSi)	31	53	1	11

Simulation results: SSM1

$1\,000$ data points, 100 data sets

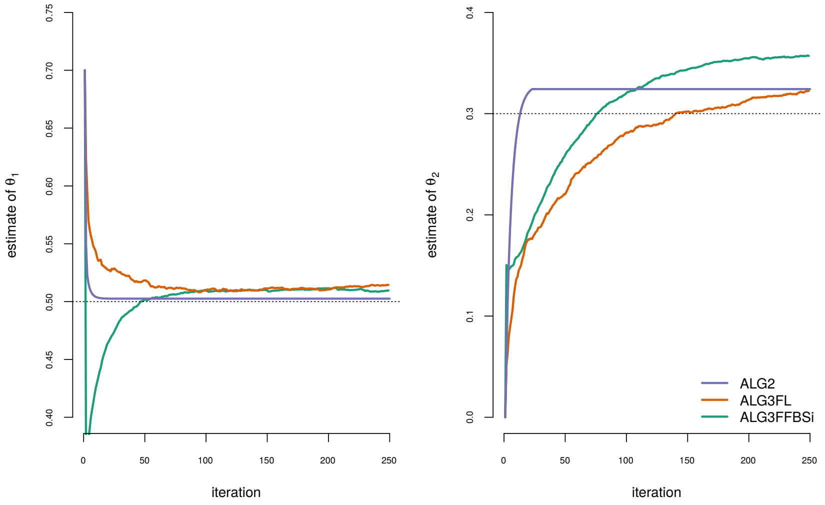
$$\begin{aligned} x_{t+1} &= \arctan x_t + v_t, & v_t &\sim \mathcal{N}(0,1), \\ y_t &= \theta_1 x_t + \theta_2 + e_t, & e_t &\sim \mathcal{N}(0, 0.1^2). \end{aligned}$$

Parameters only in the linear part of the SSM.

$\Rightarrow$ The linearization approach works very well.

Alg.	Bias ($\cdot 10^{-4}$)		MSE ($\cdot 10^{-4}$)	
	θ_1	θ_2	θ_1	θ_2
Linearization	10	10	1	10
Sampling (FL)	38	-214	2	16
Sampling (FFBSi)	31	53	1	11

Trace plot



0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
estimate of θ_1
0
50
100
150
200
250
iteration
0.4
0.3
0.2
0.1
0.0
estimate of θ_2
0
50
100
150
200
250
iteration
ALG2
ALG3FL
ALG3FFBSi

Simulation results: SSM2

$1\,000$ data points, 100 data sets

$$\begin{aligned} x_{t+1} &= \theta_1 \arctan x_t + v_t, && v_t \sim \mathcal{N}(0, 1) \\ y_t &= \theta_2 x_t + e_t, && e_t \sim \mathcal{N}(0, 0.1^2), \end{aligned}$$

Simulation results: SSM2

$1\,000$ data points, 100 data sets

$$x_{t+1} = \theta_1 \arctan x_t + v_t, \qquad v_t \sim \mathcal{N}(0,1)$$
$$y_t = \theta_2 x_t + e_t, \qquad e_t \sim \mathcal{N}(0,0.1^2),$$

Parameters also in the nonlinear part of the SSM.

$\Rightarrow$ The linearization approach needs more approximations.

Simulation results: SSM2

$1\,000$ data points, 100 data sets

$$x_{t+1} = \theta_1 \arctan x_t + v_t, \qquad v_t \sim \mathcal{N}(0,1)$$
$$y_t = \theta_2 x_t + e_t, \qquad e_t \sim \mathcal{N}(0, 0.1^2),$$

Parameters also in the nonlinear part of the SSM.

$\Rightarrow$ The linearization approach needs more approximations.

Alg.	Bias ($\cdot 10^{-4}$)		MSE ($\cdot 10^{-4}$)	
	θ_1	θ_2	θ_1	θ_2
Linearization	284	-55	28	2
Sampling (FL)	53	35	24	2
Sampling (FFBSi)	55	31	24	2

Simulation results: SSM2

1 000 data points, 100 data sets

$$x_{t+1} = \theta_1 \arctan x_t + v_t, \qquad v_t \sim \mathcal{N}(0, 1)$$
$$y_t = \theta_2 x_t + e_t, \qquad e_t \sim \mathcal{N}(0, 0.1^2),$$

Parameters also in the nonlinear part of the SSM.

$\Rightarrow$ The linearization approach needs more approximations.

Alg.	Bias ($\cdot 10^{-4}$)		MSE ($\cdot 10^{-4}$)	
	θ_1	θ_2	θ_1	θ_2
Linearization	284	-55	28	2
Sampling (FL)	53	35	24	2
Sampling (FFBSi)	55	31	24	2

Conclusions

Conclusions

- We studied the problem of ML parameter estimation in nonlinear SSMs using Newton methods.

Conclusions

- We studied the problem of ML parameter estimation in nonlinear SSMs using Newton methods.
- We determine the gradient and Hessian of the log-likelihood using Fisher's identity in combination with an algorithm to obtain smoothed state estimates.

Conclusions

- We studied the problem of ML parameter estimation in nonlinear SSMs using Newton methods.
- We determine the gradient and Hessian of the log-likelihood using Fisher's identity in combination with an algorithm to obtain smoothed state estimates.
- We approximate the log-likelihood and its gradient and Hessian using *linearizations* and using *sampling methods*.

Conclusions

- We studied the problem of ML parameter estimation in nonlinear SSMs using Newton methods.
- We determine the gradient and Hessian of the log-likelihood using Fisher's identity in combination with an algorithm to obtain smoothed state estimates.
- We approximate the log-likelihood and its gradient and Hessian using *linearizations* and using *sampling methods*.
- The *linearization approach* is computationally cheap. However, the quality of its estimates highly depends on the structure of the SSM.

Future work

Future work

- Study the quality of the estimates for a wider range of nonlinear models.

Future work

- Study the quality of the estimates for a wider range of nonlinear models.
- Compare the method from this work to a solution based on expectation maximization.

Future work

- Study the quality of the estimates for a wider range of nonlinear models.
- Compare the method from this work to a solution based on expectation maximization.
- Study optimization problems where noisy estimates of the gradient and the Hessian are provided by a particle smoother.

Future work

- Study the quality of the estimates for a wider range of nonlinear models.
- Compare the method from this work to a solution based on expectation maximization.
- Study optimization problems where noisy estimates of the gradient and the Hessian are provided by a particle smoother.

For more information: Manon Kok, Johan Dahlin, Thomas B. Schön and Adrian Wills, Newton-based maximum likelihood estimation in nonlinear state space models. Proceedings of the 17th IFAC Symposium on System Identification, Beijing, China, October 2015.
Source code, data, slides and paper online on
http://users.isy.liu.se/en/rt/manko/research.html

Thank you for your attention!

Questions?

CADICS

This work is supported by CADICS, a Linnaeus Center, and by the project *Probabilistic modeling of dynamical systems* (Contract number: 621-2013-5524), both funded by the Swedish Research Council (VR).

Vetenskapsrådet