Bayesian parameter and state estimation for partially observed nonlinear Markov process models using particle MCMC (with application to sysbio)

Darren Wilkinson
http://tinyurl.com/darrenjw
School of Mathematics & Statistics, Newcastle University, UK

LMS/EPSRC Durham Symposium:
Mathematics of Data Assimilation
Durham, UK
8th August, 2011

Systems biology modelling

- Using accurate high-resolution time-course data on a relatively small number of bio-molecules to parametrise carefully constructed mechanistic dynamic models of a process of interest based on current biological understanding
- Traditionally, models were typically deterministic, based on a system of ODEs known as the Reaction Rate Equations (RREs)
- It is now increasingly accepted that biochemical network dynamics at the single-cell level are intrinsically stochastic
- The theory of stochastic chemical kinetics provides a solid foundation for describing network dynamics using a Markov jump process

Systems Biology models

- Typically consist of a list of (bio-)chemical reactions, together with associated rate equations which govern their "speed"
- The rate equations are usually a function of the current system state, as well as parameters (rate constants)
- From this, we need to make some assumptions about the nature of the kinetics, then end up with a stochastic or deterministic kinetic model

Stochastic Chemical Kinetics

Stochastic molecular approach:

- Statistical mechanical arguments lead to a Markov jump process in continuous time whose instantaneous reaction rates are directly proportional to the number of molecules of each reacting species
- Such dynamics can be simulated (exactly) on a computer using standard discrete-event simulation techniques
- Standard implementation of this strategy is known as the "Gillespie algorithm" (just discrete event simulation), but there are several exact and approximate variants of this basic approach

Lotka-Volterra system

Trivial (familiar) example from population dynamics (in reality, the "reactions" will be elementary biochemical reactions taking place inside a cell)

Reactions

$$X \longrightarrow 2X \qquad \text{(prey reproduction)}$$
$$X + Y \longrightarrow 2Y \qquad \text{(prey-predator interaction)}$$
$$Y \longrightarrow \emptyset \qquad \text{(predator death)}$$

- X – Prey, Y – Predator
- We can re-write this using matrix notation

Forming the matrix representation

The L-V system in tabular form

	Rate Law	LHS		RHS		Net-effect	
	$h(\cdot, c)$	X	Y	X	Y	X	Y
R_1	$c_1 x$	1	0	2	0	1	0
R_2	$c_2 xy$	1	1	0	2	-1	1
R_3	$c_3 y$	0	1	0	0	0	-1

Call the 3×2 net-effect (or reaction) matrix N. The matrix $S = N'$ is the stoichiometry matrix of the system. Typically both are sparse. The SVD of S (or N) is of interest for structural analysis of the system dynamics...

Stochastic chemical kinetics

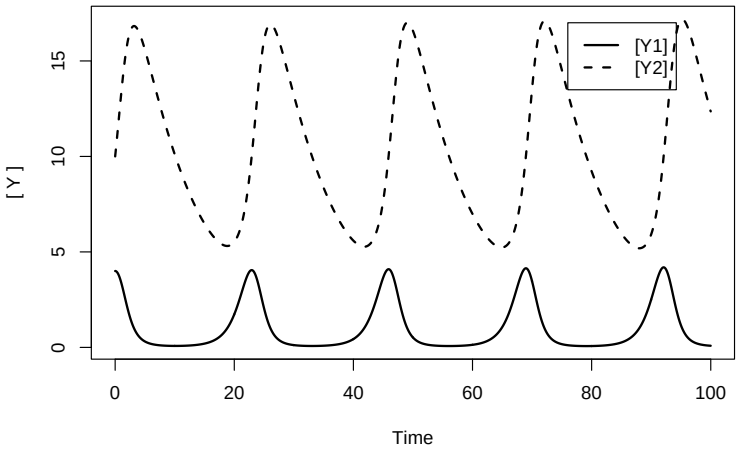
- u species: $\mathcal{X}_1, \ldots, \mathcal{X}_u$, and v reactions: $\mathcal{R}_1, \ldots, \mathcal{R}_v$
- $\mathcal{R}_i: \; p_{i1}\mathcal{X}_1 + \cdots + p_{iu}\mathcal{X}_u \longrightarrow q_{i1}\mathcal{X}_1 + \cdots + q_{iu}\mathcal{X}_u, \; i = 1, \ldots, v$
- In matrix form: $P\mathcal{X} \longrightarrow Q\mathcal{X}$ (P and Q are sparse)
- $S = (Q - P)'$ is the stoichiometry matrix of the system
- X_{jt}: # molecules of $\mathcal{X}_j$ at time t. $X_t = (X_{1t}, \ldots, X_{ut})'$
- Reaction $\mathcal{R}_i$ has hazard (or rate law, or propensity) $h_i(X_t, c_i)$, where c_i is a rate parameter, $c = (c_1, \ldots, c_v)'$, $h(X_t, c) = (h_1(X_t, c_1), \ldots, h_v(X_t, c_v))'$ and the system evolves as a Markov jump process
- For mass-action stochastic kinetics,

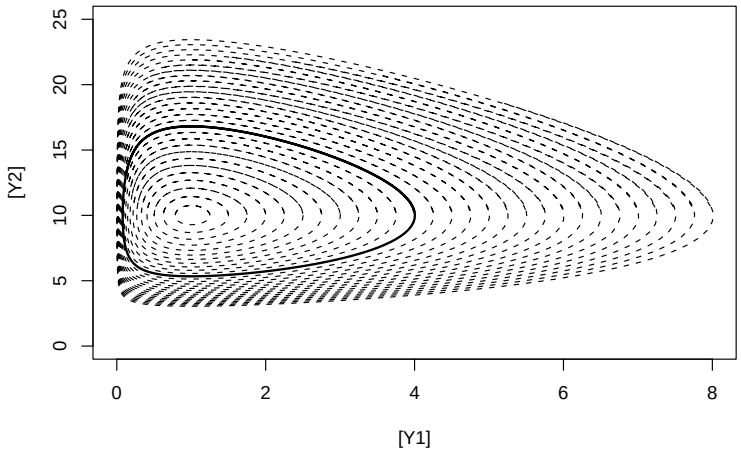
$$h_i(X_t, c_i) = c_i \prod_{j=1}^{u} \binom{X_{jt}}{p_{ij}}, \quad i = 1, \ldots, v$$

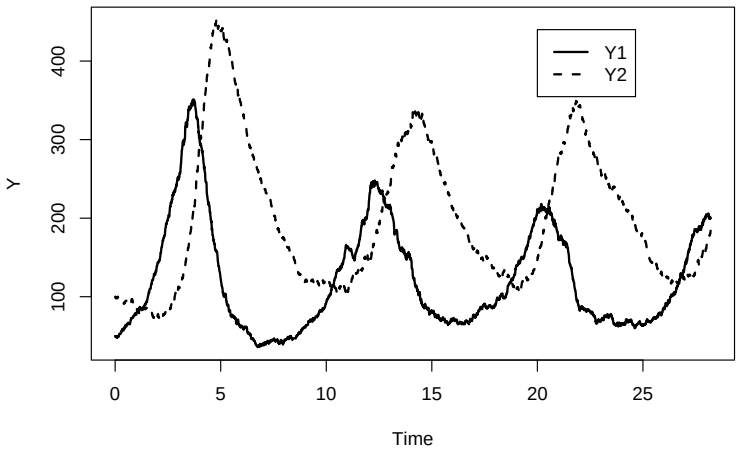
The Gillespie algorithm

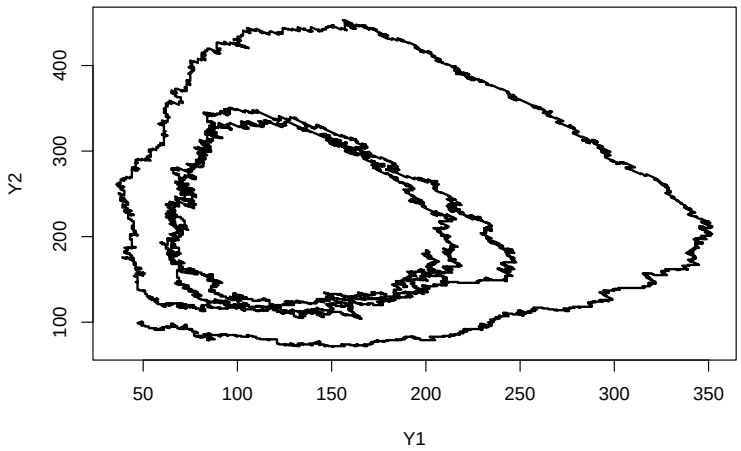
1. Initialise the system at $t = 0$ with rate constants $c_1, c_2, \ldots, c_v$ and initial numbers of molecules for each species, $x = (x_1, x_2, \ldots, x_u)'$.
2. For each $i = 1, 2, \ldots, v$, calculate $h_i(x, c_i)$ based on the current state, x.
3. Calculate $h_0(x, c) \equiv \sum_{i=1}^{v} h_i(x, c_i)$, the combined reaction hazard.
4. Simulate time to next event, τ, as an $Exp(h_0(x, c))$ random quantity, and put $t := t + \tau$.
5. Simulate the reaction index, j, as a discrete random quantity with probabilities $h_i(x, c_i)$ / $h_0(x, c)$, $i = 1, 2, \ldots, v$.
6. Update x according to reaction j. That is, put $x := x + S^{(j)}$, where $S^{(j)}$ denotes the jth column of the stoichiometry matrix S.
7. Output x and t.
8. If $t < T_{max}$, return to step 2.

The Lotka-Volterra model





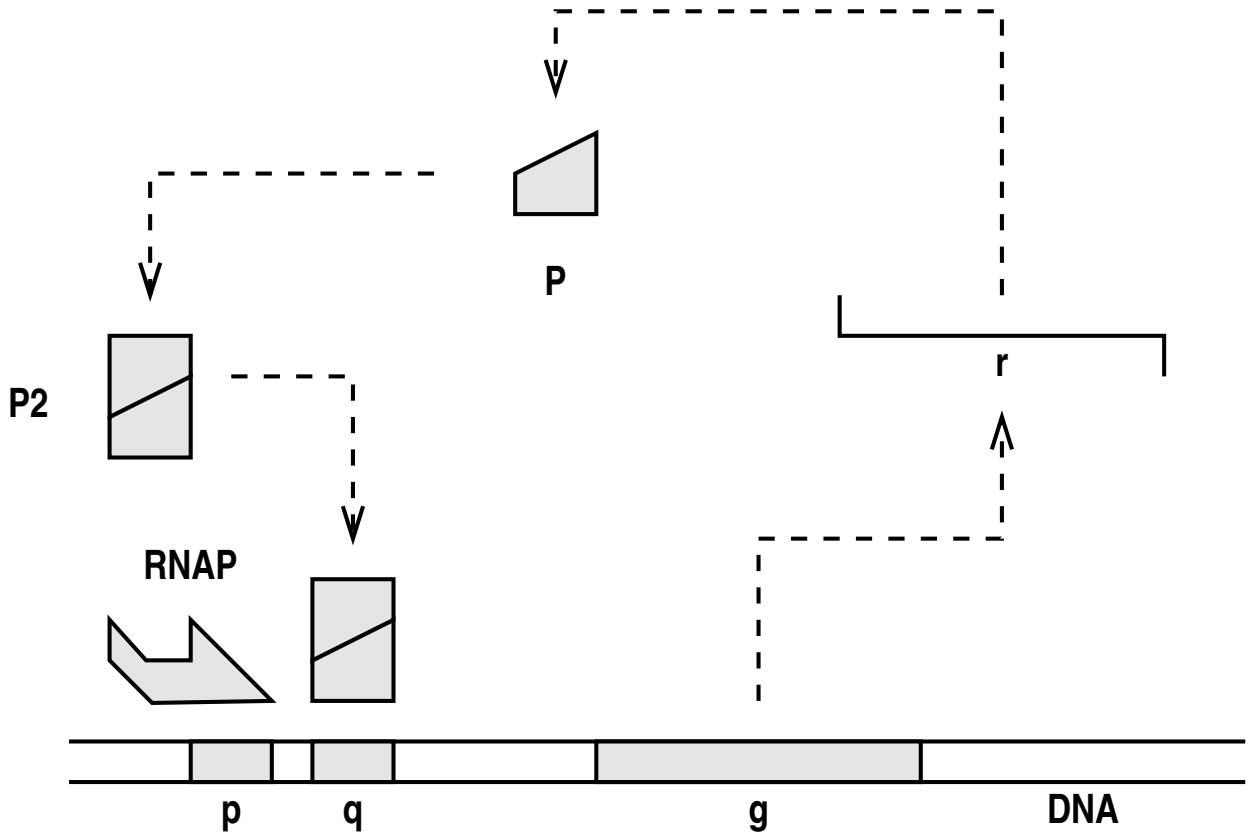




Key differences

- Deterministic solution is exactly periodic with perfectly repeating oscillations, carrying on indefinitely
- Stochastic solution oscillates, but in a random, unpredictable way (wandering from orbit to orbit in phase space)
- Stochastic solution will end in disaster! Either prey or predator numbers will hit zero...
- Either way, predators will end up extinct, so expected number of predators will tend to zero — qualitatively different to the deterministic solution
- So, in general the deterministic solution does not provide reliable information about either the stochastic process or its average behaviour

Example — genetic auto-regulation



Biochemical reactions

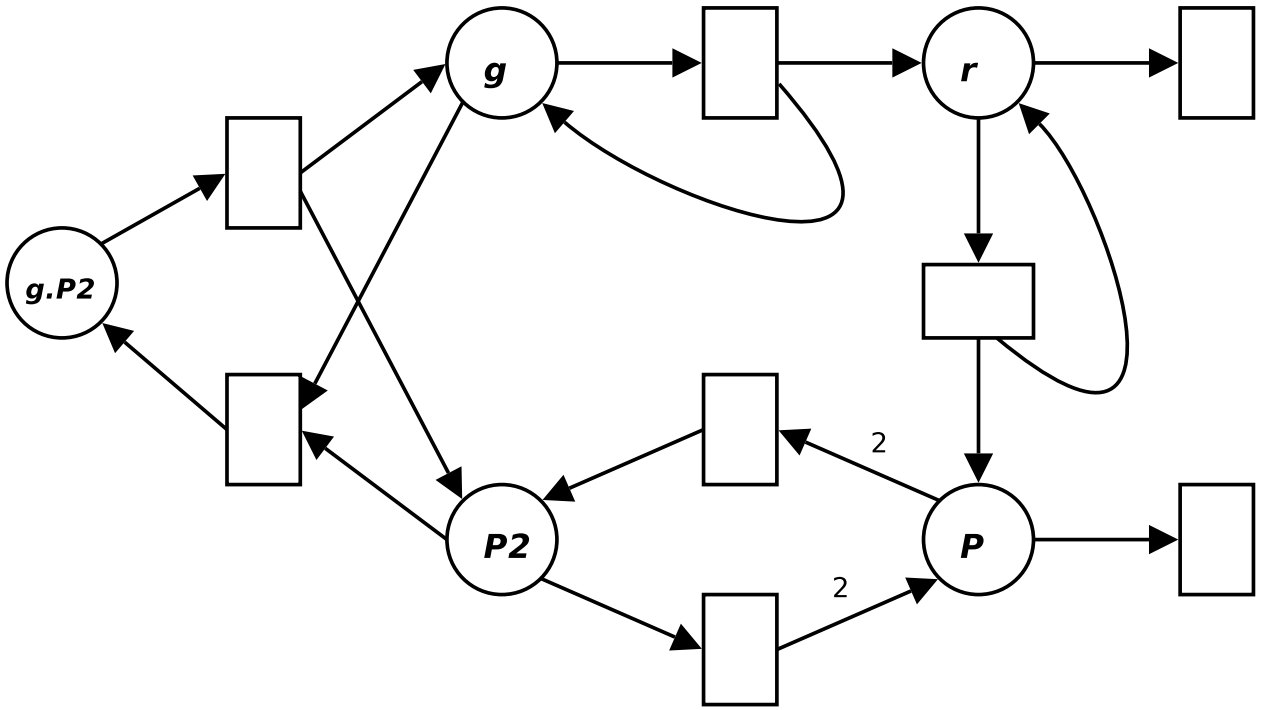
Simplified view:

Reactions

$g + P_2 \longleftrightarrow g \cdot P_2$	Repression
$g \longrightarrow g + r$	Transcription
$r \longrightarrow r + P$	Translation
$2P \longleftrightarrow P_2$	Dimerisation
$r \longrightarrow \emptyset$	mRNA degradation
$P \longrightarrow \emptyset$	Protein degradation

But these aren't as nice to look at as the picture...

Petri net representation



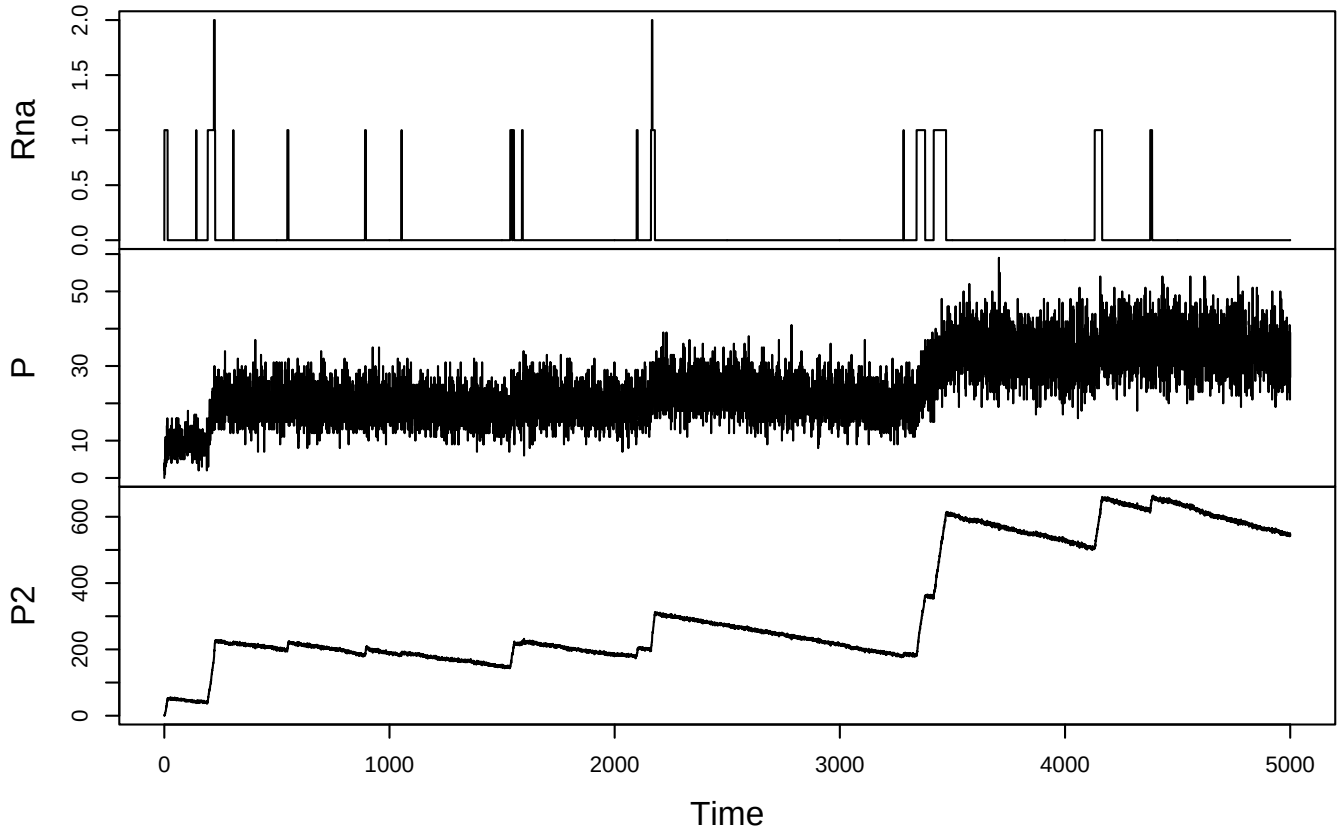
Simple bipartite digraph representation of the reaction network — useful both for visualisation and computational analysis

Matrix representation of the Petri net

	Reactants (*Pre*)					Products (*Post*)				
Species	$g \cdot P_2$	g	r	P	P_2	$g \cdot P_2$	g	r	P	P_2
Repression		1			1	1				
Reverse repression	1						1			1
Transcription		1					1	1		
Translation			1					1	1	
Dimerisation				2						1
Dissociation					1				2	
mRNA degradation			1							
Protein degradation				1						

Define $N = Post - Pre$ and $S = N'$, the stoichiometry matrix of the system

Simulated realisation of the auto-regulatory network



The chemical Langevin equation (CLE)

- The CLE is a diffusion approximation to the true Markov jump process
- Start with the time change representation

$$X_t - X_0 = S\,N\left(\int_0^t h(X_\tau, c)d\tau\right)$$

and approximate $N_i(t) \simeq t + W_i(t)$, where $W_i(t)$ is an independent Wiener process for each i

- Substituting in and using a little stochastic calculus gives:

The CLE as an Itô SDE:

$$dX_t = Sh(X_t, c)\,dt + \sqrt{S\,\mathrm{diag}\{h(X_t, c)\}S'}\,dW_t$$

Bayesian inference

Tuning model parameters so that output from the model "better matches" experimental data is a standard optimisation problem, but is problematic and unsatisfactory for a number of reasons:

- Defining an appropriate "objective function" is not straightforward if the model is stochastic or the measurement error has a complex structure (not IID Gaussian)
- The statistical concept of likelihood provides the "correct" way of measuring the evidence in favour of a set of model parameters, but typically requires computationally intensive Monte Carlo procedures for evaluation in complex settings
- Simple optimisation of the likelihood (the maximum likelihood approach) is also unsatisfactory, as there are typically many parameter combinations with very similar likelihoods (and the likelihood surface is typically multi-modal, making global optimisation difficult)

Markov chain Monte Carlo (MCMC)

- Additionally, likelihood ignores any existing information known about likely parameter values *a priori*, which can be very useful for regularising the inference problem — better to base inference on the posterior distribution
- MCMC algorithms can be used to explore plausible regions of parameter space in accordance with the posterior distribution — these provide rich information
- eg. rather than simple point estimates for parameter values, can get plausible ranges of values, together with information on parameter identifiability and confounding
- MCMC algorithms are computationally intensive, but given that evaluation of the likelihood is typically computationally intensive anyway, nothing to lose and everything to gain by doing a Bayesian analysis

Partially observed Markov process (POMP) models

- Continuous-time Markov process: $\mathbf{X} = \{X_s | s \geq 0\}$ (for now, we suppress dependence on parameters, θ)
- Think about integer time observations (extension to arbitrary times is trivial): for $t \in \mathbb{N}$, $\quad \mathbf{X}_t = \{X_s | t-1 < s \leq t\}$
- Sample-path likelihoods such as $\pi(\mathbf{x}_t | x_{t-1})$ can often (but not always) be computed (but are often computationally difficult), but discrete time transitions such as $\pi(x_t | x_{t-1})$ are typically intractable
- Partial observations: $\mathcal{D} = \{d_t | t = 1, 2, \ldots, T\}$ where

$$d_t | X_t = x_t \sim \pi(d_t | x_t), \qquad t = 1, \ldots, T,$$

where we assume that $\pi(d_t | x_t)$ can be evaluated directly (simple measurement error model)

The problem which pMCMC solves

- The particular pMCMC algorithm we will examine later requires only that we are able to forward-simulate trajectories from the model $\mathbf{x}_t|x_{t-1}$ (which must be Markovian) and evaluate the likelihood of data points $\pi(d_t|x_t)$ (conditional on a given set of model parameters, θ)
- The algorithm is exact in the sense that the equilibrium distribution of the Markov chain is the exact Bayesian posterior distribution $\pi(\theta, \mathbf{x}|\mathcal{D})$
  - No linearisations or Gaussian assumptions, and θ may contain parameters of the noise model
  - This algorithm simultaneously solves the (model, model error and noise) parameter estimation problem (computation of $\pi(\theta|\mathcal{D})$), the smoothing problem $\pi(\mathbf{x}|\mathcal{D})$, and the initial value problem $\pi(x_0|\mathcal{D})$
- The catch?! Need the model to be fast to simulate, and (for the simplest version), need measurement error

Bayesian inference for POMP models

- Most "obvious" MCMC algorithms will attempt to impute (at least) the skeleton of the Markov process: $X_0, X_1, \ldots, X_T$
- This will typically require evaluation of the intractable discrete time transition likelihoods, and this is the problem...
- Two related strategies:
  - Data augmentation: "fill in" the entire process in some way, typically exploiting the fact that the sample path likelihoods are tractable — works in principle, but difficult to "automate", and exceptionally computationally intensive due to the need to store and evaluate likelihoods of cts sample paths
  - Likelihood-free (AKA plug-and-play): exploits the fact that it is possible to forward simulate from $\pi(x_t|x_{t-1})$ (typically by simulating from $\pi(\mathbf{x}_t|x_{t-1})$), even if it can't be evaluated
- Likelihood-free is really just a special kind of augmentation strategy

Bayesian inference

- Let $\pi(\mathbf{x}|c)$ denote the (complex) likelihood of the simulation model
- Let $\pi(\mathcal{D}|\mathbf{x},\tau)$ denote the (simple) measurement error model
- Put $\theta = (c,\tau)$, and let $\pi(\theta)$ be the prior for the model parameters
- The joint density can be written

$$\pi(\theta,\mathbf{x},\mathcal{D}) = \pi(\theta)\pi(\mathbf{x}|\theta)\pi(\mathcal{D}|\mathbf{x},\theta).$$

- Interest is in the posterior distribution $\pi(\theta,\mathbf{x}|\mathcal{D})$

Marginal MH MCMC scheme

- Full model: $\pi(\theta, \mathbf{x}, \mathcal{D}) = \pi(\theta)\pi(\mathbf{x}|\theta)\pi(\mathcal{D}|\mathbf{x}, \theta)$
- Target: $\pi(\theta|\mathcal{D})$ (with $\mathbf{x}$ marginalised out)
- Generic MCMC scheme:
  - Propose $\theta^\star \sim f(\theta^\star|\theta)$
  - Accept with probability $\min\{1, A\}$, where

$$A = \frac{\pi(\theta^\star)}{\pi(\theta)} \times \frac{f(\theta|\theta^\star)}{f(\theta^\star|\theta)} \times \frac{\pi(\mathcal{D}|\theta^\star)}{\pi(\mathcal{D}|\theta)}$$

- $\pi(\mathcal{D}|\theta)$ is the "marginal likelihood" (or "observed data likelihood", or...)

LF-MCMC

- Posterior distribution $\pi(\theta, \mathbf{x}|\mathcal{D})$
- Propose a joint update for θ and $\mathbf{x}$ as follows:
  - Current state of the chain is $(\theta, \mathbf{x})$
  - First sample $\theta^\star \sim f(\theta^\star|\theta)$
  - Then sample a new path, $\mathbf{x}^\star \sim \pi(\mathbf{x}^\star|\theta^\star)$
  - Accept the pair $(\theta^\star, \mathbf{x}^\star)$ with probability $\min\{1, A\}$, where

$$A = \frac{\pi(\theta^\star)}{\pi(\theta)} \times \frac{f(\theta|\theta^\star)}{f(\theta^\star|\theta)} \times \frac{\pi(\mathcal{D}|\mathbf{x}^\star, \theta^\star)}{\pi(\mathcal{D}|\mathbf{x}, \theta)}.$$

- Note that choosing a prior independence proposal of the form $f(\theta^\star|\theta) = \pi(\theta^\star)$ leads to the simpler acceptance ratio

$$A = \frac{\pi(\mathcal{D}|\mathbf{x}^\star, \theta^\star)}{\pi(\mathcal{D}|\mathbf{x}, \theta)}$$

"Ideal" joint MCMC scheme

- LF-MCMC works by making the proposed sample path consistent with the proposed new parameters, but unfortunately not with the data
- Ideally, we would do the joint update as follows
  - First sample $\theta^\star \sim f(\theta^\star|\theta)$
  - Then sample a new path, $\mathbf{x}^\star \sim \pi(\mathbf{x}^\star|\theta^\star, \mathcal{D})$
  - Accept the pair $(\theta^\star, \mathbf{x}^\star)$ with probability $\min\{1, A\}$, where

$$A = \frac{\pi(\theta^\star)}{\pi(\theta)} \frac{\pi(\mathbf{x}^\star|\theta^\star)}{\pi(\mathbf{x}|\theta)} \frac{f(\theta|\theta^\star)}{f(\theta^\star|\theta)} \frac{\pi(\mathcal{D}|\mathbf{x}^\star, \theta^\star)}{\pi(\mathcal{D}|\mathbf{x}, \theta)} \frac{\pi(\mathbf{x}|\mathcal{D}, \theta)}{\pi(\mathbf{x}^\star|\mathcal{D}, \theta^\star)}$$
$$= \frac{\pi(\theta^\star)}{\pi(\theta)} \frac{\pi(\mathcal{D}|\theta^\star)}{\pi(\mathcal{D}|\theta)} \frac{f(\theta|\theta^\star)}{f(\theta^\star|\theta)}$$

  - This joint scheme reduces down to the marginal scheme (Chib (1995)), but will be intractable for complex models...

Particle MCMC (pMCMC)

- Of the various alternatives, pMCMC is the only obvious practical option for constructing global likelihood-free MCMC algorithms which are exact (Andreiu et al, 2010)
- Start by considering a basic marginal MH MCMC scheme with target $\pi(\theta|\mathcal{D})$ and proposal $f(\theta^\star|\theta)$ — the acceptance probability is $\min\{1, A\}$ where

$$A = \frac{\pi(\theta^\star)}{\pi(\theta)} \times \frac{f(\theta|\theta^\star)}{f(\theta^\star|\theta)} \times \frac{\pi(\mathcal{D}|\theta^\star)}{\pi(\mathcal{D}|\theta)}$$

- We can't evaluate the final terms, but if we had a way to construct a Monte Carlo estimate of the likelihood, $\hat{\pi}(\mathcal{D}|\theta)$, we could just plug this in and hope for the best:

$$A = \frac{\pi(\theta^\star)}{\pi(\theta)} \times \frac{f(\theta|\theta^\star)}{f(\theta^\star|\theta)} \times \frac{\hat{\pi}(\mathcal{D}|\theta^\star)}{\hat{\pi}(\mathcal{D}|\theta)}$$

"Exact approximate" MCMC (the pseudo-marginal approach)

- Remarkably, provided only that $\mathrm{E}[\hat{\pi}(\mathcal{D}|\theta)] = \pi(\mathcal{D}|\theta)$, the stationary distribution of the Markov chain will be exactly correct (Beaumont, 2003, Andreiu & Roberts, 2009)
- Putting $W = \hat{\pi}(\mathcal{D}|\theta)/\pi(\mathcal{D}|\theta)$ and augmenting the state space of the chain to include W, we find that the target of the chain must be

$$\propto \pi(\theta)\hat{\pi}(\mathcal{D}|\theta)\pi(w|\theta) \propto \pi(\theta|\mathcal{D})w\pi(w|\theta)$$

  and so then the above "unbiasedness" property implies that $\mathrm{E}(W|\theta) = 1$, which guarantees that the marginal for θ is exactly $\pi(\theta|\mathcal{D})$
- Blog post: http://tinyurl.com/6ex4xqw

Why the pseudo-marginal idea is important

- Many MCMC algorithms suffer from slow convergence due to high dependence between parameters, or between parameters and "missing data"
- In principle it is possible to greatly improve the rate of convergence by "integrating out" or marginalising over parameters not of direct inferential interest
- Marginalising over parameters or missing data leads to the occurrence of (often intractable) marginal likelihood terms in MCMC algorithms
- The pseudo-marginal approach gives the possibility of developing exact MCMC algorithms using approximate (unbiased, Monte Carlo) estimates of marginal likelihoods
- There are several well-known methods for constructing unbiased estimates of marginal likelihood

Particle marginal Metropolis-Hastings (PMMH)

- Likelihood estimates constructed via importance sampling typically have this "unbiasedness" property, as do estimates constructed using a particle filter
- If a particle filter is used to construct the Monte Carlo estimate of likelihood to plug in to the acceptance probability, we get (a simple version of) the particle Marginal Metropolis Hastings (PMMH) pMCMC algorithm
- The full PMMH algorithm also uses the particle filter to construct a proposal for $\mathbf{x}$, and has target $\pi(\theta, \mathbf{x}|\mathcal{D})$ — not just $\pi(\theta|\mathcal{D})$
- The (bootstrap) particle filter relies only on the ability to forward simulate from the process, and hence the entire procedure is "likelihood-free"

The bootstrap particle filter

- "Particle cloud": $\mathbf{x}_t = \{x_t^k | k = 1, \ldots, M\}$, $\boldsymbol{\pi}_t = \{\pi_t^k | k = 1, \ldots, M\}$, $\tilde{\mathbf{x}}_t = \{(x_t^k, \pi_t^k) | k = 1, \ldots, M\}$
- Initialise with $\tilde{\mathbf{x}}_0$, where $x_0^k \sim p(x_0)$ and $\pi_0^k = 1/M$ (note that w_0^k is undefined)
- Suppose at time t we have a sample from $p(x_t|y_{1:t})$: $\tilde{\mathbf{x}}_t$
  - Sample $a_t^k \sim \mathcal{F}(a_t^k|\boldsymbol{\pi}_t)$, $k = 1, \ldots, M$
  - Sample $x_{t+1}^k \sim p(x_{t+1}^k | x_t^{a_t^k})$
  - Set $w_{t+1}^k = p(y_{t+1}|x_{t+1}^k)$ and $\pi_{t+1}^k = w_{t+1}^k / \sum_{i=1}^M w_{t+1}^i$
  - Propagate $\tilde{\mathbf{x}}_{t+1}$ to the next step...

Define $\hat{p}(y_t|y_{1:t-1}) = \frac{1}{M}\sum_{i=1}^M w_t^i$ and $\hat{p}(y_{1:T}) = \prod_{i=1}^T \hat{p}(y_t|y_{1:t-1})$. Clear that $\hat{p}(y_{1:T})$ is a consistent estimator of $p(y_{1:T})$, but not obvious that it is in fact also unbiased (Pitt et al, 2011).

Blog post: http://bit.ly/jgo30P

Particles and genealogies

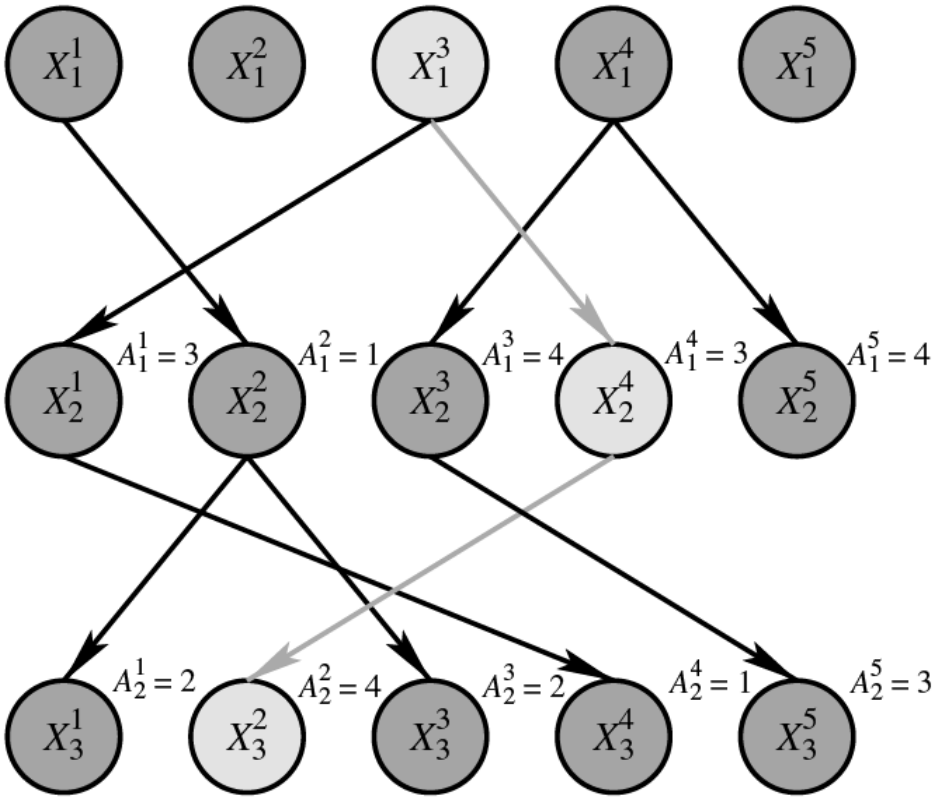


Fig. 1. Example of ancestral lineages generated by an SMC algorithm for $N = 5$ and $T = 3$: the lighter path is $X_{1:3}^2 = (X_1^3, X_2^4, X_3^2)$ and its ancestral lineage is $B_{1:3}^2 = (3,4,2)$

Joint density

It is important to know the full joint density of all random variables generated by the particle filter, which we write as

$$\tilde{q}(\mathbf{x}_0,\ldots,\mathbf{x}_T,\mathbf{a}_0,\ldots,\mathbf{a}_{T-1}) = \left[\prod_{k=1}^{M} p(x_0^k)\right]\left[\prod_{t=0}^{T-1}\prod_{k=1}^{M} \pi_t^{a_t^k} p(x_{t+1}^k|x_t^{a_t^k})\right]$$

on $X^{M(T+1)} \times \{1:M\}^{MT}$. For PMMH we also sample a final index k' from $\mathcal{F}(k'|\boldsymbol{\pi}_T)$ giving the joint density

$$\tilde{q}(\mathbf{x}_0,\ldots,\mathbf{x}_T,\mathbf{a}_0,\ldots,\mathbf{a}_{T-1})\pi_T^{k'}$$

on $X^{M(T+1)} \times \{1:M\}^{MT+1}$. We write the final selected trajectory as

$$x_{0:T}^{k'} = (x_0^{b_0^k},\ldots,x_T^{b_T^{k'}}),$$

where $b_t^{k'} = a_t^{b_{t+1}^{k'}}$, and $b_T^{k'} = k'$.

PMMH algorithm

- Propose $\theta^\star \sim q(\theta^\star|\theta)$
- Sample $\mathbf{x}^\star_{0:T}$ by running a bootstrap particle filter and picking a trajectory using the final set of weights, π_T
- Evaluate the estimate of marginal likelihood of the data, $\hat{p}_{\theta^\star}(y_{1:T})$
- Accept proposed new values with ratio

$$A = \frac{\hat{p}_{\theta^\star}(y_{1:T})p(\theta^\star)q(\theta|\theta^\star)}{\hat{p}_{\theta}(y_{1:T})p(\theta)q(\theta^\star|\theta)}$$

- On the fully augmented space $(\Theta \times X^{M(T+1)} \times \{1:M\}^{MT+1})$, the proposal is

$$q(\theta^\star|\theta)\tilde{q}_{\theta^\star}(\mathbf{x}^\star_0,\ldots,\mathbf{x}^\star_T,\mathbf{a}^\star_0,\ldots,\mathbf{a}^\star_{T-1})\pi_T^{k'^\star}$$

PMMH augmented target

- By considering the proposal and the acceptance ratio, it is clear that the augmented target must be proportional to

$$p(\theta)\hat{p}_\theta(y_{1:T})\tilde{q}_\theta(\mathbf{x}_0,\ldots,\mathbf{x}_T,\mathbf{a}_0,\ldots,\mathbf{a}_{T-1})\pi_T^{k'}$$

- We want to show that when we marginalise this down to the accepted trajectory, it is proportional to $p(\theta, x_{0:T}|y_{1:T})$
- Consider the terms in the joint distribution of the particle filter random variables

$$\tilde{q}_\theta(\mathbf{x}_0,\ldots,\mathbf{x}_T,\mathbf{a}_0,\ldots,\mathbf{a}_{T-1})\pi_T^{k'}$$

corresponding to the trajectory selected by final index k',

$$x_{0:T}^{k'} = (x_0^{b_0^{k'}},\ldots,x_T^{b_T^{k'}})$$

Likelihood of selected trajectory

- This is given by

$$p_\theta(x_0^{b_0^{k'}}) \left[\prod_{t=0}^{T-1} \pi_t^{b_t^{k'}} p_\theta(x_{t+1}^{b_{t+1}^{k'}} | x_t^{b_t^{k'}}) \right] \pi_T^{k'} = p_\theta(x_{0:T}^{k'}) \prod_{t=0}^{T} \pi_t^{b_t^{k'}}$$

- This probability simplifies to

$$\frac{p_\theta(x_{0:T}^{k'}) p_\theta(y_{1:T} | x_{0:T}^{k'})}{M^{T+1} \hat{p}_\theta(y_{1:T})}$$

- This is why PMMH works!

The selected trajectory and the target

- Consequently, the normalising constant cancels and the target is (proportional to)

$$\frac{p(\theta)p_\theta(x^{k'}_{0:T}, y_{1:T})}{M^{T+1}} \times (\text{Other terms...})$$

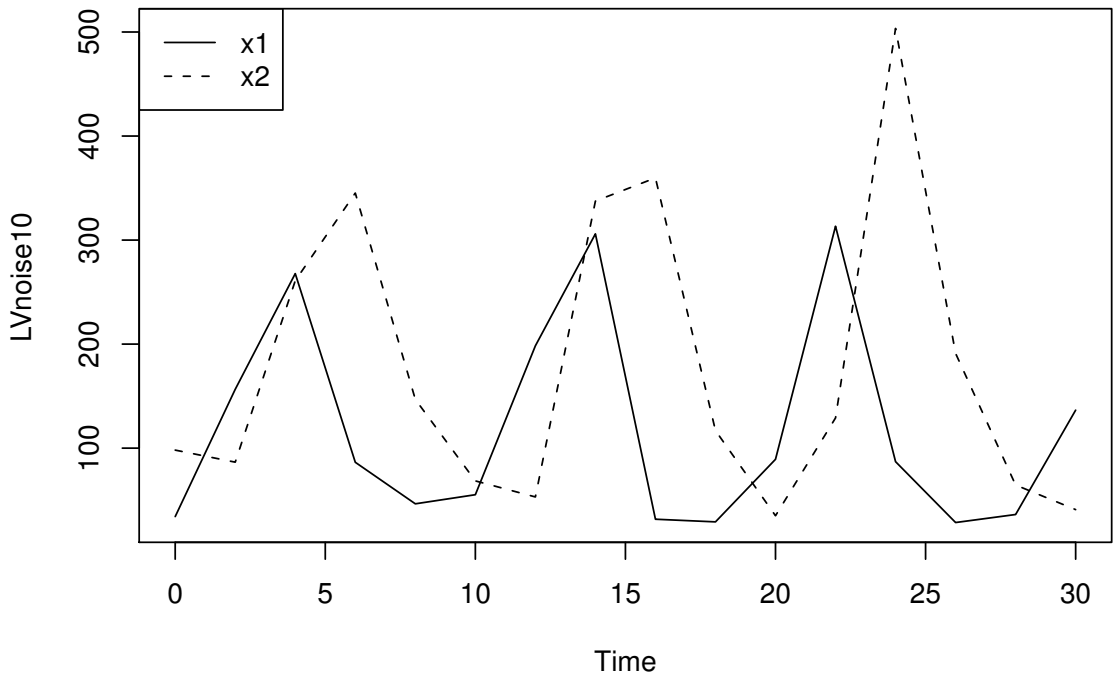
- The other terms are all probabilities of random variables which do not occur elsewhere in the target, and hence can all be marginalised away to leave the correct posterior

$$p(\theta, x_{0:T}|y_{1:T})$$

(also note the uniform distribution on the selected indices)

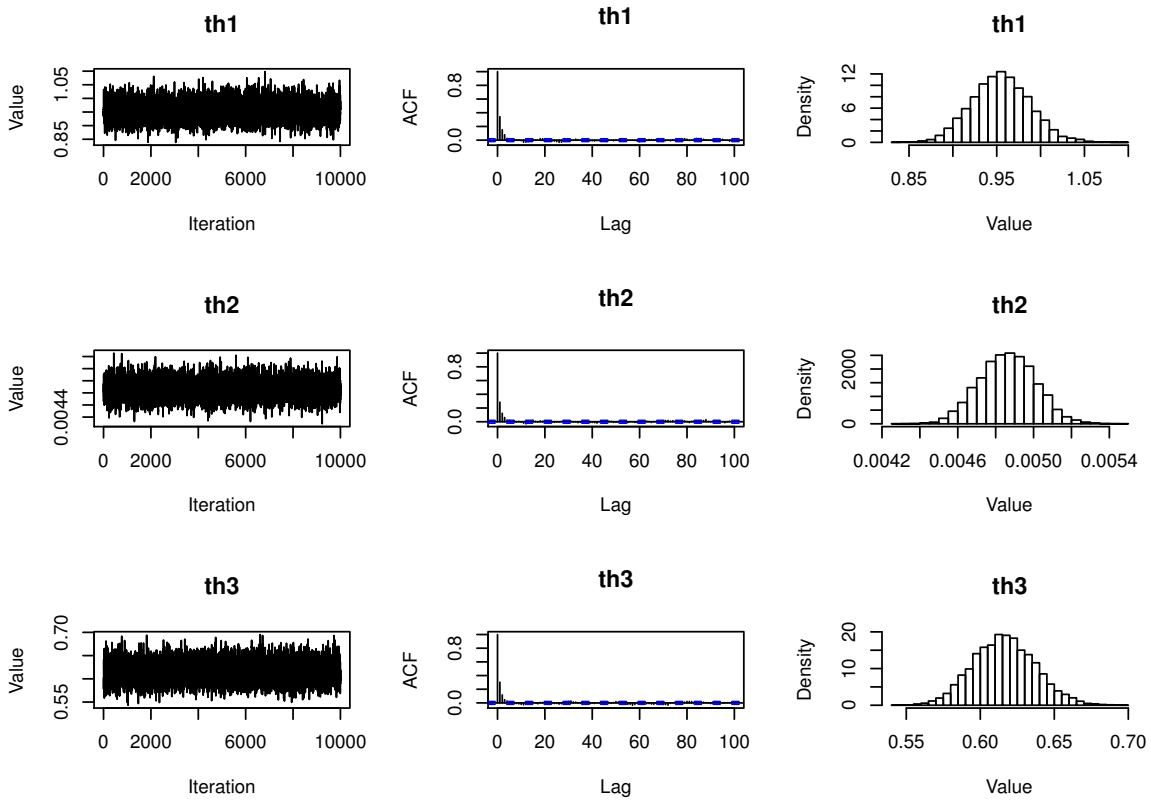
Blog post: http://bit.ly/kvznmq

Test problem: Lotka-Volterra model



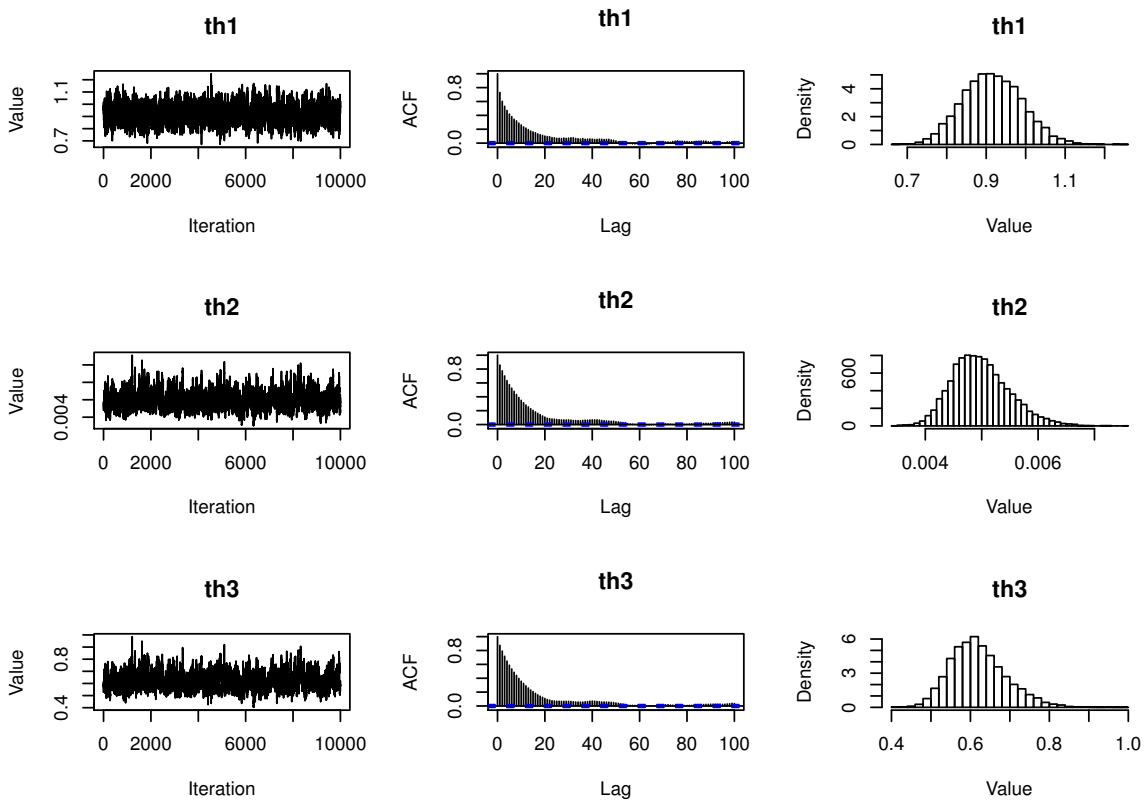
Simulated time series data set consisting of 16 equally spaced observations subject to Gaussian measurement error with a standard deviation of 10.

Marginal posteriors for the Lotka-Volterra model



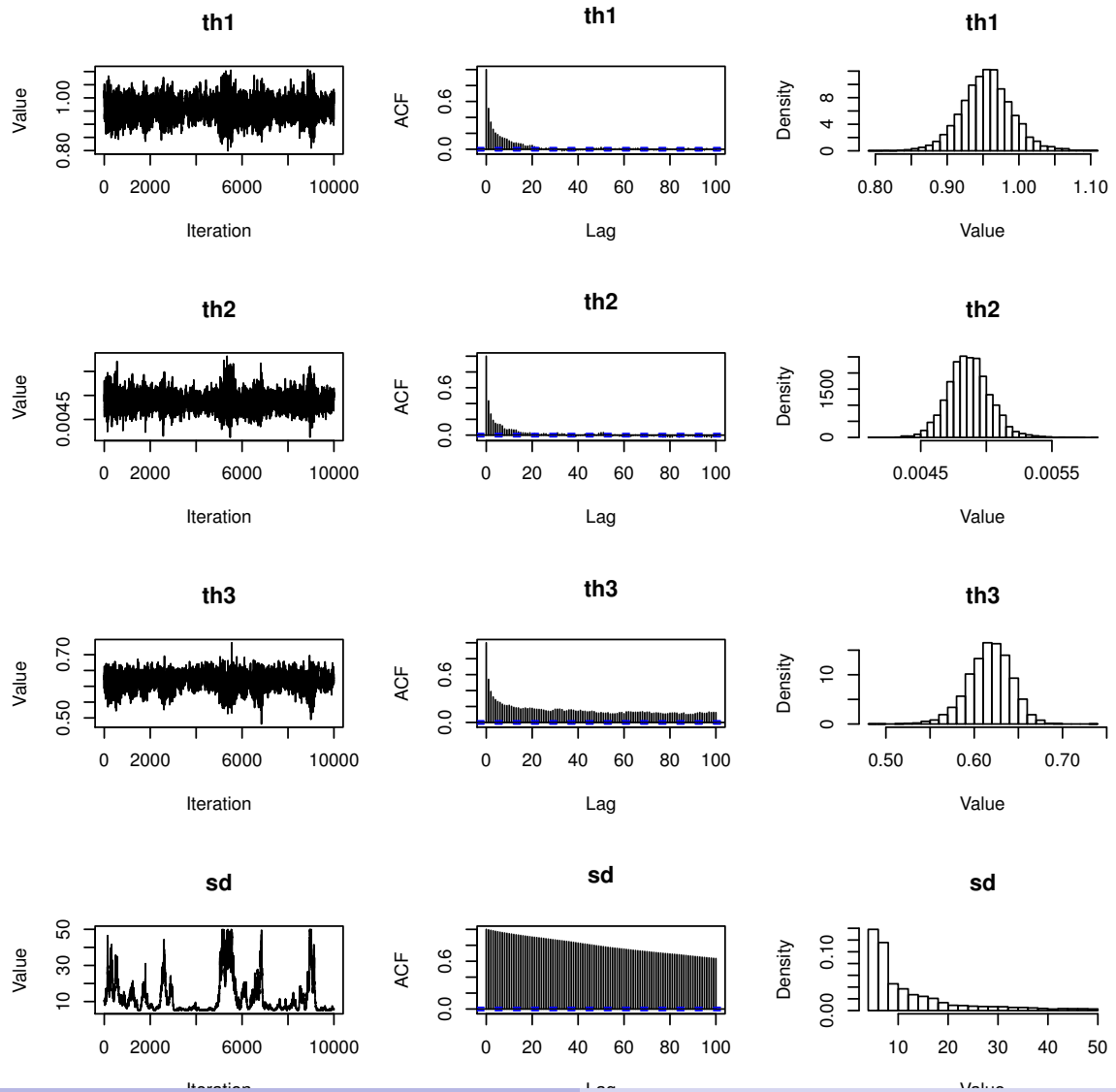
Note that the true parameters, $\theta = (1, 0.005, 0.6)$ are well identified by the data

Marginal posteriors observing only prey



Note that the mixing of the MCMC sampler is reasonable, and that the true parameters, $\theta = (1, 0.005, 0.6)$ are quite well identified by the data

Marginal posteriors for unknown measurement error



R package: smfsb

- Free, open source, well-documented software package for R, `smfsb`, associated with the forthcoming second edition of "Stochastic modelling for systems biology"
- Code for stochastic simulation and of (biochemical) reaction networks (Markov jump processes and chemical Langevin), and pMCMC-based Bayesian inference for POMP models
- Full installation and "getting started" instructions at `http://tinyurl.com/smfsb2e`
- Once the package is installed and loaded, running `demo("PMCMC")` at the R prompt will run a PMMH algorithm for the Lotka-Volterra model discussed here

Alternative: R package: pomp

- pomp is an R package for POMP models developed by King, Ionides, et al
- Provides a framework for inference based on "plug-and-play" (likelihood free) algorithms
- Primarily intended for iterative filtering (a plug-and-play SMC method for maximum likelihood estimation), but has other algorithms, including (a simple version of) PMMH, as part of the distribution
- Relatively easy to install and get working for simple Markov process models such as Immigration-Death and SIR

Hitting the data...

- The above algorithm works well in many cases, and is extremely general (works for any Markov process)
- In the case of no measurement error, the probability of hitting the data (and accepting the proposal) is very small (possibly zero), and so the mixing of the MCMC scheme is very poor
- ABC (approximate Bayesian computation) strategy is to accept if

$$\|x^{\star}_{t+1} - d_{t+1}\| < \varepsilon$$

  but this forces a trade-off between accuracy and efficiency which can be unpleasant (cf. noisy ABC)
- Same problem in the case of low measurement error
- Particularly problematic in the context of high-dimensional data
- Would like a strategy which copes better in this case

Improved particle filters for SDEs

- The "bootstrap" particle filter uses blind forward simulation from the model
- If we are able to evaluate the "likelihood" of sample paths, we can use other proposals
- The particle filter weights then depend on the Radon-Nikodym derivative of law of the proposed path wrt the true conditioned process
- For SDEs, the weight will degenerate unless the proposed process is absolutely continuous wrt the true conditioned process
- Ideally we would like to sample from $\pi(\mathbf{x}^\star_{t+1}|c^\star, x^\star_t, d_{t+1})$, but this is not tractable for nonlinear SDEs such as the CLE

Modified diffusion bridge (MDB)

- Need a tractable process $q(\mathbf{x}^\star_{t+1}|c^\star, x^\star_t, d_{t+1})$ that is locally equivalent to $\pi(\mathbf{x}^\star_{t+1}|c^\star, x^\star_t, d_{t+1})$
- Diffusion $dX_t = \mu(X_t)dt + \beta(X_t)^{\frac{1}{2}}dW_t$
- The nonlinear diffusion bridge

$$dX_t = \frac{x_1 - X_t}{1-t}dt + \beta(X_t)^{\frac{1}{2}}dW_t$$

  hits x_1 at $t = 1$, yet is locally equivalent to the true diffusion as it has the same diffusion coefficient
- This forms the basis of an efficient proposal; see Durham & Gallant (2002), Chib, Pitt & Shephard (2004), Delyon & Hu (2006), and Stramer & Yan (2007) for technical details

Summary

- POMP models form a large, important and interesting class of models, with many applications
- It is possible, and often desirable, to develop inferential algorithms which are "likelihood free" or "plug-and-play", as this allows the separation of the modelling from the inferential algorithm, allowing more rapid model exploration
- Many likelihood free approaches are possible, including sequential LF-MCMC, PMMH (pMCMC), (sequential) ABC for Bayesian inference and iterative filtering for maximum likelihood estimation
- Much work needs to be done to properly understand the strengths and weaknesses of these competing approaches


Golightly, A. and D. J. Wilkinson (2008) Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics and Data Analysis,* **52**(3), 1674–1693.


Henderson, D. A., Boys, R. J., Krishnan, K. J., Lawless, C., Wilkinson, D. J. (2009) Bayesian emulation and calibration of a stochastic computer model of mitochondrial DNA deletions in substantia nigra neurons, *Journal of the American Statistical Association.* **104**(485):76-87.


Wilkinson, D. J. (2009) Stochastic modelling for quantitative description of heterogeneous biological systems, *Nature Reviews Genetics.* **10**(2):122-133.


Wilkinson, D. J. (2010) Parameter inference for stochastic kinetic models of bacterial gene regulation: a Bayesian approach to systems biology (with discussion), in J.-M. Bernardo et al (eds) *Bayesian Statistics 9,* OUP, in press.


Wilkinson, D. J. (2006) *Stochastic Modelling for Systems Biology*. Chapman & Hall/CRC Press. (second edition in press)

Contact details...

email: darren.wilkinson@ncl.ac.uk

www: tinyurl.com/darrenjw