Supplement to "A second-order iterated smoothing algorithm"

D. Nguyen[1] **and Edward L. Ionides**[1]

August 19, 2015

[1] Department of Statistics
The University of Michigan, Ann Arbor, Michigan, USA.
email: nguyenxd@umich.edu, ionides@umich.edu

Supplementary Content

S1 Comparison of methods on the toy example

We continue with the toy example in the main text, a bivariate, linear, Gaussian discrete-time process. Fig. S-1 shows the results of 40 Monte Carlo replications so that we can see the clustering of the MLE estimates around the true MLE, corresponding to Fig. 1 in the main text. The computations in Fig. S-1 match the setup in the main text. For IS2, most of the replications clustered near the true MLE while none of them stays in a lower likelihood region. Fig. 1, in the main text, can be viewed as a statistical summary of Fig. S-1, with 200 Monte Carlo replications. These results indicate that IS2 is clearly the best of the investigated methods for this test.

We also checked how the methods compared when given additional computational resources, setting $M = 100$ iterations and $J = 10000$ particles, with the random walk standard deviation decreasing geometrically from 0.23 down to 0.0207 for RIS1 and from 0.02 down to 0.0018 for other methods. In this situation, IS2 is better than both IF2 and RIS1, and IF1 performed substantially worse than the other methods (Fig. S-2). All four of these methods have comparable computational demands for given M and J. IS1 requires substantially more computational resources, and we did not compute it for this comparison.

S2 Algorithms IS1 and RIS1

The pseudo-code in Algorithm S2 corresponds to the iterated smoothing algorithm of [1]. The computational complexity of approach in [1] is $\mathcal{O}(LN)$, the algorithm is expected to be slow, especially when computing covariance of every pair of time points with distance smaller than L. We also propose a variant of IS1 using a computationally convenient approximation to this covariance; we call this method reduced IS1 (RIS1). reduced iterated smoothing algorithm of [1], called RIS1. RIS1 avoids the computational expense of computing covariances at different lags by simply ignoring these terms. This makes pseudo-code for RIS1 look more like IS2, but with white noise parameter perturbations in place of random walk perturbations. Specifically, RIS1 is a modification of IS2 for which, at line 5 in Algorithm 1, we do not update $\Theta^F_{t-1,n}$. For IS2, these covariance terms cancel in the theoretical analysis. However, there is no theorem to support RIS1 and it is only justified heuristically based on the observation that covariance between different time points may be small in practice. RIS1 is not presented for its theoretical interest, but for empirical interest in providing a computationally efficient benchmark for comparing between white noise and random walk noise.

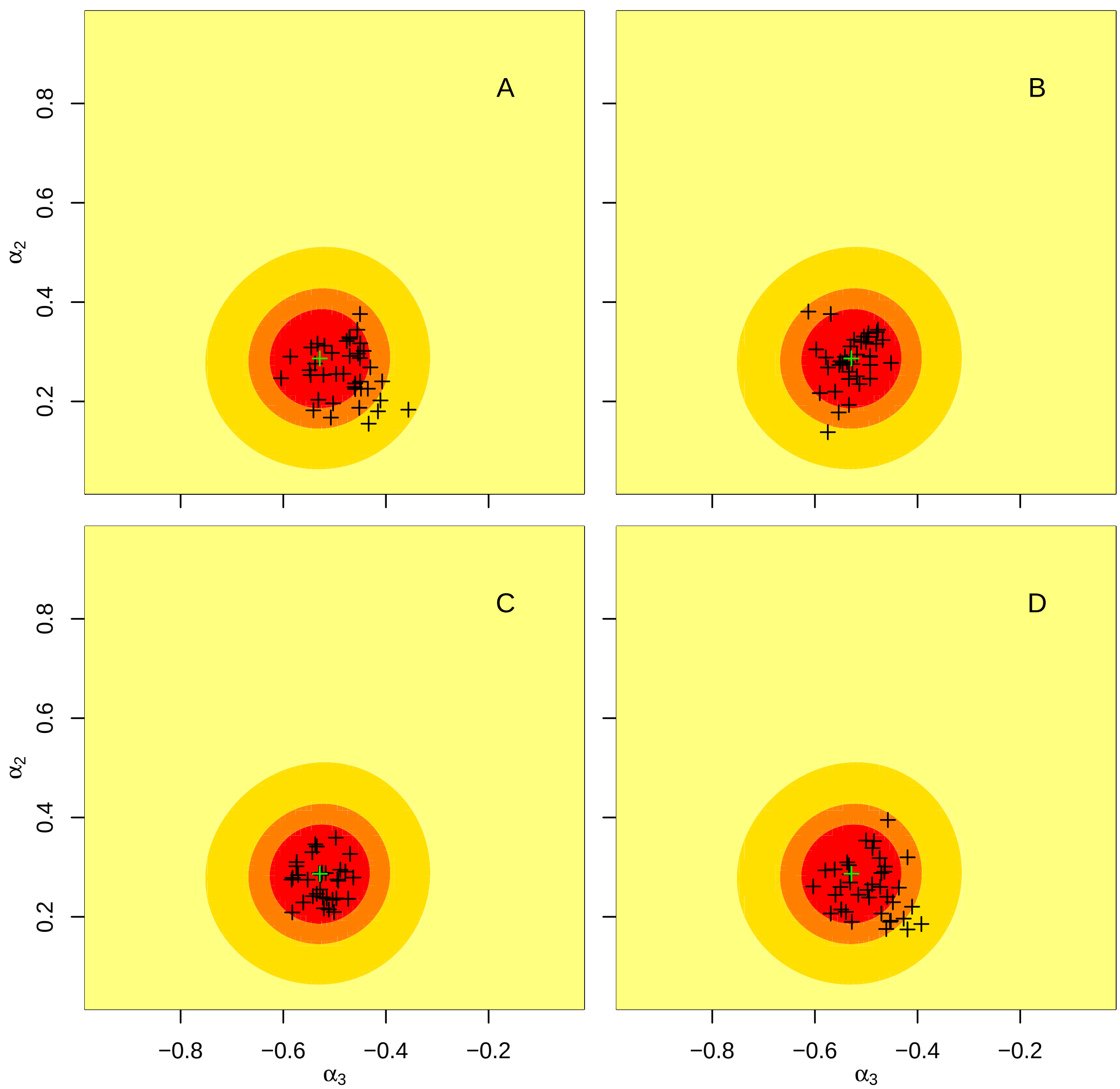

Figure S-1: Comparison of different estimators. The likelihood surface for the linear, Gaussian model, with likelihood within 2 log units of the maximum shown in red, within 4 log units in orange, within 10 log units in yellow, and lower in light yellow. The location of the MLE is marked with a green cross. The black crosses show final points from 40 Monte Carlo replications of the estimators: (A) IF1 method; (B) IF2 method; (C) IS2 method; (D) RIS1 method. Each method, except RIS1, was started uniformly over the rectangle shown, with $M = 25$ iterations, $N = 1000$ particles, and a random walk standard deviation decreasing from 0.02 geometrically to 0.011 for both α_2 and α_3. We use bigger random walk standard deviations for RIS1. Specifically random walk standard deviations decrease from 0.23 geometrically to 0.125 for both α_2 and α_3.

Algorithm S1 Iterating smoothing using white noise perturbations (IS1)

Input:

starting parameter, θ_0

simulator for $f_n(x_n|x_{n-1};\theta)$

initial value parameters, $I \subset \{1, .., d\}$

number of iteration, M

perturbation scales, $\sigma_{1:d}$, $\Psi = \text{diag}(\sigma^2_{1:d})$

simulator for $\mu(x_0;\theta)$

evaluator for $g_n(y_n|x_n;\theta)$

data, $y^*_{1:N}$

number of particles, J

cooling rate, $0 < c < 1$, lag, L

Output:

Monte Carlo maximum likelihood estimate, θ_M

for m in $1:M$ **do**

  initialize parameters: $\left[\Theta^F_{0,j}\right]_i \sim \mathcal{N}\big(\left[\theta_{m-1}\right]_i, (c^{m-1}\sigma_i)^2\big)$ for j in $1:J$, i in $1:d$

  initialize states: simulate $X^F_{0,j} \sim \mu\big(x_0;\Theta^F_{0,j}\big)$ for j in $1:J$

  for n in $1:N$ **do**

    perturb at time n: $\left[\Theta^P_{n,j}\right]_i \sim \mathcal{N}\big(\left[\Theta^F_{n-1,j}\right]_i, (c^{m-1}\sigma_i)^2\big)$ for $i \notin I$, j in $1:J$

    simulate prediction particles: $X^P_{n,j} \sim f_n\big(x_n|X^F_{n-1,j};\Theta^P_{n,j}\big)$ for j in $1:J$

    evaluate weights: $w(n,j) = g_n(y^*_n|X^P_{n,j};\Theta^P_{n,j})$ for j in $1:J$

    normalize weights: $\breve{w}(n,j) = w(n,j)/\sum_{u=1}^J w(n,u)$

    apply re-sampling to select indices $k_{1:J}$ with $P\{k_u = j\} = \breve{w}(n,j)$

    re-sample particles: $X^F_{n,j} = X^P_{n,k_j}$ and $\Theta^F_{n,j} = \Theta^P_{n,k_j}$ for j in $1:J$

    let $a_1(n,k_j) = j$, $a_{l+1}(n,j) = a_1(n-l, a_l(n,j))$ for j in $1:J$, l in $0:L-1$

  end for

  for n in $1:N$ **do**

    smoothed mean: $\bar{\theta}^L_{n-L} = \sum_{j=1}^J \breve{w}(n,j)\Theta^P_{n-L,a_L(n,j)}$ if $n > L$

    for l in $n:\min(n+L,N)$ **do**

      Covariance: $C^m_{n-L,l-L} = \sum_j \breve{w}(n,j)\big(\Theta^P_{n-L,a_L(n,j)} - \bar{\theta}^L_{n-L}\big)\big(\Theta^P_{l-L,a_L(n,j)} - \bar{\theta}^L_{l-L}\big)^\top$ if $n > L$

    end for

  end for

  for j in $0:L$ **do**

    smoothed mean: $\bar{\theta}^L_{N+j-L} = \sum_{j=1}^J \breve{w}(N,j)\Theta^P_{N+j-L,a_{L-j}(N,j)}$

    for l in $N+j-L:N$ **do**

      Covariance: $C^m_{N+j-L,l} = \sum_j \breve{w}(N,j)\big(\Theta^P_{N+j-L,a_{L-j}(N,j)} - \bar{\theta}^L_{N+j-L}\big)\big(\Theta^P_{l,a_{N-l}(N,j)} - \bar{\theta}^L_l\big)^\top$

    end for

  end for

  update: $S_m = c^{-2(m-1)}\Psi^{-1}\sum_{n=1}^N \left[\,\big(\bar{\theta}^L_n - \theta_{m-1}\big)\,\right]$

  $I_m = -c^{-4(m-1)}\Psi^{-1}\left[\sum_{n=1}^N \left(C^m_{n,n} - c^{2(m-1)}\Psi + 2\sum_{s=n+1}^{(s+L)\wedge N} C^m_{s,n}\right)\right]\Psi^{-1}$

  update parameters: $\theta_m = \theta_{m-1} + I_m^{-1}S_m$

end for

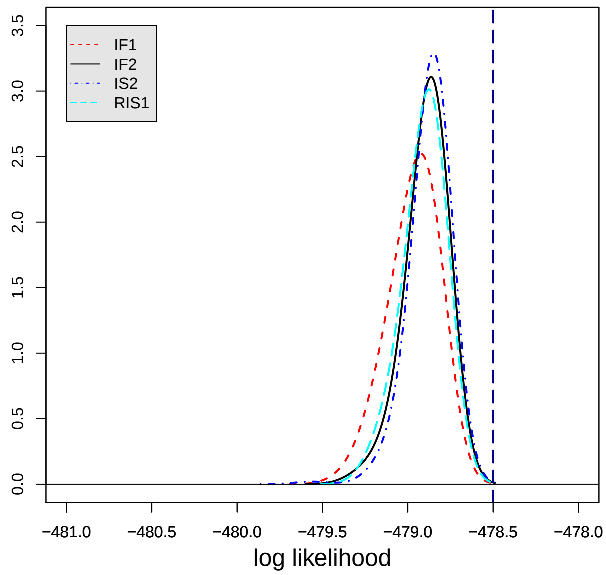

Figure S-2: The distributions of likelihoods corresponding to Monte Carlo MLE approximations estimated by IF1, IF2, RIS1 and IS2 methods for toy model. The MLE is shown as a dashed vertical line (dark blue in electronic version). The optimizations were started from 200 randomly uniform initial values over a rectangle.

S3 Parameters definitions and starting ranges for the malaria model

Table S-4. Parameters for the malaria $SEIH^3Q$ model.

Symbol	Definition	Units	θ_{low}	θ_{high}
μ_{EI} $(*)$	E $\rightarrow$ I transition rate	yr^{-1}	24	24
μ_{IH}	I $\rightarrow$ H transition rate	yr^{-1}	1.00	5.00
μ_{HI}	H $\rightarrow$ I transition rate	yr^{-1}	1.00	5.00
μ_{IS}	I $\rightarrow$ S transition rate	yr^{-1}	0.5	2.00
μ_{IQ}	I $\rightarrow$ Q transition rate	yr^{-1}	1.00	2.00
μ_{QS}	Q $\rightarrow$ S transition rate	yr^{-1}	10.00	20.00
q $(*)$	relative infectivity of Q class	–	0.001	0.001
τ	mean lag for mosquitoes	month	0.10	0.50
ρ	case reporting fraction	–	0.001	0.01
σ_{pro}	s.d. of dynamic noise	$\text{yr}^{0.5}$	0.1	0.5
σ_{obs}	s.d. of measurement noise	–	0.1	0.5
b_r	coefficient of rainfall covariate	–	0.5	0.9
S_0	initial fraction in S class	–	0	1
E_0	initial fraction in E class	–	0	1
I_0	initial fraction in I class	–	0	1
$H_{i,0}$	initial fraction in H_i class	–	0	1
Q_0	initial fraction in Q class	–	0	1
κ_0	initial value, $\kappa(t_0)$	–	0.1	0.5
$\mu_{SE,0}$	initial value, $\mu_{SE}(t_0)$	–	0.1	0.5
b_1	1^{st} spline coefficient	–	-5	5
b_2	2^{nd} spline coefficient	–	-5	5
b_3	3^{rd} spline coefficient	–	-5	5
b_4	4^{th} spline coefficient	–	-5	5
b_5	5^{th} spline coefficient	–	-5	5
b_6	6^{th} spline coefficient	–	-5	5
$1/\delta$ $(*)$	mean human life span	yr	0.02	0.02

We follow definitions as in [2]. θ_{low} and θ_{high} are the lower and upper bounds for a hyper-rectangle used to generate starting points for the search. Parameters labeled with $(*)$ were set at fixed values. Non-negative parameters were logarithmically transformed for optimization.

Supplementary References

[1] Doucet, A, Jacob, P. E, & Rubenthaler, S. (2013) Derivative-free estimation of the score vector and observed information matrix with application to state-space models. *ArXiv:1304.5768*.

[2] Roy, M, Bouma, M. J, Ionides, E. L, Dhiman, R. C, & Pascual, M. (2013) The potential elimination of plasmodium vivax malaria by relapse treatment: Insights from a transmission model and surveillance data from NW India. *PLoS Neglected Tropical Diseases* **7**, e1979.