Efficient (Exact/Approximate) Monte Carlo Inference for Stochastic Epidemic Models

Theo Kypraios
http://www.maths.nott.ac.uk/∼tk

School of Mathematical Sciences, University of Nottingham

Warwick, Infer 2011 Workshop

What this talk is about

- Most of the content is (joint) work in progress . . .
- Interested in exploring cases (i.e. models/data) where **exact** “likelihood-free” Monte-Carlo inference can be drawn.
- If *exact* inference is not possible, interested in **efficient, but approximate** methods.
- **Discuss**
  - how one can take advantage of the **model’s structure**?
  - how some, if any, of the **available (theoretical) results** can aid the construction of efficient simulation-based algorithms?

Exact Bayesian Computation (EBC)

Exact Bayesian Computation (EBC)

- Suppose we have discrete data $\mathcal{D}$, prior $\pi(\theta)$ for parameter(s) θ.

- Aim: Draw samples from the posterior distribution of the parameters, $\pi(\theta|\mathcal{D})$.

- Bayes Theorem gives

$$\pi(\theta|\mathcal{D}) = \pi(\mathcal{D}|\theta)\pi(\theta)/\pi(\mathcal{D})$$

where $\pi(\mathcal{D})$ is a normalising constant:

$$\pi(\mathcal{D}) = \int_\theta \pi(\mathcal{D}|\theta)\pi(\theta)\ \mathrm{d}\theta$$

and therefore

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

Rejection Sampling

Consider the following algorithm:

ER1

1. Draw $\theta \sim \pi(\theta)$;
2. Accept θ with probability equal to $\pi(\mathcal{D}|\theta)$;
3. Repeat until the required number of samples is drawn.

* It is trivial to show that the accepted values will be distributed according to the posterior distribution $\pi(\theta|\mathcal{D})$.

* Note that this algorithm requires that the likelihood, $\pi(D|\theta)$, can be computed for any θ (Step 2).

Rejection Sampling

Can do better: if $\pi(\mathcal{D}|\theta) < c$ for all θ, substitute Step 2 with:

2. Accept θ with probability equal $\pi(\mathcal{D}|\theta)/c$

i.e. **increased acceptance rate** at zero cost provided that we can find an upper bound for the likelihood.

Rejection Sampling (cont.)

Consider the following algorithm:

ER2

1. Draw $\theta \sim \pi(\theta)$;
2. Draw $\mathcal{D}' \sim \pi(\cdot|\theta)$, i.e. simulate data $\mathcal{D}'$ from the model;
3. Accept θ if $\mathcal{D} = \mathcal{D}'$;
4. Repeat until the required number of samples is drawn.

* Algorithm **ER2** is equivalent to algorithm **ER1**.

* The main difference is that the computation of the likelihood $\pi(\mathcal{D}|\theta)$ (**ER1**, Step 2) is substituted by the simulation of an event which occurs with that probability (**ER2**, Steps 2 and 3).

* That means that the calculation of the likelihood is unnecessary as long as we can simulate from our stochastic model.

EBC and Monte Carlo Maximum Likelihood

- Suppose that we are interested in finding the MLE, say $\widehat{\theta}_{MLE}$.
- One needs to be able to compute the likelihood, $\pi(\mathcal{D}|\theta)$ for any value θ.
- If this cannot be done directly, one can approximate the likelihood function using Monte Carlo (Diggle and Gratton, 1984):

ML1

1. Choose a grid of values for θ, i.e. $(\theta_1, \ldots, \theta_K)$;
2. Estimate the value of the likelihood for each θ_j, $j = 1, \ldots, K$ by

$$\widehat{\pi(\mathcal{D}|\theta_j)} = \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}(\mathcal{D}'_{\theta_j} = D)$$

where D'_{θ_j} is a simulated dataset from the model with parameter θ_j.

EBC and Monte Carlo Maximum Likelihood

- Instead of maximising the likelihood function $\pi(\mathcal{D}|\theta)$ we are maximising a Monte Carlo estimator of it.
- One could either maximise the likelihood in brute-force manner (expensive) or by optimization (much cheaper).
- Although such a MCMLE scheme enjoys desired properties (such as unbiasness etc) we have to be careful, since we might get misleading results due to large Monte Carlo error $\rightarrow$ increase Monte Carlo samples M.
- One can also try and do a Kernel Density Estimation (KDE) of the Monte Carlo likelihood function and do optimization on the KDE.

Some Remarks

Both algorithms ER1 & ER2 are **very** easy to implement but they both suffer in terms of efficiency when

- the prior $\pi(\theta)$ is **very different** from the posterior $\pi(\theta|\mathcal{D})$;

while algorithm ML1 can be problematic if

- there is **huge Monte Carlo variation** and one needs a very large M to accurately approximate the likelihood function.

How does all this relate to epidemic modelling?

EBC for Epidemic Models

- Consider some final size data $\mathcal{D}$ (i.e. total number of infectives out of the initial susceptible population of size $\mathcal{N}$) from a standard SIR model.

- Having assumed a distribution for the infectious period, we wish to make inference for the infection rate λ (or R_0), i.e. derive (or draw samples from) $\pi(\lambda|\mathcal{D})$ (or $\pi(R_0|\mathcal{D})$).

- When using standard (double) precision arithmetic, numerical problems appear when computing the likelihood $\pi(\mathcal{D}|\theta)$ (using for example the triangular equations) due to rounding errors even for $\mathcal{N}$ of the order 50–100.

EBC for Epidemic Models (2)

- Therefore, the algorithm **ER1** cannot be implemented unless Multiple Precision Arithmetic (MPA) is employed (Demiris and O'Neill, ST&COM).
- On the other hand, algorithm **ER2** can be used instead; simulating final size data from an SIR model is trivial using either the *Gillespie algorithm* or the *Sellke construction* etc.
- The Gillespie algorithm can be computationally very expensive for large $\mathcal{N}$ and therefore we prefer using the Sellke construction.
- Note that **ER2** is *very straightforward* to implement (i.e. few lines of code) and it results in drawing *exact* samples from the posterior distribution of interest; **MCMC approaches available:** e.g. Demiris' thesis, (2004) O'Neill (2009), Ball(201?), White's thesis (2010).

Accounting for observational error

- It is very easy to allow for **error in observation** of the final size, e.g. account for underreporting.

- In other words, we are interested in the posterior distribution of $\pi(\lambda|\mathcal{D} \pm k)$ for some k.

- This can be done either retrospectively, i.e. if we had stored all the pairs $(\theta, \mathcal{D}')$ we can look at those that satisfy the criterion, . . .

- . . . or by modifying the existing **ER1** algorithm, i.e. only store *on the fly* the values of θ which satisfy the criterion.

An Exact Algorithm for Sample Data

Consider the case where we have *sample* final size data, i.e. we only observe the final size of a proportion of the population, but wish to make inference for the whole population.

- Total population size of initial susceptibles $\mathcal{N}$
- (Random) Sampled population size of initial susceptibles m
- Observed final size d (out of m).

ER3

1. Draw $\theta \sim \pi(\theta)$;
2. Draw $\mathcal{D}' \sim \pi(\cdot|\theta)$, where $\mathcal{D}'$ is **total** final size.
3. Accept θ with probability $p(m, \mathcal{N}, d, \mathcal{D})$.
4. Repeat until the required number of samples are drawn.

An Exact Algorithm for Sample Data (2)

- The probability $p(m, \mathcal{N}, d, \mathcal{D})$ in Step 3 can be derived from a hypergeometric distribution and can be computed explicitly.

- Instead of *accepting* θ w.p. $p(m, \mathcal{N}, d, \mathcal{D})$ one could the following:

AR3

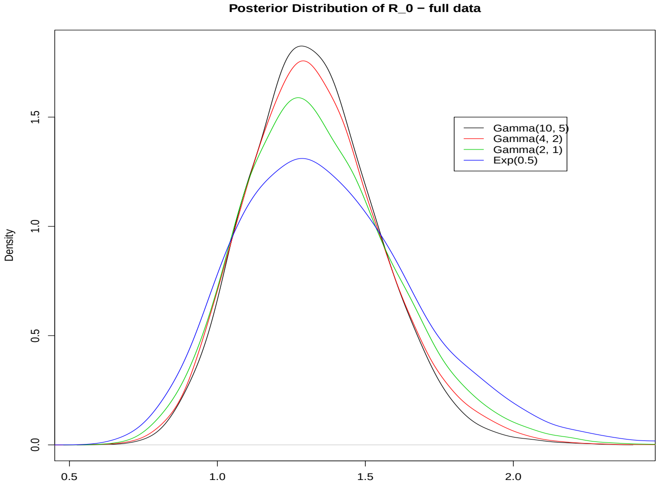
1. Draw $\theta \sim \pi(\theta)$;
2. Draw $\mathcal{D}' \sim \pi(\cdot|\theta)$, where $\mathcal{D}'$ is **total** final size.
3. Store the pair $\{\theta,\ p_\theta(m, N, d, D)\}$.
4. Repeat until the required number of samples are drawn.

One can think of $p_\theta(m, \mathcal{N}, d, \mathcal{D})$ as a weight for each drawn θ and then approximate $\pi(\theta|\mathcal{D})$ in an Importance Sampling manner.

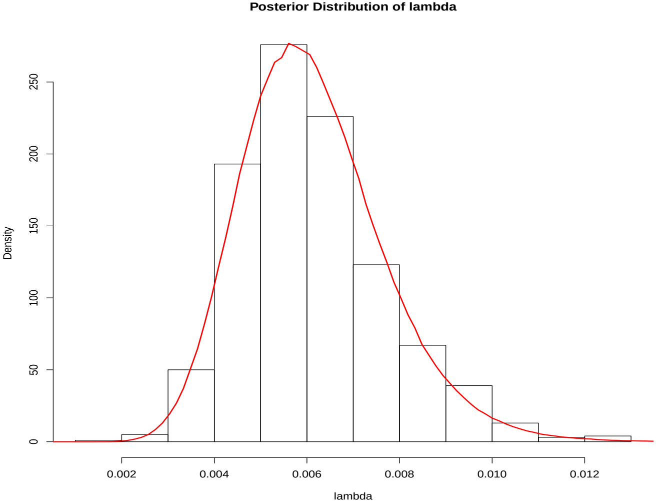
Do these algorithm really work (well) in practice?

Some Examples – Full Data

Suppose we observe a final size of 40 out of 100 initially susceptible individuals.

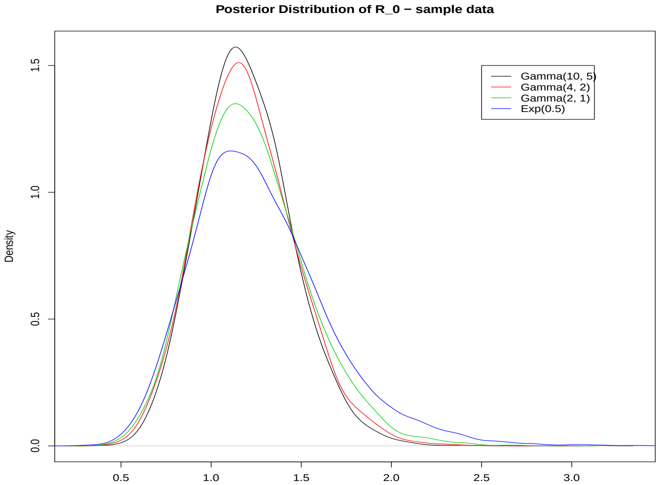


Some Examples – MCMLE



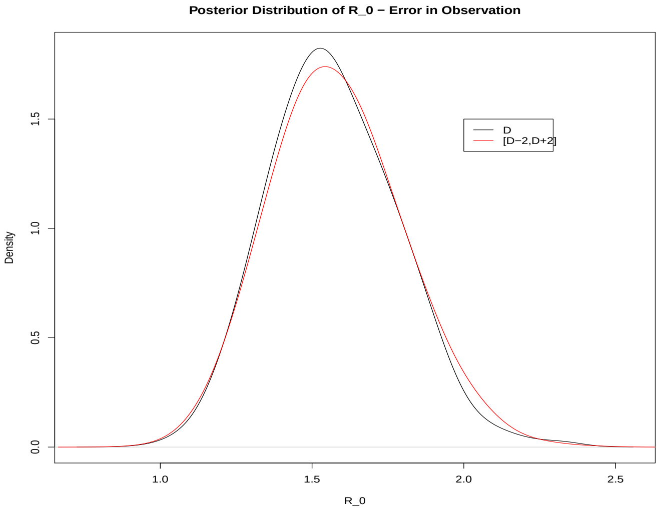
Some Examples – Sample Data

Suppose we observe a final size of 10 out of a sample population of 40 initially susceptible individuals while the size of the total population is 100 individuals.



Some Examples – Error in Observations

Suppose we observe 60 infected individuals out of 100 but we want to allow for some error.



Challenges

- If the prior is very vague, e.g. $\lambda \sim Exp(10^{-6})$, then **ER2** and **ER3** will not be efficient (we rarely hit the target).

- The problem can be overcome by obtaining a Monte Carlo estimate for the likelihood to get an idea where the most likely values of λ lie. Then use this information to draw from a *constrained* "prior". Problem when we have a pretty informative prior?

    - This is some sort of an *adaptive* algorithm → strictly speaking we are not doing *exact* inference anymore but in practice we loose very little!

- When the size of initially susceptible population is large, then the EBC algorithm will very inefficient, i.e. it will take ages to hit the target.

How Can we Overcome These Problems?

1. **Sequential Monte Carlo - Exact Bayesian Computation**
   - One "solution" to the problem might be to apply an SMC-ABC type algorithm ... but, is there an alternative way?
2. **"Likelihood-Free Importance Sampling"?**
   - Suppose that, some-how, we have some sort of *approximation* to our posterior of interest, $\pi(\theta|\mathcal{D})$, say $q(\theta)$.
   - Can we make use of $q(h)$ and draw **honest** Bayesian inference?

“Likelihood-Free” Importance Sampling

How Can we Overcome These Problems? (2)

Recall, **ER1** algorithm:

1. Draw $\theta \sim \pi(\theta)$;
2. Accept θ with probability equal to $\pi(\mathcal{D}|\theta)$ or if $\pi(\mathcal{D}|\theta)/c$ if c known;
3. Repeat until the required number of samples is drawn.

and **ER2** algorithm

1. Draw $\theta \sim \pi(\theta)$;
2. Draw $\mathcal{D}' \sim \pi(\cdot|\theta)$, i.e. simulate data $\mathcal{D}'$ from the model;
3. Accept θ if $\mathcal{D} = \mathcal{D}'$;
4. Repeat until the required number of samples is drawn.

Approach I: Use ideas from the “Bernoulli Factory”

Although both algorithms are equivalent, ER2, relies on the fact we can simulate events with probability equal to the likelihood (Steps 2 and 3 of ER2).

If c is known (see ER1) then the *equivalent* ER2 algorithm should take into account the fact that we want to simulate events with probability $\pi(\mathcal{D}|\theta)/c$ or $k \cdot \pi(\mathcal{D}|\theta)$ for $k = 1/c$.

But, we only know how to simulate events with probability equal to the likelihood.

The question then is: Given that we can simulate events with some probability p, how can we simulate events with probability $f(p)$? → Bernoulli Factory! It is a (very old) and hard problem! See Nacu and Peres (2005), Latuszynski *et al.* (2010) etc

Approach I: Use ideas from the "Bernoulli Factory" (2)

- Briefly, there is no exact algorithm to simulate events with probability $f(p)$, . . . for any $f(\cdot)$
- nevertheless, there is an algorithm (see previous refs) to simulate events with probability when $f(p) = \min(kp, 1 - k\epsilon)$ for some $\epsilon > 0$.
- The larger the ϵ the more inefficient this algorithm becomes!
- **Question:** If don't want to do exact inference, are the there any fast, but approximate solutions to the Bernoulli factory problem?

Sequential Monte Carlo

Approach II: Sequential Monte Carlo

SMC-E/ABC type algorithms (eg. Sisson *et al* (2010), Toni *et al.* (2010)) can offer a good alternative. The idea is:

1. We use samples from the priors (called *particles*) ...
2. ..., which are propagated through a sequence of intermediate distributions, $\pi(\theta|\Delta(\mathcal{D},\mathcal{D}^*) \leq \varepsilon_i)$, where the tolerances ϵ_i for

   $i = 1, \ldots, T$ are chosen such that $\epsilon_1 > \ldots \epsilon_T \geq 0$.
3. In other words, these intermediate distributions evolve towards $\pi(\theta|\mathcal{D})$.

Approach II: Sequential Monte Carlo

In general:

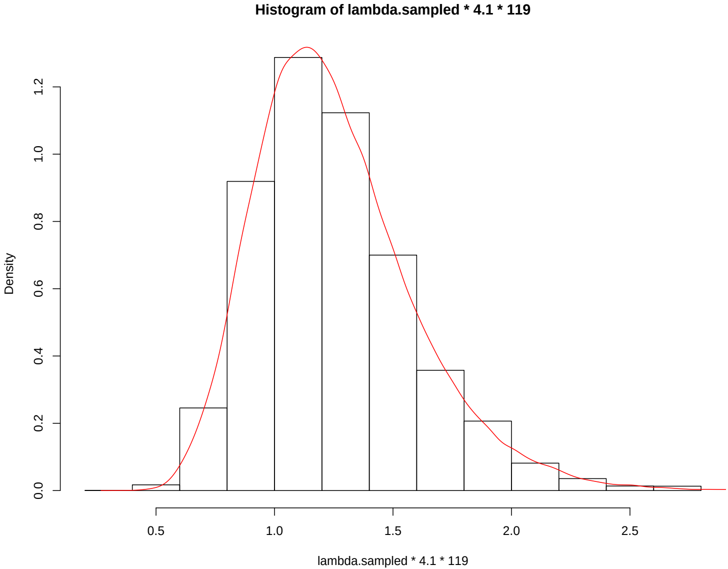
- Such algorithms can work very well;
- although their efficiency largely depends on certain tunable aspects, such as the choice of the tolerances ϵ_i,$i = 1, \ldots, T$
- and the kernel via which the particles are propagated.
- Moreover, without care the approach can even fail altogether due to “particle degeneracy”.

In the epidemic modelling context:

- At least in the case of a single-type one level mixing model, one could start the series of the ϵ’s with large values and drag it down to zero.
- Effectively, this will lead to *exact* Monte Carlo samples from the posterior of interest.

Approach II: Sequential Monte Carlo

Abakaliki Data: Posterior distribution of R_0:



What about models with more population structure?

A Model with Two Levels of Mixing

- We consider epidemic models in populations which mix at two levels, “global” and “local”.
- More precisely, we consider models in which each infectious individual i has a ‘global’ rate λ_G for infecting each other individual in the population ...
- ... plus a ‘local’ rate λ_L of infecting each other individual among a set of neighbours, $N(i)$.
- Our main concern is the case where the population is partitioned into local groups or households (or farms/yards etc).

EBC for the 2-level Mixing Model

ER2

1. Draw $\theta \sim \pi(\theta)$;
2. Draw $\mathcal{D}' \sim \pi(\cdot|\theta)$, i.e. simulate data $\mathcal{D}'$ from the model;
3. Accept θ if $\mathcal{D} = \mathcal{D}'$;
4. Repeat until the required number of samples are drawn.

- Step 3 now is more ambitious than before. We need to hit the target of the total final size of the epidemic as well as the the final size in each group (households/farms/yards).

- If the households are all of the same size then it is not too bad → permutations or groupings.

- Otherwise the above algorithm is (or can be) **very** inefficient and an EBC-SMC type algorithm may not offer that great advantages.

ABC for the 2-level Mixing Model

Doing exact inference seems very challenging and therefore, we seek for alternative, but **approximate ways** to draw inference for $\boldsymbol{\theta} = (\lambda_L, \lambda_G)$.

ABC

1. Draw $\theta \sim \pi(\theta)$;
2. Draw $\mathcal{D}' \sim \pi(\cdot|\theta)$, i.e. simulate data $\mathcal{D}'$ from the model;
3. Accept θ if $\Delta(\mathcal{D}, \mathcal{D}') < \epsilon$;
4. Repeat until the required number of samples are drawn.

Questions:

- What do we mean by $\mathcal{D}$?
- How small to choose ϵ?
- What distance metric $\Delta(\cdot)$ should we use?
- How good the approximate posteriors are to the true ones?

Types of data

At least, two different types of data:

1. Suppose that the data $\mathcal{D}$ consist of the final size at each group (household/farm/yard), i.e. a series of $(d_1, d_2, \ldots d_n)$ as well as the number of susceptibles in each group $(N_1, N_2, \ldots N_n)$.
2. Alternatively, if the population consists of various groups of similar size then, often the data appear in such a form:

	Susceptibles per household				
No of Infected	1	2	3	4	5
0	40	49	22	20	3
1	93	137	83	80	23
2		3	2	4	2
3			1	1	1
4				1	1
5					1
Total	133	189	108	106	31

Various Choices of Metrics

- Just compare the final size and ignore the group structure (not very clever):

$$\Delta(\mathcal{D}, \mathcal{D}') = \left| \sum_{i=1}^{n} d_i - \sum_{i=1}^{n} d_i' \right|$$

- Just compare the final sizes in each groups (a bit clever):

$$\Delta(\mathcal{D}, \mathcal{D}') = \sum_{i=1}^{n} \left| d_i - d_i' \right|$$

- Look at proportions instead (quite clever, especially when N_i's are **very** different):

$$\Delta(\mathcal{D}, \mathcal{D}') = \sum_{i=1}^{n} \left| \frac{d_i - d_i'}{N_i} \right|$$

- A combination of the above? (see Baguelin *et. al* (2009))

- ...

Practicalities – Challenges

- Our experience reveals that this metric

  $$\Delta(\mathcal{D}, \mathcal{D}') = \sum_{i=1}^{n} \left| \frac{d_i - d_i'}{N_i} \right|$$

  works pretty well when compared with the results from a gold-standard approach, i.e. MCMC.

- Adopting an SMC-ABC algorithm one can approximate the true posteriors very well (see graphs later).

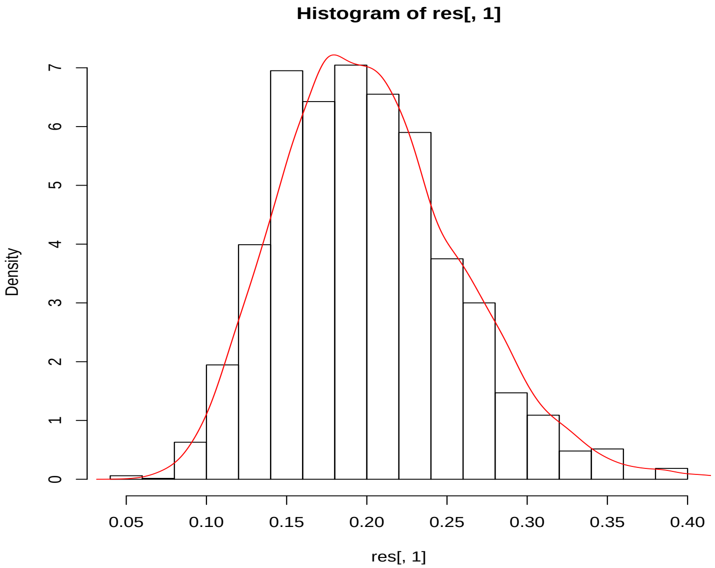
- In theory, one could drag the ϵ_i in the series of errors to be close to zero, but some times this could be problematic, i.e. particles degenerate. *When do we stop?*

- In this case, there are only two parameters and it is possible to **move the particles in a block** (e.g. using a random-walk kernel). What happens when there are more model-parameters? Use a Gibbs kernel?

Practicalities – Challenges (2)

- One needs to be **very careful** how to choose the different values for $\epsilon_1, \ldots, \epsilon_k$.
- A bad selection could lead to a **very inefficient algorithm**. Of course, the larger the *k* is, the lower the chance of having particles to become degenerate . . . (making slick moves) . . .
- . . . the more costly the algorithm becomes.
- Borrow results from Simulated Annealing/Parellel Tempering such that we place $\epsilon_1, \ldots, \epsilon_k$ efficiently, i.e in an *optimal way*? see: Behrens, Friel, Hurn (ST&COM, to appear)

Some Results

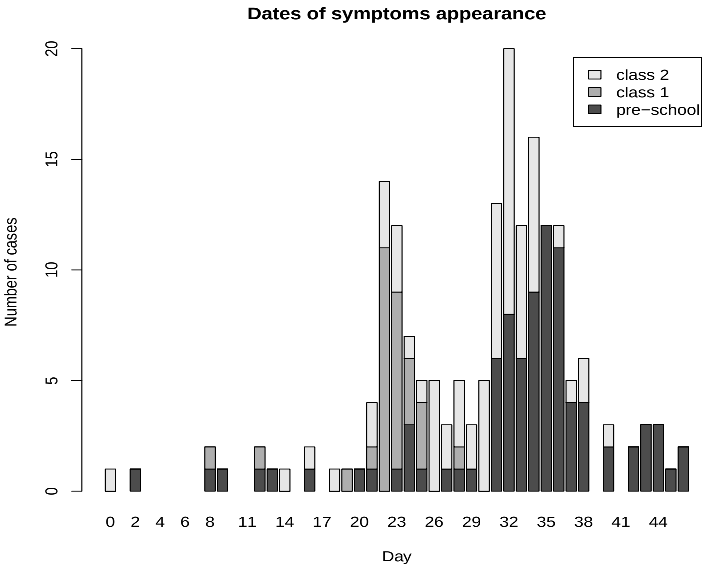


What about temporal data?

Temporal Data

- So far we have only considered final outcome data which allowed us to draw *exact* Monte Carlo inference for the parameters.
- When temporal data are available, it is often the case that a model (defined) in *continuous time* will be fitted; although this is not necessary always the case.
- In these, due the continuity, there are many more things to worry about, e.g. which summary statistics to use etc.
- However, in practice, real data are only available in discrete time, i.e. number of removals per day etc.
- An alternative approach to the one presented in Cook *et al* (2010) would be to involve some sort of data discretization.

Temporal Data – Hagelloch Dataset



Discretizing the data

- The idea is, that in the Step where data are simulated in continuous time (i.e. removal times). we will perform some sort of discretization.

- Obviously, this has to be in accordance with the observed data (days$\leftrightarrow$days, months$\leftrightarrow$months)

- Similar distance metrics to the ones used for final size data could be used and enjoy similar properties.

- Challenges:
  - Bins of equal width?
  - Weighted bins?
  - Effect of amount of discretization?

Piecewise Exact (Approximate) Bayesian Computation

Piecewise Exact (Approximate) Bayesian Computation: PW-EBC (PW-ABC)

White, K, Preston and Crowe (2011)

- For simulation-based methods, computing resources are always a limiting factor,
- and a guiding principle is to take every opportunity to exploit model structure to minimise computational costs.

Piecewise Exact (Approximate) Bayesian Computation: PW-EBC (PW-ABC) (2)

For a particular (but fairly broad) class of models,

- namely those that have the **Markov property**, and
- and whose vector of state variables is **observable at discrete time points**

we would like to have a (likelihood free) simulation-based method which

- **takes advantage** of the models **Markov structure**,
- requires **minimal** amount of **tuning**, and
- enables **less approximate**, or even exact, Bayesian inference

PW-EBC & PW-ABC

- Suppose our data are a set of observations denoted $\mathcal{X} = \{x_1, \dots, x_n\} = \{x(t_1), \dots, x(t_n)\}$ of state variable $x \in \mathbb{R}^m$ at time points $t_1, \dots, t_n$.

- The posterior density is:

$$\pi(\theta|\mathcal{X}) = \frac{\pi(\mathcal{X}|\theta)\pi(\theta)}{\int_\theta \pi(\mathcal{X}|\theta)\pi(\theta)\ \mathrm{d}\theta} \propto \pi(\mathcal{X}|\theta)\pi(\theta),$$

- EBC (ER1/ER2) or ABC can be applied in the usual way.

  1. Draw θ^* from $\pi(\theta)$.
  2. Simulate dataset $\mathcal{X}^* = \{x^*(t_1), \dots, x^*(t_n)\}$ from the model using parameters θ^*.
  3. Accept θ^* if $\mathcal{X}^* = \mathcal{X}$, otherwise reject **or** Accept θ^* if $d(\mathcal{X}, \mathcal{X}^*) \leq \varepsilon$, otherwise reject

Exploting the Markovian Structure

The Markov property enables the likelihood to be written as

$$\begin{aligned}\pi(\mathcal{X}|\theta) &= \left(\prod_{i=2}^{n} \pi(x_i|x_{i-1},\ldots,x_1,\theta)\right)\pi(x_1|\theta) \\ &= \left(\prod_{i=2}^{n} \pi(x_i|x_{i-1},\theta)\right)\pi(x_1|\theta), \qquad (1)\end{aligned}$$

and hence the posterior as

$$\begin{aligned}\pi(\theta|\mathcal{X}) &\propto \pi(\mathcal{X}|\theta)\pi(\theta) \\ &\propto \prod_{i=2}^{n}\left(\frac{\pi(x_i|x_{i-1},\theta)\pi(\theta)}{\pi(\theta)}\right)\pi(x_1|\theta)\pi(\theta) \\ &\propto \pi(\theta)^{(1-n)}\pi(\theta|x_1)\left(\prod_{i=2}^{n}\pi(\theta|x_i,x_{i-1})\right). \qquad (2)\end{aligned}$$

Essentially, the density of the posterior distribution of interest has been decomposed into a product of the densities of the posteriors $\theta|(x_i, x_{i-1})$, for $i = 2, \ldots, n$.

Exploting the Markovian Structure (2)

We can estimate

$$\pi(\theta|\mathcal{X}) \propto \pi(\theta)^{(1-n)}\pi(\theta|x_1)\left(\prod_{i=2}^{n}\pi(\theta|x_i, x_{i-1})\right)$$

as follows:

<u>PW-EBC</u> (PW-ABC)

1. For $i = 2$ to n do:
   (a) Apply EBC (ABC) Algorithm to draw exact (approximate) samples from $\pi(\theta|x_i, x_{i-1})$;
   (b) Calculate the kernel density estimate (KDE), $\hat{\pi}(\theta|x_i, x_{i-1})$
2. Calculate an estimate, $\hat{\pi}(\theta|\mathcal{X})$, of $\pi(\theta|\mathcal{X})$ by replacing in the equation above the densities $\pi(\theta|x_i, x_{i-1})$ with their corresponding KDEs, $\hat{\pi}(\theta|x_i, x_{i-1})$.

PW-EBC/PW-ABC

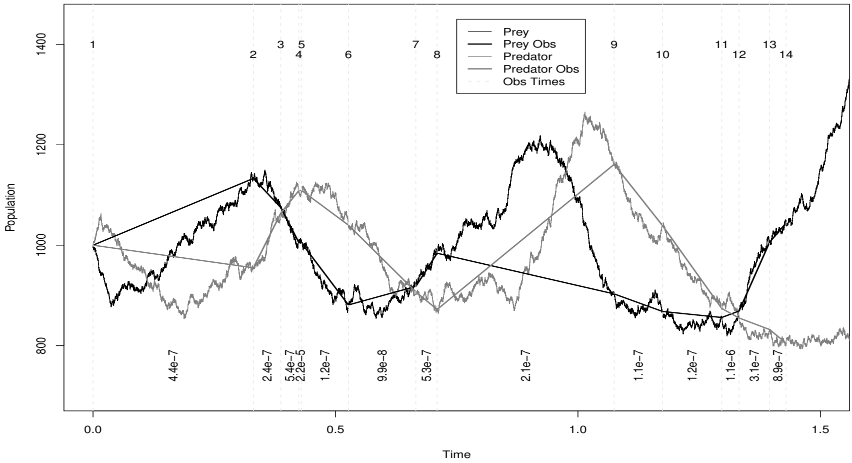
The good news:

- for a given tolerance ϵ, the **rejection sampling acceptance probability** is **substantially higher** when drawing samples from $\pi(\theta|x_i, x_{i-1})$ than it is from $\pi(\theta|\mathcal{X})$.
- Also benefit from being **trivial to parallelise:** factors can be calculated in parallel, and so can samples within factors.

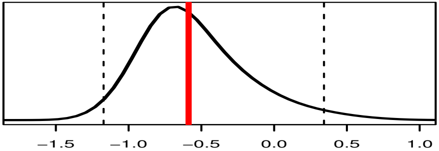
The not so good news:

- PW-EBC and PW-ABC involve having to **calculate kernel density estimates** (KDEs).
- *Sophisticated methods exist* for fast calculation (eg. Fast Fourier Transform) but can be problematic . . .
- For parameters with bounded support (e.g. parameters that can only be positive) to avoid *"edge effects"* we calculated KDEs for simple transformed versions of the parameters.

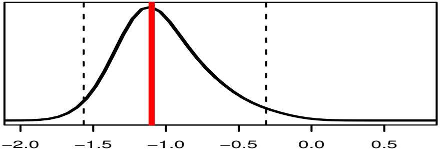
Application to a Stochastic Lotka-Voltera Model



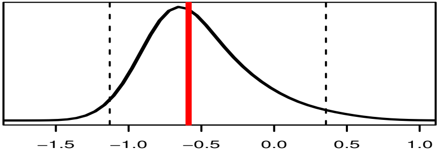
LV Model: An Example



(a) scaled logit Prey, logit($r_1/28$)



(b) scaled logit Interaction, logit($r_2/0.04$)



(c) scaled logit Predator, logit($r_3/28$)

Connection with Epidemic Models

- Some of the most commonly used epidemic models exhibit a Markovian structure (e.g. SI, SIR etc) ...
- therefore, such an approach seems suitable. → see Tj's talk later in the week.
- Note that we *are not drawing samples* from the posterior, but instead we calculate it pointwise.
- One can incorporate a sequential **sampling-based** framework similar to the one proposed by Chopin (2002) and Del Moral *et al* (2006).

Conclusions

Conclusions and Some Thoughts

- MCMC methods for stochastic epidemic models are considered as **"gold standard"**. Nevertheless, they can be *computationally expensive* and non-standard algorithms may be needed for efficiency {see for example, Neal and Roberts (2005), K (2007)}

- On the other hand, *exact* Monte-Carlo non MCMC inference for (some) final size data stochastic epidemic models is possible.

- The greatest at advantage of EBC/ABC-type methods is that they are **much much easier to** implement; on the other hand, careful tuning and care is required in most cases.

- It would be great if there was a way to quantify the error of approximation in ABC, or in other words to know how far we are from the *true* posterior ...

Acknowledgments

- **Final size data:** Frank Ball (Nottingham), Nikos Demiris (Athens)
- **Temporal data:** Phil O'Neill (Nottingham), Eleni Elia (Nottingham)
- **PW-EBC/ABC:** Simon White (BSU-MRC, Cambridge), Simon Preston (Nottingham).