Model selection with ABC

Alex J H Fedorec

October 1, 2025

1 Model selection with ABC SMC

Bayesian inference

Let $\theta \in \Theta$ be a parameter vector with a prior $\pi(\theta)$. Given an objective of x_0, where x_0 exists in the solution space, $x_0 \in \mathcal{D}$. We define the likelihood function for the objective behaviour as $f(x_0|\theta)$

$$\pi(\theta|x_0) = \frac{f(x_0|\theta)\pi(\theta)}{\pi(x_0)}$$

We can rewrite $\pi(x_0)$ where a and b represent the lower and upper bounds of the parameter value

$$\pi(x_0) = \int_a^b f(x_0, \theta) d\theta = \int_a^b f(x_0|\theta)\pi(\theta) d\theta$$

The posterior distribution represents the parameter distribution that gives rise to the objective

$$\pi(\theta|x_0) = \frac{f(x_0|\theta)\pi(\theta)}{\int_a^b f(x_0|\theta)\pi(\theta) d\theta}$$

Let m be a model from vector of competing models, M, such that $m \in M = \{m_1, m_2 ... m_q\}$. Each model has its own parameter space, allowing us to define a joint space, $(m, \theta) \in M \times \Theta_M$. We can write Bayes' theorem in the context of a model space

$$\pi(m|x_0) = \frac{f(x_0|m)\pi(m)}{\int_M f(x_0|m')\pi(m') dm'}$$

Since the M is discrete, we can rewrite this

$$\pi(m|x_0) = \frac{f(x_0|m)\pi(m)}{\sum_M f(x_0|m')\pi(m')}$$

The marginal likelihood of the model, $f(x_0|m)$, is the expectation of the likelihood function taken over the model parameter prior distribution

$$f(x_0|m) = \int_\Theta \pi(\theta|m) f(x_0|\theta, m) d\theta$$

Approximate Bayesian computation

Writing the likelihood function, $f(x_0|\theta)$, in terms of complex summary statistics can be difficult. We bypass this and approximate the posterior by generating data from a model. We can sample a parameter vector from the prior, $\theta^* \sim \pi(\theta)$, which is simulated to yield a data vector, x^*. This can be written as a conditional, $x^* \sim f(x|\theta^*)$, which also gives the joint density, $\pi(\theta, x)$. In order to obtain the posterior distribution that satisfies our objective behaviour, x_0, we apply a conditional to define whether a generated data vector, x^* belongs to the objective x_0.

Algorithm 1: ABC rejection algorithm

```
Set particle indicator $i = 0$
while $i < N$ do
    Sample model $m$, from model space prior, $M$
    Sample parameter $\theta^*$, from prior distribution $\pi(\theta)$
    Simulate model, $f(x|\theta^*)$, giving simulation data, $x^*$
    Calculate distance between simulation data and objective, $d(x^*, x)$
     if $d(x^*, x) \leq \epsilon$ then
        Set $\theta_t^i = \theta^*$
        i = i + 1
    else
        Reject $\theta^*$
end while
Posterior distribution generated from accepted particles, $\pi(\theta^*|d(x^*, x)) \leq \epsilon$
```

If $x = x_0$

$$\pi(\theta|x, x_0) = \frac{\pi(\theta)f(x|\theta)}{\pi(\theta)f(x|\theta)dxd\theta}$$

Else

$$\pi(\theta|x, x_0) = 0$$

Let $\rho(x, x_0)$ be a distance function that compares a simulation to the objective. Using distance thresholds (ϵ), we can define values below which the distance is acceptably small. We can redefine $\pi(\theta|x, x_0)$ in the context of thresholds to obtain an approximation of the posterior:

If $\rho(x, x_0) < \epsilon$

$$\pi_\epsilon(\theta|x, x_0) = \frac{\pi(\theta)f(x|\theta)}{\pi(\theta)f(x|\theta)dxd\theta}$$

Else

$$\pi_\epsilon(\theta|x, x_0) = 0$$

The smaller ϵ is and the larger the number of simulations conducted, the more accurate the representation of the true posterior will be

$$\pi(\theta^*|\rho(x^*, x_0)) \leq \epsilon \approx \pi(\theta|x_0)$$

ABC rejection algorithm

The most basic ABC algorithm is the ABC rejection algorithm. Let ϵ be the distance threshold defining the necessary level of agreement between the objective, x_0, and a given simulation, x^* (Algorithm 1). Particles refer to a model and vector of parameters necessary to simulate the model. The ABC rejection algorithm repeatedly samples particles from the prior distribution, $\pi(m, \theta)$, generating data by simulation and testing for its representation of an objective data. Particles that are able to reproduce the objective data are accepted. By repeating this process we build a population of accepted particles that represent the posterior distribution for the objective.

Algorithm 2: Model selection with ABC SMC

```
Set population indicator, $t = 0$
Set initial epsilon, $\epsilon_t = \inf$
Set final epsilon, $\epsilon_T = [x, y, z]$
Set particle indicator, $i = 0$
if $t = 0$ then
    Sample $m^*$ from $\pi(m)$
    Sample $\theta^{**}$ from $\pi(\theta(m^*))$
else if $t > 0$ then
    Sample particle $\theta^*$ from previous population $\{\theta(m^*)^i_{t-1}\}$ with weights $w(m^*)_{t-1}$
    Perturb $\theta^*$ to obtain $\theta^{**} \sim K_t(\theta|\theta^*)$
    if $\pi(\theta^{**}) = 0$ then
        go to 5

Simulate, $x^* \sim f(x|\theta^{**}, m^*)$
    if $d(x^*, x_0) > \epsilon_t$ then
        go to 5

Set $m^i_t = m^*$
Set $\theta^i_t = \theta^{**}$
Calculate particle weight, $w^i_t$
    if $t = 0$ then
        $w^i_t = 1$
    else
        $w^i_t = \frac{\pi(\theta^{**})}{\sum_{j=1}^{N} w^j_{t-1} K_t(\theta^i_{t-1}|\theta^{**})}$
    if $i < N$ then
        Set $i = i + 1$
        go to 5

Normalise weights for every $m$.
if $\epsilon_t \neq \epsilon_T$ then
    Update population number, $t = t + 1$
    Update $\epsilon$ according to accepted particle distances, $\epsilon_t = f_\epsilon()$
    go to 5
```

Model selection with ABC SMC

In Toni et al. 2009, they use a variant of ABC, ABC Sequential Monte Carlo (ABC SMC). ABC SMC evolves the prior distribution through a series of intermediate distributions, each of which more closely resembles the posterior distribution than the previous (Algorithm 2). The distance threshold (ϵ) is decreased between distributions, moving the acceptance criteria closer to the objective. The gradual evolution reduces the tendency of becoming focused on local areas of minimal distance. In these experiments we use a component-wise Gaussian perturbation kernel. This produces a random walk from a particle of the previous population to a particle of the next population

$$K_t(\theta|\theta^*) = \mathcal{N}(\theta^*, 2x)$$

Where, x, is the variance of the previous population

$$x = \sigma(\theta_{t-1})^2$$

Defining stable steady state objective

We define the stable steady state objective with three summary statistics, where x is the time series data of a strain abundance. Each distance function has been chosen to filter out undesirable behaviours leaving the remaining desired stable steady state behaviour. d_1 is the final gradient of x, capturing the fundamental feature of stable steady state

$$d_1(x) = |\Delta x(t-1)|$$

d_2 is the standard deviation of the signal, this filters out simulations which are oscillating

$$d_2(x) = \sigma(x)$$

d_3 is the reciprocal the final value of the simulation, this allows us to define a minimum population threshold

$$d_3(x) = \frac{1}{x(t-1)}$$

Using these distances we can define the final conditional ϵ_F.

$$\begin{aligned} d_1 &< \epsilon_{F_1} \\ d_2 &< \epsilon_{F_2} \\ d_3 &< \epsilon_{F_3} \end{aligned}$$

Auto ϵ generation

The next ϵ for each new population is generated based on the accepted particle distances from previous population until we reach the final ϵ. We calculate distances for each species we are fitting to the objective. Let X be time series data of a simulated particle and $\hat{X}$ contain all accepted particles in a population

$$X \in \hat{X}$$

We set α as a proportion of the distances we should to generate the next ϵ. Generally speaking, the smaller α is the faster we will progress to ϵ_F

$$n = [\text{length}(\hat{X})/\alpha]$$

The threshold for distance d_i in population j is given by the n-th smallest distance d_i in population $j-1$.