Bayesian Inference for Systems Biology Models via a Diffusion Approximation

Andrew Golightly
School of Mathematics & Statistics

Darren Wilkinson
School of Mathematics & Statistics
and
Centre for Integrated Systems Biology of Ageing and Nutrition
Newcastle University, UK

Parameter Estimation in Systems Biology,
University of Manchester 28-29th March 2007
Overview

- Introduction
- Markov process models of biochemical network dynamics
- A diffusion approximation
  - Estimating diffusion parameters
- Application: Toy prokaryotic auto-regulatory network
- Summary & future directions
Concerned with building models of complex biological pathways, then validating and analysing those models using a variety of methods, including time-course simulation.

The traditional approach involves working with continuous deterministic models (e.g. coupled ODEs).

There is increasing evidence that much intra-cellular behaviour (including gene expression) is intrinsically stochastic, and that systems cannot be properly understood unless stochastic effects are incorporated into the models.

Stochastic models are harder to build, estimate, validate, analyse and simulate than deterministic models...
Start with a set of (pseudo-)biochemical reactions
Specify the rate laws and rate parameters of the reactions
Run some stochastic or deterministic computer simulator of the system dynamics
Straightforward using the Gillespie algorithm. The reverse problem is trickier – given time course data, and a set of reactions, can we recover the rates?
Mass Action Kinetics

Second Order Reaction

\[ Y_1 + Y_2 \rightarrow Y_3 \]

This will occur when a molecule of \( Y_1 \) collides with a molecule of \( Y_2 \).

- For a small, fixed volume (\( V \)) and assuming thermal equilibrium, the hazard of molecules colliding is constant (Gillespie, 1992).
- We assume the law of mass action such that the hazard of the above reaction \( \propto Y_1 Y_2 \).
Mass Action Kinetics (2)

Generically

\( k \) species and \( r \) reactions with a typical reaction

\[ R_i : \quad u_{i1} Y_1 + \ldots + u_{ik} Y_k \quad \rightarrow \quad v_{i1} Y_1 + \ldots + v_{ik} Y_k \]

- Each \( R_i \) has a stochastic rate constant, \( c_i \) and hazard \( h_i(Y, c_i) \) where \( Y = (Y_1, \ldots, Y_k)' \) is the current state of the system.
- Every system has a \( r \times k \) net effect matrix, \( A = (a_{ij}) \) where

\[ a_{ij} = v_{ij} - u_{ij} \]
Markov Process Models

Traditionally based on solving the “chemical master equation” for

\[ P(Y; t) = P(Y_1, \ldots, Y_k \text{ molecules in } V \text{ at time } t) \]

Derive the M-eq. by noting that

\[
P(Y; t + \Delta t) = \sum_{i=1}^{r} h_i(Y - A_i', c_i)P(Y - A_i'; t)\Delta t + \left\{ 1 - \sum_{i=1}^{r} h_i(Y, c_i)\Delta t \right\} P(Y; t)
\]

which leads to the M-eq.

\[
\frac{\partial}{\partial t} P(Y; t) = \sum_{i=1}^{r} \{ h_i(Y - A_i', c_i)P(Y - A_i'; t) - h_i(Y, c_i)P(Y; t) \}
\]

However

- M-eq is only tractable for a handful of cases
- Therefore stochastic models are typically examined using the Gillespie algorithm
The Gillespie algorithm

1. Initialise the system at $t = 0$ with rate constants $c_1, c_2, \ldots, c_r$ and initial numbers of molecules for each species, $Y_1, Y_2, \ldots, Y_k$.

2. Calculate $h_0(Y, c) \equiv \sum_{i=1}^{r} h_i(Y, c_i)$, the combined reaction hazard.

3. Simulate time to next event, $t' \sim \text{Exp}(h_0(Y, c))$ random quantity, and put $t := t + t'$.

4. Simulate the reaction index, $j$, as a discrete random quantity with probabilities $h_i(Y, c_i) / h_0(Y, c)$, $i = 1, 2, \ldots, r$.

5. Update $Y$ according to reaction $j$. That is, put $Y := Y + A_j'$, where $A_j$ denotes the $j$th row of the net effect matrix $A$.

6. Output $Y$ and $t$.

7. If $t < T_{\text{max}}$, return to step 2.
Example: Lotka-Volterra

Reactions

\begin{align*}
R_1 & : \ Y_1 & \longrightarrow & 2Y_1 & \text{Prey reproduction} \\
R_2 & : \ Y_1 + Y_2 & \longrightarrow & 2Y_2 & \text{Predator eats prey} \\
R_3 & : \ Y_2 & \longrightarrow & \emptyset & \text{Predator dies}
\end{align*}

If the discreteness and stochasticity are ignored, then it is straightforward to deduce the mass-action ODE system:

Lotka-Volterra: ODE Model

\begin{align*}
\frac{dY_1}{dt} & = c_1 Y_1 - c_2 Y_1 Y_2 \\
\frac{dY_2}{dt} & = c_2 Y_1 Y_2 - c_3 Y_2
\end{align*}

Analytic solutions are rarely available, but good numerical solvers can generate time course behaviour.
The Lotka-Volterra model
The Lotka-Volterra model

![Lotka-Volterra model graph]

The Lotka-Volterra model is a pair of first-order, nonlinear ordinary differential equations that describe the dynamics of predator-prey interactions. The model was first published by Alfred J. Lotka in 1925 and Vito Volterra in 1928. The equations are:

\[
\begin{align*}
\frac{dx}{dt} &= \alpha x - \beta x y \\
\frac{dy}{dt} &= -\gamma y + \delta x y
\end{align*}
\]

where:

- \(x\) is the number of prey (for example, a species of small fish) and \(y\) is the number of predators (for example, a species of fish)
- \(\alpha\) is the prey growth rate
- \(\beta\) is the predation rate of the prey by the predator
- \(\gamma\) is the predator death rate
- \(\delta\) is the efficiency of turning predated prey into predators

The model can be solved analytically in certain cases, and its solutions are characterized by cycles or limit cycles, which means that the populations of both species oscillate over time. The graph shows the concentration of predators (solid line) and prey (dashed line) over time.
The Lotka-Volterra model

![Graph showing the Lotka-Volterra model with time on the x-axis and molecules on the y-axis. The graph displays oscillatory behavior with peaks at regular intervals.](image-url)
The Lotka-Volterra model

The Lotka-Volterra model is a pair of first-order, non-linear, differential equations that describe the dynamics of biological systems in which two species interact, one as a predator and the other as prey. The equations are:

\[
\begin{align*}
\frac{dN_1}{dt} &= r_1 N_1 - a_1 N_1 N_2, \\
\frac{dN_2}{dt} &= a_2 N_1 N_2 - r_2 N_2,
\end{align*}
\]

where:
- \(N_1\) and \(N_2\) represent the number of individuals of species 1 and 2, respectively.
- \(r_1\) and \(r_2\) are the intrinsic growth rates of species 1 and 2, respectively.
- \(a_1\) and \(a_2\) are the interaction coefficients, representing the predation rate of species 1 on species 2 and the consumption rate of species 2 on species 1, respectively.

The model exhibits oscillatory behavior, with the populations of the two species fluctuating over time.

The Lotka-Volterra equations also underlie the classical predator-prey models in ecology.
The Lotka-Volterra model
The Lotka-Volterra model

![Graph showing the Lotka-Volterra model with multiple curves representing different scenarios over time. The x-axis is labeled "Time" and the y-axis is labeled "Molecules." The graph includes data points and lines connecting them, illustrating the dynamics of the model over a 50-time unit period.]
Key differences

- Deterministic solution is exactly periodic with perfectly repeating oscillations, carrying on indefinitely
- Stochastic solution oscillates, but in a random, unpredictable way
- 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
In principle it is possible to carry out rigorous statistical inference for the parameters of the stochastic process model.

Techniques for exact inference for the true discrete model (Boys, Wilkinson, Kirkwood 2004) do not scale well to problems of realistic size and complexity.

True process is discrete and stochastic — stochasticity is vital — what about discreteness?

Apply the Fokker-Planck equation to the Master equation for the true process to obtain an SDE known as the Chemical Langevin Equation (CLE).
The Stochastic-Kinetic Diffusion Approximation

**Chemical Langevin Equation (Itô SDE)**

\[ dY_t = A' h(Y_t, c) dt + [A' \text{ diag}\{h(Y_t, c)\} A]^{1/2} dW_t \]

- Fairly general class of non-linear multivariate SDEs
- The net effect matrix \( A \) is typically rank-degenerate, which complicates things slightly
- \( A \) is known and \( Y \) (or a subset) is observed at discrete times (subject to error)
- Inference is for \( c \) (the vector of rate constants parameterising the reaction rate vector, \( h(\cdot, \cdot) \))
Inference for Diffusions

Set $\mu(Y_t, c) = A' h(Y_t, c)$, $\beta(Y_t, c) = A' \text{diag}\{h(Y_t, c)\} A$

Need to consider the general problem of inferring parameters $c$ governing

$$dY_t = \mu(Y_t, c)dt + \beta^{\frac{1}{2}}(Y_t, c)dW_t$$

using observations (that may be incomplete and subject to error) at discrete times

**Problem:** For $\mu$ and $\beta$ nonlinear, analytic solutions rarely available
  - Can’t obtain underlying transition densities!
  - Likelihood inference non-trivial
Bayesian Imputation approach

Work with the Euler discretisation

\[ \Delta Y_t = \mu(Y_t, c) \Delta t + \beta^{\frac{1}{2}}(Y_t, c) \Delta W_t, \quad \Delta W_t \sim N(d(0, I \Delta t)) \]

- Inter-obs. time, \( \Delta^* \), usually too big to use as \( \Delta t \)!
- Set \( \Delta t = \Delta^*/m \), choose \( m \) large so that \( \Delta t \) is small
- Gives \( m - 1 \) latent values between every pair of obs
- Augmented data in matrix form,

\[ \hat{Y} = \begin{pmatrix} y_{t_0} & Y_{t_1} & \cdots & Y_{t_{m-1}} & y_{t_m} & Y_{t_{m+1}} & \cdots & Y_{t_{n-1}} & y_{t_n} \end{pmatrix} \]

- For data, \( D_n \), formulate joint posterior for \( c \) and missing values \( \hat{Y} \setminus \{D_n\} \)

\[ \pi(c, \hat{Y} \setminus \{D_n\} | D_n) \propto \pi(c) \times \prod_{i=0}^{n-1} \pi(Y_{i+1} | Y_i, c) \]

- Integrate over our uncertainty for \( \hat{Y} \) using MCMC
Could sample $\pi(c, \mathbf{\hat{Y}}\backslash\{D_n\}|D_n)$ by alternating between

- draws of missing data (e.g. one column at a time) conditional on $c$ and $D_n$ (Metropolis step)
- draws of $c$ conditional on augmented data, $\mathbf{\hat{Y}}$ (Metropolis step)

However, if the diffusion coefficient is not free of $c$, the algorithm is reducible

- For $m \to \infty$, there is an infinite amount of information in the augmented sample $\mathbf{\hat{Y}}$

Solution (due to Roberts & Stramer, ’01): Find an analytic transformation of the diffusion to constant volatility

- Typically impossible to implement for interesting nonlinear diffusions
Irreducible Global MCMC Schemes

Idea (Chib, Pitt & Shephard, ’06). Gibbs sampler: Draw from \( c|\hat{W} \) rather than \( c|\hat{Y} \) thereby breaking the problematic dependence. Target:

\[
\pi(c|\hat{W}) \propto \pi(c) \pi(g(\hat{W}, c)|c) \times \text{Jacobian}
\]

Conditional on \( c \), there is a one-to-one relationship between \( \hat{Y} \) and \( \hat{W} \) – the skeleton of the driving B.M.

Numerically map between the diffusion sample paths and the corresponding sample paths of the driving Brownian motion, for example using the Euler-Maruyama discretisation

\[
\Delta Y_t = \mu(Y_t, c) \Delta t + \beta^{1/2} \left(Y_t, c \right) \Delta W_t
\]

\[
\Rightarrow \Delta W_t = \beta^{-1/2} \left(Y_t, c \right) [\Delta Y_t - \mu(Y_t, c) \Delta t]
\]

Problem: unless the diffusion is observed very indirectly, changing the parameters causes the sample paths to “miss” the data points, rendering it impractical
(Golightly & Wilkinson, ’06): Use the modified diffusion bridge MDB construct of Durham and Gallant ’02 as a template for building sample paths, and use the Wiener processes driving the MDB as our sampler components.

Thinking just about a discretisation of [0, 1] and the fully observed case, we can map back and forth using the deterministic transformations

\[
\Delta Y_t = \frac{y_1 - Y_t}{1 - t} \Delta t + \left( \frac{1 - t - \Delta t}{1 - t} \beta(Y_t, c) \right)^{\frac{1}{2}} \Delta W_t
\]

\[
\Rightarrow \Delta W_t = \left( \frac{1 - t}{1 - t - \Delta t} \right) \beta^{-\frac{1}{2}}(Y_t, c) \left[ \Delta Y_t - \frac{y_1 - Y_t}{1 - t} \Delta t \right]
\]

Crucially, there is no problem with failing to “hit” data points after transforming back to the observed diffusion.
Algorithm

1. Initialise parameters $c$, and latent data $\hat{Y}\backslash\{D_n\}$
2. For times $t_0, t_m, \ldots, t_{n-m}$ update latent data in blocks of size $m - 1$ using the MDB, and accept/reject with a M-H step
3. Map from $\hat{Y}$ to $\hat{W}$ using the MDB transformation on each interval
4. Propose a new parameter $c^*$. Using $c^*$ with fixed $\hat{W}$, deterministically construct the corresponding sample path $\hat{Y}^*$, and accept/reject the pair jointly with a M-H step
5. Output state and return to step 2

Generalisations to noisy/imperfect observations are straightforward
Acceptance probabilities

- Let $Y_m$ denote all latent values in $(t_j, t_{j+m})$
- The acceptance probability for a single interval path update on $(t_j, t_{j+m})$ takes the form
  \[ A = \frac{\pi(Y_m^*|c, y_j, y_{j+m})}{\pi(Y_m|c, y_j, y_{j+m})} \times \frac{q(Y_m|c, y_j, y_{j+m})}{q(Y_m^*|c, y_j, y_{j+m})} \]
- The acceptance probability for a proposed update to $c^*$ takes the form
  \[ A = \frac{\pi(c^*)}{\pi(c)} \times \frac{f(c|c^*)}{f(c^*|c)} \times \frac{\pi(\hat{Y}^*|c^*)}{q(\hat{Y}^*|c^*)} \times \frac{\pi(\hat{Y}|c)}{q(\hat{Y}|c)} \]
**Toy Application: Prokaryotic Auto-Regulation**

**Reaction list:**

- $R_1: \ DNA + P_2 \rightarrow DNA \cdot P_2$  \hspace{1cm} \text{Repression}$
- $R_2: \ DNA \cdot P_2 \rightarrow DNA + P_2$
- $R_3: \ DNA \rightarrow DNA + RNA$  \hspace{1cm} \text{Transcription}$
- $R_4: \ RNA \rightarrow RNA + P$  \hspace{1cm} \text{Translation}$
- $R_5: \ 2P \rightarrow P_2$  \hspace{1cm} \text{Dimerisation}$
- $R_6: \ P_2 \rightarrow 2P$
- $R_7: \ RNA \rightarrow \emptyset$  \hspace{1cm} \text{Degradation}$
- $R_8: \ P \rightarrow \emptyset$

- 5 species DNA, DNA $\cdot$ P$_2$, RNA, P, P$_2$ and 8 reactions with rate constants $c = (c_1, \ldots, c_8)$

- Note that DNA and DNA $\cdot$ P$_2$ are deterministically related
- Induces a 4-dimensional diffusion process parameterised by $c$
Simulation Study

- 50 obs simulated using the Gillespie algorithm

Rate constants $c = (0.1, 0.7, 0.35, 0.2, 0.1, 0.9, 0.3, 0.1)'$

Run the innovation scheme to recover these values
Results, $m = 10$, Gibbs Sampler
Results, $m = 10$, Innovation Scheme

- $c_1$
- $c_3$
- $c_5$
- $c_7$
Results, $m = 10$, Innovation Scheme
## Results, $m = 10$, Innovation Scheme

<table>
<thead>
<tr>
<th></th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
<th>$C_6$</th>
<th>$C_7$</th>
<th>$C_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Values</td>
<td>0.1</td>
<td>0.7</td>
<td>0.35</td>
<td>0.2</td>
<td>0.1</td>
<td>0.9</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>Observe (DNA, RNA, P, $P_2$)</td>
<td>Mean</td>
<td>0.087</td>
<td>0.655</td>
<td>0.547</td>
<td>0.055</td>
<td>0.078</td>
<td>0.758</td>
<td>0.437</td>
</tr>
</tbody>
</table>
### Results, $m = 10$, Innovation Scheme

<table>
<thead>
<tr>
<th></th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
<th>$C_6$</th>
<th>$C_7$</th>
<th>$C_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>True Values</strong></td>
<td>0.1</td>
<td>0.7</td>
<td>0.35</td>
<td>0.2</td>
<td>0.1</td>
<td>0.9</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td><strong>Observe (DNA, RNA, P, P_2)</strong></td>
<td>Mean 0.087</td>
<td>0.655</td>
<td>0.547</td>
<td>0.055</td>
<td>0.078</td>
<td>0.758</td>
<td>0.437</td>
<td>0.038</td>
</tr>
<tr>
<td><strong>Observe (RNA, P, P_2)</strong></td>
<td>Mean 0.061</td>
<td>0.451</td>
<td>0.497</td>
<td>0.024</td>
<td>0.072</td>
<td>0.702</td>
<td>0.393</td>
<td>0.020</td>
</tr>
</tbody>
</table>
Results, $m = 10$, Innovation Scheme

<table>
<thead>
<tr>
<th></th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
<th>$C_6$</th>
<th>$C_7$</th>
<th>$C_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Values</td>
<td>0.1</td>
<td>0.7</td>
<td>0.35</td>
<td>0.2</td>
<td>0.1</td>
<td>0.9</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>Observe (DNA, RNA, $P$, $P_2$)</td>
<td>Mean</td>
<td>0.087</td>
<td>0.655</td>
<td>0.547</td>
<td>0.055</td>
<td>0.078</td>
<td>0.758</td>
<td>0.437</td>
</tr>
<tr>
<td>Observe (RNA, $P$, $P_2$)</td>
<td>Mean</td>
<td>0.061</td>
<td>0.451</td>
<td>0.497</td>
<td>0.024</td>
<td>0.072</td>
<td>0.702</td>
<td>0.393</td>
</tr>
<tr>
<td>Observe (RNA)</td>
<td>Mean</td>
<td>0.047</td>
<td>0.262</td>
<td>0.540</td>
<td>0.049</td>
<td>0.023</td>
<td>0.153</td>
<td>0.461</td>
</tr>
</tbody>
</table>
Summary

- Systems Biology and post-genomics are full of interesting (hard) statistical problems.
- It appears promising to consider the problem of understanding biochemical network dynamics in terms of inference for the Chemical Langevin Equation.
- Inference for arbitrary multivariate diffusions observed partially, discretely and with error is non-trivial.
- It is possible, however, to implement global MCMC schemes which do not break down for large amounts of augmentation.


Contact details...

e-mail: a.golightly@ncl.ac.uk

www: http://www.mas.ncl.ac.uk/~nag48/