What to do when exact Bayes is impossible? Some tools for approximate Bayesian inference

Umberto Picchini
picchini@chalmers.se
@uPicchini
https://umbertopicchini.github.io

Chalmers University of Technology and University of Gothenburg
Sweden

Notice, all text in pink (except this one) represents a URL that you can click.

For questions on today's topic I open 1 office hour on Wednesday 21 October at 11.00-12.00 (formally "lecture time"). Just connect at my zoom room `https://chalmers.zoom.us/my/picchini`

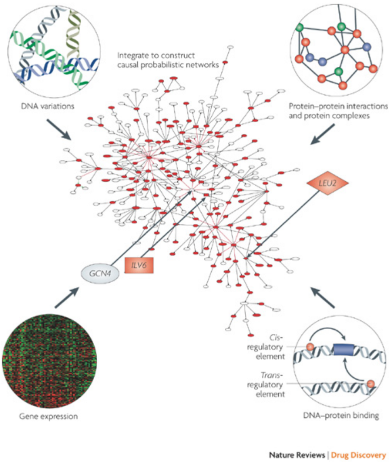
Also lookup other material (eg code) on the course page.

- I will briefly introduce a methodology that has literally revolutionised statistical inference for complex models in the last 10-15 years.
- For the last 30 years advancements in computer hardware have enabled modellers to become more and more ambitious.
- Complex models are needed to make sense of advanced experiments and multivariate (large) datasets.

However the advancements of statistical algorithms didn’t proceed at the same (fast) pace as hardware and modelling advancements.

We wanted to consider realistic model for our data, but often **we could not** because of the lack of flexible statistical methods.

Most real-life modelling is way more complex than examples from courses textbooks. The likelihood of the object below might be totally out of reach.


DNA variations
Integrate to construct causal probabilistic networks
Protein–protein interactions and protein complexes
LEU2
GCN4
ILV6
Cis-regulatory element
Trans-regulatory element
Gene expression
DNA–protein binding
Nature Reviews | Drug Discovery

[Pic from Schadt et al. (2009) doi:10.1038/nrd2826]

Say that you have a complex network, with many edges, as in the previous figure.

You would like to run a Gibbs sampler...however you do not know how to sample from all the conditionals.

Ok, fear not, you can introduce a Metropolis-Hastings step to sample from those conditional that are unknown.

...oops, it turns out those conditionals have an unknown density. We can't run Metropolis-within-Gibbs!.

More generally **we may not know the likelihood function explicitly** or it could be too computationally expensive to approximate.

Apparently we are stuck and can't use Bayesian methodology (or any likelihood-based methodology).
So what do we do?

What we typically want is the **likelihood function** for model parameters θ:

- We have some data: $\mathbf{y}^o$.
- the likelihood function: $p(\mathbf{y}^o|\theta)$
- We consider data as the outcome of some probabilistic model, and write $\mathbf{y}^o \sim p(\mathbf{y}|\theta = \theta_0)$
- θ_0 is the **unknown** *ground-truth* value of θ that generated the data.

Main issue

For realistically complex models, the likelihood function is **unavailable** in closed form.

Hence exact likelihood based inference is often **not possible**.

A paradigm shift is the concept of **generative model or simulator**.

You code a mathematical model $\mathcal{M}(\theta)$ as an idealized representation of the phenomenon under study.

$$\theta^* \to \mathcal{M}(\theta^*) \to \mathbf{y}^*$$

As long as we are able to run an instance of the model, we simulate/generate artificial data $\mathbf{y}^*$with $\mathbf{y}^* \sim p(\mathbf{y}^*|\theta = \theta^*)$.

So we have obtained a random *realization* $\mathbf{y}^*$ of the generative model $\mathcal{M}(\theta)$

Therefore the simulator $\mathcal{M}(\theta)$ defines the model pdf $p(\mathbf{y}|\theta)$ **implicitly**!

Trivial examples of generative models

Example 1: say that you get told that $y \sim N(\mu, \sigma^2)$ but **assume you do not know how to sample from a general Gaussian** (just assume...).

However, given knowledge of how to sample $u \sim N(0,1)$, then we also know how to **sample** from a generic Gaussian.

$$\mathcal{M} : y = \mu + \sigma \times u, \qquad u \sim N(0,1) \quad \text{independently}$$

then $y \sim N(\mu, \sigma^2)$ with $\theta = (\mu, \sigma)$.

So the equation above is a **generative model** $\mathcal{M}(\theta)$ for iid Gaussian draws. We only need some random input (u) and then we can generate draws.

Trivial examples of generative models

Example 2: stochastic Ricker model.

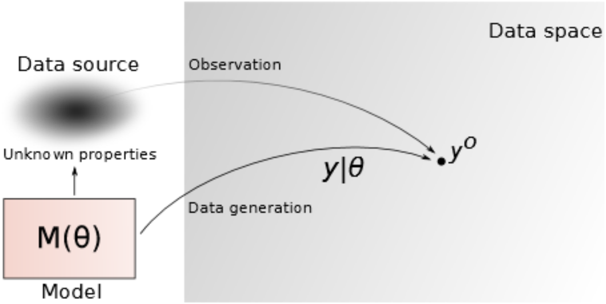
$$\mathcal{M}(\theta) = \begin{cases} \textbf{(observations):}\ y_t \sim_{indep.} \text{Poisson}(\phi N_t), & t = 1, ..., T \\ \textbf{(unobservable process):}\ N_t = r \cdot N_{t-1} \cdot e^{-N_{t-1}+e_t}, & e_t \sim_{iid} \mathcal{N}(0, \sigma^2) \end{cases}$$

This **state-space model** can be used to describe the evolution in time of a population of size N_t. We study this model later.

Even though its likelihood is **analytically intractable** and given by

$$p(y_{1:T}|r, \phi, \sigma) = \int \prod_{t=1}^{T} \big(p(y_t|N_t, r, \sigma)p(N_t|N_{t-1})\big) \cdot dN_1 \cdots dN_T$$

we can still simulate the $y_t \sim p(y|r, \phi, \sigma)$ from the (unknown) likelihood, because we can just run a computer model implementing $\mathcal{M}(\theta)$ at any $\theta = (r, \Phi, \sigma)$.

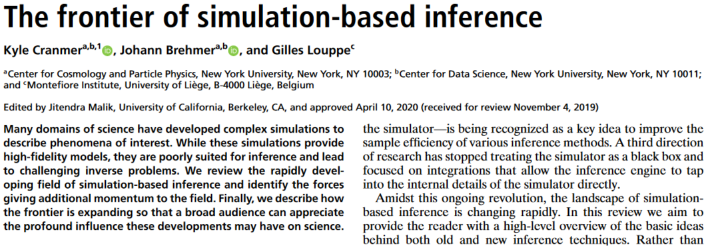


We can use simulations from the generative model to produce inference about θ, **without explicit knowledge of the likelihood $p(\mathbf{y}|\theta)$.**

This is at the basis of **likelihood-free methods**

We are entering simulation based inference. Lots of literature available.

Some interesting discussions in this review (a bit biased towards the authors' own research):



The frontier of simulation-based inference

Kyle Cranmer[a,b,1], Johann Brehmer[a,b], and Gilles Louppe[c]

[a]Center for Cosmology and Particle Physics, New York University, New York, NY 10003; [b]Center for Data Science, New York University, New York, NY 10011; and [c]Montefiore Institute, University of Liège, B-4000 Liège, Belgium

Edited by Jitendra Malik, University of California, Berkeley, CA, and approved April 10, 2020 (received for review November 4, 2019)

Many domains of science have developed complex simulations to describe phenomena of interest. While these simulations provide high-fidelity models, they are poorly suited for inference and lead to challenging inverse problems. We review the rapidly developing field of simulation-based inference and identify the forces giving additional momentum to the field. Finally, we describe how the frontier is expanding so that a broad audience can appreciate the profound influence these developments may have on science.

the simulator—is being recognized as a key idea to improve the sample efficiency of various inference methods. A third direction of research has stopped treating the simulator as a black box and focused on integrations that allow the inference engine to tap into the internal details of the simulator directly.

Amidst this ongoing revolution, the landscape of simulation-based inference is changing rapidly. In this review we aim to provide the reader with a high-level overview of the basic ideas behind both old and new inference techniques. Rather than

One of the possible simulation-based methods is ABC (approximate Bayesian Computation).

ABC, **a**pproximate **B**ayesian **c**omputation

ABC is probably the most studied likelihood-free methodology.

The first and simplest ABC algorithm is **acceptance-rejection sampling**.

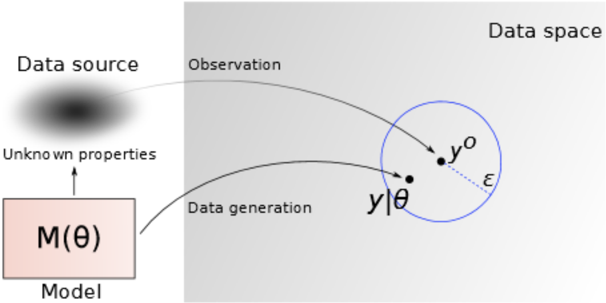
1. simulate from the prior $\theta^* \sim \pi(\theta)$
2. plug $\theta^* \rightarrow \mathcal{M}(\theta^*) \rightarrow \mathbf{y}^*$
3. if $\| \mathbf{y}^* - \mathbf{y}^o \| < \epsilon$ accept θ^* otherwise discard. Go to step 1 and repeat as many times as needed to obtain N accepted draws.

Each accepted pair $(\theta^*, \mathbf{y}^*)$ is from the augmented-posterior $\pi_\epsilon(\theta, \mathbf{y}^* | \mathbf{y}^o)$.

But we do not really care for $\mathbf{y}^*$, so if we retain only accepted θ^* then

$$\theta^* \sim \pi_\epsilon(\theta | \mathbf{y}^o)$$

No likelihood was explicitly involved, only implicitly via simulation!



Simulated data $\mathbf{y}^*$ inside the blue circle correspond to accepted parameters θ^*.

Which posterior are we targeting?

Acceptance-rejection sampling produces draws from the joint "augmented posterior" $\pi_\epsilon(\theta, \mathbf{y}^*|\mathbf{y}^o)$ where

$$\pi_\epsilon(\theta, \mathbf{y}^*|\mathbf{y}^o) \propto \mathbb{I}_\epsilon(\mathbf{y}^*, \mathbf{y}^o) p(\mathbf{y}^*|\theta^*)\pi(\theta^*)$$

where $\mathbb{I}_\epsilon(\mathbf{y}^*, \mathbf{y}^o)$ equals 1 if $\| \mathbf{y}^* - \mathbf{y}^o \| < \epsilon$ and 0 otherwise.

However, in reality we do not need to store the $\mathbf{y}^*$ (we can just discard those immediately after we have evaluated $\| \mathbf{y}^* - \mathbf{y}^o \| < \epsilon$), and then $\theta^* \sim \pi_\epsilon(\theta|\mathbf{y}^o)$ where

$$\pi_\epsilon(\theta|\mathbf{y}^o) \propto \pi(\theta^*) \int_{\mathcal{Y}} \mathbb{I}_\epsilon(\mathbf{y}^*, \mathbf{y}^o) p(\mathbf{y}^*|\theta^*) d\mathbf{y}^*$$

Say that θ^* has been accepted by the ABC acceptance-rejection sampling.
Then:

- if $\epsilon = 0$ then $\theta^* \sim \pi(\theta|\mathbf{y}^o)$, the **exact posterior**
- if $\epsilon = \infty$ then $\theta^* \sim \pi(\theta)$, the **prior**

Toy model

Let’s try something really trivial. We show how acceptance-rejection can easily become inefficient.

The Weibull distribution has two positive parameters, $a =$“shape” and $b =$“scale”.

This is a **tractable distribution**. We use a tractable case study as it is pedagogically useful, since we can compare ABC inference with exact inference.

Its pdf for $x > 0$ is

$$f(x) = (a/b)(x/b)^{a-1}\exp(-(x/b)^{a})$$

and 0 otherwise.

Suppose we have $n = 5$ i.i.d. observations $y_i \sim \mathit{Weibull}(2, 5)$.

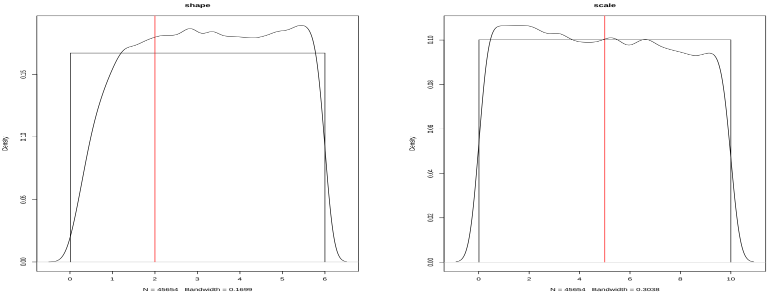
Want to estimate parameters of the Weibull, so $\theta_0 = (2, 5) = (a, b)$ are the true values.

- take $\parallel \mathbf{y}^o - \mathbf{y}^* \parallel = \sqrt{\sum_{i=1}^{n}(y_i^o - y_i^*)^2}$ (you can try a different distance, this is not really crucial).
- We'll use different thresholds ϵ.
- Run 50,000 iterations of acceptance-rejection.

Notice, in this case we prefix the number of iterations. So we do not know in advance how many accepted parameters we get. Instead in the algorithm we previously defined, I wrote that we keep repeating until a number N of acceptances is obtained.

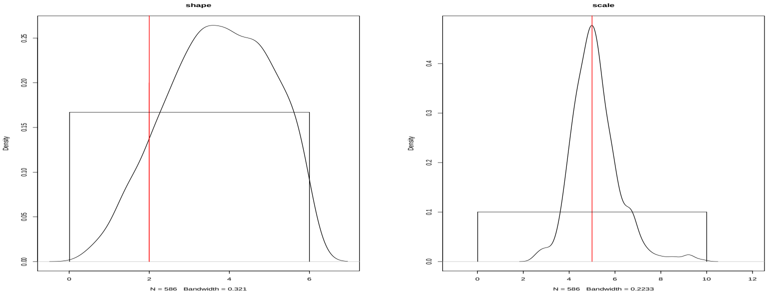
Wide priors for the “shape” parameter $a \sim U(0.01, 6)$ and “scale” $b \sim U(0.01, 10)$.

Try $\epsilon = 20$. True parameter values in red.



We are evidently sampling from the prior. Must reduce ϵ. About 92% draws were accepted. Way too large percentage!

Reduce ϵ from $\epsilon = 20$ to $\epsilon = 3$



About 1% of the produced simulations has been accepted.

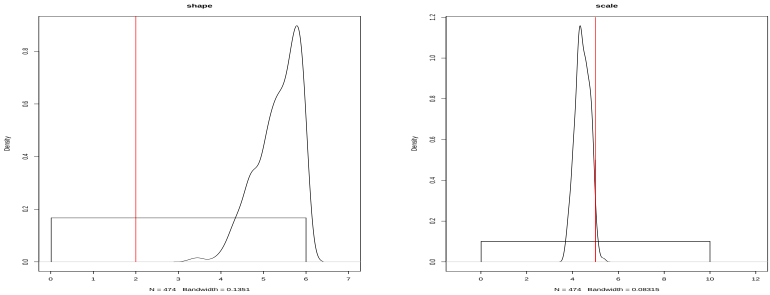
Of course $n = 5$ is a very small sample size, so inference quality is necessarily limited, but you got an idea of the ABC method.

An acceptance rate of about 1% is often implemented in many studies as a good tradeoff between computational effort and statistical accuracy.

Curse of dimensionality

- results will degrade for a larger sample size n because of a "necessarily too large" ϵ;
- even for a moderately long dataset $\mathbf{y}^o$, how likely is that we simulate a $\mathbf{y}^*$ such that $\sum_{i=1}^{n}(y_i - y_i^*)^2 < \epsilon$ for **small** ϵ? Very unlikely.
- inevitably, we'll be forced to enlarge ϵ thus degrading the quality of the inference.
- Serious trade-off between computational efficience and statistical precision.

Here we take $n = 200$. To compare with our “best” previous result, we use $\epsilon = 31$ (to obtain again a 1% acceptance rate on 50,000 iterations).



Notice `shape` is completely off!

The approach is just **not going to be of any practical use with large datasets**.

Break the curse of dimensionality

Compress data information using some **summary statistics** $S(\mathbf{y})$.

Example: $S(\mathbf{y})$ may contain sample mean, standard deviation, autocorrelations, quantiles etc.

Idea: instead of comparing $\mathbf{y}^o$ with $\mathbf{y}^*$, compare $S(\mathbf{y}^o)$ with $S(\mathbf{y}^*)$.

Requirements:

- $\mathbf{S}(\cdot)$ should be "informative" regarding θ, as we give up on using the full data $\mathbf{y}$.
- $\mathbf{S}(\cdot)$ should not be too large. Ideally $\dim(\mathbf{S}) \equiv \dim(\theta)$ [Fearnhed& Prangle '12].[1]

[1] Fearnhead & Prangle (2012). JRSS-B, 74(3), 419-474.

Acceptance-rejection with summaries (Pritchard et al.[2])

1. simulate from the prior $\theta^* \sim \pi(\theta)$
2. simulate $\mathcal{M}(\theta^*) \rightarrow \mathbf{y}^*$, compute $S(\mathbf{y}^*)$
3. if $\| S(\mathbf{y}^*) - S(\mathbf{y}^o) \| < \epsilon$ store θ^*. Go to step 1 and repeat as many times as needed to obtain N accepted draws.

Samples are from $\pi_\epsilon(\theta | S(\mathbf{y}^o))$

with

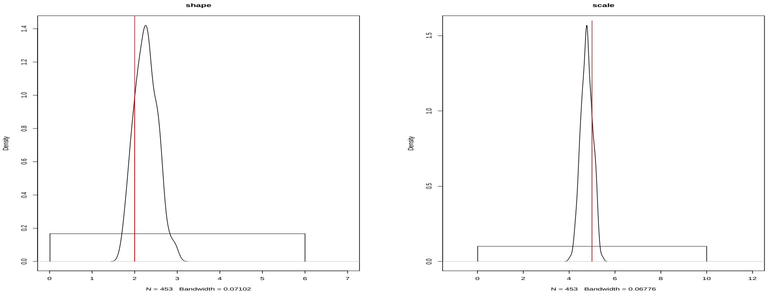
$$\pi_\epsilon(\theta | S(\mathbf{y}^o)) \propto \pi(\theta^*) \int_{\mathcal{Y}} \mathbb{I}_{A_{\epsilon,\mathbf{y}^o}}(\mathbf{y}^*) p(\mathbf{y}^* | \theta^*) d\mathbf{y}^*$$

$$A_{\epsilon,\mathbf{y}^o}(\mathbf{y}^*) = \{\mathbf{y}^* \in \mathcal{Y}; \| S(\mathbf{y}^*) - S(\mathbf{y}^o) \| < \epsilon\}.$$

[2] Pritchard et al. 1999, Molecular Biology and Evolution, 16:1791-1798.

Weibull example, reprise with $n = 200$

Set $S(\mathbf{y}) = ($sample mean of $\mathbf{y},$ sample SD of $\mathbf{y})$. Set $n = 200$ observations. Use $\epsilon = 0.35$.



This time we have captured **both shape and scale** (with 1% acceptance).

Also, enlarging n would not cause problems, thanks to $S(\cdot)$.

Using summary statistics clearly introduces a further level of approximation. Except when $S(\cdot)$ is *sufficient* for θ (carries the same info about θ as the whole $\mathbf{y}^o$).

When $S(\cdot)$ is a set of sufficient statistics for θ,

$$\pi_\epsilon(\theta|S(\mathbf{y}^o)) = \pi_\epsilon(\theta|\mathbf{y}^o)$$

However when the distribution of $\mathbf{y}^o$ is not in the *exponential family*, we basically have **no hope to construct sufficient statistics**.

For the sake of completeness: we could achieve exact inference under the rather stringent hypotheses that

1. S is sufficient, and simultaneously that
2. $\epsilon = 0$

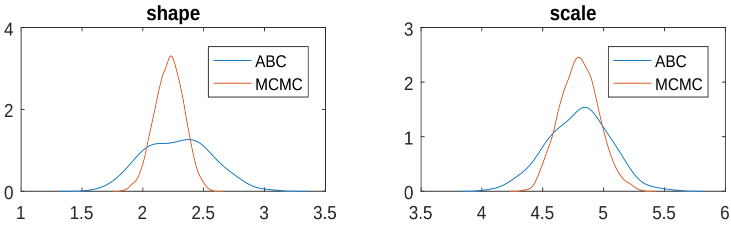
Note, (2) is **completely impractical** unless $\mathbf{y} \in \mathcal{Y}$, with $\mathcal{Y}$ a discrete set having few possible values.

ABC methodology is only apparently simple.

A large amount of research has been produced in the last 20 years to improve over the basic algorithm.

We will see an example soon, but first...

For this toy model, exact inference is possible.
Did we obtain an accurate approximation to the exact posterior?



Exact posteriors (via MCMC) are in red.

ABC has 2 (+1) sources of approximation:

- we used arbitrary non-sufficient statistics.
- $\epsilon > 0$
- Monte Carlo approximation due to finite samples (but that's obvious)

The apparent simplicity of ABC acceptance-rejection should not promote “lazy science”, just because it can be easily run.

At any time you should strive to use (or search for) the methods that allow you to implement exact inference.

ABC should be a last-resort if better methods are unavailable, since it is unknown how much approximate the ABC results will be (you typically cannot compare with exact inference, unlike with the previous toy model).

Beyond ABC rejection

ABC rejection is the simplest example of ABC algorithm.

It generates independent draws and can be coded into an **embarrassingly parallel algorithm**. However in can be very inefficient.

Parameters are proposed from the prior $\pi(\theta)$. **A prior does not exploit the information of already accepted parameters.**

Unless $\pi(\theta)$ is similar to the posterior, many proposals will be rejected for moderately small ϵ. This worsen with increasing dimension of θ.

A natural approach is to consider ABC within an MCMC algorithm.

In a MCMC with random walk proposals, the proposed parameter explores a neighbourhood of the last accepted parameter.

ABC-MCMC [Marjoram et al. 2003]

Integrating ABC within MCMC is **very simple** [Marjoram et al. 2003]

Notation: write $\mathbf{s}^* \equiv S(\mathbf{y}^*)$, $\mathbf{s}^o \equiv S(\mathbf{y}^o)$.
and $\mathbb{I}_\epsilon(\mathbf{s}^*, \mathbf{s}^o)$ equals 1 if $\|\mathbf{s}^* - \mathbf{s}^o\| < \epsilon$, and 0 otherwise.

1. sample proposal $\theta^* \sim q(\theta^*|\theta^\#)$
2. plug $\theta^* \to \mathcal{M}(\theta^*) \to \mathbf{y}^* \to \mathbf{s}^*$
3. compute acceptance ratio:

$$\text{ratio} := \frac{\mathbb{I}_\epsilon(\mathbf{s}^*, \mathbf{s}^o)\pi(\theta^*)}{\mathbb{I}_\epsilon(\mathbf{s}^\#, \mathbf{s}^o)\pi(\theta^\#)} \times \frac{q(\theta^\#|\theta^*)}{q(\theta^*|\theta^\#)}$$

4. simulate $u \sim U(0, 1)$, accept θ^* if $u <$ ratio.

The previous algorithm produces dependent samples from the "augmented" posterior $\pi_\epsilon(\theta, \mathbf{s}^*|\mathbf{s}^o)$.

This means that if we disregard $\mathbf{s}^*$, and retain only θ^*, we have

$$\theta^* \sim \pi_\epsilon(\theta|\mathbf{s}^o)$$

This is just another way to sample from an approximated posterior. Using a more informed proposal function than the prior.

Example: stochastic Ricker model

$$\begin{cases} \textbf{(observations):}\ y_t \sim \text{Poisson}(\phi N_t), \qquad t = 1, ..., T \\ \textbf{(unobservable process):}\ N_t = r \cdot N_{t-1} \cdot e^{-N_{t-1}+e_t}, \quad e_t \sim_{iid} \mathcal{N}(0, \sigma^2) \end{cases}$$

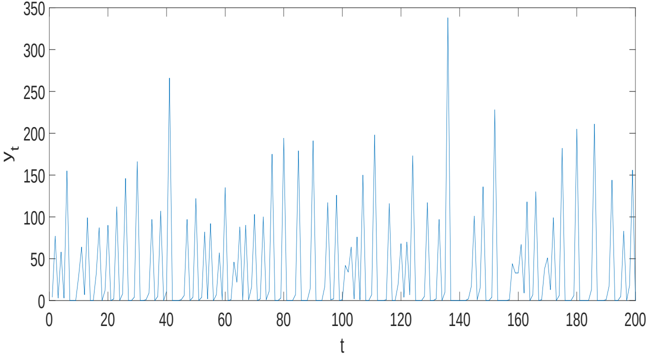
It can be used to describe the evolution in time of a population of size N_t.

- r is the intrinsic growth rate of the population;
- ϕ is a scale parameter
- e_t interpreted as environmental noise.

This is a state-space model, as the dynamics of $\{N_t\}$ are Markovian and we assume measurements $y_{1:T}$ to be conditionally independent given $\{N_t\}$.

the data

We simulated 200 time points from the model, with
$\log r = 3.8$, $\log \phi = 2.3$, $\log \sigma = -1.2$



Summary statistics

We used the 13 summary statistics suggested in Wood 2010. These include:

- the sample mean of observations $\bar{y}$;
- number of zeros in the dataset;
- autocovariances up to lag 5;
- and six more summaries...(not important to be mentioned here, see the reference above or the provided MATLAB code).

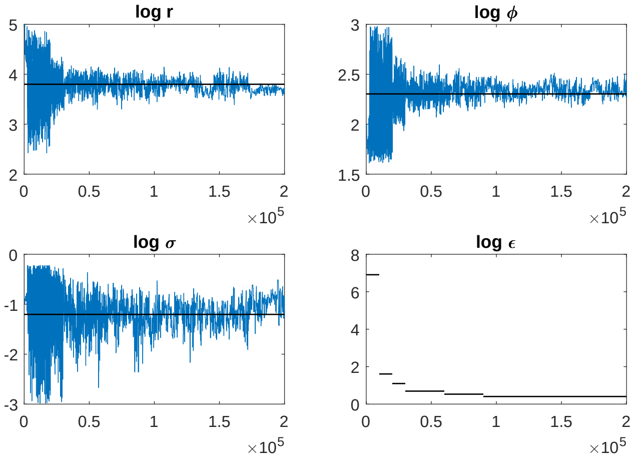
So we have $s(\mathbf{y}^o) = (\bar{y}, \#\text{zeros}, \text{autocov lag1}, ..., \text{autocov lag5}, ...)$.

Priors:

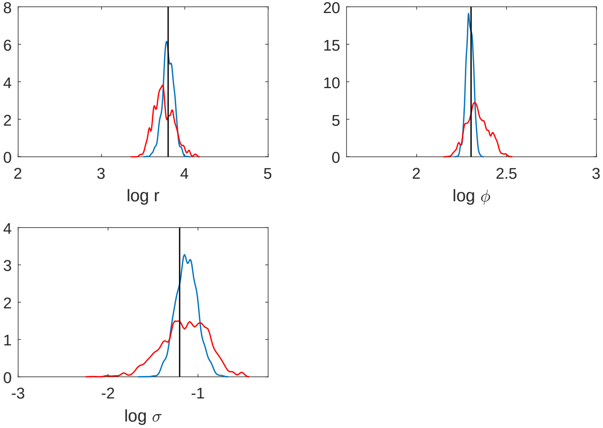
- $\log r \sim U(2,5)$
- $\log \Phi \sim U(1.61,3)$
- $\log \sigma \sim U(-3,-0.22)$

ABC-MCMC traces

We performed 200,000 ABC-MCMC iterations with decreasing ϵ_t.



ABC-MCMC is in red.



In blue is inference via “particle MCMC”. No time to talk about this but the blue one is basically exact inference (up to Monte Carlo error).

Preliminary conclusions

- ABC allows you to produce approximate inference for models having an intractable/unknown likelihood function.
- in our examples **we never needed to know the likelihood**;
- ABC only requires the ability to simulate artificial data, but is not a silver bullet.
- the main difficulty is how to specify summary statistics that are “informative” for θ.
- when summaries are not informative and ϵ is too large results will be poor.
- tuning ABC is not straightforward. Many available resources though, see the final slides...

Semi-automatic summary statistics

To date the most important study on the construction of summaries in ABC is in Fearnhead-Prangle 2012[3]. Recall a well-known result: consider the class of quadratic losses

$$L(\theta_0, \hat{\theta}; A) = (\theta_0 - \hat{\theta})^T \cdot A \cdot (\theta_0 - \hat{\theta})$$

with θ_0 true value of a parameter and $\hat{\theta}$ an estimator of θ. A is a positive definite matrix.

The minimal expected quadratic loss $E(L(\theta_0, \hat{\theta}; A) \mid y)$ is achieved via $\hat{\theta} = E_{ABC}(\theta \mid S(y))$ as $\epsilon \to 0$, where $S(y) = E(\theta \mid y)$.

That is to say, as $\epsilon \to 0$, we minimize the expected posterior loss using the ABC posterior expectation (if $S(y) = E(\theta|y)$). However $E(\theta \mid y)$ **is unknown**.

[3] Fearnhead and Prangle (2012).

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

- Consider vector $\theta = (\theta_1, ..., \theta_j, ..., \theta_p)$
- for the jth parameter in θ fit **separately** the linear regression models

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

  [e.g. $S_j(y) = \hat{\beta}_0^{(j)} + \hat{\beta}^{(j)}\eta(y) = \hat{\beta}_0^{(j)} + \hat{\beta}_1^{(j)}y_0 + \cdots + \hat{\beta}_n^{(j)}y_n$ or you can let $\eta(\cdot)$ contain powers of y, say $\eta(y, y^2, y^3, ...)$]

- repeat the fitting separately for each θ_j.
- hopefully $S_j(y) = \hat{\beta}_0^{(j)} + \hat{\beta}^{(j)}\eta(y)$ will be "informative" for θ_j.
- Do the above separately for each j. The full vector of summary statistics will be $S(y) = (S_1(y), ..., S_p(y))$.

An example (run before ABC-MCMC):

1. $p = \dim(\theta)$. Simulate from the prior $\theta^* \sim \pi(\theta)$ (not very efficient...)
2. using θ^*, generate $y^* = (y_1^*, ..., y_n^*)$ from your model.

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

$$\begin{bmatrix} \theta_1^{(1)} & \theta_2^{(1)} & \cdots \theta_p^{(1)} \\ \theta_1^{(2)} & \theta_2^{(2)} & \cdots \theta_p^{(2)} \\ \vdots & & \end{bmatrix}, \qquad \begin{bmatrix} y_1^{*(1)} & y_2^{(*1)} & \cdots & y_n^{(*1)} \\ y_1^{(*2)} & y_2^{(*2)} & \cdots & y_n^{(*2)} \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix}$$

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

$$\begin{bmatrix} \theta_j^{(1)} \\ \theta_j^{(2)} \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & y_1^{(*1)} & y_2^{(*1)} & \cdots & y_n^{(*1)} \\ 1 & y_1^{(*2)} & y_2^{(*2)} & \cdots & y_n^{(*2)} \\ \vdots & \vdots & \cdots & \vdots & \end{bmatrix} \times \beta_j \qquad (j = 1, ..., p),$$

and obtain a statistic for θ_j, defined as $S_j(\cdot) = \hat{\beta}_0^{(j)} + \hat{\beta}^{(j)}\eta(\cdot)$.

You need to do the previous procedure only once and for all, before executing ABC.

Once you have obtained the $\hat{\beta}$, use these same coefficients when calculating summaries for any new simulated data y^* and actual data y^o, i.e.

$$S_j(y^o) = \hat{\beta}_0^{(j)} + \hat{\beta}^{(j)}\eta(y^o)$$
$$S_j(y^*) = \hat{\beta}_0^{(j)} + \hat{\beta}^{(j)}\eta(y^*)$$

In Picchini 2013 I used this approach to select summaries for state-space models defined by stochastic differential equations.

A (trivial) generalization of ABC-MCMC

Marjoram et. al used $\mathbb{I}_\epsilon(\mathbf{s}^*, \mathbf{s}^o)$ to compare summaries, however any positive kernel $J_\epsilon(\mathbf{s}^*, \mathbf{s}^o)$ will do.

Examples:

- Gaussian kernel: $J_\epsilon(\mathbf{s}^*, \mathbf{s}^o) \propto \frac{1}{\epsilon} \cdot \exp(-(\mathbf{s}^* - \mathbf{s}^o)'W^{-1}(\mathbf{s}^* - \mathbf{s}^o)/2\epsilon^2)$
- Epanechnikov kernel (look it up);

And of course the value of ϵ may change depending on the kernel you use.

Then the acceptance probability trivially generalizes to[4]

$$\alpha = \min\left\{1, \frac{J_\epsilon(\mathbf{s}^*, \mathbf{s}^o)\pi(\theta^*)}{J_\epsilon(\mathbf{s}^\#, \mathbf{s}^o)\pi(\theta^\#)} \times \frac{q(\theta^\#|\theta^*)}{q(\theta^*|\theta^\#)}\right\}.$$

This is still a likelihood-free approach.

[4] Sisson and Fan (2010), chapter in Handbook of Markov chain Monte Carlo.

Choice of the threshold ϵ

We would like to use a “small” $\epsilon > 0$, however it turns out that if you start at a bad value of θ a small ϵ will cause many rejections.

- start with a fairly large ϵ allowing the chain to move in the parameters space.
- after some iterations reduce ϵ so the chain will explore a (narrower) and more precise approximation to $\pi(\theta|\mathbf{s}^o)$
- keep reducing (slowly) ϵ. Use the set of θ’s accepted using the smallest ϵ to report inference results.

It’s not obvious how to determine the sequence of $\epsilon_1 > \epsilon_2 > ... > \epsilon_k > 0$. If the sequence decreases too fast there will be many rejections (chain suddenly trapped in some tail).

It’s a problem similar to tuning the “temperature” in optimization via simulated annealing.

Choice of the threshold ϵ

A possibility:

- Say that you have completed a number of iterations via ABC-MCMC or via rejection sampling using ϵ_1, and say that you stored all distances $d_{\epsilon_1} = \| S(\mathbf{y}^o) - S(\mathbf{y}^*) \|$ obtained using ϵ_1.
- Take the xth percentile of such d_{ϵ_1} distances and set a new threshold ϵ_2 as $\epsilon_2 := \min\{\epsilon_1, x\text{th percentile of } d_{\epsilon_1}\}$.
- this way $\epsilon_2 < \epsilon_1$. So now you can use ϵ_2 to conduct more simulations, then similarly obtain $\epsilon_3 := \min\{\epsilon_2, x\text{th percentile of } d_{\epsilon_2}\}$ etc.
- Depending on x the decrease from a ϵ to another ϵ' will be more or less fast. Setting say $x = 20$ will cause a sharp decrease, while $x = 90$ will let the threshold decrease more slowly.
- A slow decrease of ϵ is safer but implies longer simulations before reaching acceptable results.

Alternatively just to set the sequence of ϵ's by trial and error.

When do we stop decreasing ϵ?

- Several studies have shown that a 1% acceptance rate (at the smallest ϵ) is a good compromise between accuracy and computational needs.

  This is also my experience.
- However recall that a “small” ϵ implies many rejections→you’ll have to run a longer simulation to obtain enough acceptances to enable inference.
- ABC, unlike exact MCMC, *does require* a small acceptance rate. This is needed by its own nature as we are not happy to use a large ϵ producing inaccurate inferences.
- A high acceptance rate denotes that your ϵ is way too large.

Other likelihood-free methods

Likelihood-free methods date back to at least Diggle and Gratton (1984) and Rubin (1984, p. 1160)

More recent examples:

- **Indirect Inference** (Gourieroux and Ronchetti 1993);
- for state-space models (Markov processes observed with noise) the **bootstrap filter** of Gordon, Salmond and Smith (1993)
- **Synthetic Likelihoods** method of Wood (2010)
- Lots of more recent machine learning literature: see this review.

Are approximations any worth?

Why should we care about approximate methods?

Well, we know the most obvious answer: it's because this is what we do when exact methods are impractical. No big news...

But I am more interested in the following phenomenon, which I noticed by direct experience:

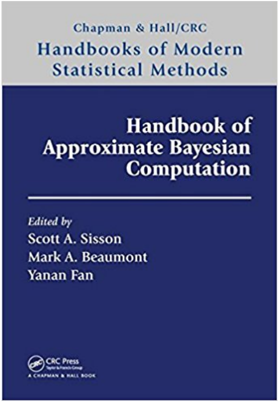
- Many scientists seem to get intellectual fulfilment by using exact methods, leading to exact inference.
- What we might not see is when they fail to communicate that they (consciously or unconsciously) pushed themselves to formulate simpler models (too simple?!), **so that exact inference could be achieved**.

Gelman and Rubin, 1996

“[...] as emphasized in Rubin (1984), one of the great scientific advantages of simulation analysis of Bayesian methods is the freedom it gives the researcher to formulate appropriate models **rather than be overly interested in analytically neat but scientifically inappropriate models**.”

The ABC methodology is quite mature but still evolving.
Comprehensive 2018 monography (many chapters available on arxiv):

Sisson, Fan, Beaumont. (2018). Handbook of approximate Bayesian computation. Chapman and Hall/CRC.



Chapman & Hall/CRC
Handbooks of Modern Statistical Methods

Handbook of Approximate Bayesian Computation

Edited by
Scott A. Sisson
Mark A. Beaumont
Yanan Fan

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Blog posts and slides (coloured links are clickable)

1. Christian P. Robert often blogs about ABC (and beyond: it's a fantastic blog!)
2. an intro to ABC by Darren J. Wilkinson
3. Two blog posts by Rasmus Bååth here and here
4. Tons of slides at Slideshare.

Software (coloured links are clickable)

- EasyABC, R package. Research article.
- abc, R package. Research article
- abctools, R package. Research article. Focusses on tuning.
- A list with more options here .
- examples with implemented model simulators (useful to incorporate in your programs).

Reviews

Accessible reviews:

1. Sunnåker et al. 2013
2. (with applications in ecology) Hartig et al. 2013

Fairly extensive reviews:

1. Sisson and Fan 2010
2. (with applications in ecology) Beaumont 2010
3. Marin et al. 2010

Review specific for dynamical models:

1. Jasra 2015

Determination of summary statistics

1. review paper by Blum et al. 2013 on dimension reduction methods for ABC;
2. Fearnhed and Prangle 2012 (a JRSS-B discussion paper).
3. Wiqvist et al. 2019 using deep learning.