An intro to ABC – approximate Bayesian computation

PhD course FMS020F–NAMS002 "Statistical inference for partially observed stochastic processes", Lund University

http://goo.gl/sX8vU9

Umberto Picchini
Centre for Mathematical Sciences,
Lund University
www.maths.lth.se/matstat/staff/umberto/

In this lecture we consider the case where it is not possible to pursue exact inference for model parameters θ, nor it is possible to approximate the likelihood function of θ within a given computational budget and available time.

The above is not a rare circumstance.

Since the advent of affordable computers and the introduction of advanced statistical methods, researchers have become increasingly ambitious, and try to formulate and fit very complex models.

Example: MCMC (Markov chain Monte Carlo) has provided a universal machinery for Bayesian inference since its rediscovery in the statistical community in the early 90's.

Thanks to MCMC (and related methods) scientists' ambitions have been pushed further and further.

However for complex models (and/or large datasets) MCMC is often impractical. Calculating the likelihood, or an approximation thereof might be impossible.

For example in spatial statistics INLA (integrated nested Laplace approximation) is a welcome alternative to the more expensive MCMC.

Also MCMC is not *online*: when new observations arrive we have to re-compute the *whole* likelihood for the total set of observations, i.e. we can't make use of the likelihood computed at previous observations.

Particle marginal methods (particle MCMC) are a fantastic possibility for exact Bayesian inference for state-space models. But what can we do for non-state space models?

And what can we do when the dimension of the mathematical system is large and the implementation of particle filters with millions of particles is infeasible?

There is an increasingly interest in statistical methods for models that are easy to simulate from, but for which it is impossible to calculate transition densities or likelihoods.

General set-up: we have a complex stochastic process $\{X_t\}$ with unknown parameters θ. For any θ we can simulate from this process.

We have observations $y = f(\{X_{0:T}\})$.

We want to estimate θ but we cannot calculate $p(y|\theta)$, as this involves integrating over the realisations of $\{X_{0:T}\}$.

Notice we are not specifying the probabilistic properties of $X_{0:T}$ nor Y. We are certainly not restricting ourselves to state-space models.

The likelihood-free idea

Likelihood-free inference motivating idea:

- Easy to simulate from model conditional on parameters.
- So run simulations for many parameters.
- See for which parameter value the simulated data sets match observed data best

Different 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);
- Approximate Bayesian Computation (ABC) (a review is Marin et al. 2011);
- bootstrap filter of Gordon, Salmond and Smith (1993)
- Synthetic Likelihoods method of Wood (2010)

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 on 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, **so that exact inference could be achieved**.

So the pattern I often notice is:

1. You have a complex scenario, noisy data, unobserved variables etc
2. you formulate a pretty realistic model... which you can't fit to data (i.e. exact inference is not possible)
3. you simplify the model (a lot) so it is now tractable with exact methods.
4. You are happy.

However you might have simplified the model a *wee* too much to be realistic/useful/sound.

John Tukey – 1962

"Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise. "

If a complex model is the one I want to use to answer the right question, then I prefer to obtain an approximative answer using approximate inference, than fooling myself with a simpler model using exact inference.

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."

Approximate Bayesian Computation and *Synthetic Likelihoods* are two approximate methods for inference, with ABC vastly more popular and with older origins.

We will discuss ABC only.

Features of ABC

- only need a *generative model*, i.e. the model we assumed having generated available data y.
- only need to be able to simulate from such a model.
- in other words, we do not need to assume anything regarding the probabilistic features of the model components.
- particle marginal methods also assume the ability to simulate from the model, but also **assume a specific model structure**, usually a state-space model (SSM).
- also, particle marginal methods for SSM **require at least knowledge of** $p(y_t|x_t;\theta)$ (to compute importance weights). What do we do without such requirement?

For the moment we can denote data with $\mathbf{y}$ instead of, say, $\mathbf{y_{1:T}}$ as what we are going to introduce is not specific to dynamical models.

Bayesian setting: target is $\pi(\theta|y) \propto p(y|\theta)\pi(\theta)$

What to do when (1) the likelihood $p(y|\theta)$ is unknown in closed form and/or (2) it is expensive to approximate?

Notice that if we are able to simulate observations y^* by *running the generative model*, then we have

$$y^* \sim p(y|\theta)$$

That is y^* is produced by the statistical model that generated observed data y.

(i) Therefore if $\mathcal{Y}$ is the space where y takes values, then $y^* \in \mathcal{Y}$.

(ii) y and y^* have the same dimension.

Loosely speaking...

Example: if we have a SSM and given a parameter value θ and x_{t-1} simulate x_t, then plug x_t in the observation equation ans simulate y_t^*, then I have that $y_t^* \sim p(y_t|\theta)$.

This is because if I have two random variables x and y with joint distribution (conditional on θ) $p(y,x|\theta)$ then $p(y,x|\theta) = p(y|x;\theta)p(x|\theta)$.

I first simulate x^* from $p(x|\theta)$, then conditional on x^* I simulate y^* from $p(y|x^*,\theta)$.

What I obtain is a draw (x^*,y^*) from $p(y,x|\theta)$ hence y^* alone must be a draw from the marginal $p(y|\theta)$.

Likelihood free rejection sampling

1. simulate from the prior $\theta^* \sim \pi(\theta)$
2. plug θ^* in your model and simulate a y^* [this is the same as writing $y^* \sim p(y|\theta^*)$]
3. if $y^* = y$ store θ^*. Go to step 1 and repeat.

The above is a *likelihood free* algorithm: it **does not require knowledge of the expression of** $p(y|\theta)$.

Each accepted θ^* is such that $\theta^* \sim \pi(\theta|y)$ **exactly**.

We justify the result in next slide.

Justification

The previous algorithm is exact. Let's see why.

Denote with $f(\theta^*, y^*)$ the joint distribution of the **accepted** (θ^*, y^*). We have that

$$f(\theta^*, y^*) = p(y^*|\theta^*)\pi(\theta^*)\mathbb{I}_y(y^*)$$

with $\mathbb{I}_y(y^*) = 1$ iff $y^* = y$ and zero otherwise. Marginalizing y^* we have

$$f(\theta^*) = \int_{\mathcal{Y}} p(y^*|\theta^*)\pi(\theta^*)\mathbb{I}_y(y^*)dy^* = p(y|\theta^*)\pi(\theta^*) \propto \pi(\theta^*|y)$$

hence all accepted θ^* are drawn from the exact posterior.

Curse of dimensionality

Algorithmically the rejection algorithm could be coded in a `while loop`, that would repeat itself until the equality condition is satisfied.

For y taking discrete values in a "small" set of states this is manageable.

For y a long sequence of observations from a discrete random variables with many states this is very challenging.

For y a continuous variable the equality happens with probability zero.

ABC rejection sampling (Tavare et al.[1])

Attack the curse of dimensionality by introducing an approximation.
Take an arbitrary distance $\| \cdot \|$ and a threshold $\epsilon > 0$.

1. simulate from the prior $\theta^* \sim \pi(\theta)$
2. simulate a $y^* \sim p(y|\theta^*)$
3. if $\| y^* - y \| < \epsilon$ store θ^*. Go to step 1 and repeat.

Each accepted θ^* is such that $\theta^* \sim \pi_\epsilon(\theta|y)$.

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

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

[1] Tavare et al. 1997. Genetics;145(2)

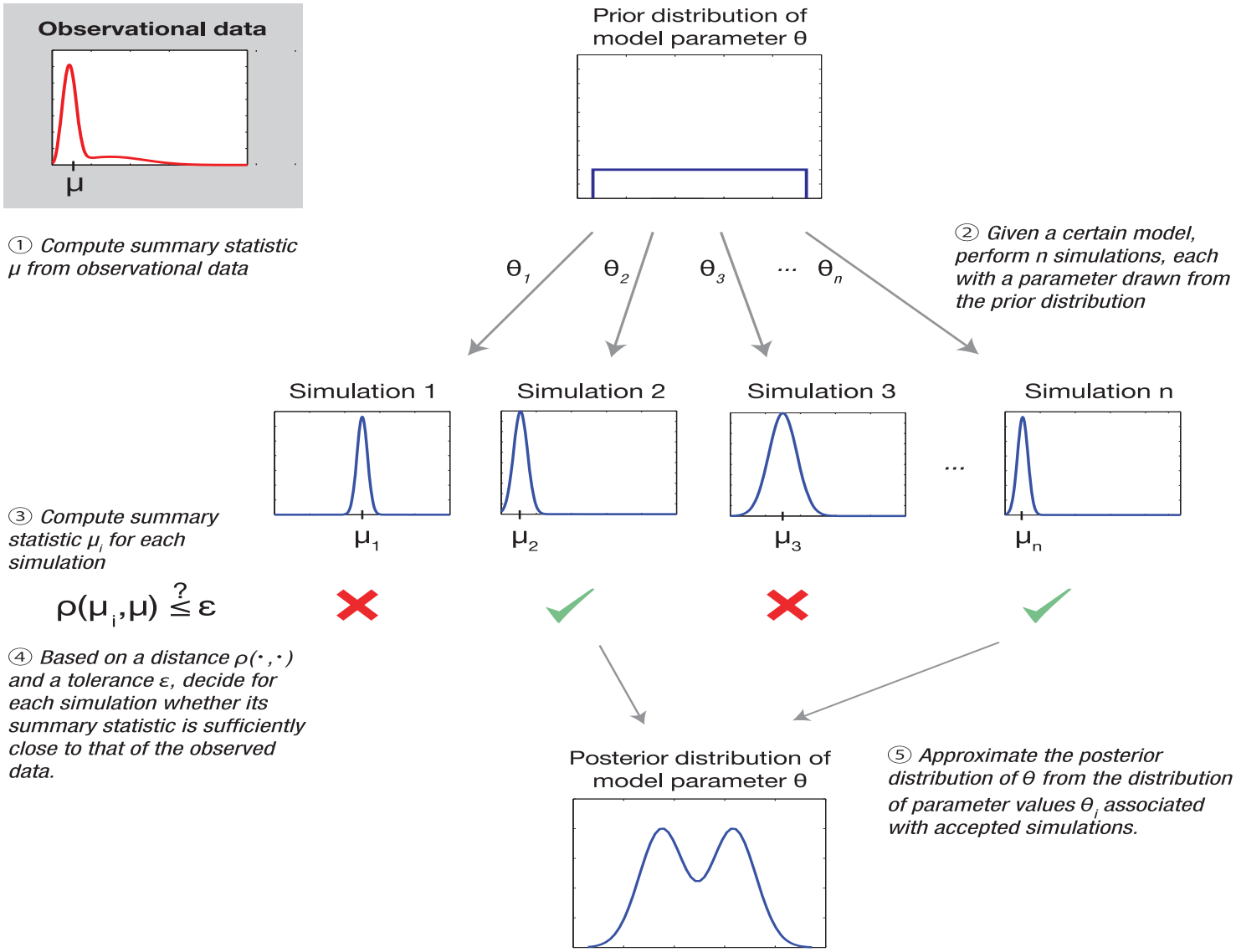


Figure 1. Parameter estimation by Approximate Bayesian Computation: a conceptual overview.
doi:10.1371/journal.pcbi.1002803.g001

It is self evident that when imposing $\epsilon = 0$ we force $y^* = y$ thus implying that draws will be, again, from the true posterior.

However in practice imposing $\epsilon = 0$ might require unbearable computational times to obtain a single acceptance. In practice we have to set $\epsilon > 0$, so that draws are from the approximate posterior $\pi_\epsilon(\theta|y)$.

Important ABC result

Convergence "in distribution":

- when $\epsilon \to 0$, $\quad \pi_\epsilon(\theta|y) \to \pi(\theta|y)$
- when $\epsilon \to \infty$, $\quad \pi_\epsilon(\theta|y) \to \pi(\theta)$

Essentially for a too large ϵ we learn nothing.

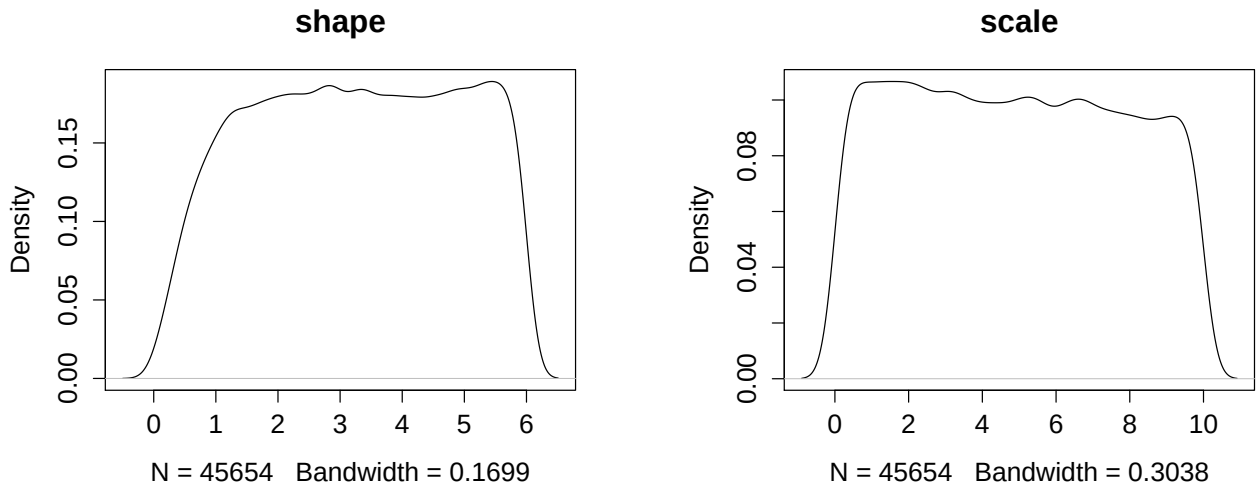
Toy model

Let's try something really trivial. We show how ABC rejection can become easily inefficient.

- $n = 5$ i.i.d. observations $y_i \sim Weibull(2, 5)$
- want to estimate parameters of the Weibull, so $\theta = (2, 5) = (a, b)$ are the true values.
- take $\| y - y^* \| = \sum_{i=1}^{n} (y_i - y_i^*)^2$ (you can try a different distance, this is not really crucial)
- let's use different values of ϵ
- run 50,000 iterations of the algorithm.

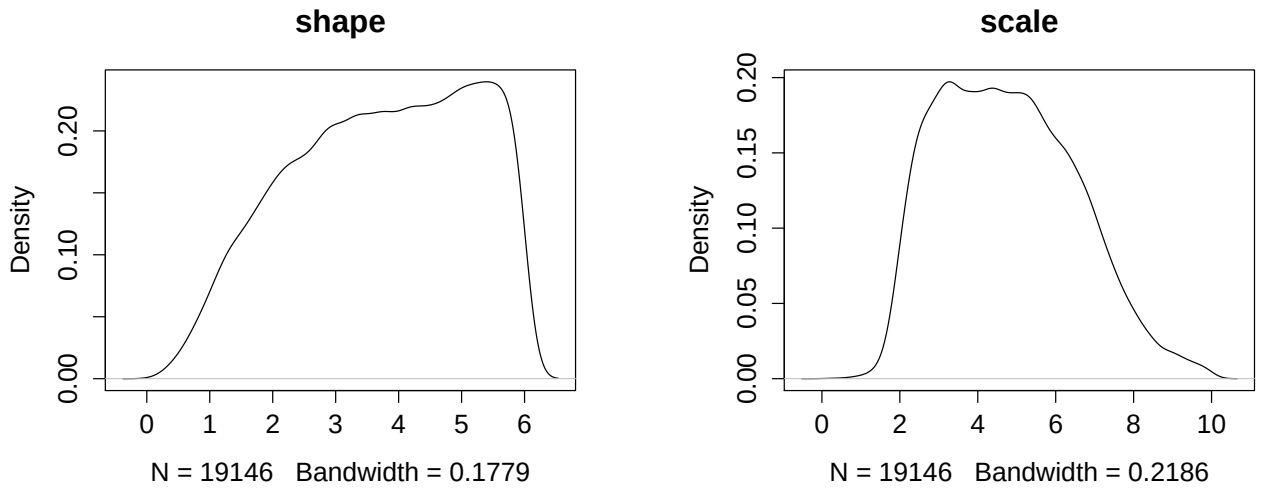
We assume wide priors for the "shape" parameter $a \sim U(0.01, 6)$ and for the "scale" $b \sim U(0.01, 10)$.

Try $\epsilon = 20$



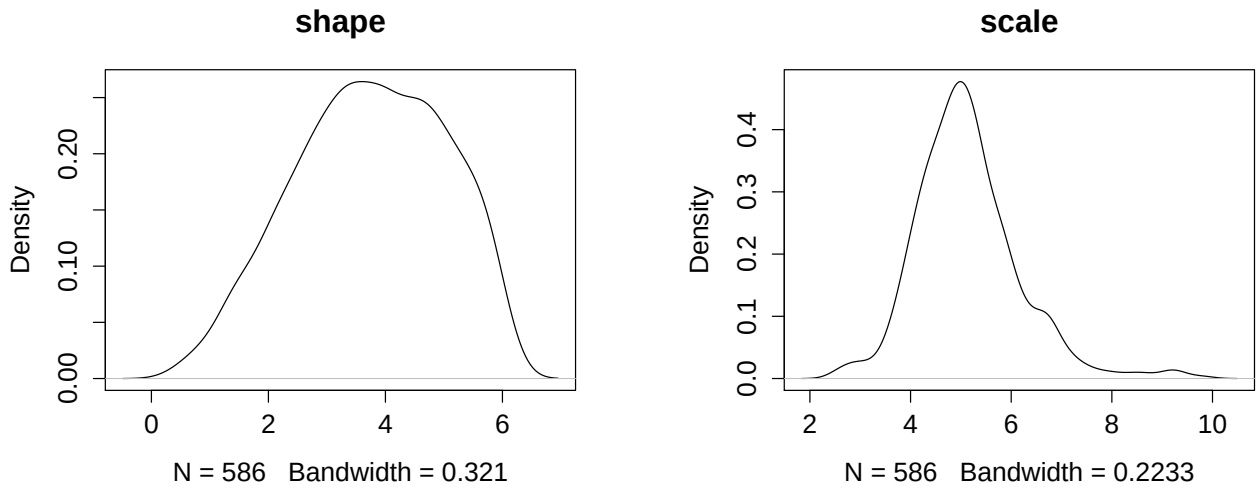
We are evidently sampling from the prior. Must reduce ϵ. In fact notice about 46,000 draws were accepted.

Try $\epsilon = 7$



Here about 19,000 draws were accepted (38%).

Try $\epsilon = 3$



Here about 1% of the produced simulations has been accepted. Recall true values are $(a, b) = (2, 5)$.

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

An idea for self-study

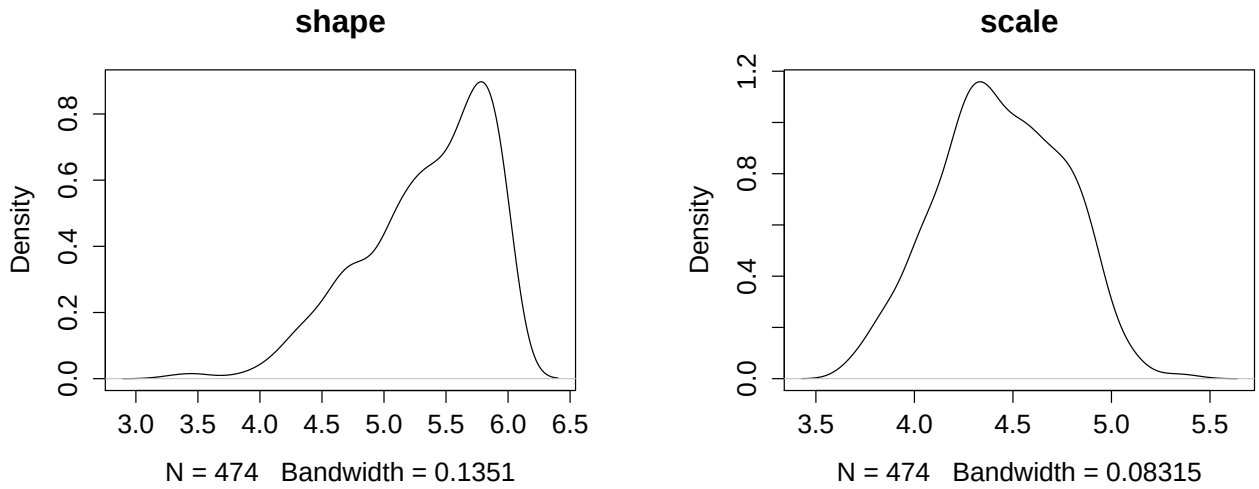
Compare the ABC (marginal) posteriors with exact posteriors from some experiment using conjugate priors.

For example see http://www.johndcook.com/CompendiumOfConjugatePriors.pdf

Curse of dimensionality

- It becomes immediately evident that results will soon degrade for a larger sample size n
- even for a moderately long dataset y, how likely is that we produce a 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.

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 (true value is 2).

The approach is just **not going to be of any practical use with continuous data**.

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

Same as before, but comparing $S(y)$ with $S(y^*)$ for "appropriate" summary statistics $S(\cdot)$.

1. simulate from the prior $\theta^* \sim \pi(\theta)$
2. simulate a $y^* \sim p(y|\theta^*)$, compute $S(y^*)$
3. if $\| S(y^*) - S(y) \| < \epsilon$ store θ^*. Go to step 1 and repeat.

Samples are from $\pi_\epsilon(\theta|S(y))$ with

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

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

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

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 y).

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

$$\pi_\epsilon(\theta|S(y)) = \pi_\epsilon(\theta|y)$$

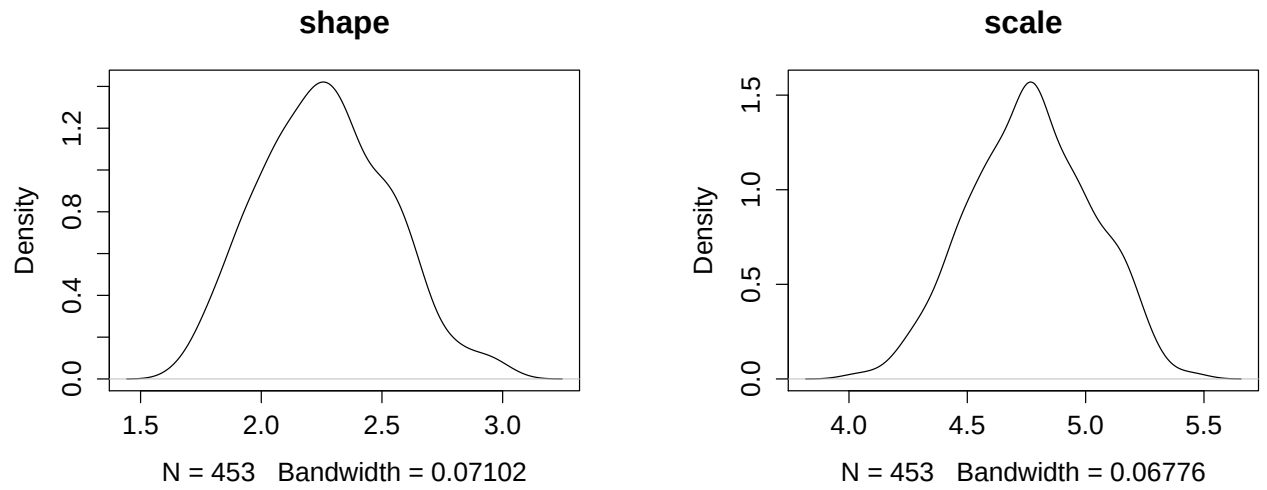
But then again when y is not in the exponential family, we basically have no hope to construct sufficient statistics.

- A central topic in ABC is to construct "informative" statistics, as a replacement for the (unattainable) sufficient ones.
- Important paper, Fearnhead and Prangle 2012 (discussed later).

If we have "good summaries" we can bypass the curse of dimensionality problem.

Weibull example, reprise

Take $n = 200$. Set $S(y) = (\text{sample mean y, sample SD y})$ and similarly for y^*. Use $\epsilon = 0.35$.



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

Also, enlarging n would not cause problems $\rightarrow$ robust comparisons

From now on we silently assume working with $S(y^*)$ and $S(y)$, and if we wish not to summarize anything we can always set $S(y) := y$.

A main issue in ABC research is that when we use an arbitrary $S(\cdot)$ we can't quantify "how much off" we are from the ideal sufficient statistic.

Important work on constructing "informative" statistics:

- Fearnhead and Prangle 2012, JRSS-B 74(3).
- review by Blum et al 2013, Statistical Science 28(2).

Michael Blum will give a free workshop in Lund on 10 March. Sign up here!

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 massively inefficient.

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

Unless $\pi(\theta)$ is somehow similar to $\pi_\epsilon(\theta|y)$ many proposals will be rejected for moderately small ϵ.

This is especially true for a large dimensional 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

Consider the approximated **augmented** posterior:

$$\pi_\epsilon(\theta, y^*|y) \propto J_\epsilon(y^*, y) \underbrace{p(y^*|\theta)\pi(\theta)}_{\propto \pi(\theta|y^*)}$$

- $J_\epsilon(y^*, y)$ a function which is a positive constant when $y = y^*$ (or $S(y) = S(y^*)$) and takes large positive values when $y^* \approx y$ (or $S(y) \approx S(y^*)$).
- $\pi(\theta|y^*)$ the (intractable) posterior corresponding to artificial observations y^*.
- when $\epsilon = 0$ we have $J_\epsilon(y^*, y)$ constant and $\pi_\epsilon(\theta, y^*|y) = \pi(\theta|y)$.

Without loss of generality, let's assume that $J_\epsilon(y^*, y) \propto \mathbb{I}_y(y^*)$, the indicator function.

ABC-MCMC (Marjoram et al. [3])

We wish to simulate from the posterior $\pi_\epsilon(\theta, y^*|y)$: hence construct proposals for both θ and y^*.

- Present state is $\theta^\#$ (and corresponding $y^\#$). Propose $\theta^* \sim q(\theta^*|\theta^\#)$.
- Simulate y^* from the model given θ^* hence the proposal is the model itself, $y^* \sim p(y^*|\theta^*)$.

The acceptance probability is thus:

$$\alpha = \min\left\{1, \frac{\mathbb{I}_y(y^*)p(y^*|\theta^*)\pi(\theta^*)}{1 \times p(y^\#|\theta)\pi(\theta^\#)} \times \frac{q(\theta^\#|\theta^*)p(y^\#|\theta^\#)}{q(\theta^*|\theta^\#)p(y^*|\theta^*)}\right\}$$

The "1" at the denominator it's there because of course we must start the algorithm at some admissible (accepted) $y^\#$, hence the denominator will always have $\mathbb{I}_y(y^\#) = 1$.

[3] Marjoram et al. 2003, PNAS 100(26).

By considering the simplification in the previous acceptance probability we have the ABC-MCMC:

1. Last accepted parameter is $\theta^{\#}$ (and corresponding $y^{\#}$). Propose $\theta^* \sim q(\theta^*|\theta^{\#})$.
2. generate y^* conditionally on θ^* and compute $I_y(y^*)$
3. if $I_y(y^*) = 1$ go to step 4 else stay at $\theta^{\#}$ and return to step 1.
4. Calculate

$$\alpha = \min\left\{1, \frac{\pi(\theta^*)}{\pi(\theta^{\#})} \times \frac{q(\theta^{\#}|\theta^*)}{q(\theta^*|\theta^{\#})}\right\}$$

   generate $u \sim U(0,1)$. If $u < \alpha$ set $\theta^{\#} := \theta^*$ otherwise stay at $\theta^{\#}$. Return to step 1.

During the algorithm there is no need to retain the generated y^* hence the set of accepted θ form a Markov chain with stationary distribution $\pi_\epsilon(\theta|y)$.

- The previous ABC-MCMC algorithm is also denoted as "likelihood-free MCMC".
- Notice that likelihoods do not appear in the algorithm.
- Likelihoods are substituted by sampling of artificial observations from the data-generating model.
- The Handbook of MCMC (CRC press) has a very good chapter on Likelihood-free Markov chain Monte Carlo.

Blackboard: proof that the algorithm targets the correct distribution

A (trivial) generalization of ABC-MCMC

Marjoram et. al used $J_\epsilon(y^*, y) \equiv \mathbb{I}_y(y^*)$. This implies that we consider equally ok those y^* such that $|y^* - y| < \epsilon$ (or such that $|S(y^*) - S(y)| < \epsilon$)

However we might also reward y^* in different ways depending on their distance to y.

Examples:

- Gaussian kernel: $J_\epsilon(y^*, y) \propto e^{-\sum_{i=1}^n (y_i - y_i^*)^2 / 2\epsilon^2}$, or...
- for vector $S(\cdot)$: $J_\epsilon(y^*, y) \propto e^{-(S(y) - S(y^*))' W^{-1} (S(y) - S(y^*))/2\epsilon^2}$

And of course the ϵ in the two formulations above are different.

Then the acceptance probability trivially generalizes to[4]

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

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 $\pi(\theta|y)$
- 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 the distances $d_{\epsilon_1} = \| S(y) - S(y^*) \|$ obtained using ϵ_1.
- Take the xth percentile of such distances and set a new threshold ϵ_2 as $\epsilon_2 :=$ xth percentile of d_{ϵ_1}.
- this way $\epsilon_2 < \epsilon_1$. So now you can use ϵ_2 to conduct more simulations, then similarly obtain $\epsilon_3 :=$ xth percentile of d_{ϵ_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.

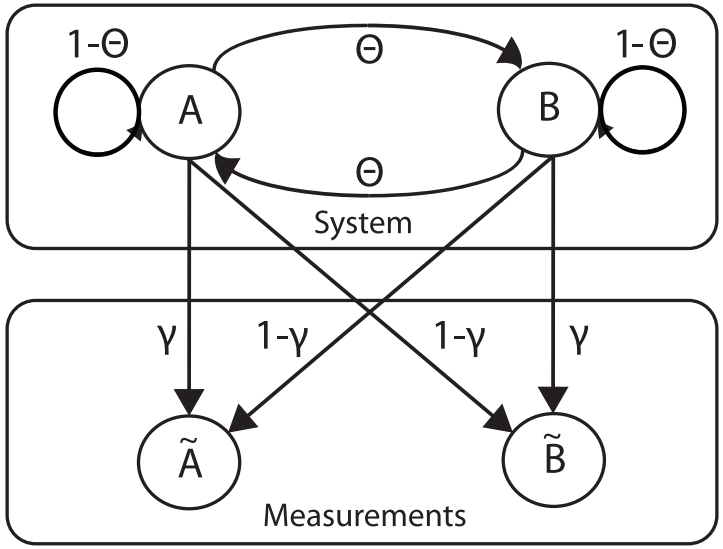
Alternatively just to set the sequence of ϵ's by trial and error.

When do we stop decreasing ϵ?

- Several studies have shown that when using ABC-MCMC obtain a chain resulting in 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$\rightarrow$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.
- A high acceptance rate denotes that your ϵ is way too large and you are probably sampling from the prior $\pi(\theta)$ (!)

Example from Sunnåker et al. 2013

[Large chunks from the cited article constitute the ABC entry in Wikipedia.]



- We have a hidden system state, moving between states {A,B} with probability θ, and stays in the current state with probability $1 - \theta$.
- Actual observations affected by measurement errors: probability to misread system states is $1 - \gamma$ for both A and B.

Example of application: the behavior of the Sonic Hedgehog (Shh) transcription factor in Drosophila melanogaster can be modeled by the given model.

Not surprisingly, the example is a hidden Markov model:

- $p(x_t|x_{t-1}) = \theta$ when $x_t \neq x_{t-1}$ and $1 - \theta$ otherwise.
- $p(y_t|x_t) = \gamma$ when $y_t = x_t$ and $1 - \gamma$ otherwise.

In other words a typical simulation pattern looks like:

A,B,B,B,A,B,A,A,A,B (states $x_{1:T}$)

A,A,B,B,B,A,A,A,A,A (observations $y_{1:T}$)

Misrecorded states are flagged in red.

The example could be certainly solved via exact methods, but just for the sake of illustration, assume we are only able to simulate random sequences from our model.

Here is how we simulate a sequence of length T:

1. given θ, generate $x_t^* \sim Bin(1, \theta)$
2. conditionally on x_t, y_t is Bernoulli: generate a $u \sim U(0,1)$ if $u < \gamma$ set $y_t^* := x_t^*$ otherwise take the other value.
3. set $t := t + 1$ go to 1 and repeat until we have collected $y_1, ..., y_T$.

So we are totally set to generate sequences of A's and B's given parameter values.

We generate a sequence of size $T = 150$ with $\theta = 0.25$ and $\gamma = 0.9$.

The states are discrete and only two (A and B) hence with datasets of moderate size we could do without summary statistics. But not for large T.

- Take $S(\cdot) =$ number of switches between *observed* states. Example: if $y = (A, B, B, A, A, B)$ we switched 3 times so $S(y) = 3$.

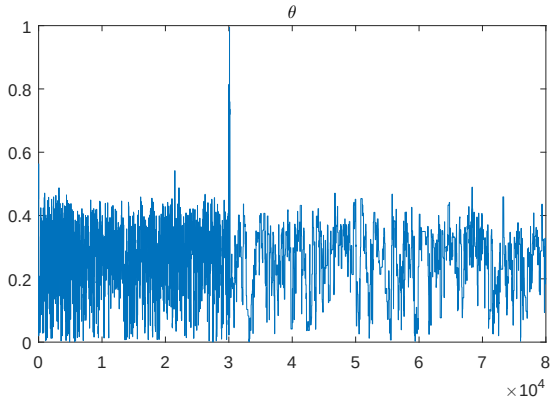
We only need to set a metric and then we are done:

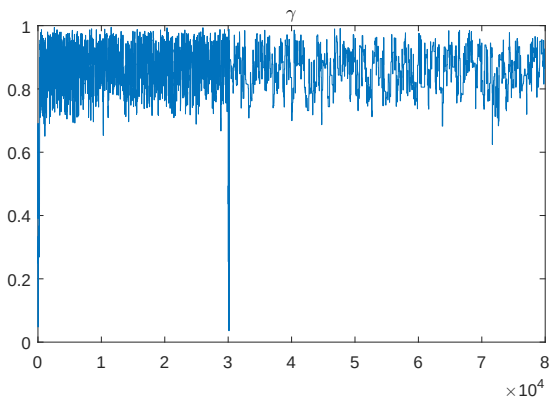
- Example (you can choose a different metric): $J_\epsilon(y^*, y) = \mathbb{I}_y(y^*)$ with

$$\mathbb{I}_y(y^*) = \begin{cases} 1, & |S(y^*) - S(y)| < \epsilon \\ 0, & \text{otherwise} \end{cases}$$

Plug this setup into an ABC-MCMC and we are essentially using Marjoram et al. original algorithm.

Priors: $\theta \sim U(0,1)$ and $\gamma \sim Beta(20,3)$.
Starting values for the ABC-MCMC: $\theta = \gamma = 0.5$



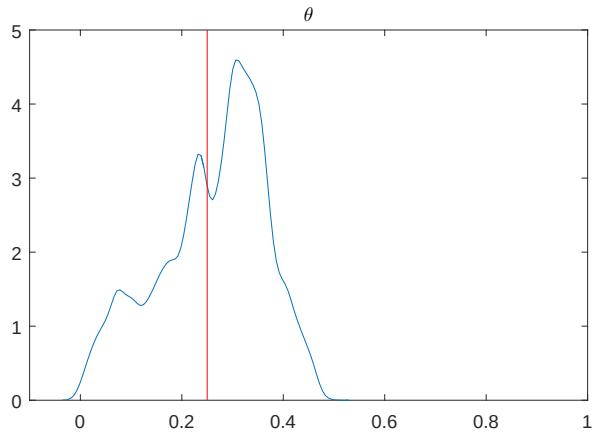


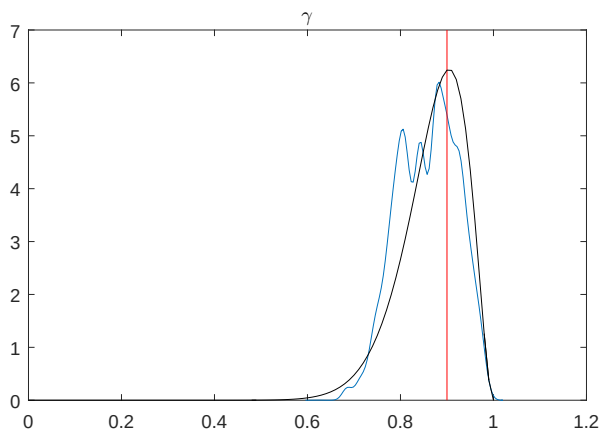
- Used $\epsilon = 6$ (first 5,000 iterations) then $\epsilon = 2$ for further 25,000 iterations and $\epsilon = 0$ for the remaining 50,000 iterations.
- When $\epsilon = 6$ accept. rate 20%, when $\epsilon = 2$ accept. rate 9% and when $\epsilon = 0$ accept. rate 2%.

Results at $\epsilon = 0$

Dealing with a discrete state-space model allows the luxury to obtain results at $\epsilon = 0$ (impossible with continuous states).

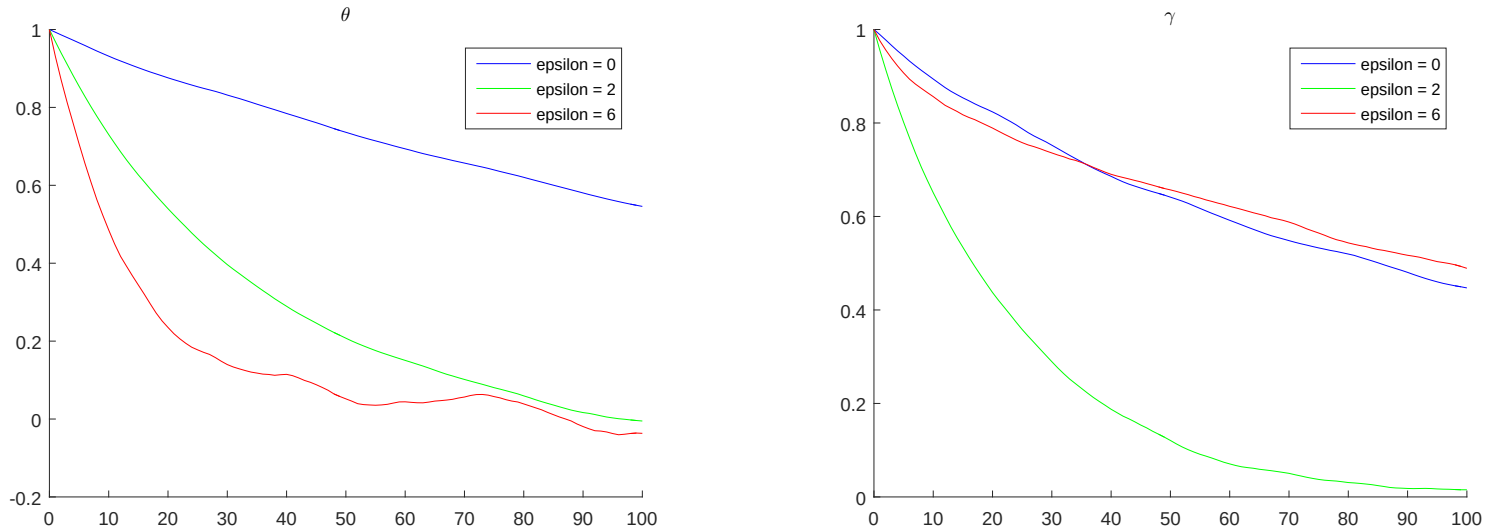
Below: ABC posteriors (blue), true parameters (vertical red lines) and Beta prior (black). For θ we used a uniform prior in [0,1].





Remember: **when using non-sufficient statistics results will be biased even with $\epsilon = 0$.**

A price to be paid when using ABC with a small ϵ is that, because of the many rejections, autocorrelations are very high.



This implies the need for longer simulations.

An apology

Paradoxically, all the (trivial) examples I have shown do not require ABC.

I considered simple examples because it's easier to illustrate the method, but you will receive an homework having (really) intractable likelihoods :-b

Weighting summary statistics

Consider a **vector** of summaries $S(\cdot) \in \mathbb{R}^d$, not much literature discuss how to assign weights to the components in $S(\cdot)$.

For example consider

$$J_\epsilon(y^*, y) \propto e^{-\| (S(y) - S(y^*) \| / 2\epsilon^2}$$

with $\| S(y) - S(y^*) \| = (S(y) - S(y^*))' \cdot W^{-1} \cdot (S(y) - S(y^*))$

Prangle[5] notes that if $S(y) = (S_1(y), ..., S_d(y))$ and if we give to all S_j the same weight (hence W is the identity matrix) then the distance $\| \cdot \|$ is dominated by the most variable summary S_j.

Only the component of θ "explained" by such S_j will be nicely estimated.

[5] D. Prangle (2015) arXiv:1507.00874

Useful to have a diagonal W, say $W = \text{diag}(\sigma_1^2, ..., \sigma_d^2)$.

The σ_j could be determined from some pilot study. Say that we are using ABC-MCMC, after some appropriate burnin say that we have stored R realizations of $S(y^*)$ corresponding to the R parameters θ^* into a $R \times d$ matrix.

For each column j extract the unique values from $(S_j^{(1)}(y^*), .., S_j^{(R)}(y^*))'$ then compute its mad_j (median absolute deviation).

Set $\sigma_j^2 := \text{mad}_j^2$.

(The median absolute deviation is a robust measure of dispersion.)

Rerun ABC-MCMC with the updated W, and an adjustment to ϵ will probably be required.

ABC for dynamical models

It is trickier to select intuitive (i.e. without the Fearnhead-Prangle approach) summaries for dynamical models.

However, we can bypass the need for $S(\cdot)$ if we use an ABC version of sequential Monte Carlo.

A very good review of methods for dynamical models is given in Jasra 2015.

ABC-SMC

A simple ABC-SMC algorithm is in Jasra et al. 2010, presented in next slide (with some minor modifications).

For the sake of brevity, just consider a bootstrap filter approach with N particles.

Recall in in ABC we assume that if observation $y_t \in \mathcal{Y}$ then also $y_t^{i*} \in \mathcal{Y}$.

As usual, we assume $t \in \{1, 2, ..., T\}$.

Step 0.
Set $t = 1$. For $i = 1, ..., N$ sample $x_1^i \sim \pi(x_0)$, $y_1^{*i} \sim p(y_1|x_1^i)$, compute weights $w_1^i = J_{1,\epsilon}(y_1, y_1^{*i})$ and normalize weights $\tilde{w}_1^i := w_1^i / \sum_{i=1}^N w_1^i$.

Step 1.
resample N particles $\{x_t^i, \tilde{w}_t^i\}$. Set $w_t^i = 1/N$.
Set $t := t + 1$ and if $t = T + 1$, stop.

Step 2.
For $i = 1, ..., N$ sample $x_t^i \sim p(x_t|x_{t-1}^i)$ and $y_t^{*i} \sim p(y_t|x_t^i)$. Compute

$$w_t^i := J_{t,\epsilon}(y_t, y_t^{*i})$$

normalize weights $\tilde{w}_t^i := w_t^i / \sum_{i=1}^N w_t^i$ and go to step 1.

The previous algorithm is not as general as the one actually given in Jasra et al. 2010.

I assumed that resampling is performed at every t (not strictly necessary). If resampling is not performed at every t in step 2 we have

$$w_t^i := w_{t-1}^i J_{t,\epsilon}(y_t, y_t^{*i}).$$

Specifically Jasra et al. use $J_{t,\epsilon}(y_t, y_t^{*i}) \equiv \mathbb{I}_{\|y_t^{*i} - y_t\| < \epsilon}$ but that's not essential for the method to work.

What is important to realize is that in SMC methods the **comparison is "local"**, that is we compare particles at time t vs. the observation at t. So we can avoid summaries and use data directly.

That is instead of comparing a length T vector y^* with a length T vector y we perform separately T comparisons $\| y_t^{*i} - y_t \|$. This is very feasible and clearly does not require an $S(\cdot)$.

So you can form an approximation to the likelihood as we explained in the particle marginal methods lecture, then plug it into a standard MCMC (*not* ABC-MCMC) algorithm for parameter estimation.

This is a topic for a final project.

Construction of $S(\cdot)$

We have somehow postponed an important issue in ABC practice: the choice/construction of $S(\cdot)$.

This is the most serious open-problem in ABC and one often determining the success or failure of the simulation.

- We are ready to accept non-sufficiency (available only for data in the exponential family) in exchange of an "informative statistic".
- Statistics are somehow easier to identify for static models. For dynamical models their identification is rather arbitrary, but see Martin et al[6] for state space models.

[6] Martin et al. 2014, arXiv:1409.8363.

Semi-automatic summary statistics

To date the most important study on the construction of summaries in ABC is in Fearnehad-Prangle 2012[7] which is a discussion paper on JRSS-B. Recall a well-known result: consider the class of quadratic losses

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

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

If we set $S(y) = E(\theta \mid y)$ then 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 $\epsilon \to 0$.

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**.

[7] Fearnhead and Prangle (2012).

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

- for the jth parameter in θ fit **separately** the linear regression models

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

  [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.
- Clearly, in the end we have as many summaries as the number of unknown parameters $\dim(\theta)$.

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^* 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, $S_j(\cdot) = \hat{\beta}_0^{(j)} + \hat{\beta}^{(j)}\eta(\cdot)$.

Use the same coefficients when calculating summaries for simulated data and actual data, i.e.

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

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

Software (coloured links are clickable)

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

Reviews

Fairly extensive but accessible reviews:

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

Simpler introductions:

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

Review specific for dynamical models:

1. Jasra 2015

Non-reviews, specific for dynamical models

1. SMC for Parameter estimation and model comparison: Toni et al. 2009
2. Markov models: White et al. 2015
3. SMC: Sisson et al. 2007
4. SMC: Dean et al. 2014
5. SMC: Jasra et al. 2010
6. MCMC: Picchini 2013

More specialistic resources

- selection of summary statistics: Fearnhead and Prangle 2012.
- review on summary statistics selection: Blum et al. 2013
- expectation-propagation ABC: Barthelme and Chopin 2012
- Gaussian Processes ABC: Meeds and Welling 2014
- ABC model choice: Pudlo et al 2015

Blog posts and slides

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 posts by Rasmus Bååth here and here
4. Tons of slides at Slideshare.