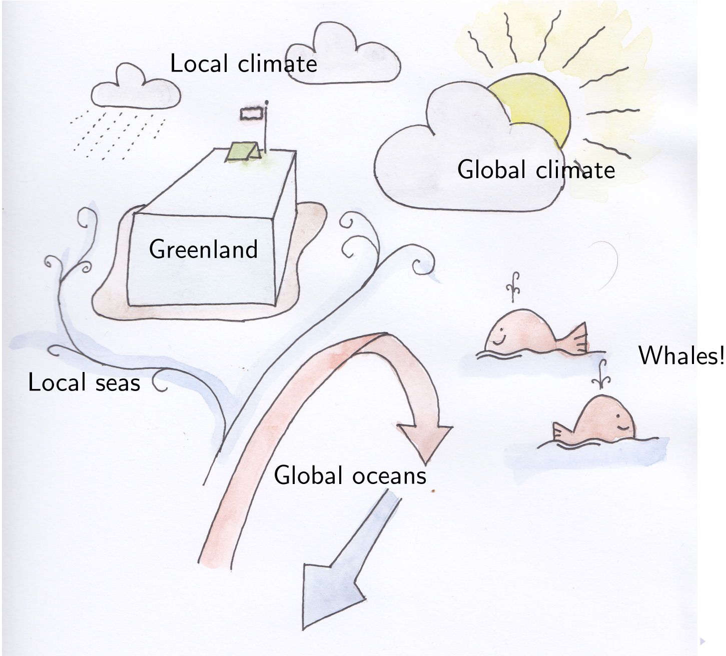
Complex systems: Accounting for model limitations

Jonathan Rougier

Department of Mathematics
University of Bristol, UK

St Andrews, Monday 22 March 2010

Illustration: the Greenland ice-sheet



Simplest interesting example

Conditional on θ:

$$x_0 \sim \pi_{x_0}(\theta) \qquad \text{(init. cond. unc.)}$$

$$x_t = g(x_{t-1}; \theta) + Q(x_{t-1}; \theta)\, \omega_t \qquad \text{(state eqn.)}$$

$$y_t = f(x_t; \theta) + \nu_t \qquad \text{(obs. eqn.)}$$

where

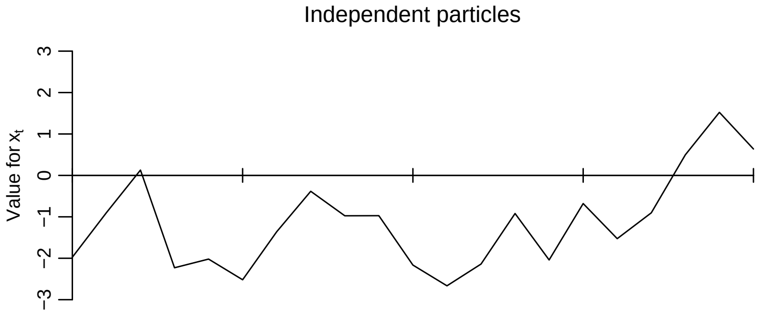
$$\omega_t \overset{\text{iid}}{\sim} \mathrm{N}(0, I) \qquad \text{(structural uncertainty)}$$

$$\nu_t \overset{\text{iid}}{\sim} \mathrm{N}(0, v^2) \qquad \text{(measurement unc.)}$$

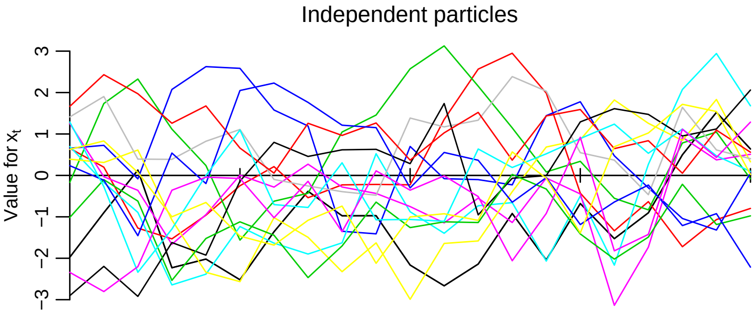
and then let $\theta \sim \pi_\theta$, to account for parametric uncertainty. The functions f, g, and Q are given, likewise the measurement uncertainty standard deviation, v.

Sampling from $\{x_{0:T}, \theta\} \mid y_{1:T}$ "intractable and unsolved" (C. Andrieu)

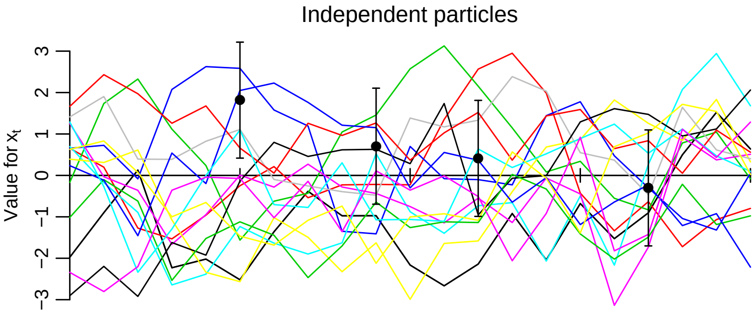
Particle filters, for given θ



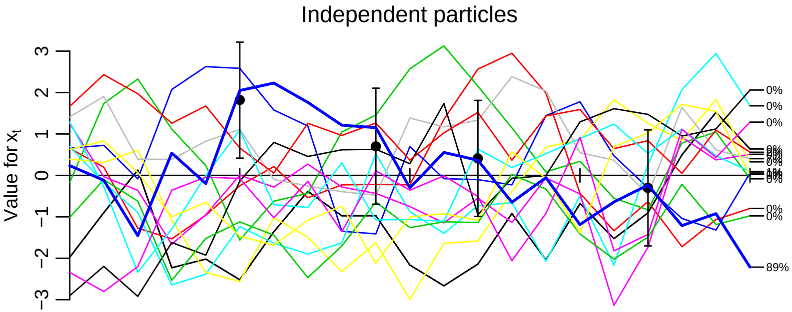
Particle filters, for given θ



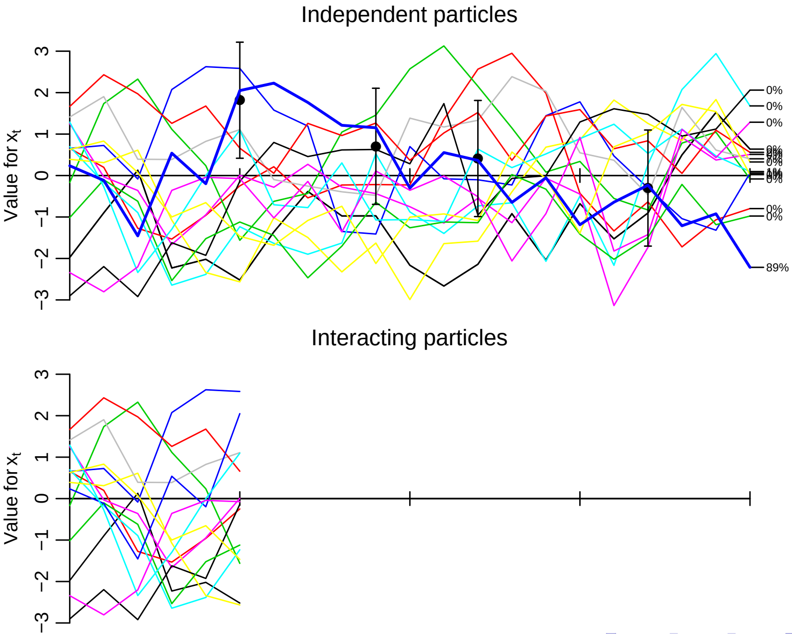
Particle filters, for given θ



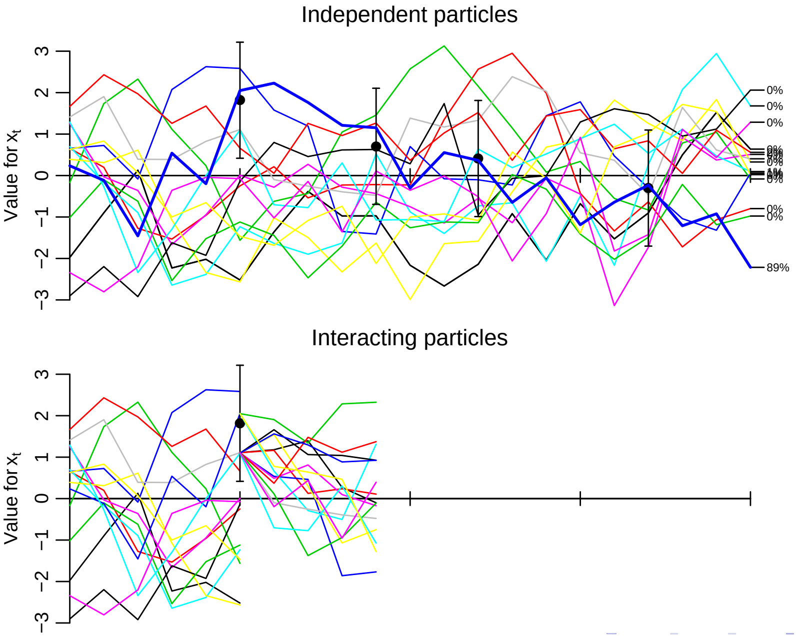
Particle filters, for given θ



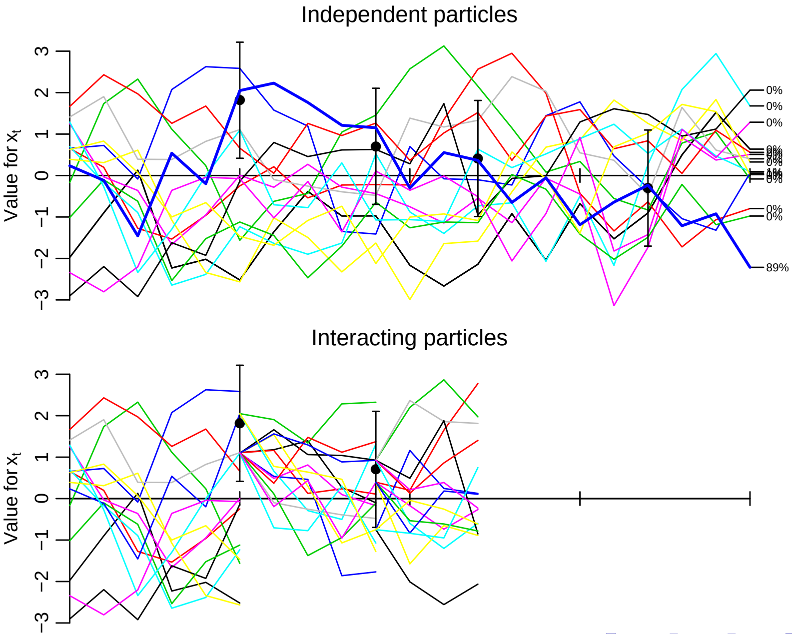
Particle filters, for given θ



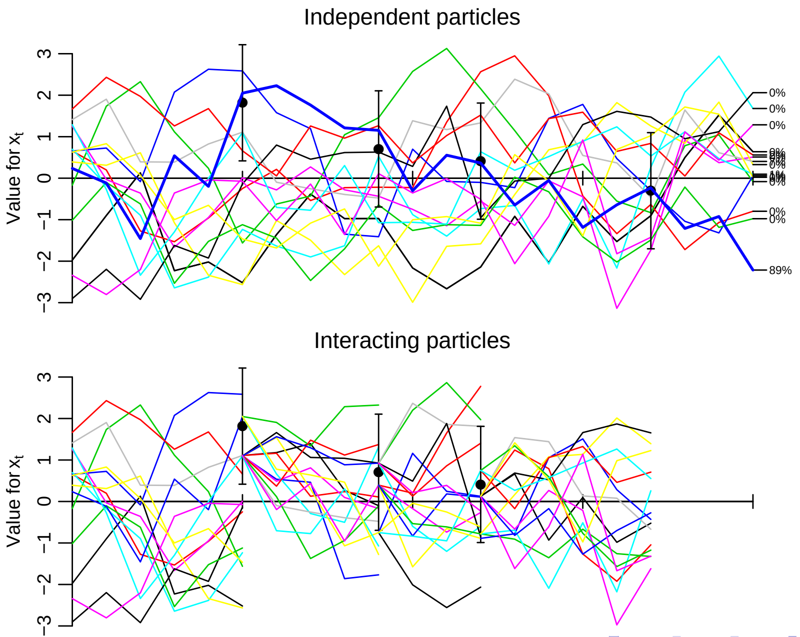
Particle filters, for given θ



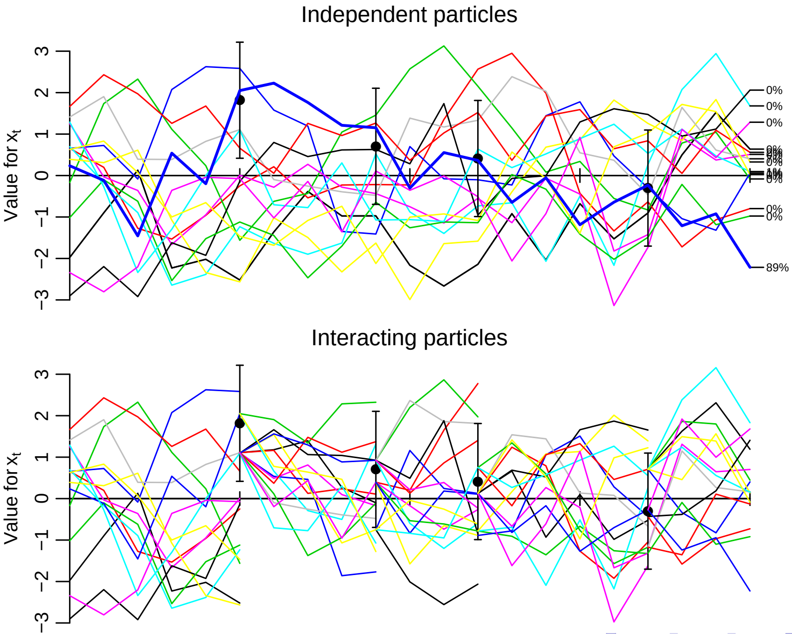
Particle filters, for given θ



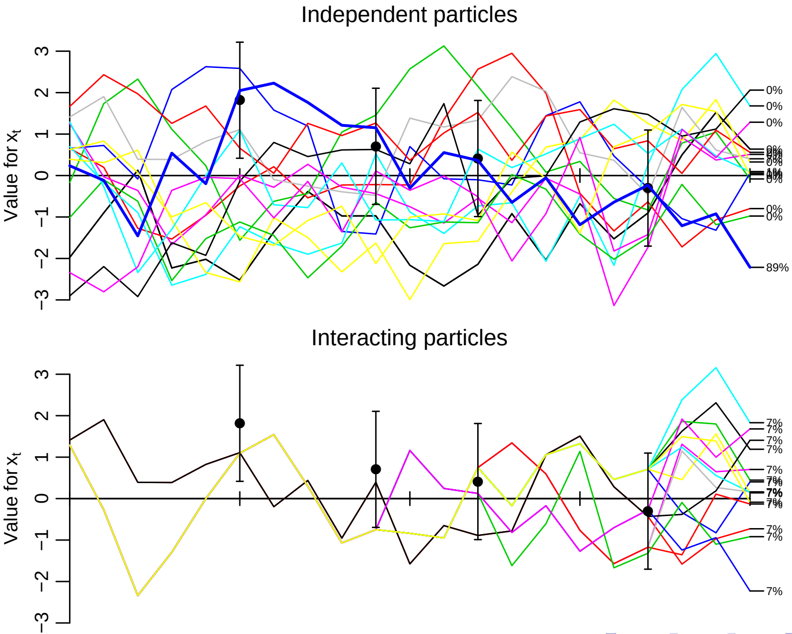
Particle filters, for given θ



Particle filters, for given θ



Particle filters, for given θ



The difficulties with uncertainty θ

- One simple idea is to attach a realisation from π_θ to each particle, in order to sample jointly from $\{x_{0:T}, \theta\} \mid y_{1:T}$.

  However, static parameters do not evolve in time, so every interaction that culls particles reduces the resolution of the θ distribution. Too many observations, and the θ distribution becomes degenerate, unless we have `<DrEvil>`one million`</DrEvil>` particles.

The difficulties with uncertainty θ

- One simple idea is to attach a realisation from π_θ to each particle, in order to sample jointly from $\{x_{0:T}, \theta\} \mid y_{1:T}$.

  However, static parameters do not evolve in time, so every interaction that culls particles reduces the resolution of the θ distribution. Too many observations, and the θ distribution becomes degenerate, unless we have <DrEvil>one million</DrEvil> particles.

- The solution is to 'integrate out' the state vector in some form. The two approaches are

  1. Gaussian (Laplace) approximation for $x_{1:T} \mid \{\theta, y_{1:T}\}$ turning a high-dimensional integration into a high-dimensional optimisation;
  2. Particle Markov chain Monte Carlo (P-MCMC), which uses a Gibbs sampler to swap between sampling $x_{1:T} \mid \{\theta, y_{1:T}\}$ and $\theta \mid \{x_{1:T}, y_{1:T}\}$.

Candidate's formula

For integrating out nuisance parameters; in our case, the state vector $\mathbf{x} = x_{0:T}$. 'Discovered' by a Durham undergraduate during a Bayesian Modelling exam; recorded by Julian Besag:

$$\pi(\theta) = \pi(\theta)\, \frac{\pi(\theta, \mathbf{x})}{\pi(\theta, \mathbf{x})} \qquad \text{for all } \mathbf{x}$$

Candidate's formula

For integrating out nuisance parameters; in our case, the state vector $\mathbf{x} = x_{0:T}$. 'Discovered' by a Durham undergraduate during a Bayesian Modelling exam; recorded by Julian Besag:

$$
\begin{aligned}
\pi(\theta) &= \pi(\theta)\, \frac{\pi(\theta, \mathbf{x})}{\pi(\theta, \mathbf{x})} && \text{for all } \mathbf{x} \\
&= \pi(\theta)\, \frac{\pi(\theta, \mathbf{x})}{\pi(\mathbf{x} \mid \theta)\, \pi(\theta)} && \text{for all } \mathbf{x}
\end{aligned}
$$

Candidate's formula

For integrating out nuisance parameters; in our case, the state vector $\mathbf{x} = x_{0:T}$. 'Discovered' by a Durham undergraduate during a Bayesian Modelling exam; recorded by Julian Besag:

$$\begin{aligned}
\pi(\theta) &= \pi(\theta)\,\frac{\pi(\theta,\mathbf{x})}{\pi(\theta,\mathbf{x})} && \text{for all } \mathbf{x} \\
&= \pi(\theta)\,\frac{\pi(\theta,\mathbf{x})}{\pi(\mathbf{x}\mid\theta)\,\pi(\theta)} && \text{for all } \mathbf{x} \\
&= \frac{\pi(\theta,\mathbf{x})}{\pi(\mathbf{x}\mid\theta)} && \text{for all } \mathbf{x}.
\end{aligned}$$

Candidate's formula

For integrating out nuisance parameters; in our case, the state vector $\mathbf{x} = x_{0:T}$. 'Discovered' by a Durham undergraduate during a Bayesian Modelling exam; recorded by Julian Besag:

$$\begin{aligned} \pi(\theta) &= \pi(\theta)\, \frac{\pi(\theta, \mathbf{x})}{\pi(\theta, \mathbf{x})} && \text{for all } \mathbf{x} \\ &= \pi(\theta)\, \frac{\pi(\theta, \mathbf{x})}{\pi(\mathbf{x} \mid \theta)\, \pi(\theta)} && \text{for all } \mathbf{x} \\ &= \frac{\pi(\theta, \mathbf{x})}{\pi(\mathbf{x} \mid \theta)} && \text{for all } \mathbf{x}. \end{aligned}$$

In our case, writing $\mathbf{y} = y_{1:T}$,

$$\pi(\theta \mid \mathbf{y}) \propto \frac{\pi(\theta, \mathbf{x}, \mathbf{y})}{\pi(\mathbf{x} \mid \theta, \mathbf{y})} \quad \text{for all } \mathbf{x},$$

where the constant of proportionality is $1/\pi(\mathbf{y})$.

'Cool' derivation of Laplace approximation

We start with Candidate's formula, use a Gaussian approximation in the denominator, and then plug in for $\mathbf{x}$:

$$\pi(\theta \mid \mathbf{y}) \propto \frac{\pi(\theta, \mathbf{x}, \mathbf{y})}{\pi(\mathbf{x} \mid \theta, \mathbf{y})} \qquad \text{for all } \mathbf{x}$$

‘Cool’ derivation of Laplace approximation

We start with Candidate’s formula, use a Gaussian approximation in the denominator, and then plug in for $\mathbf{x}$:

$$\pi(\theta \mid \mathbf{y}) \propto \frac{\pi(\theta, \mathbf{x}, \mathbf{y})}{\pi(\mathbf{x} \mid \theta, \mathbf{y})} \qquad \text{for all } \mathbf{x}$$

$$\approx \frac{\pi(\theta, \mathbf{x}, \mathbf{y})}{\phi(\mathbf{x}; \mu(\theta), \Sigma(\theta))} \quad \text{for all } \mathbf{x} \text{ (?), Gaussian approx.}$$

where

$$\mu(\theta) = \operatorname*{argmin}_{\mathbf{x}} \{-\log \pi(\mathbf{x} \mid \theta, \mathbf{y})\}$$

$$\Sigma(\theta) = \left[-\nabla^2 \log \pi(\mathbf{x} \mid \theta, \mathbf{y})\right]^{-1} \quad \text{at } \mathbf{x} = \mu(\theta).$$

'Cool' derivation of Laplace approximation

We start with Candidate's formula, use a Gaussian approximation in the denominator, and then plug in for $\mathbf{x}$:

$$
\begin{aligned}
\pi(\theta \mid \mathbf{y}) &\propto \frac{\pi(\theta, \mathbf{x}, \mathbf{y})}{\pi(\mathbf{x} \mid \theta, \mathbf{y})} && \text{for all } \mathbf{x} \\
&\approx \frac{\pi(\theta, \mathbf{x}, \mathbf{y})}{\phi(\mathbf{x}; \mu(\theta), \Sigma(\theta))} && \text{for all } \mathbf{x} \text{ (?), Gaussian approx.} \\
&= \frac{\pi(\theta, \mu(\theta), \mathbf{y})}{|2\pi\Sigma(\theta)|^{-1/2}} && \text{plug in } \mathbf{x} = \mu(\theta),
\end{aligned}
$$

where

$$
\begin{aligned}
\mu(\theta) &= \operatorname*{argmin}_{\mathbf{x}}\{-\log \pi(\mathbf{x} \mid \theta, \mathbf{y})\} \\
\Sigma(\theta) &= \left[-\nabla^2 \log \pi(\mathbf{x} \mid \theta, \mathbf{y})\right]^{-1} \quad \text{at } \mathbf{x} = \mu(\theta).
\end{aligned}
$$

'Cool' derivation of Laplace approximation

We start with Candidate's formula, use a Gaussian approximation in the denominator, and then plug in for $\mathbf{x}$:

$$\begin{aligned}
\pi(\theta \mid \mathbf{y}) &\propto \frac{\pi(\theta, \mathbf{x}, \mathbf{y})}{\pi(\mathbf{x} \mid \theta, \mathbf{y})} && \text{for all } \mathbf{x} \\
&\approx \frac{\pi(\theta, \mathbf{x}, \mathbf{y})}{\phi(\mathbf{x}; \mu(\theta), \Sigma(\theta))} && \text{for all } \mathbf{x} \text{ (?), Gaussian approx.} \\
&= \frac{\pi(\theta, \mu(\theta), \mathbf{y})}{|2\pi\Sigma(\theta)|^{-1/2}} && \text{plug in } \mathbf{x} = \mu(\theta),
\end{aligned}$$

where

$$\begin{aligned}
\mu(\theta) &= \underset{\mathbf{x}}{\operatorname{argmin}}\{-\log \pi(\mathbf{x} \mid \theta, \mathbf{y})\} \\
\Sigma(\theta) &= \left[-\nabla^2 \log \pi(\mathbf{x} \mid \theta, \mathbf{y})\right]^{-1} \quad \text{at } \mathbf{x} = \mu(\theta).
\end{aligned}$$

Crucial for large problems:

Must use exact gradient function in the optimisation. And must not be sanguine about finding an optimum value.

In practice . . .

Recollect:

$$x_t = g(x_{t-1};\theta) + Q(x_{t-1};\theta)\,\omega_t \qquad t = 1, \ldots, T.$$

- One does not optimise over $\mathbf{x}$. Instead, put x_0 into θ, in which case $\mathbf{x}$ is completely determined by $\omega_{1:T}$ and θ. Thus $\boldsymbol{\omega} = \omega_{1:T}$ becomes the nuisance parameter.

In practice ...

Recollect:

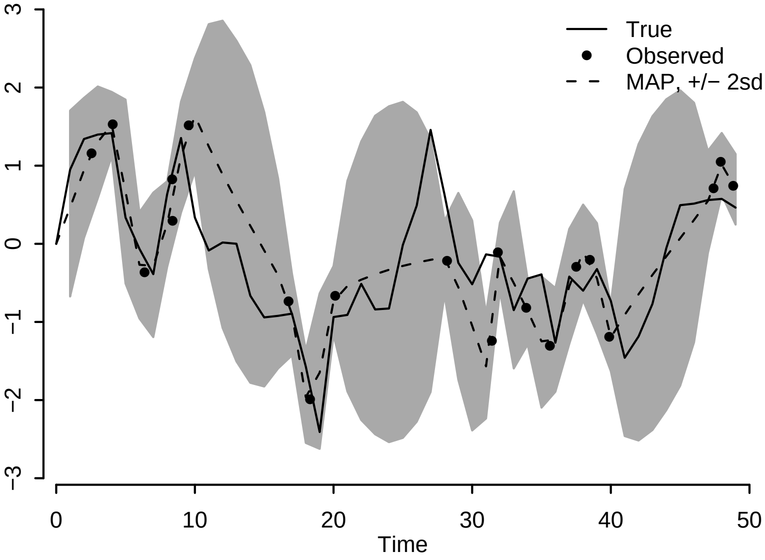
$$x_t = g(x_{t-1}; \theta) + Q(x_{t-1}; \theta)\, \omega_t \qquad t = 1, \dots, T.$$

- One does not optimise over $\mathbf{x}$. Instead, put x_0 into θ, in which case $\mathbf{x}$ is completely determined by $\omega_{1:T}$ and θ. Thus $\boldsymbol{\omega} = \omega_{1:T}$ becomes the nuisance parameter.
- Computing the gradient function of

$$\log \pi(\boldsymbol{\omega} \mid \theta, \mathbf{y}) = c + \log \pi(\mathbf{y} \mid \mathbf{x}(\boldsymbol{\omega}, \theta), \theta) + \log \pi(\boldsymbol{\omega})$$

  is *brutal*, because of the recursive structure of the state equation. But it can be done, in terms of the gradient matrix of g and the gradient tensor of Q.

Proof of concept



Summary

- In inference for environmental systems, model limitations require us to account for both parametric and structural uncertainty.
- The generic problem for dynamical systems is therefore *non-linear data assimilation with uncertain static parameters.*
- Only approximate solutions are available to this notoriously intractable problem. They involve *integrating out the state vector* $\mathbf{x}$, to focus attention on the parameter θ.
- ‘Quick and dirty’ is to use a *Laplace approximation*. This may or may not work. An explicit and exact value for the gradient function is strongly recommended.