Inference for SDE models via Approximate Bayesian Computation

Umberto Picchini
Centre for Mathematical Sciences,
Lund University, Sweden

PEDS-II, Eindhoven, 4 June 2012

Motivations

- The study of complex systems demands for multidimensional models, sometimes defined on different scales, with several unknown parameters to be estimated from data.
- Systems are often "partially observed" $\rightarrow$ latent variables;
- Measurement error is often a non-negligible source of variability affecting data;
- Dynamical systems might evolve randomly $\rightarrow$ stochastic differential equations (SDEs);
- Systems biology applications are examples of problems involving the "complications" above.

Goal of this talk

- Inference for stochastic systems modelled by SDEs observed at discrete time points is complicated and has generated a large amount of literature in the past 20 years.
- the exploitation of MCMC methodology produced in the '90s has broken several barriers and allowed exploration of complex inferential problems like never before;
- the availability of more affordable computational resources has made the application of intense computational algorithms more viable.
- however MCMC Bayesian methodology (e.g. Metropolis-Hastings) does not scale well for large systems, i.e. the resulting chain might have poor mixing, resulting in unacceptably long computational times.

Goal is to consider Approximate Bayesian Computation to alleviate inferential problems in relatively "large"/complex SDE models.

We are interested in estimating parameters from observation $\{y_i\}_{i=1,\dots,n}$ generated from the following state-space model:

$$\begin{cases} dX_t = \mu(X_t, t, \psi)dt + \sigma(X_t, t, \psi)dW_t \\ Y_{t_i} = f(X_{t_i}, \varepsilon_{t_i}), \quad \varepsilon_{t_i} \sim N(0, \sigma_\varepsilon^2 I_d), \qquad i = 1, .., n \end{cases}$$

- X_{t_i} may be multidimensional $\in \mathbb{R}^d$ and partially observed;
- the ε_i are to be interpreted as i.i.d measurement error terms independent of W_t.

And w.l.g we assume:

- $y_i = X_{t_i} + \varepsilon_{t_i}, \qquad i = 1, ..., n$

Our goal is to estimate $\theta = (\psi, \sigma_\varepsilon)$ conditionally on $y = \{y_0, y_1, ..., y_n\}$ using Bayesian methodology.

Bayesian inference about θ is based on the (marginal) posterior density

$$\pi(\theta \mid y) \propto l(\theta; y)\pi(\theta)$$

We assume that the likelihood function $l(\theta; y)$ may be unavailable for mathematical reasons (not available in closed from) or for computational reasons (too expensive to calculate).

In our case

- data: $y = (y_0, y_1, ..., y_n)$ obtained at times $\{t_0, ..., t_n\}$ i.e. $y_i \equiv y_{t_i}$
- and $x = (x_0, x_1, ..., x_n)$ is the SDE solution at time-points $\{t_0, ..., t_n\}$.

$$l(\theta; y) = \pi(y \mid \theta) = \int \pi(y \mid x; \theta)\pi(x \mid \theta)dx \rightarrow \text{“data augmentation”}$$

from θ to (x, θ)

- use MCMC to deal with the multiple integration problem.
- Because of increased dimension (from θ to (θ, x)) convergence properties of the MCMC algorithms are too poor for the algorithm to be considered. [Marin, Pudlo, Robert and Ryder 2011]

Bayesian inference about θ is based on the (marginal) posterior density

$$\pi(\theta \mid y) \propto l(\theta; y)\pi(\theta)$$

We assume that the likelihood function $l(\theta; y)$ may be unavailable for mathematical reasons (not available in closed from) or for computational reasons (too expensive to calculate).
In our case

- data: $y = (y_0, y_1, ..., y_n)$ obtained at times $\{t_0, ..., t_n\}$ i.e. $y_i \equiv y_{t_i}$
- and $x = (x_0, x_1, ..., x_n)$ is the SDE solution at time-points $\{t_0, ..., t_n\}$.

$$l(\theta; y) = \pi(y \mid \theta) = \int \pi(y \mid x; \theta)\pi(x \mid \theta)dx \rightarrow \text{“data augmentation”}$$

from θ to (x, θ)

- use MCMC to deal with the multiple integration problem.
- Because of increased dimension (from θ to (θ, x)) convergence properties of the MCMC algorithms are too poor for the algorithm to be considered. [Marin, Pudlo, Robert and Ryder 2011]

Approximate Bayesian Computation (ABC, first proposed in Tavaré et al., 1997) bypass the computation of the likelihood function.

Basic ABC idea (no measurement error here):
for an observation $x_{\text{obs}} \sim \pi(x \mid \theta)$, under the prior $\pi(\theta)$, keep jointly simulating

$$\theta' \sim \pi(\theta), \quad x' \sim \pi(x \mid \theta')$$

until the simulated variable x' is equal to the observed one, $x' = x_{\text{obs}}$.
Then a θ' satisfying such condition is such that $\theta' \sim \pi(\theta \mid x_{\text{obs}})$.
[Tavaré et al., 1997]. ▸ proof

In general equality $x = x_{\text{obs}}$ is often hardly verified, and it's even more unlikely for vectors $x = (x_{t_0}, ..., x_{t_n})$ resulting from a diffusion process.

- Choose instead an accuracy threshold δ;
- substitute "accept if $x' = x_{\text{obs}}$" with "accept if $\rho(S(x'), S(x_{\text{obs}})) < \delta$" for some measure $\rho(\cdot)$ and statistics $S(\cdot)$.

Acceptance probability in Metropolis-Hastings

Suppose at a given iteration of Metropolis-Hastings we are in the (augmented)-state position $(\theta_\#, x_\#)$ and wonder whether to move (or not) to a new state (θ', x'). The move is generated via a proposal distribution "$q((\theta_\#, x_\#) \to (x', \theta'))$".

- e.g. "$q((\theta_\#, x_\#) \to (x', \theta'))$" $= u(\theta'|\theta_\#)v(x' \mid \theta')$;
- move "$(\theta^\#, x^\#) \to (\theta', x')$" accepted with probability

$$\alpha_{(\theta_\#, x_\#) \to (x', \theta')} = \min\left(1, \frac{\pi(\theta')\pi(x'|\theta')\pi(y|x', \theta')q((\theta', x') \to (\theta_\#, x_\#))}{\pi(\theta_\#)\pi(x_\#|\theta_\#)\pi(y|x_\#, \theta_\#)q((\theta_\#, x_\#) \to (\theta', x'))}\right)$$

$$= \min\left(1, \frac{\pi(\theta')\pi(x'|\theta')\pi(y|x', \theta')u(\theta_\#|\theta')v(x_\# \mid \theta_\#)}{\pi(\theta_\#)\pi(x_\#|\theta_\#)\pi(y|x_\#, \theta_\#)u(\theta'|\theta_\#)v(x' \mid \theta')}\right)$$

now choose $v(x \mid \theta) \equiv \pi(x \mid \theta)$

$$= \min\left(1, \frac{\pi(\theta')\cancel{\pi(x'|\theta')}\pi(y|x', \theta')u(\theta_\#|\theta')\cancel{\pi(x_\# \mid \theta_\#)}}{\pi(\theta_\#)\cancel{\pi(x_\#|\theta_\#)}\pi(y|x_\#, \theta_\#)u(\theta'|\theta_\#)\cancel{\pi(x' \mid \theta')}}\right)$$

This is likelihood–free! And we only need to know *how* to generate x'

Acceptance probability in Metropolis-Hastings

Suppose at a given iteration of Metropolis-Hastings we are in the (augmented)-state position $(\theta_\#, x_\#)$ and wonder whether to move (or not) to a new state (θ', x'). The move is generated via a proposal distribution "$q((\theta_\#, x_\#) \to (x', \theta'))$".

- e.g. "$q((\theta_\#, x_\#) \to (x', \theta'))$" $= u(\theta'|\theta_\#)v(x' \mid \theta')$;
- move "$(\theta^\#, x^\#) \to (\theta', x')$" accepted with probability

$$\alpha_{(\theta_\#, x_\#) \to (x', \theta')} = \min\left(1, \frac{\pi(\theta')\pi(x'|\theta')\pi(y|x', \theta')q((\theta', x') \to (\theta_\#, x_\#))}{\pi(\theta_\#)\pi(x_\#|\theta_\#)\pi(y|x_\#, \theta_\#)q((\theta_\#, x_\#) \to (\theta', x'))}\right)$$

$$= \min\left(1, \frac{\pi(\theta')\pi(x'|\theta')\pi(y|x', \theta')u(\theta_\#|\theta')v(x_\# \mid \theta_\#)}{\pi(\theta_\#)\pi(x_\#|\theta_\#)\pi(y|x_\#, \theta_\#)u(\theta'|\theta_\#)v(x' \mid \theta')}\right)$$

now choose $v(x \mid \theta) \equiv \pi(x \mid \theta)$

$$= \min\left(1, \frac{\pi(\theta')\cancel{\pi(x'|\theta')}\pi(y|x', \theta')u(\theta_\#|\theta')\cancel{\pi(x_\# \mid \theta_\#)}}{\pi(\theta_\#)\cancel{\pi(x_\#|\theta_\#)}\pi(y|x_\#, \theta_\#)u(\theta'|\theta_\#)\cancel{\pi(x' \mid \theta')}}\right)$$

This is likelihood–free! And we only need to know *how* to generate x'

Acceptance probability in Metropolis-Hastings

Suppose at a given iteration of Metropolis-Hastings we are in the (augmented)-state position $(\theta_\#, x_\#)$ and wonder whether to move (or not) to a new state (θ', x'). The move is generated via a proposal distribution "$q((\theta_\#, x_\#) \to (x', \theta'))$".

- e.g. "$q((\theta_\#, x_\#) \to (x', \theta'))$" $= u(\theta'|\theta_\#)v(x' \mid \theta')$;
- move "$(\theta^\#, x^\#) \to (\theta', x')$" accepted with probability

$$\alpha_{(\theta_\#, x_\#) \to (x', \theta')} = \min\left(1, \frac{\pi(\theta')\pi(x'|\theta')\pi(y|x', \theta')q((\theta', x') \to (\theta_\#, x_\#))}{\pi(\theta_\#)\pi(x_\#|\theta_\#)\pi(y|x_\#, \theta_\#)q((\theta_\#, x_\#) \to (\theta', x'))}\right)$$

$$= \min\left(1, \frac{\pi(\theta')\pi(x'|\theta')\pi(y|x', \theta')u(\theta_\#|\theta')v(x_\# \mid \theta_\#)}{\pi(\theta_\#)\pi(x_\#|\theta_\#)\pi(y|x_\#, \theta_\#)u(\theta'|\theta_\#)v(x' \mid \theta')}\right)$$

now choose $v(x \mid \theta) \equiv \pi(x \mid \theta)$

$$= \min\left(1, \frac{\pi(\theta')\cancel{\pi(x'|\theta')}\pi(y|x', \theta')u(\theta_\#|\theta')\cancel{\pi(x_\# \mid \theta_\#)}}{\pi(\theta_\#)\cancel{\pi(x_\#|\theta_\#)}\pi(y|x_\#, \theta_\#)u(\theta'|\theta_\#)\cancel{\pi(x' \mid \theta')}}\right)$$

This is likelihood–free! And we only need to know *how* to generate x'

As from the previous slide, for dynamical problems we only need to know how to generate a proposal $x' = (x'_0, ..., x'_n) \Rightarrow$ for SDEs: Euler-Maruyama, Milstein, Stochastic RK etc.

We now plug the previous ideas in a MCMC ABC algorithm [Sisson-Fan, 2011]

1. Choose or simulate $\theta_{start} \sim \pi(\theta)$ and $x_{start} \sim \overbrace{\pi(x|\theta_{start})}^{\text{it can be unknown!}}$. Fix $\delta > 0$ and $r = 0$. Put $\theta_r = \theta_{start}$ and (supposedly known!) statistics $S(x_r) \equiv S(x_{start})$.

At $(r+1)$th MCMC iteration:

2. generate $\theta' \sim u(\theta'|\theta_r)$ from its proposal distribution;
3. generate $x' \sim \pi(x'|\theta')$ and calculate $S(x')$;
4. with probability $\min\left(1, \frac{\pi(\theta')\pi_\delta(y|x',\theta')u(\theta_r|\theta')}{\pi(\theta_r)\pi_\delta(y|x_r,\theta_r)u(\theta'|\theta_r)}\right)$ set $(\theta_{r+1}, S(x_{r+1})) = (\theta', S(x'))$ otherwise set $(\theta_{r+1}, S(x_{r+1})) = (\theta_r, S(x_r))$;
5. increment r to $r+1$ and go to step 2.

with e.g. $\pi_\delta(y \mid x, \theta) = \frac{1}{\delta} K\left(\frac{|S(x) - S(y)|}{\delta}\right)$ [Bloom 2010].

As from the previous slide, for dynamical problems we only need to know how to generate a proposal $x' = (x'_0, ..., x'_n) \Rightarrow$ for SDEs: Euler-Maruyama, Milstein, Stochastic RK etc.
We now plug the previous ideas in a MCMC ABC algorithm [Sisson-Fan, 2011]

1. Choose or simulate $\theta_{start} \sim \pi(\theta)$ and $x_{start} \sim \overbrace{\pi(x|\theta_{start})}^{\text{it can be unknown!}}$. Fix $\delta > 0$ and $r = 0$. Put $\theta_r = \theta_{start}$ and (supposedly known!) statistics $S(x_r) \equiv S(x_{start})$.

At $(r+1)$th MCMC iteration:

2. generate $\theta' \sim u(\theta'|\theta_r)$ from its proposal distribution;
3. generate $x' \sim \pi(x'|\theta')$ and calculate $S(x')$;
4. with probability $\min\left(1, \frac{\pi(\theta')\pi_\delta(y|x',\theta')u(\theta_r|\theta')}{\pi(\theta_r)\pi_\delta(y|x_r,\theta_r)u(\theta'|\theta_r)}\right)$ set $(\theta_{r+1}, S(x_{r+1})) = (\theta', S(x'))$ otherwise set $(\theta_{r+1}, S(x_{r+1})) = (\theta_r, S(x_r))$;
5. increment r to $r+1$ and go to step 2.

with e.g. $\pi_\delta(y \mid x, \theta) = \frac{1}{\delta} K\left(\frac{|S(x) - S(y)|}{\delta}\right)$ [Bloom 2010].

The previous algorithm generates a sequence $\{\theta_r, x_r\}$ from the (ABC) posterior $\pi_\delta(\theta, x_r \mid y)$. We only retain the $\{\theta_r\}$ which are thus from $\pi_\delta(\theta \mid y)$, the (marginal) ABC posterior of θ.

In order to apply ABC methodology we are required to specify the statistic $S(\cdot)$ for θ.

[Fearnhead & Prangle]

(Classic result of Bayesian stats: for quadratic losses) the "optimal" choice of $S(y)$ is given by $S(y) = E(\theta \mid y)$, the (unknown) posterior of θ.

So Fearnhead & Prangle propose a regression-based approach to determine $S(\cdot)$ (prior to MCMC start):

- for the jth parameter in θ fit the following linear regression models

$$S_j(y) = \hat{E}(\theta_j|y) = \hat{\beta}_0^{(j)} + \hat{\beta}^{(j)}\eta(y), \qquad j = 1, 2, ..., dim(\theta)$$

- repeat the fitting separately for each θ_j.
- hopefully $S_j(y) = \hat{\beta}_0^{(j)} + \hat{\beta}^{(j)}\eta(y)$ will be "almost sufficient" for θ_j;

An example (run prior to MCMC start):

1. simulate from the prior $\theta' \sim \pi(\theta)$ (very inefficient!)
2. generate simulated data y_{sim} via $x_{sim} \sim \pi(x \mid \theta')$, $y_{sim} = x_{sim} + \varepsilon$

repeat (1)-(2) many times to get the following matrices:

$$\begin{bmatrix} \theta_1^{(1)} & \theta_2^{(1)} & \cdots \theta_p^{(1)} \\ \theta_1^{(2)} & \theta_2^{(2)} & \cdots \theta_p^{(2)} \\ \vdots & & \end{bmatrix}, \qquad \begin{bmatrix} y_{sim,t_0}^{(1)} & y_{sim,t_1}^{(1)} & \cdots & y_{sim,t_n}^{(1)} \\ y_{sim,t_0}^{(2)} & y_{sim,t_1}^{(2)} & \cdots & y_{sim,t_n}^{(2)} \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix}$$

and for each column of the left matrix do a multivariate linear regression (or Lasso, or MARS,...)

$$\begin{bmatrix} \theta_j^{(1)} \\ \theta_j^{(2)} \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & y_{sim,t_0}^{(1)} & y_{sim,t_1}^{(1)} & \cdots & y_{sim,t_n}^{(1)} \\ 1 & y_{sim,t_0}^{(2)} & y_{sim,t_1}^{(2)} & \cdots & y_{sim,t_n}^{(2)} \\ \vdots & \vdots & \cdots & \vdots & \end{bmatrix} \times \beta_j \qquad (j = 1, ..., p),$$

and obtain an (hopefully almost sufficient!) statistic for θ_j

$S_j(y) = \hat{\beta}_0^{(j)} + \hat{\beta}^{(j)} \eta(y) = \hat{\beta}_0^{(j)} + \hat{\beta}_1^{(j)} y_0 + \cdots + \hat{\beta}_n^{(j)} y_n \Rightarrow$ plug this into MCMC alg.

We have also obtained much better results (than with linear regression) using statistics based on:

- regression via MARS (multivariate adaptive regression splines [Friedman 1991]);
- Lasso-like estimation via `glmnet` [Friedman, Hastie, Tibshirani 2001]: estimates are found via

$$\hat{\beta} = \text{argmin}_{\beta} \left(\frac{1}{2} \sum_{i=0}^{n} (y_i - \sum_{j=1}^{p} \eta_{ij} \beta_j)^2 + \gamma \sum_{j=1}^{p} \mid \beta_j \mid \right)$$

  and the penality γ selected via cross validation methods (the one giving the smallest mean prediction error is selected).

An application: Stochastic kinetic networks

[also in Golightly-Wilkinson, 2010]

Consider a set of reactions $\{R_1, R_2, ..., R_8\}$:

$R_1 : DNA + P_2 \rightarrow DNA \cdot P_2$ $\quad$ $R_2 : DNA \cdot P_2 \rightarrow DNA + P_2$

$R_3 : DNA \rightarrow DNA + RNA$ $\quad$ $R_4 : RNA \rightarrow RNA + P$

$R_5 : 2P \rightarrow P_2$ (dimerization) $\quad$ $R_6 : P_2 \rightarrow 2P$ (dimerization)

$R_7 : RNA \rightarrow \emptyset$ (degradation) $\quad$ $R_8 : P \rightarrow \emptyset$ (degradation)

These reactions represent a simplified model for prokaryotic auto-regulation.
"Reactants" are on the left side of $\rightarrow$;
"Products" are on the right side of $\rightarrow$;
There are several ways to simulate biochemical networks, e.g. the "exact" Gillespie algorithm (computationally intense for inferential purposes).
Or by using a diffusion approximation $\Rightarrow$ SDE.

An application: Stochastic kinetic networks

[also in Golightly-Wilkinson, 2010]

Consider a set of reactions $\{R_1, R_2, ..., R_8\}$:

$R_1 : DNA + P_2 \to DNA \cdot P_2$ $\quad$ $R_2 : DNA \cdot P_2 \to DNA + P_2$

$R_3 : DNA \to DNA + RNA$ $\quad$ $R_4 : RNA \to RNA + P$

$R_5 : 2P \to P_2$ (dimerization) $\quad$ $R_6 : P_2 \to 2P$ (dimerization)

$R_7 : RNA \to \emptyset$ (degradation) $\quad$ $R_8 : P \to \emptyset$ (degradation)

These reactions represent a simplified model for prokaryotic auto-regulation.
"Reactants" are on the left side of $\to$;
"Products" are on the right side of $\to$;
There are several ways to simulate biochemical networks, e.g. the "exact" Gillespie algorithm (computationally intense for inferential purposes).
Or by using a diffusion approximation $\Rightarrow$ SDE.

Each one of the 8 reactions $\{R_1, R_2, ..., R_8\}$ has an associated rate constant c_i and a "propensity function" $h_i(X_t, c_i)$. ▸ uh?!

We want to consider the evolution of the "species"
$X_t = (RNA_t, P_t, P_{2,t}, DNA_t)$ each representing # of molecules (integers!).

A *continuous*–time approximation for the (innerly discrete) X_t process is given by:

$$dX_t = Sh(X_t, c)dt + \sqrt{S\text{diag}(h(X_t, c))S^T}dW_t, \qquad X_t \in \mathbb{R}^4_+$$

This is the *Chemical Langevin equation*.

$$S = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 2 & 0 & -1 \\ -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and hazard function

$$h(X_t, c) = (c_1 DNA_t \times P_{2,t}, c_2(k - DNA_t), c_3 DNA_t, c_4 RNA_t, c_5 P_t(P_t - 1)/2, \\ c_6 P_{2,t}, c_7 RNA_t, c_8 P_t)$$

The inferential problem

data $\mathcal{D}_1$ fully observed without error

– all coordinates of $\{X_t\} \in \mathbb{R}^4_+$ are observed;
– $y_{t_i} \equiv X_{t_i}$.
– 50 measurements for each coordinate, $n = 200$ total observations.
– we want to estimate $\theta = (c_1, c_2, ..., c_8)$.

data $\mathcal{D}_2$ fully observed with *known* error

– all coordinates of $\{X_t\} \in \mathbb{R}^4_+$ are observed;
– $y_{t_i} = X_{t_i} + \varepsilon_{t_i}, \qquad \varepsilon_{t_i} \sim N(0, \sigma_\varepsilon I)$;
– we want to estimate $\theta = (c_1, c_2, ..., c_8)$ (notice σ_ε is *known*).

$\mathcal{D}_3$ partially observed with *unknown* error

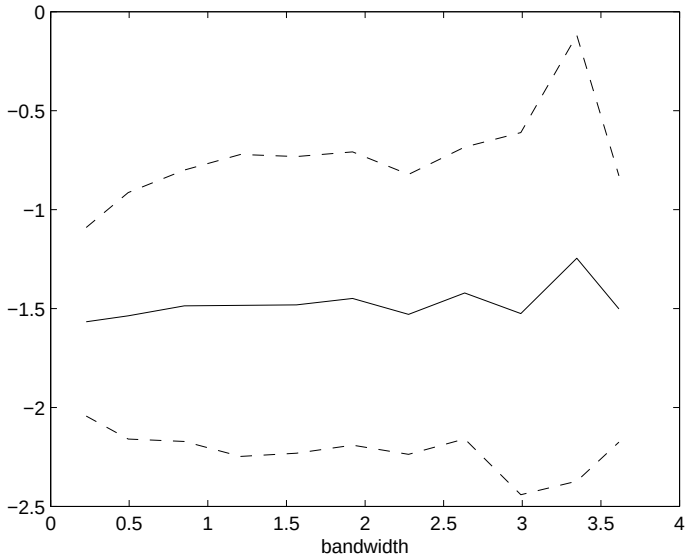
- the DNA_t coordinate is *not observed* $\Rightarrow y_{t_i} \in \mathbb{R}^3_+$
– want to estimate $\theta = (DNA_0, c_1, c_2, ..., c_8, \sigma_\varepsilon)$.

Priors: $\log c_j \sim U(-3, 0)$ and $\log DNA_0 \sim U(0, 2.3)$ (when needed).

	True parameters	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$
DNA_0	5	—	—	3.110
				[1.441, 6.526]
c_1	0.1	0.074	0.074	0.074
		[0.057, 0.097]	[0.055, 0.099]	[0.056, 0.097]
c_2	0.7	0.521	0.552	0.536
		[0.282, 0.970]	[0.294, 1.029]	[0.345, 0.832]
c_1/c_2	0.143	0.142	0.135	0.138
c_3	0.35	0.216	0.214	0.244
		[0.114, 0.4081]	[0.109, 0.417]	[0.121, 0.492]
c_4	0.2	0.164	0.168	0.174
		[0.088, 0.308]	[0.088, 0.312]	[0.089, 0.331]
c_5	0.1	0.088	0.088	0.087
		[0.070, 0.112]	[0.069, 0.114]	[0.063, 0.120]
c_6	0.9	0.352	0.355	0.379
		[0.139, 0.851]	[0.136, 0.880]	[0.147, 0.937]
c_5/c_6	0.111	0.250	0.248	0.228
c_7	0.3	0.158	0.158	0.157
		[0.113, 0.221]	[0.108, 0.229]	[0.103, 0.240]
c_8	0.1	0.160	0.169	0.165
		[0.098, 0.266]	[0.097, 0.286]	[0.091, 0.297]
σ_ε	1.414	—	—	0.652
				[0.385, 1.106]

HOWTO: post-hoc selection of δ (the "precision" parameter) [Bortot et al. 2007]

As previously explained, during the MCMC we let δ vary (according to a MRW): at rth iteration $\delta_r = \delta_{r-1} + \Delta$, $\Delta \sim N(0, \nu^2)$.
After the end of the MCMC we have a sequence $\{\theta_r, \delta_r\}_{r=0,1,2\ldots}$ and for each parameter $\{\theta_{j,r}\}_{r=0,1,2\ldots}$ we produce a plot like the following (e.g. $\log c_3$ vs δ):



Post-hoc selection of the bandwidth δ, cont'd...

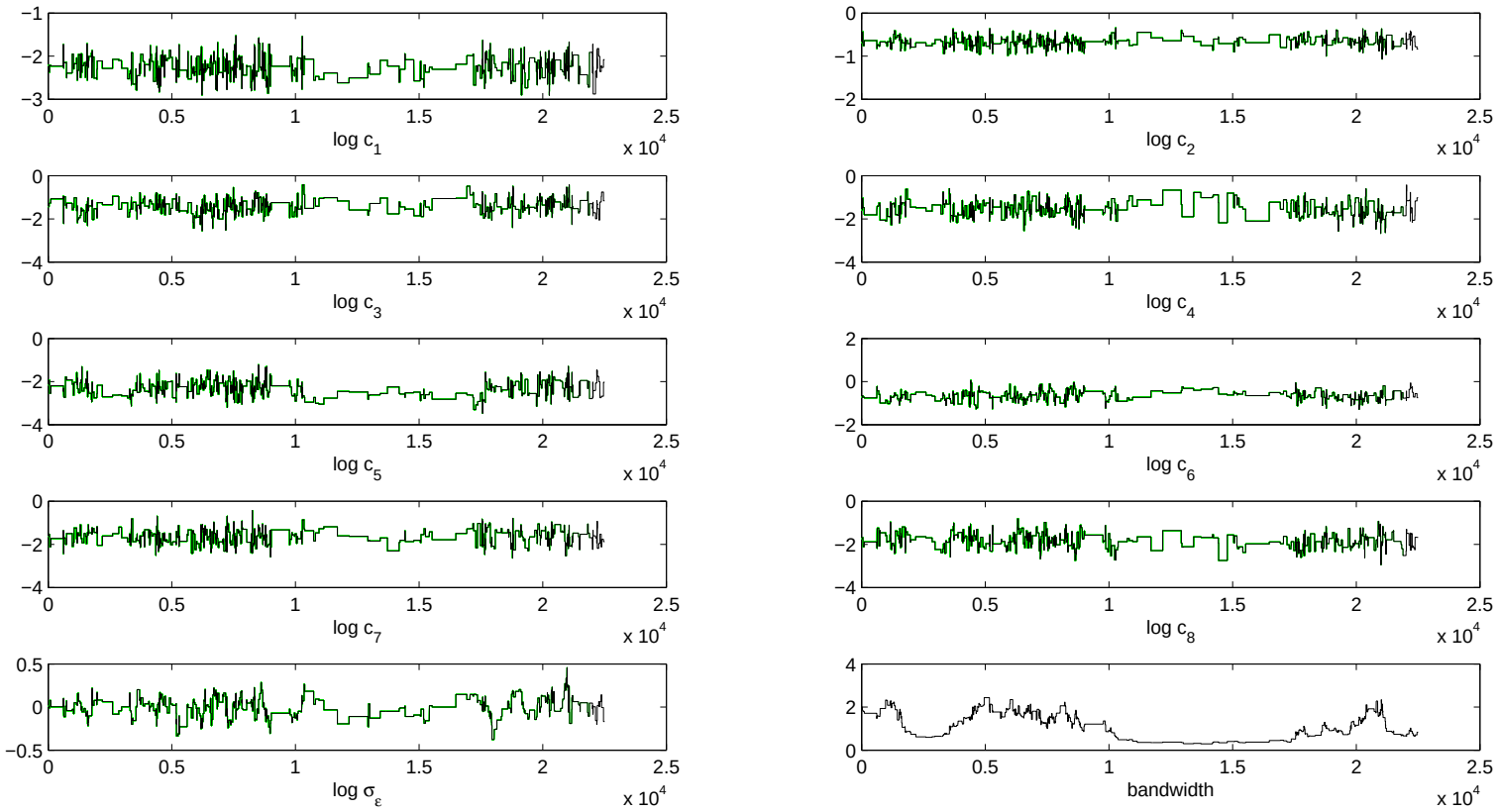


Figure : Stochastic networks example using $\mathcal{D}_3$: a (non-thinned) sample from the first 23,000 draws from the MCMC. The bandwidth δ is in the bottom-right panel.

Post-hoc selection of the bandwidth δ, cont'd...

Therefore in practice:

- we filter out of the analyses those draws $\{\theta_r\}_{r=0,1,2,\ldots}$ corresponding to "large" δ, for statistical precision;
- we retain only those $\{\theta_r\}_{r=0,1,2,\ldots}$ corresponding to a low δ (but not too low, because of previous considerations).
- in the example we retain $\{\theta_r; \delta_r < 1.5\}$.
- PRO: this is useful as it allows an *ex-post* selection of δ, i.e. we do not need to know in advance a suitable value for δ.
- CON: by filtering out some of the draws, a disadvantage of the approach is the need to run very long MCMC simulations in order to have enough "material" on which to base our posterior inference.
- PRO: also notice that by letting δ vary we are effectively considering a global optimization method (similar to *simulated tempering* & *tempered transitions*).

Conclusions

- Approximate Bayesian Computation allows inference from a wide class of models which would otherwise be unavailable.
  ..but it's not a silver bullet!
- free ABC from the choice of summary statistics would be a huge step forward (but would it still be ABC?!)
- construct the statistics $S(\cdot)$ by simulating parameters from the prior is highly inefficient;
- long simulations are needed as a price to be paid in any algorithm that avoids likelihood calculations (~ millions of MCMC iterations, ***but*** this is not necessarily very time consuming as we avoid the most computationally intense part, that is likelihood calculation!);
- a general MATLAB implementation for fully and partially observed SDEs is in preparation and will be made available on `www.maths.lth.se/matstat/staff/umberto/`.

Thank you!

Conclusions

- Approximate Bayesian Computation allows inference from a wide class of models which would otherwise be unavailable.
  ..but it's not a silver bullet!
- free ABC from the choice of summary statistics would be a huge step forward (but would it still be ABC?!)
- construct the statistics $S(\cdot)$ by simulating parameters from the prior is highly inefficient;
- long simulations are needed as a price to be paid in any algorithm that avoids likelihood calculations (~ millions of MCMC iterations, ***but*** this is not necessarily very time consuming as we avoid the most computationally intense part, that is likelihood calculation!);
- a general MATLAB implementation for fully and partially observed SDEs is in preparation and will be made available on `www.maths.lth.se/matstat/staff/umberto/`.

Thank you!

Proof that the basic ABC algorithm works

The proof is straightforward.
We know that a draw (θ', x') produced by the algorithm is such that (i) $\theta' \sim \pi(\theta)$, *and* (ii) such that $x' = x_{\text{obs}}$, where $x' \sim \pi(x \mid \theta')$.

Thus let's call $f(\theta')$ the (unknown) density for such θ', then because of (i) and (ii)

$$f(\theta') \propto \sum_x \pi(\theta')\pi(x|\theta')I_{x_{\text{obs}}}(x) = \sum_{x=x_{\text{obs}}} \pi(\theta', x) \propto \pi(\theta|x_{\text{obs}}).$$

Therefore $\theta' \sim \pi(\theta|x_{\text{obs}})$.

A theoretical motivation to consider ABC

An important (known) result

A fundamental consequence is that if $S(\cdot)$ is a sufficient statistic for θ then $\lim_{\delta \to 0} \pi_\delta(\theta \mid y) = \pi(\theta \mid y)$ the exact (marginal) posterior!!!

▸ uh?!

Otherwise (in general) the algorithm draws from the approximation $\pi(\theta \mid \rho(S(x), S(y)) < \delta)$.

The straightforward motivation is the following:
consider the (ABC) posterior $\pi_\delta(\theta \mid y)$ then

$$\pi_\delta(\theta \mid y) = \int \pi_\delta(\theta, x \mid y)dx \propto \pi(\theta) \int \frac{1}{\delta} K\left(\frac{|S(x) - S(y)|}{\delta}\right) \pi(x \mid \theta)dx$$
$$\to \pi(\theta)\pi(S(x) = S(y) \mid \theta) \qquad (\delta \to 0).$$

Therefore if $S(\cdot)$ is a sufficient statistic for θ then

$$\lim_{\delta \to 0} \pi_\delta(\theta \mid y) = \pi(\theta \mid y)$$

the exact (marginal) posterior!!!

Some remarks on SysBio concepts

- a "propensity function" $h_i(X_t, c_i)$ gives the overall hazard of a type i reaction occurring, that is the probability of a type i reaction occurring in the time interval $(t, t+dt]$ is $h_i(X_t, c_i)dt$.
- a "conservation law" (from redundant rows in S): $DNA \cdot P_2 + DNA = k$.
- In a stoichiometry matrix S reactants appear as negative values and products appear as positive values.

Choice of kernel function

We follow Fearnhead-Prangle(2012):

- we consider bounded kernels $K(\cdot) \leqslant 1$;
- specifically we use the uniform kernel $K(z)$ returning 1 when $z^T A z < c$ and 0 otherwise. In our case $z = (S(y_{sim}) - S(y))/\delta$ and A is chosen to be a $p \times p$ diagonal matrix defining the relative weighting of the parameters in the loss function.
- The uniform kernel is defined on a region $z^T A z$ bounded by a volume c, with $c = V_p |A|^{1/p}$, where $V_p = \pi^{-1} [\Gamma(p/2)p/2]^{2/p}$ and $p = dim(\theta)$;
- such c is the unique value producing a valid probability density function K i.e. such that the volume of the region $z^T A z < c$ equals 1.