Introduction to Approximate Bayesian Computation with Sequential Monte Carlo Methods

Kerry Liu

November 17, 2023

Abstract

This paper offers an overview of Approximate Bayesian Computation (ABC) and its advancements via Sequential Monte Carlo (SMC) methods. It examines the foundational theories behind ABC, focusing on the algorithmic aspects of both ABC Rejection and ABC-SMC. Furthermore, the paper presents a practical application of these methods in the study of population dynamics. The results highlight the strengths and weaknesses of the ABC-SMC method, which provide insights into the importance of user-defined choices in constructing effective estimations.

1 Introduction

Advances in Bayesian computational methods have been pivotal in addressing complex physical problems in dynamics settings where traditional approaches fail. Approximate Bayesian Computation (ABC) methods is one such adaptation, offering a robust framework for analyzing datasets where the likelihood function is intractable or unknown. [1] Among its variants, the ABC-Sequential Monte Carlo (ABC-SMC) method, designed for addressing challenges inherent in simpler ABC sampling approaches. This method has gained prominence in fields such as epidemiology [2], specifically in disease modeling. In this paper, we present an introduction to ABC Rejection and ABC-SMC methods, illustrating their respective algorithm mechanisms with R examples. The paper also showcases a practical application of ABC-SMC in the context of population dynamics. Section 2 explains the theories behind ABC followed by a general ABC Rejection Sampling algorithm case study. Section 3 then introduces ABC-SMC algorithm with a comparative example, with a subsequent real-world application in Section 4 using the logistic growth model to investigate the population growth of England and Wales from 1801 to 2021.

2 Approximate Bayesian Computation (ABC)

In Bayesian statistics, given parameters θ and sets of observed data $\mathbf{y_1}, \ldots, \mathbf{y_n}$, [3] we define the following: $p(\theta)$ represents the *prior distribution*, the initial beliefs about the parameters before any data is observed. These prior beliefs are updated to result in the *posterior distribution* $\pi(\theta|y)$, which incorporates information from the data through the *likelihood* $f(y|\theta)$. The likelihood represents the probability of observing the data given the parameters θ. Bayes' Theorem mathematically expresses this update as:

$$\pi(\theta|y) = \frac{f(y|\theta)p(\theta)}{f(y)}, \tag{1}$$

where $f(y)$ acts as a normalizing constant, also known as the *marginal likelihood* Since $f(y)$ is independent of θ, Equation (1) can be simplified to:

$$\pi(\theta|y) \propto f(y|\theta)p(\theta). \tag{2}$$

Approximate Bayesian Computation (ABC) is a computational method that approximates posterior distributions when the likelihood function is intractable or unknown. ABC is particularly suited for complex stochastic processes characterized by unknown parameters θ. It enables the generation of simulations from these processes, circumventing the need for difficult integration across all possible realizations. [4] ABC is also useful when dealing with large or high-dimensional data sets. In these cases, the likelihood $f(y|\theta)$ can be challenging to derived analytically. Unlike traditional Bayesian frameworks, such as Markov chain Monte Carlo (MCMC) and Gibbs sampling, which depend on the availability of a likelihood function to evaluate acceptance probabilities, ABC employs a distance metric to selectively accept or reject parameter values based on their 'closeness' with observed data. This method facilitates Bayesian inference for a broader range of models where standard computational approaches may not be viable. The steps of ABC Rejection Sampling Algorithm are as follows:

1. **Initialize Parameters:** A prior distribution $p(\theta)$, Observed data y_{obs}, A distance measure $d(\cdot, \cdot)$, A tolerance threshold ε, and an integer $N > 0$.

2. **Sampling:** For $i = 1, \ldots, N$:

    (a) Sample a parameter vector $\theta^{(i)}$ from the prior distribution $p(\theta)$.

    (b) Simulate a data set y^* from the model using the sampled parameter vector $\theta^{(i)}$, i.e., $y^* \sim p(y|\theta^{(i)})$.

    (c) If $d(y_{obs}, y^*) \leq \varepsilon$, accept $\theta^{(i)}$; otherwise, go to 1.

3. **Output:** A set of accepted parameter vectors $\{\theta^{(i)}; i = 1, \ldots, N\}$, approximate the posterior distribution $\hat{\pi}(\theta|y_{obs})$ given the tolerance threshold ε.

Hence, the N accepted parameter vectors θ^* provide an approximation to the posterior distribution $\pi(\theta|y_{obs})$ as

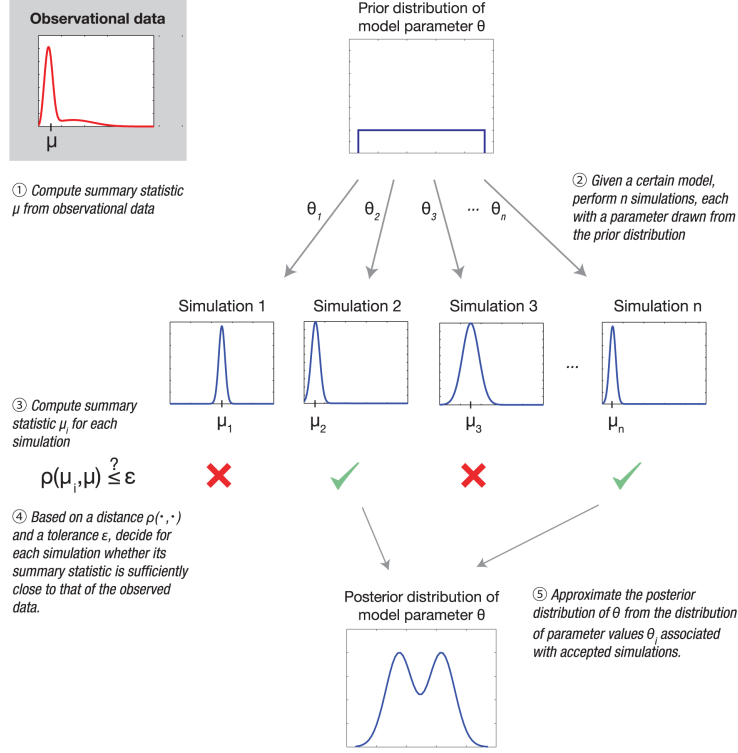$$\pi(\theta|y_{obs}) \approx \hat{\pi}(\theta|d(y_{obs}, y^*) \leq \varepsilon). \tag{3}$$

Figure 1: Illustration of the ABC rejection algorithm.
[3]

The distance function $d(\cdot,\cdot)$ measures the closeness between the observed dataset y_{obs} and the sampled data y^* under a set of proposed parameters θ. Common choices of distance function include Euclidean distance $\sqrt{(y_{obs}-y^*)^2}$ and the absolute distance $|y_{obs}-y^*|$, the latter of which is used in Section 4. As the tolerance ε approaches zero, the marginal distribution of the rejection sampler converges to the true posterior. Conversely, as ε approaches ∞ the sampler effectively becomes the prior distribution, implying no learning from the data (i.e., all samples are accepted).

$$\lim_{N \to \infty} \hat{\pi}(\theta|y_{obs}) = \pi(\theta|y_{obs}), \quad \text{for } \varepsilon \to 0. \tag{4}$$

The ABC Sampler may alternatively employ summary statistics $S(D)$ to reduce the data to lower-dimensional features such as mean or variance. [1] In this case, Step 2-(c) above is replaced by accepting a parameter vector $\theta^{(i)}$ if $d(S(y_{obs}), S(y^*)) \le \varepsilon$. Furthermore, when multiple summary statistics are used, the acceptance criterion requires all conditions to be met for a parameter vector

to be accepted.

$$\hat{\pi}(\theta|y_{obs}) \approx P\left(\bigcap_{m=1}^{M} \{d_m(S_m(y_{obs}), S_m(y^*)) \leq \varepsilon_m\}\right), \qquad (5)$$

where d_m is the distance measure associated with the m-th summary statistic, $S_m(\cdot)$ denotes the m-th summary statistic, M is the total number of summary statistics employed, and ε_m is the tolerance level for the m-th summary statistic.

3 ABC-SMC Algorithm

The simple ABC Rejection Sampling often suffers from low acceptance rates and may fail to adequately represent regions of high posterior density. The Sequential Monte Carlo (SMC) algorithm refines parameter samples through a sequence of intermediate distributions (generations), each with a decreasing tolerance ε. ABC-SMC initiates with particles sampled from the prior distribution $P(\theta)$, and each subsequent generation is refined via a weighted perturbation using a transition kernel $K(\theta|\theta^*)$. This approach concentrates the samples in regions of higher posterior density, thereby significantly reducing the computational complexity compared to traditional ABC Rejection Sampling. The ABC-SMC Algorithm is as follows:

1. **Initialize Parameters:** Set the total number of generations G, the number of particles N, and a sequence of decreasing tolerances $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_G\}$, which serve as distance thresholds for each generation.

2. **Start Generations:** Begin with the first generation $g = 1$.

3. **Start Particle Evaluation:** Initialize the particle counter $i = 1$.

4. **Sample Parameters:**

   - For the first generation $g = 1$, sample a parameter set θ^{**} independently of the prior distribution $P(\theta)$.

   - For subsequent generations $g > 1$, sample a parameter set θ^* from the previous generation θ_{g-1} weighted by w_{g-1}, and perturb it using a kernel K to obtain a new set θ^{**}.

5. **Prior Check:** If $P(\theta^{**}) = 0$, indicating the parameters are implausible according to the prior, return to Step 4.

6. **Generate Synthetic Data and Calculate Distance:** Generate n synthetic datasets y_j^* using θ^{**} through generative model $y^* \sim p(y|\theta^{**})$. Then, approximate the likelihood of observed data y_{obs} through a Monte Carlo estimate

$$\hat{P}(y_{obs}|y^*) = \left(\frac{1}{n}\right)\sum_{j=1}^{n} I(d(y_{obs}, y_j^*) \leq \varepsilon_g).$$

The indicator function I assigns 1 to the particles if $d(y_{\text{obs}}, y^*) \leq \varepsilon_g$ and 0 otherwise. For deterministic models, only 1 dataset needs to be generated.

7. **Posterior Check:** If $\hat{P}(y_{obs}|y^*) = 0$, return to Step 4.

8. **Acceptance and Weight Calculation:**

   - Accept the parameter and set $\theta_g^{(i)} = \theta^{**}$ for particle i.
   - Calculate the weight $w_g^{(i)}$ for particle i, considering the likelihood obtained in step 6 and the prior $P(\theta^*)$, adjusted by previous generation weights if $g > 1$.

$$w_g^{(i)} = \begin{cases} \hat{P}(y_{obs}|y^*)P(\theta^*), & \text{if } g = 1. \\ \dfrac{\hat{P}(y_{obs}|y^*)P(\theta^*)}{\sum_{j=1}^{N} w_{g-1}^{(j)} K(\theta_g^{(i)}|\theta_{g-1}^{(j)})}, & \text{if } g > 1. \end{cases}$$

9. **Particle Iteration:** Increment i by 1 if $i < N$, and go to Step 4.

10. **Weight Normalization:** Normalize the weights of all N particles in generation g such that the sum of the weights equals one.

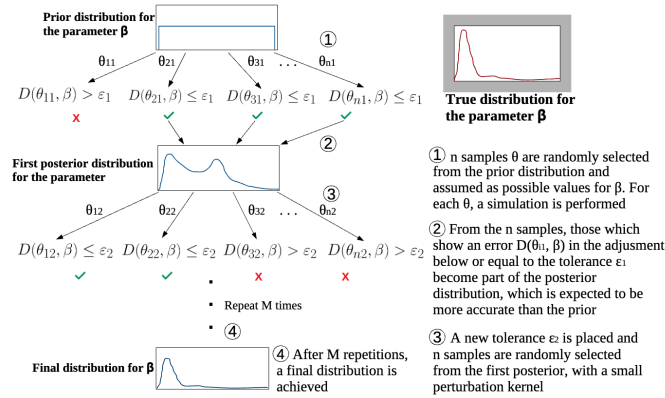11. **Generation Iteration:** If $g < G$, increment g by 1, and return to step 3.

Figure 2: Illustration of the ABC-SMC algorithm.
[3]

The efficiency of the ABC-SMC algorithm depends on several key factors: the predetermined sequence of decreasing tolerance values ε_g, the number of generations G, the number of simulations per parameter set n, and the perturbation kernel K. The Effective Sample Size (ESS), given by $\left(\sum_{i=1}^{N} \left[w^{(i)} \right]^2 \right)^{-1}$,

is calculated from the normalized weight $w^{(i)}$ and reflects the count of independent particles at the end of each generation (before resampling applied). A low ESS, often less than half total particles count N [1], can indicate *particle degeneracy*, where a small subset of particles dominates due to excessively large weights. This necessitates the adjustments of K, ε_g or resampling (replenishing samples based on their empirical distribution, See Section 5 Example) [1], to promote particle variance for a more representative posterior distribution. The perturbation kernel K adjusts particles' weights, facilitating their progression from θ to θ^*. A common choice for K is a multivariate normal distribution $\mathcal{N}$, defined as $K(\theta_g^{(i)}|\theta_{g-1}^{(i)}) = \mathcal{N}(\theta_{g-1}^{(i)}, \Sigma)$, where Σ is a scaled covariance matrix, informed by the covariance of the M nearest neighbors (MNN) [3]. For the examples following in Sections 3 and 4, we simplify Step 8, the Weight Calculation, by using the inverse of the distance as the weight.

4 Algorithmic Implementation

A simple ABC Rejection and an ABC-SMC example in R are presented.

4.1 ABC Rejection Sampling Code Setup

The observed data is simulated from an $\mathcal{N}(5, 1^2)$ distribution. We define a Uniform $U[-10, 10]$ prior for the parameter θ. The code iteratively samples within a `while` loop until $N = 1000$ particles are accepted. At each iteration, it simulates data from a `rnorm()` model, calculates the distance between the simulated and observed data via `calc_distance` function, and accepts the parameter if the distance is below $\varepsilon = 0.1$ The results (accepted parameters and their distances) are stored in *ABC_Rejection_Sampling_Results*. The acceptance rate, defined as the ratio of accepted proposals to total proposals, is displayed at the end of each iteration.

4.2 ABC-SMC Sampling Code Setup

Observations are assumed to be normally distributed, $y_{obs} \sim \mathcal{N}(5, 1^2)$, and the parameter of interest μ is estimated using 5 sequential tolerance levels (generations), ε_g, decreasing from 3 to 0.1. At each level, particles θ are sampled from a uniform prior $U[-10, 10]$ until $N = 200$ of them are accepted, resulting in a total of 200×5=1000 particles, aligning with the number used in Rejection Sampling. The acceptance is determined by the proximity to the observed data, measured by the distance function `calc_distance`. In the SMC approach, each accepted particle is weighted inversely proportional to its distance, calculated as $w_i = \frac{1}{d(y_{obs}, y_i^*)}$ (see Section 3 for details on the likelihood and kernel perturbation approach), resampling them according to these weights, and iteratively moving through decreasing ε_g to refine the posterior estimation of μ. The final posterior is represented as a weighted mean of accepted particles. The algorithm's progression is tracked by logging the current mean estimate after each

tolerance level. The result with accepted θ values, corresponding distances and weights are saved to *ABC_SMC_Results.csv*.
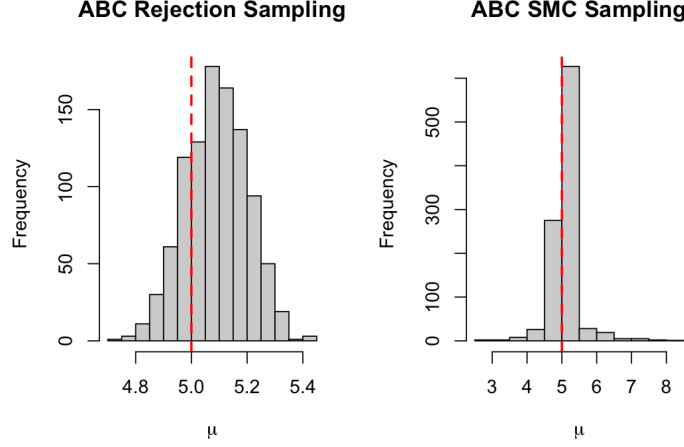

Figure 3: Posterior distributions of the θ obtained from ABC Rejection and ABC SMC Sampling, the red dashed line indicates the true parameter value

5 ABC-SMC for Population Dynamics

This ABC-SMC case study focuses on modeling population dynamics in England and Wales from 1801 to 2021, with 23 data points sourced from the Office for National Statistics [5]. The logistic growth model, characterized by the differential equation

$$\frac{dP}{dt} = rP\left(1 - \frac{P}{K}\right), \tag{6}$$

[6] is employed, where P represents the population at time t, r is the intrinsic growth rate, and K is the carrying capacity. In R, we have implemented two function for generating population data based on the above logistic growth model: one using the exact analytical solution of the equation, and another employing a numerical approximation method. To maximize variability, the approximate method and the more sensitive distance measure $(y_{obs} - y^*)^2$ are used. Accepted particles are resampled at the end of each generation.

In the ABC-SMC process, we use priors for r and K, simulate population data, and compare the simulations to the observed data using a distance metric. The algorithm proceeds through a series of decreasing tolerance levels (ε_g) for each generation g, refining the parameter estimates (r and K) iteratively. Pseudocode as follows:

- For $g = 1$, sample θ^* from priors of r and K. For $g > 1$, generate θ^* by perturbing the previous generation's particles θ_{g-1}, i.e., we sample $\theta_i^* \sim \mathcal{N}(\theta_{i,g-1}, \sigma_i^2)$, where σ_i^2 is the variance of the i-th parameter estimated from the previous generation (simplifying the kernel perturbation).

- Simulate population data using the logistic growth model with θ^* and calculate the distance between y^* and y_{obs}. Accept particles within the tolerance level ϵ and assigning weights inversely proportional to the distance, $w_i = \frac{1}{d(y_{obs}, y_i^*)}$ (simplifying the calculation of weights from the full kernel.).

- If no particles are accepted after 10 attempts, the tolerance level ϵ_g is increased by a factor of 1.1 to increase the likelihood of acceptance. This adjustment is made to avoid the algorithm becoming stuck if the tolerance is too stringent and the number of attempts is logged for each g.

- At the end of each generation, resample particles based on normalized weights, emphasizing those closer to the observed data.
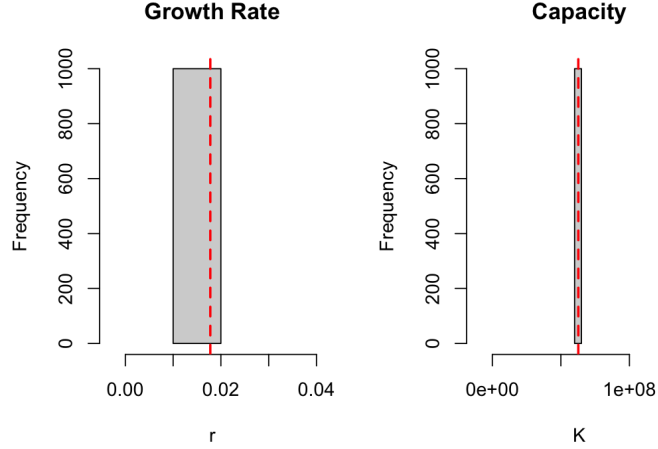

Figure 4: Posterior distributions of the growth rate r and carrying capacity K parameters, obtained after $g = 10$ generations with $N = 1000$ particles using a numerical logistic growth model. The red dashed line indicates the solution from non-linear least squares fitting of the growth equation for reference.

6   Discussion

In Section 4, we demonstrated the efficacy of the ABC-SMC method in concentrating posterior distributions into regions of high posterior density, particularly

in cases with sufficient data. This method notably improves the acceptance rate over ABC rejection sampling, offering significant efficiency gains in handling large datasets. However, our analysis in Section 5 revealed a tendency of the ABC-SMC algorithm to limit particle diversity, especially in scenarios with sparse observations, such as the population time-series data we examined. Specifically, with only 23 observations, the Effective Sample Size (ESS) reduced to 1 after the initial generation, and the high number of attempts required for acceptance suggested challenges in achieving effective sampling. These links back to the *particle degeneracy* issue mentioned in Section 3. The concentration can reduce the algorithm's ability to fully capture the posterior distribution's variability. A critical aspect of implementing ABC-SMC is finding the right balance between maintaining a high acceptance rate and ensuring a sufficient Effective Sample Size (ESS). Alternative kernels, while necessitating adjustments to importance weights, might offer a solution to this trade-off [1]. In conclusion, while ABC-SMC exhibits potential in efficiently processing large datasets and refining posterior estimations, its application, especially in cases with limited observations, necessitates careful consideration of intermediary distributions, as well as balancing the reweighing, resampling, and moving steps. Future research should focus on optimizing these aspects to enhance the algorithm's robustness across various data contexts.

References

[1] Scott A. Sisson, Yanan Fan, and Mark Beaumont, editors (2018), *Handbook of Approximate Bayesian Computation*. 1st edition, Chapman and Hall/CRC. DOI: `10.1201/9781315117195`.

[2] Amanda Minter, Renata Retkute (2019), *Approximate Bayesian Computation for infectious disease modelling*. Epidemics, Volume 29, 100368, ISSN 1755-4365. DOI: `10.1016/j.epidem.2019.100368`. Available from: `https://www.sciencedirect.com/science/article/pii/S175543651930026X`.

[3] Mikael Sunnåker, Alberto Giovanni Busetto, Elina Numminen, Jukka Corander, Matthieu Foll, et al. (2013), *Approximate Bayesian Computation*. PLOS Computational Biology, 9(1), e1002803. DOI: `10.1371/journal.pcbi.1002803`.

[4] Gael M. Martin, Brendan P. M. McCabe, Worapree Maneesoonthorn, and Christian P. Robert (2014), *Approximate Bayesian Computation in State Space Models*. arXiv:1409.8363 [math.ST].

[5] Office for National Statistics (2021), *Census 2021*.

[6] Leonard Lipkin and David Smith (2001), *Mathematical Association of America - Logistic Growth Model*. [Online; accessed 16-November-2023]. Available from: `https://maa.org/book/export/html/115630`.