Algorithms for Approximate Bayesian Computation

D

Tom Burr
Statistical Sciences, Los Alamos National Laboratory, USA

Alexei Skurikhin
Space Data Systems, Los Alamos National Laboratory, USA

INTRODUCTION

Computer models can be broadly categorized as deterministic or stochastic. Deterministic models output the same predictions for the same inputs. Stochastic models output different predictions for the same inputs. Our focus is stochastic computer models (SM), and in particular, on a new option to calibrate SMs using approximate Bayesian computation (ABC).

An example used later in this article is a SM to model neuronal loss in a region of the human brain that is associated with Parkinson's disease. Deletion mutations in the mitochondrial DNA (mtDNA) in that brain region are observed to accumulate with age. A deletion mutation converts a healthy copy of mtDNA to the mutant (unhealthy) variant. The number of mutant copies in cases with Parkinson's disease tends to be higher than in controls without Parkinson's disease. The role that mtDNA deletions play in neuronal loss is not yet fully understood, so better understanding of how mtDNA deletions accumulate is an area of active research. Henderson et al. (2009) use a simple SM that allows for any of five reactions, occurring at rates to be estimates. The five reactions are mutation, synthesis, degradation, mutant synthesis, and mutant degradation.

Approximate Bayesian computation (ABC) is an approach for using data to calibrate a SM and is especially useful when the likelihood for the data is unknown or intractable. ABC requires a set of summary statistics computed from real data. Then, the same set of summary statistics is computed from the SM for each of many candidate model parameter values. In a parameter acceptance/rejection loop, the candidate SM parameter values that are accepted provide an approximation to the posterior distribution of model parameters given the summary statistics computed from the real data. In a nutshell, ABC favors model parameters for which simulated summary statistics roughly agree with the corresponding summary statistics computed from the observed data. Because ABC relies on user-chosen summary statistics rather than on full data, it becomes computationally feasible. ABC is therefore appealing when the data dimension and/or parameter dimension is large.

This article describes applications of ABC and illustrates the challenges with ABC related to the quality of the approximation to the posterior distribution of model parameters. The challenges involve the fact that the user must choose (1) summary statistics, (2) a distance measure to calculate the distance between summary statistics in the real data and in the model-simulated data, and (3) the acceptance threshold used to accept or reject candidate parameter values in the acceptance/rejection sampling loop.

For a model with parameters θ and data D, a key quantity in Bayesian inference is the posterior distribution of model parameters given by Bayes theorem as

$$p_{\text{post}}(\theta \mid D) = \frac{p(D \mid \theta) p_{\text{prior}}(\theta)}{p(D)},$$

where

$$p_{\text{prior}}(\theta)$$

is the probability distribution for θ before observing data D, $p(D \mid \theta)$ is the likelihood, and

$$p(D) = \int_{\theta} p(D \mid \theta) p_{\text{prior}}(\theta)$$

DOI: 10.4018/978-1-4666-5888-2.ch149

is the marginal probability of the data that is used to normalize the posterior probability $p_{\text{post}}(\theta \mid D)$ to integrate to 1 (Aitken, 2010). The likelihood $p(D \mid \theta)$ can be regarded as the "data model" for a given value of θ. Alternatively, when the data *D* is considered fixed, $p(D \mid \theta)$ is regarded as a function of θ, and non-Bayesian methods such as maximum likelihood find the value of θ that maximizes $p(D \mid \theta)$ (Aitken, 2010).

In many applications, the data model $p(D \mid \theta)$ is computationally intractable but can be implemented in a stochastic model (SM); thus, many realizations from $p(D \mid \theta)$ are available by running the model many times at each of many trial values of θ. In a bioinformatics example, Tavaré et al. (1997) considered the classic problem of inferring the time to the most recent common ancestor of a random sample of *n* DNA sequences. The full likelihood of the data *D* involves the branching order and branch lengths, which is known to be computationally intractable because the number of possible branching orders of a sample of *n* DNA sequences grows approximately as *n*!. Tavaré et al. greatly facilitated the analysis by replacing *D* with the number of segregating sites (a segregating site exhibits variation in the DNA character across the sample) S_n in the sample of *n* sequences. The key simplification was that the distribution of S_n does not depend on the branching order or individual branch lengths, but rather on the total length of the phylogenetic tree, which is the sum of all branch lengths. One question that illustrates this article's focus is—how effective is such replacement of original data *D* with S_n for estimating the posterior distribution of the time to the most recent common ancestor of the sample? This situation is typical of many biology and epidemiology applications in that an unknown tree structure (such as the evolutionary tree relating species or the transmission tree in a disease outbreak chain) is involved, the likelihood of which is complicated or intractable. This article describes applications of ABC and illustrates challenges with ABC related to the quality of the approximation to the posterior distribution of model parameters, such as the choice of effective summary statistics, a distance measure, and an acceptance threshold.

In the contexts of interest here, the SM provides the data generation mechanism, so no explicit functional form exists for $p(D \mid \theta)$. Likelihood-free inference dates to at least Diggle and Gratton (1984), but the name ABC originated in Beaumont et al. (2002), when they referred to an approach to likelihood-free inference methods. Effective values of input parameters for stochastic computer models are typically chosen by some type of comparison to measured data. The numerically intense loop is often Markov Chain Monte Carlo (MCMC), which is a method used to simulate observations from the posterior distribution of model parameters (Aitken, 2010). Parameter estimation for stochastic models for which an explicit likelihood is not available is most commonly done using ABC. For examples of ABC applied to calibrate SMs, see Marjoram et al. (2003), Tanaka et al. (2006), Joyce and Marjoram (2008), Wegman et al. (2009), Csillery et al. (2010), Blum (2010), Beaumont (2010), Nunes and Balding (2010), Toni and Stumpf (2010), Robert et al. (2011), Fearnhead and Prangle (2012), Blum et al. (2013), Burr and Skurikhin (2013), and Weyant et al. (2013).

The following sections provide background on ABC and then describe the challenges in ABC related to choosing effective summary statistics, a distance measure, and an acceptance threshold. We describe current ABC research that addresses the challenges.

BACKGROUND

In this article, we assume that the SM provides the data generation mechanism, so no explicit functional form exists for $p(D \mid \theta)$. Effective values of input parameters for SMs are typically chosen by some type of comparison to measured data. To avoid possible confusion, we note here that parameter estimation for deterministic models is frequently done by running the model at multiple values of the input parameters, constructing an approximator to the model, and using the approximator inside a numerically intense loop that examines many trial values for the input parameters (Marjoram et al., 2003; Csillery et al., 2010; Blum, 2010; Toni and Stumpf, 2010; and Beaumont, 2012). The numerically intense loop is often MCMC, which is a method used to simulate observations from the posterior distribution of model parameters (Aitken, 2010; and Toni and Stumpf, 2010). Parameter estimation for SMs for which an explicit likelihood is not available has been attempted at least once using MCMC with a model approximator but is far more commonly