Hacettepe Journal of Mathematics & Statistics

Hacet. J. Math. Stat.
Volume 52 (4) (2023), 1120 – 1134
DOI : 10.15672/hujms.1055050

Research Article

Conditional density estimation using population Monte Carlo based approximate Bayesian computation

Faiza Afzaal*, Maryam Ilyas

College of Statistical Sciences, University of the Punjab, Lahore, Pakistan

Abstract

Most statistical methods require likelihood evaluation to draw a statistical inference. However, in some situations, likelihood evaluation becomes difficult analytically or computationally. Different likelihood-free methods are available that eliminate the need to compute the likelihood function. Approximate Bayesian Computation (ABC) is a framework that implements likelihood-free inference and replaces the likelihood evaluation with simulations by using forward modeling. The goal of ABC methods is to approximate the posterior distribution. However, posterior approximation via ABC methods is still considerably expensive for high dimensions. ABC requires many simulations that become computationally infeasible for complex models. Here, a technique is proposed that combines a somewhat more efficient form of ABC (Population Monte Carlo, PMC) with a Conditional Density Estimation (CDE) approach. The proposed framework provides an estimation of the posterior distribution which is referred to as PMC-CDE. A simulation study is performed that provides empirical evidence to show the efficiency of PMC-CDE in terms of integrated squared error loss. Furthermore, real-life datasets manifest the application of the proposed method.

Mathematics Subject Classification (2020). 60E05, 62E17

Keywords. Approximate Bayesian computation, sequential Monte Carlo, population Monte Carlo, conditional density estimation

1. Introduction

The main concern in statistical inference is to make inferences about population parameters θ given a finite set of observations $\mathbf{y_0} = (y_0^{(1)}, y_0^{(2)}, \ldots, y_0^{(n)})$. These inferences can be formed via probability distribution of the parameters of interest θ. The Bayesian approach is usually implemented to obtain the probability distribution of θ given data. This technique incorporates the likelihood function and the prior distribution to evaluate the posterior distribution of model parameters θ. Bayesian statistics implements the Bayes theorem (e.g., [42, 43]) to calculate the posterior probability distribution. Bayes theorem revises the prior distribution based on the given data $\mathbf{y_0}$. The updated posterior

*Corresponding Author.
Email addresses: faizaafzaalbaig@gmail.com (F. Afzaal), maryam.stat@pu.edu.pk (M. Ilyas)
Received: 28.05.2022; Accepted: 13.02.2023

distribution is computed using Equation (1.1). Here, $f(\mathbf{y_0}|\theta)$ is the likelihood function of $\theta, \pi(\theta)$ is the prior probability distribution and $\int f(\mathbf{y_0}|\theta)\ \pi(\theta)\, d\theta$ shows marginal likelihood or normalising constant. Typically, the normalizing constant does not affect the shape of the posterior distribution [24]. Therefore, posterior distribution can be written as Equation (1.2).

$$f(\theta|\mathbf{y_0}) \ = \ \frac{f(\mathbf{y_0}|\theta)\ \pi(\theta)}{\int f(\mathbf{y_0}|\theta)\ \pi(\theta)\, d\theta}, \tag{1.1}$$

$$f(\theta|\mathbf{y_0}) \propto f(\mathbf{y_0}|\theta)\ \pi(\theta). \tag{1.2}$$

The closed-form expression of the posterior distribution is generally difficult. Therefore, the respected Markov Chain Monte Carlo (MCMC) (e.g., [1, 12]) can be implemented. This technique draws samples from the posterior. However, the implementation of the MCMC approach requires likelihood evaluation. Likelihood evaluation becomes difficult for large data settings [41]. Therefore, many approaches are recommended that bypass the computation of the likelihood. For example, in spatial statistics, Integrated Nested Laplace Approximation (INLA) [7] is a good alternative to MCMC. Approximate Bayesian Computation (ABC) is another alternative to MCMC. In recent likelihood-free procedures, ABC [44] is the most commonly used family of algorithms (e.g., [3, 4, 26, 29, 33, 40]).

1.1. Approximate Bayesian computation (ABC)

Until recently, different ABC-based techniques have been proposed. The first algorithm of ABC is known as the rejection sampler. It was introduced by [44]. This algorithm was restricted only to discrete settings. A rejection sampling algorithm for continuous settings was introduced by [33]. Instead of an exact match between the observed and simulated datasets, simulated values of θ were accepted if the difference between observed and synthetic datasets became smaller than the fixed threshold value θ. Beaumont et al. [3] suggested two improvements in the rejection sampler using kernel weighting and regression adjustments. These changes provoke observations that follow $f(\theta|y)$ more closely. The curse of dimensionality in the ABC approach was resolved to some extent via local linear regression [3]. Blum et al. [9] proposed an approach that deals with the curse of dimensionality and computational cost more effectively. This machine learning approach was based on two-stage nonlinear regression. Marjoram et al. [26] used the MCMC algorithm that does not depend on the likelihood ratios.

Sequential Monte Carlo (SMC) methods (e.g., [13]) play the role of work-horses for ABC approach. SMC samplers are used to draw samples from the approximate posterior. Initially, algorithms for sequential Monte Carlo samplers were introduced by [8]. This approach was based on Importance Sampling (IS) and resampling moves. Sisson et al. [40] introduced a likelihood-free approach by making some modifications to the algorithm [8]. A sequence of distributions approximating the posterior was specified with a decreasing sequence of discrepancy thresholds. This technique was found to be effective when there was a mismatch between sampling and target distributions. However, this approach results in biased posterior samples. Beaumont et al. [2] and Toni et al. [46] provided the appropriate solution and proposed sequential approaches that could implement suboptimal backward kernels. This step caused computational complexity that was quadratic in the number of generated particles. Selection of the appropriate tolerance threshold sequence can also affect the approximate posterior. Different techniques were introduced for the adaptive selection of tolerance thresholds. For example, thresholds based on fixed quantiles of distances corresponding to the accepted particles (e.g., [20, 37, 38, 50]), thresholds based on quantiles of effective-sample size [28, 30], thresholds based on an adaptive choice of quantiles instead of fixed ones [37, 38]. Naive ABC approaches cannot accommodate high-dimensional hierarchical models. Turner and Zandt [47] suggested a new

MCMC-based algorithm known as Gibbs ABC. Bayesian techniques have been used to increase the efficiency and accuracy of ABC methods to estimate hierarchical models. Biau et al. [5] suggested an ABC algorithm based on a k-Nearest Neighbor (KNN) [5]. The KNN approach asks for the number of neighboring values that are closest to the observed values. The main focus of the KNN-based ABC approach was to describe asymptotic features of conditional density linked with ABC. Prescott and Baker [32] introduced an SMC approach that replaces the importance sampling step with a multi-fidelity importance sampling. It was introduced to reduce the cost of repeated simulations in the ABC context.

1.2. Conditional density estimation (CDE)

Density estimation is primarily the construction of the unobservable underlying probability function based on the information contained in the observed sample. The parametric approach assumes that the shape of the distribution is known. The parametric estimation has restrictive distributional assumptions. In this approach, distributions are typically unimodal and cannot model complex applications. However, the nonparametric approach does not make any restrictive assumptions about the shape of the unknown distribution except for the mild ones. The key idea of the nonparametric approach is that for a given data set $(y_1, y_2, \ldots, y_n \sim f(y))$ the unknown density is estimated [48]. Different methods have been introduced for the nonparametric estimation of a probability distribution. Some commonly used approaches include histograms, kernel density estimators, and orthogonal series estimators [36]. The CDE approach adopted for the proposed technique in our paper is based on orthogonal series estimators.

The idea of density estimation via orthogonal series was first presented by [11] and later on by [45]. These estimators are usually modified according to the geometric features of data and bear the useful property of dimension reduction [14]. Orthogonal series is a linear combination of expansion coefficients and an orthonormal system of functions. It is an efficient approach for univariate, multivariate, and conditional density estimation. Different types of orthogonal series, bounded or unbounded, can be employed for density estimation purposes. For simplicity, suppose that the interest lies in the estimation of density function f, over the unit interval [0,1]. However, estimation of f via this approach requires estimation of the coefficients in its Fourier expansion. Suppose $y_1, y_2, \ldots, y_n \sim f(y)$ is a random sample of observations. The Fourier expansion of f is given in Equation (1.3).

$$f(y) = \sum_{j=1}^{\infty} \phi_j \beta_j, \tag{1.3}$$

$$\beta_j = \int_0^1 \phi_j(y) f(y)\, dy = E[\phi_j(y)]. \tag{1.4}$$

The known and predetermined orthonormal functions are $\phi_j(y)$ and β_j are Fourier coefficients. A system with basis functions $\phi_j(y)$ is said to be orthonormal if $\int_0^1 \phi_u(y)\phi_v(y) = 0, u \neq v$ and $\int_0^1 (\phi_j(y))^2 = 1$, for every value of j. Hence, based on a sample of n observations, an estimate of β_j can be obtained using Equation (1.5). The sum $\sum_{j=1}^{\infty} \beta_j \phi_j$ is not a good estimate of f. To obtain a better estimate of f, it is required to truncate the expansion $\sum \beta_j \phi_j$ at some point. Choose the cutoff point k. It determines the amount of smoothing. Then, density estimate $(\hat{f})$ can be obtained using Equation (1.6).

$$\hat{\beta}_j = \frac{1}{n} \sum_{i=1}^{n} \phi(y_i), \tag{1.5}$$

$$\hat{f}(y) = \sum_{j=1}^{k} \hat{\beta}_j \phi_j(y). \tag{1.6}$$

The relationship between a dependent variable and predictor(s) can be described via conditional mean ($E(\theta|y)$). Conditional Density Estimation (CDE) is a technique to get an estimate of the full probability function ($f(\theta|y)$) of a random quantity θ for a given quantity y [15]. CDE is the generalization of the regression approach. It explains the stochastic dependence between θ and y which is modeled by a conditional probability function $f(\theta|y)$. Different nonparametric estimators of conditional density have been proposed in the literature. Most of them are based on the ratio method for estimating conditional density. In the ratio method, joint and marginal distributions are estimated by kernel density estimation [35]. Then, conditional density is estimated by taking a ratio of these two estimates. Some estimators of CDE are based on machine learning approaches e.g., mixture density networks [6]. This method implemented neural networks to learn mixture model parameters. However, it was difficult to directly fit it on conditional density estimation loss. Meinshausen [27] introduced Quantile Regression Forests (QRF). The QRF estimator estimates density as a weighted sum of quantiles attained from the tree of the random forest. Stating differently, the density is estimated using random forest weights. In recent methodologies, Random Forest Conditional Density Estimation (RFCDE) is a good CDE estimator which is given by [31]. Its target is to minimize conditional density estimation loss by training the random forest. Spectral series conditional density estimator is another method for CDE [21]. FlexCode is also a conditional density estimator formulated by [22].

In this paper, a technique is proposed that combines a sequential approach of ABC i.e., ABC-Population Monte Carlo (ABC-PMC) [2] with a Conditional Density Estimation (CDE) approach [22]. Therefore, the proposed technique is entitled PMC-CDE. This approach starts with a crude approximation of the target posterior via the ABC-PMC sampler at the first stage. The second stage estimates the posterior by incorporating a nonparametric CDE approach. The CDE approach implemented here utilizes orthogonal series for the response θ and then computes posterior based on a regression technique. In this paper, we have implemented the k-nearest neighbor regression technique. To assess the performance of the proposed method (PMC-CDE) integrated squared error loss is used. The rest of the paper is structured into four more sections. Section 2 describes the proposed method (PMC-CDE) in detail. In Section 3, details of the data generation procedure and simulation study are presented. Section 4 depicts the performance of the proposed technique via simulated and real-life datasets. Section 5 concludes this paper.

2. Statistical methodology

In this section, a method for posterior density estimation is proposed (Section 2.3). The proposed technique (PMC-CDE) is a combination of the population Monte Carlo-based ABC approach [2] and a Conditional Density Estimation (CDE) technique [22]. CDE framework is used for efficient estimation of the posterior density. The performance of the proposed method (Section 2.3) is analyzed through average integrated squared error loss.

2.1. ABC population Monte Carlo

The ABC rejection sampler draws approximate posteriors by comparing the observed and simulated data sets. The main cause of inefficiency in the ABC rejection sampler is that the same distribution is used to draw samples [49]. There is a high possibility that certain regions of prior generate simulations that are distant from observed summary statistics. Therefore, a portion of parameter values is sampled from a low region of posterior

mass [49]. This causes a low acceptance rate in the rejection sampling approach. Intelligent exploration techniques can be implemented over parameter space to decrease the number of simulations in low-likelihood regions [39]. An efficient implementation of ABC is possible by incorporating sequential approaches i.e., Sequential Monte Carlo (SMC) (e.g., [8, 13]) and Population Monte Carlo (PMC) [10]. This paper incorporates the PMC approach of ABC in the proposed technique.

Population Monte Carlo-based ABC approach was introduced by [2]. The goal of PMC methods is to improve the proposal distribution continuously for each iteration. These methods enhance the sampling efficiency and provide a better approximation of the target distribution. ABC-PMC [2] technique provides an approximation to the target posterior via a sequence of artificial distributions. This sequence of distributions is constructed for a monotonically decreasing sequence of tolerance thresholds. ABC-PMC is an iterative procedure. At each iteration, a set of weighted particles is generated by satisfying a criterion between the summaries of observed and simulated datasets. The main focus of the ABC-PMC approach is to provide an estimate of the posterior based on a set of weighted particles. This set of particles approximates the target posterior. The approximating distributions improve iteratively as the tolerance threshold is decreasing for each subsequent iteration.

ABC-PMC algorithm starts with a basic rejection sampler. A set of particles is drawn from a proposal distribution. Proposal distribution ($q_1(\theta)$) at the first iteration is equal to the prior ($\pi(\theta)$). The first approximation of the target posterior $f(\theta|s_0)$ is obtained by using the rejection sampler [3, 33] with a larger tolerance threshold (ϵ_1). A large value of ϵ_1 corresponds to a higher acceptance rate and a bad approximation of the target posterior. At the end of the first iteration, equal weights are assigned to the accepted particles that give a first approximation to $f(\theta|s_0)$ for an observed vector of summary statistics (s_0). The tolerance threshold for the next iteration is also specified. The ABC-PMC algorithm gradually reduces the value of the tolerance threshold to effectively shift from a prior distribution to a well-approximated target distribution. Initially, a fixed sequence of threshold values can be implemented [2]. Such a sequence results in a poor approximation of the target posterior (e.g., [49]). However, some approaches in the literature suggest an adaptive sequence of tolerance thresholds. It is recommended that for every iteration, the value of the tolerance threshold is set as $\alpha\%$ quantile of the distance values corresponding to the accepted particles in that iteration. The posterior distribution approximated in the previous iteration is used as the proposal distribution in the next iteration. A set of weighted samples (or particles) is drawn from the proposal distribution. These samples are then perturbed. The ABC rejection sampler is then performed with a lower tolerance threshold as compared to the previous iteration. The accepted particles are closer to the target posterior $f(\theta|s_0)$ in contrast to the previous approximation. At the end of each iteration, importance weights are calculated and the tolerance threshold is specified for the next iteration. The value of the tuning parameter is typically set using Equation (2.1). This procedure continues until the algorithm converges. The proposals consistently improve across iterations. Therefore, the approximation error decreases.

$$\sum^{(r)} = 2Cov(\theta^{(r)}). \tag{2.1}$$

2.2. Posterior density estimation by FlexCode

High-dimensional and complex data handling is a challenging problem in the statistical inference of the modern age. Often, regression ($E(\theta|\mathbf{s})$) is used to describe the relationship between a random variable (θ) and a vector of high dimensional covariates ($\mathbf{s}$). Limited literature is available to estimate full conditional density ($f(\theta|\mathbf{s})$) for high dimensional covariates ($\mathbf{s}$). If the full conditional density is implemented, bias will be reduced [21].

Conditional density is a key quantity in modern statistical inference. In most problems like multimodality, asymmetry, and heteroscedastic pattern of noise, the regression curve fails to produce useful information. Conditional density estimation is the generalization of the regression in describing the relationship between the target variable (θ) and covariates ($\mathbf{s}$). A naive method for density estimation requires a set of independent and identically distributed samples i.e., $\psi = (\mathbf{s_1}, \theta_1), (\mathbf{s_2}, \theta_2), , (\mathbf{s_N}, \theta_N)$. This set of samples ($\psi$) is incorporated in a CDE estimator to obtain an estimated density ($\hat{f}(\theta|\mathbf{s})$) at point $\mathbf{s} = \mathbf{s_0}$. However, conventional methods for estimation of conditional density ($f(\theta|\mathbf{s})$) use every ($\mathbf{s}$) for estimation while in ABC-PMC applications the interest lies in observed summary statistics $\mathbf{s_0}$ only.

A conditional density estimator, FlexCode [22], is used in this paper. FlexCode is a nonparametric flexible estimator that estimates conditional density based on regression. Flexible means that it can be applied to different data types and data structures. FlexCode converts regression estimation into a conditional density estimation. For this purpose, orthogonal series are implemented in FlexCode [22]. Orthogonal series requires the estimation of expansion coefficients. FlexCode employs different nonparametric regression techniques to estimate expansion coefficients. Different regression techniques can be selected according to the data type or nature of the problem. Also, different sparse structures can be induced into the data. Like other CDE methods, FlexCode also requires a set of independent and identically distributed data sets, i.e. $\psi = (\mathbf{s_1}, \theta_1), (\mathbf{s_2}, \theta_2), \ldots, (\mathbf{s_N}, \theta_N)$. FlexCode is usually implemented in high-dimensional settings. To keep it simple, assume that interest lies in the estimation of the posterior distribution for a single parameter ($\theta \in R$). In the FlexCode framework, orthonormal basis functions $(\phi_j)_{j \in N}$ are specified that are implemented on θ to model the conditional density ($f(\theta|\mathbf{s})$) as a function of ϕ. There are several orthogonal bases in the literature. Generally, the choice of orthogonal basis function depends on the complexity of the shape to be captured [25]. However, a reasonable choice to rationally smooth functions is the Fourier basis.

$$\begin{aligned} \phi_1(\theta) &= 1, \\ \phi_{2j+1}(\theta) &= \sqrt{2}\sin(2\pi j\theta), \quad j \in N, \\ \phi_{2j}(\theta) &= \sqrt{2}\cos(2\pi j\theta), \quad j \in N. \end{aligned}$$

In FlexCode, orthogonal series is used for the response variable (θ). FlexCode limits the estimation of complex high dimensional conditional density problems to a straightforward regression problem (Equation (2.2)). It is worthy to note that Equation (2.2) holds if for every $\mathbf{s} \in R$, $f(\theta|\mathbf{s})$ is $L^2(R)$ integrable as a function of θ. Moreover, as the basis functions $\{\phi_j\}_{j \in N}$ are orthogonal so expansion coefficients are computed using Equation (2.3). Therefore, every $\beta_j(\mathbf{s})$ is a regression estimate of the transformed variable $\phi_j(\theta)$ on $\mathbf{s}$. Stating differently, for fixed j, an estimate of $\beta_j(\mathbf{s})$ is obtained by regressing $\phi_j(\theta)$ on $\mathbf{s}$. By incorporating an estimate of $\beta_j(\mathbf{s})$, FlexCode estimator is obtained (Equations (2.4) and (2.5)).

$$f(\theta|\mathbf{s}) = \sum_{j \in N} \beta_j(\mathbf{s})\phi_j(\theta), \tag{2.2}$$

$$\beta_j(\mathbf{s}) = \int_{\mathbb{R}} \phi_j(\theta) f(\theta|\mathbf{s})\, d\theta = \mathbb{E}(\phi_j(\theta)|\mathbf{s}), \tag{2.3}$$

$$\hat{f}(\theta|\mathbf{s}) = \sum_{j=1}^{J} \hat{\beta}_j(\mathbf{s})\phi_j(\theta), \tag{2.4}$$

$$\hat{\beta}_j(\mathbf{s}) = \hat{\mathbb{E}}(\phi_j(\theta)|\mathbf{s}). \tag{2.5}$$

The cutoff (J) plays the role of tuning parameter in the series expansion and regulates the bias-variance tradeoff in the final density estimate. Broadly speaking, for a smaller value of

J, the smoothness of density increases. Regarding computational efficiency, FlexCode is a faster density estimator as compared to many existing conditional density estimators [22]. FlexCode is a consistent estimator as it is based on orthogonal series. Moreover, FlexCode is a suitable method for parallel computing as each of the regression functions is estimated separately and then combined to get a density estimate. Conditional density estimation via FlexCode is streamlined to the choice of a regression technique. FlexCode offers different regression methods for the estimation of regression functions $\beta_j(\mathbf{s})$ (Equation (2.3)). To keep things simple, the k-Nearest Neighbor (KNN) regression (e.g., [16]) technique is implemented in the proposed method PMC-CDE. KNN regression is a nonparametric approach that approximates the relationship between covariates ($\mathbf{s}$) and the continuous response variable ($\phi(\theta)$) by taking the average of the response values in the neighborhood of the observed variable ($\mathbf{s_0}$) for given $\mathbf{s}$. The neighborhood size (k) is specified by the analyst. However, the best choice is to set the neighborhood size (k) via cross-validation and choose the size that minimizes the Mean Squared Error (MSE).

2.3. Population Monte Carlo based CDE (PMC-CDE)

This section describes the step-wise procedure of the proposed method (PMC-CDE) for estimating posterior density. The PMC-CDE combines the ideas of the conditional density estimation approach [22] and approximate Bayesian computation population Monte Carlo [2]. The main focus is to estimate the posterior $f(\theta|\mathbf{s_0})$ as accurately as possible for a given vector of observed summary statistics ($\mathbf{s_0}$) and prior belief ($\pi(\theta)$). For this purpose, the proposed technique (PMC-CDE) estimates the required posterior at two stages. The first stage consists of approximating the posterior by generating approximate samples from a distribution of interest $f(\theta|\mathbf{s_0})$. To approximate the posterior, a sequential approach i.e. approximate Bayesian computation-population Monte Carlo [2] is implemented. The second stage is the post-processing of the approximate posterior obtained in the first stage. For this purpose, a conditional density estimator (FlexCode [22]) is incorporated that estimates $f(\theta|\mathbf{s_0})$. Both phases operate independently. As a result, PMC-CDE retains the asymptotic properties of both parent methods. The algorithm-1 fully demonstrates the implementation procedure of the PMC-CDE.

2.3.1. Tolerance threshold. In the algorithm 1, the tolerance threshold for the iteration $r = 1$ is specified differently than for iteration $r > 1$. For the first iteration, the distance values $(\gamma_i(s,\ s_{\mathbf{0}}), i = 1, ..., B)$ are arranged in ascending order. Then, 50% of B particles are retained corresponding to the minimum distances. For iteration $r > 1$, the tolerance threshold ϵ^r is set as the 75^{th} quantile of distance values $\{\gamma_i^{(r-1)}\}_{i=1}^N$ corresponding to the accepted particles in $(r-1)^{th}$ iteration [37, 50].

2.3.2. Convergence criterion. Different convergence criteria are given in the literature. However, a criterion given by [19] is implemented in this paper. It typically depends on the derived uncertainties of the parameters to be inferred. To check the convergence, uncertainties are measured at the end of each iteration. If the uncertainties show a balanced pattern and have negligible variation, then the algorithm attains convergence [19]. For this purpose, dispersion is computed for the accepted particles at the end of each iteration. If the dispersion stabilizes, convergence is achieved.

Algorithm 1 Population Monte Carlo based CDE

1: At first iteration, draw B particles θ_i from the prior distribution. For each particle (θ_i) simulate the synthetic data (y^*).
2: Compute the distance between summary statistics of observed and simulated datasets i.e., $\gamma = \|s(y^*) - s(y_0))\|$.

3: Arrange the distances in ascending order and retain $N = 50\%$ of B particles corresponding to smallest distances.
4: Set the initial tolerance threshold $\epsilon^{(1)} = \max(\gamma_1, ..., \gamma_N)$, sample of particles $\theta^{(1)} \leftarrow (\theta_1^*, \theta_2^*, \ldots, \theta_N^*)$, weights as $\omega^{(1)} \leftarrow \frac{1}{N}$ for all elements of $\theta^{(1)}$. and the tuning parameter as $\Sigma^{(1)} = 2Cov(\theta^{(1)})$.
5: For iteration $r > 1$, repeat the steps (a), (b), (c), and (d) until N particles satisfy $\gamma(s^*, s_0) < \epsilon^{(r-1)}$
   (a) Sample θ^* from previous iteration, i.e., $\theta^* \sim \theta^{(r-1)}$ with probabilities $\omega^{(r-1)}$.
   (b) Perturb θ^* using Gaussian kernel, $\theta^{**} \sim N(\theta^*, \Sigma^{(r-1)})$.
   (c) Simulate data y^{**} i.e., $y^{**} \sim f(y|\theta^{**})$.
   (d) Compute $\gamma = \|s(y^{**}) - s(y_0))\|$.
6: Set $\theta^{(r)} \leftarrow (\theta_1^{**}, \theta_2^{**}, \theta_3^{**}, \ldots, \theta_N^{**})$, weights as

$$\omega^{(r)} \leftarrow \frac{\pi(\theta^{(r)})}{\Sigma\omega^{(r-1)} f(\theta^{(r)}|\theta^{(r-1)}, \Sigma^{(r-1)})}$$

and tolerance as $\epsilon^{(r)} \leftarrow 75^{th}$quantile $(\gamma_1, \gamma_2, \ldots, \gamma_N)$ and set $\Sigma^{(r)} = 2Cov(\theta^{(r)})$
7: Store the particles of approximated posterior $f_{abc-pmc}(\theta|\mathbf{s})$ at R^{th} iteration in a set along with the corresponding summary statistics i.e., $\psi = \{(\mathbf{s_1}, \theta_1), \ldots, (\mathbf{s_N}, \theta_N)\}$.
8: Partition the dataset ψ into two parts for cross-validation such that 70% of ψ is training dataset and 30% of ψ is validation dataset.
9: Apply FlexCode [22] on the partitioned information of ψ to obtain an estimate of $f(\theta|\mathbf{s_0})$.

In Algorithm 1, for one-dimensional case $\Sigma^{(r)}$ represents a scalar quantity. For high dimensions, $\Sigma^{(r)}$ indicates a covariance matrix.

2.4. Integrated squared-error loss

To measure the efficiency of the proposed estimator (PMC-CDE), Integrated Squared Error (ISE) loss is computed [23]. If $\hat{f}(\theta|\mathbf{s_0})$ is an estimator of the true posterior $(f(\theta|s_0))$ then ISE loss is $\mathcal{L}_s$ (Equation (2.6)). It should be smaller than any other estimator of the true posterior.

$$L_s = \int \left(\hat{f}\,(\theta|s_0) - f(\theta|s_0)\right)^2 d\theta. \tag{2.6}$$

3. Simulation study

In the simulation study, a model with known parameters is considered. As an example, a simple model is chosen to demonstrate the performance of the proposed method (PMC-CDE, Section 2.3). The posterior distribution of the mean of the Gaussian distribution is estimated via the PMC-CDE estimator. It is compared with the actual density through integrated squared error loss. The posterior distribution of the mean of Gaussian distribution is estimated for the following two settings.

(1) Posterior of the mean (μ) of Gaussian distribution when variance (σ^2) is known. The sample size is set as 20 and the synthetic dataset is generated as $y_1, y_2, \ldots, y_{20}|\mu \sim N(\mu, 1)$. The prior distribution for μ is $\mu \sim N(0,\ \sigma_0^2)$. Different values of σ_0 are selected between 0.5 and 100, i.e., $\sigma_0 = (0.5,\ 5,\ 10,\ 20,\ 30,\ 40,\ 50,\ 60,\ 70,\ 80,\ 90,\ 100)$.
(2) Posterior of parameter μ of Gaussian distribution with unknown precision τ. For this purpose, a data set is simulated as $y_1, y_2, \ldots, y_{20} \sim N(\mu,\ 1/\tau)$. However, a joint prior distribution for μ and τ is set as $(\mu, \tau) \sim Normal{-}Gamma(\mu_0, \nu_0, \alpha_0, \beta_0)$.

To change the precision level, different values of ν_0 are chosen between 0.001 and 1, and $\mu = 0, \alpha = 2, \beta = 50$.

For both settings, observed data ($\mathbf{y_0}$) is simulated from $N(0,1)$. Sufficient summary statistics are $\mathbf{s}^* = \bar{y}$, for setting-1, and $\mathbf{s}^* = (\bar{y}, S)$ for setting-2. In each setting, the experiment is repeated for 100 Monte Carlo runs. The posterior distribution of μ is estimated in two main stages.

(a) Get a set of N-weighted particles that provide an approximation to the target posterior along with the corresponding summary statistics i.e.

$$\psi = \{(s^*_{\mathbf{1}},\ \mu_1)\,,\ (s^*_{\mathbf{2}},\ \mu_2)\,, \ldots, (s^*_N, \mu_N)\}.$$

(b) To obtain a final estimate of the posterior, apply FlexCode [22] on a set of approximate samples (ψ) obtained in the first stage.

For each setting, particles (ψ) and synthetic summary statistics are constructed via algorithm 1 (Section 2.3). For this, B=1000 and the adaptive sequence of tolerance thresholds ($\epsilon^{(r)}$) are considered. At the first iteration of the Algorithm 1, particles (μ_i) are generated from the prior distribution ($\pi(\theta)$). These particles are used to simulate data of size 20 from normal distribution i.e., $y_i \sim\ N(\mu, 1)$. Then, the Euclidean distance function ($\gamma\,(s_{\mathbf{0}},\ s^*)$) is used to measure dissimilarity between observed ($s_{\mathbf{0}}$) and synthetic data summary statistics (s^*). The value of $\gamma\,(s_{\mathbf{0}},\ s^*)$ is compared with the first tolerance threshold (ϵ_1). If $\gamma\,(s_{\mathbf{0}},\ s^*) < \epsilon_1$, the corresponding particle μ_i is retained otherwise rejected and the next value is simulated from the prior ($\pi(\mu)$). This process continues until a pool of N particles is accepted.

After the first iteration, instead of simulating from the prior the particles are sampled from the perturbed previous pool of accepted particles. The Gaussian kernel is used to perturb the accepted particles. Stating differently, instead of simulating particles from the prior distribution ($\pi(\mu)$) particles (μ_i) are sampled from the previous pool of accepted particles and perturbed by the Gaussian kernel. The tolerance threshold decreases at each iteration. To select the tolerance threshold, 75^{th} quantile of $\gamma(s^*, s_{\mathbf{0}})$ corresponding to accepted particles is taken. The set of weighted particles, obtained at the convergence of the algorithm provides an approximate posterior. The final estimate of the true posterior ($f(\mu|s_0)$) is obtained by applying the conditional density estimator FlexCode [22]. FlexCode is applied to the data set (ψ) obtained at the first stage.

This method takes the input of the particles as a response variable and the corresponding summary statistics (s^*) are treated as covariates. It transforms the response variable (μ_i) into Fourier basis ($\phi_j(\mu)$) and use it with covariates (s^*) to find regression function ($\beta_j\,(s^*) = \mathbb{E}(\phi_j(\mu)|s^*)$). In essence, to estimate the conditional mean ($\mathbb{E}(\phi_j(\mu)|s^*)$) FlexCode employs a regression technique. In this simulation study, the k-nearest neighbor regression technique is used to compute the conditional mean. When the conditional mean and Fourier transformation of response μ are combined via Equation (2.4), an efficient conditional density estimate is obtained. To show the efficiency of the PMC-CDE (Section 2.3), the average loss is used to compare the proposed PMC-CDE with the existing method ABC-CDE [23]. ABC-CDE is based on the basic ABC-rejection sampler.

4. Results

This section explains the results obtained by the implementation of the proposed technique (PMC-CDE). Densities estimated by PMC-CDE in both settings are compared with the densities obtained by applying ABC-CDE [23]. The whole experiment is repeated for different values of σ_0 and ν_0 under both settings (Section 3). Integrated squared error loss (Figure 3, Figure 4) is computed for both estimators and averaged over 100 Monte Carlo runs. Posterior densities of μ estimated by PMC-CDE (Section 2.3) and ABC-CDE [23] over a single value of σ_0 and ν_0 are shown in Figure 1 and 2. In Figure 1 and Figure 2,

true posteriors are also plotted. Figure 1 and 2 show that posterior density estimated by PMC-CDE is much closer to the true posterior [23] for both settings.

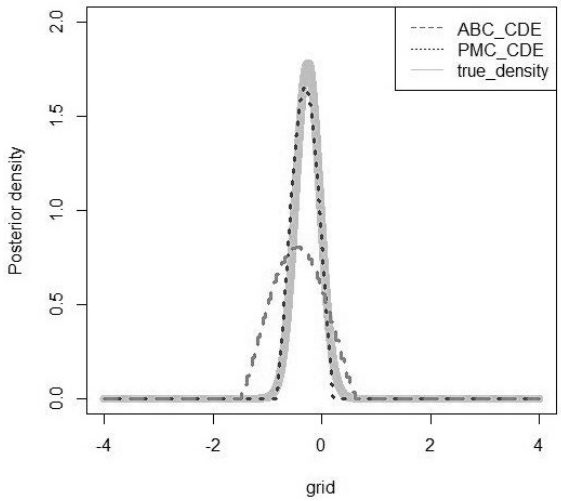

Figure 1. Posterior densities of μ estimated by ABC-CDE and PMC-CDE with $\sigma_0 = 40$.

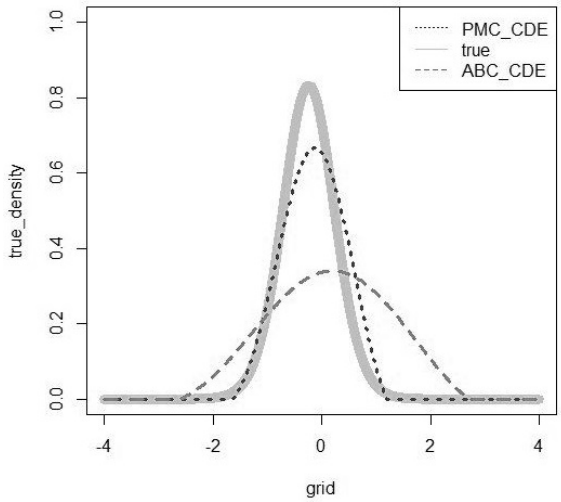

Figure 2. Posterior densities of μ estimated by ABC-CDE and PMC-CDE with $\nu_0 = 1$.

This experiment is repeated 100 times for every value of σ_0 under setting-1 and for each value of ν_0 under setting-2. Posterior densities are estimated by ABC-CDE [23] and PMC-CDE (Section 2.3) for both settings. Then, integrated squared error loss (Equation 2.6) of actual and estimated densities is computed for both settings. This loss is averaged over 100 Monte Carlo runs. The average loss of PMC-CDE is compared with that of ABC-CDE. This comparison is depicted in Figure 3 for setting-1 and Figure 4 for setting-2. These figures (Figure 3, Figure 4) demonstrate the performance of both methods in terms of average loss. For setting-1, as the value of σ_0 increases the average loss of ABC-CDE attains a higher value in contrast to the PMC-CDE approach i.e. the larger the variation in prior the better PMC-CDE performs as compared to ABC-CDE. For setting-2, as the value of ν_0 (precision) increases average loss for PMC-CDE decreases.

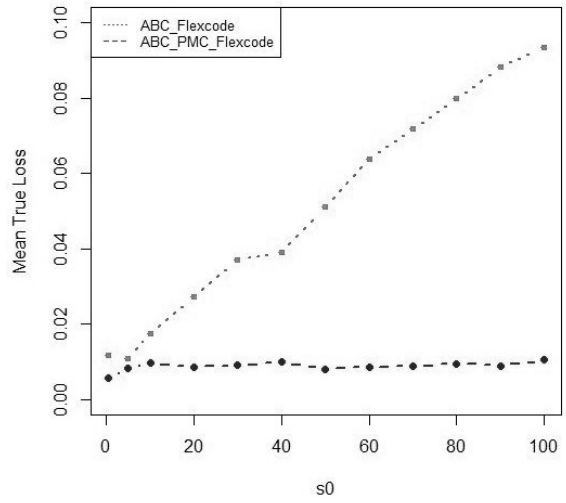

Figure 3. Average loss between estimated and true densities under setting-1.

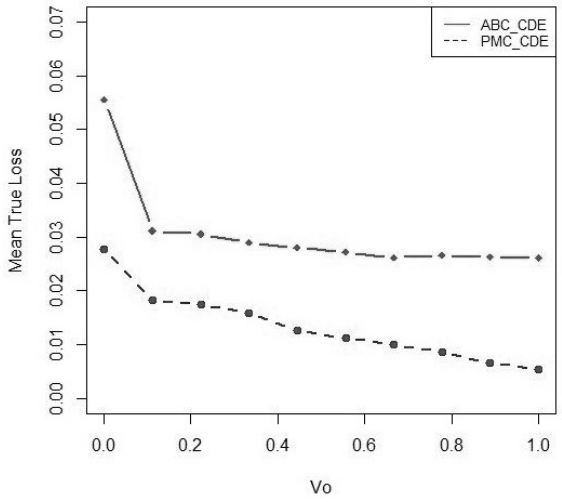

Figure 4. Average loss between estimated and true densities under setting-2.

For real-life application, the proposed methodology, PMC-CDE is applied on the cooperation variable of a real-life dataset Guyer [17]. This dataset consists of three variables. However, only one variable is considered in this study. Guyer dataset is taken from an R-package, CAR, and has 20 observations. Posterior densities are estimated based on this information and shown in Figure 5. These posterior densities are estimated by PMC-CDE and ABC-CDE. Credible intervals are also computed for these estimated posteriors. The interval (46.86445, 49.78514) is based on the PMC-CDE estimator and the interval (34.50141, 50.58128) is based on the ABC-CDE estimator. It is found that the estimated average value i.e. 48.3 of real-life datasets lies within these intervals. However, the length of the credible interval for the PMC-CDE estimator is shorter than that of ABC-CDE.

There is another real-life data application. The proposed technique PMC-CDE and ABC-CDE estimators are applied to the shrimp dataset which exists in R-package, MASS [34]. It is a univariate dataset of size 18. Densities estimated by employing this dataset are shown in Figure 6. Moreover, credible intervals based on PMC-CDE and ABC-CDE approaches are (30.72584, 32.8317) and (18.02356, 39.85572), respectively. The average

value of the shrimp dataset i.e. 31.7 lies within both intervals. Similar to the first application, the width of the credible interval based on PMC-CDE is smaller than that of ABC-CDE.

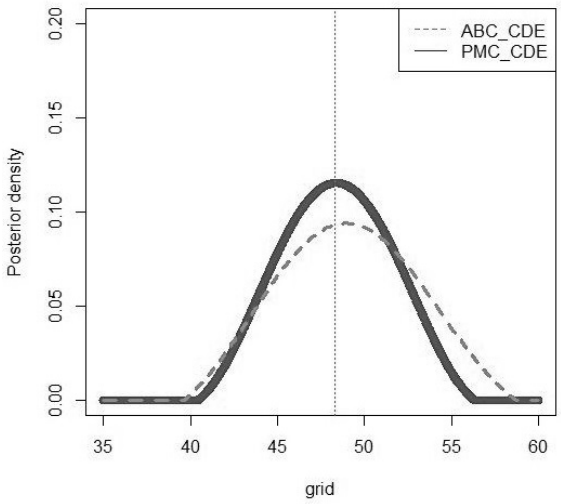

Figure 5. Posterior density of Guyer data by ABC-PMC and PMC-CDE.

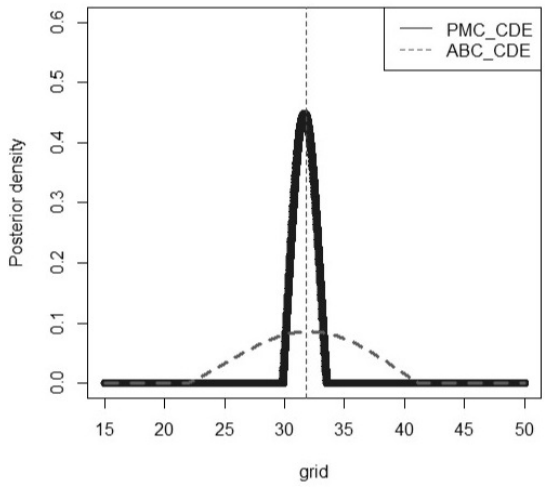

Figure 6. Posterior density of maximum stress estimated by ABC-CDE and PMC-CDE.

5. Conclusion and discussion

This paper suggests a technique for posterior density estimation. The proposed technique (PMC-CDE) is a combination of a sequential form of ABC i.e., ABC-PMC, and a conditional density estimation approach. Conditional density estimation is the generalized form of regression that explains the relationship between the response variable (θ) and covariates. PMC-CDE is a somewhat improved form of the existing method (ABC-CDE [23]) which is based on the classical rejection sampler of ABC.

PMC-CDE approach estimates the target posterior at two different stages. In the first stage, the proposed technique approximates the target posterior based on algorithm 1. The approximate posterior is obtained via iterative refinement of a sequence of intermediate

distributions using a sequence of decreasing tolerance thresholds. It gives a set of particles that are closely related to the given information. In the second stage, a conditional density estimator, FlexCode [22] is applied to obtain a final estimate of the posterior density. For this purpose, FlexCode requires a set of particles and associated synthetic summary statistics. FlexCode implements regression techniques along with the basis transformation of the response variable (θ) to provide an estimate of the posterior density. The performance of the proposed technique (PMC-CDE, Section 2.3) is demonstrated using synthetic and real-life data sets. The ISE loss of the proposed PMC-CDE is compared with that of another CDE estimator i.e., ABC-CDE [23]. Results demonstrate that PMC-CDE performs better in terms of loss as the prior variation increases. A potential use of the proposed technique is in the more complex models e.g., mixture models. Posteriors of mixture parameters can be estimated via PMC-CDE. Further, their comparative performance with other existing methods can be assessed.

References

[1] M.A. Beaumont and R. Bruce, *The Bayesian revolution in genetics*, Nat. Rev. Genet. **5** (4), 251-261, 2004.

[2] M.A. Beaumont, J.M. Cornuet, J.M. Marin and C.P. Robert, *Adaptive approximate Bayesian computation*, Biometrika **96** (4), 983-990, 2009.

[3] M.A. Beaumont, W. Zhang and D.J. Balding, *Approximate Bayesian computation in population genetics*, Genetics **162** (4), 2025-2035, 2002.

[4] J. Bi, W. Shen and W. Zhu, *Random forest adjustment for approximate Bayesian computation*, J. Comput. Graph. Statist. **31** (1), 64-73, 2022.

[5] G. Biau, F. Cérou and A. Guyader, *New insights into approximate Bayesian computation*, Ann. I. H. Poincare-P.R. **51** (1), 376-403, 2015.

[6] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[7] M. Blangiardo and C. Michela, *Spatial and Spatio-temporal Bayesian Models with R-INLA*, John Wiley & Sons, 2015.

[8] M.G. Blum and O. François, *Sequential Monte Carlo samplers*, J. R. Stat. Soc. Ser. B. Stat. Methodol. **68** (3), 411-436, 2006.

[9] M.G. Blum and O. François, *Non-linear regression models for approximate Bayesian computation*, Stat. Comput. **20** (1), 63-73, 2010.

[10] O. Cappé, A. Guillin, J.M. Marin and C.P. Robert, *Population Monte Carlo*, J. Comput. Graph. Statist. **13** (4), 907-929, 2004.

[11] N.N. Cencov, *Estimation of an unknown distribution density from observations*, Soviet Math. **9** (2), 1559-1566, 1962.

[12] R. Christian and C. George, *Monte Carlo Statistical Methods*, Springer Science & Business Media, 2013.

[13] N. de Freitas and N.J. Gordon, *An Introduction to Sequential Monte Carlo Methods*, Springer, 2001.

[14] S. Efromovich, *Orthogonal series density estimation*, Wiley Interdiscip. Rev. Comput. Stat. **2** (4), 467-476, 2010.

[15] S. Efromovich, *Fast nonparametric conditional density estimation*, arXiv: 1206.5278 [stat.ME].

[16] E. Fix, *Discriminatory Analysis: Nonparametric Discrimination, Consistency Properties*, USAF School of Aviation Medicine, 1951.

[17] J. Fox, S. Weisberg, D. Adler, D. Bates, G. Baud-Bovy, S. Ellison, D. Firth, M. Friendly, G. Gorjanc, S. Graves and R. Heiberger, Package: "car", R package version: 3.1-1, 2022.

[18] C. Grazian and Y. Fan, *A review of approximate Bayesian computation methods via density estimation: Inference for simulator-models*, Wiley Interdiscip. Rev. Comput. Stat. **12** (4), e1486, 1-16, 2020.

[19] C. Hahn, M. Vakili, K. Walsh, A.P. Hearin, D.W. Hogg and D. Campbell, *Approximate Bayesian computation in large-scale structure: constraining the galaxy-halo connection*, Mon. Notices Royal Astron. Soc. **469** (3), 2791-2805, 2017.

[20] E.E. Ishida, S.D.P. Vitenti, M. Penna-Lima, J. Cisewski, R.S. de Souza, A.M.M. Trindadei E. Cameron, V.C. Busti, for the COIN collaboration, *Cosmoabc: likelihood-free inference via population Monte Carlo approximate Bayesian computation*, Astron. Comput. **13**, 1-11, 2015.

[21] R. Izbicki and A.B. Lee, *Nonparametric conditional density estimation in a high-dimensional regression setting*, J. Comput. Graph. Statist. **25** (4), 1297-1316, 2016.

[22] R. Izbicki and A.B. Lee, *Converting high-dimensional regression to high-dimensional conditional density estimation*, Electron. J. Stat. **11** (2), 2800-2831, 2017.

[23] R. Izbicki, A.B. Lee and T. Pospisil *ABC–CDE: Toward approximate Bayesian computation with complex high-dimensional data and limited simulations*, J. Comput. Graph. Statist. **28** (3), 481-492, 2019.

[24] D.V. Lindley, *Bayesian Statistics: A Review*, SIAM Publishers, 1972.

[25] S. Mallat, *A Wavelet Tour of Signal Processing*, Elsevier, 1999.

[26] P. Marjoram, J. Molitor, V. Plagnol and S. Tavaré, *Markov chain Monte Carlo without likelihoods*, Proc. Nat. Acad. Sci. USA **100** (26), 15324-15328, 2003.

[27] N. Meinshausen, *Quantile regression forests*, J. Mach. Learn. Res. **7**, 983-999, 2006.

[28] P.D. Moral, A. Doucet and A. Jasra, *An adaptive sequential Monte Carlo method for approximate Bayesian computation*, Stat. Comput. **22** (5), 1009-1020, 2012.

[29] R. Nayek, M.A.B. Abdessalem, N. Dervilis, E.J. Cross and K. Worden, *Approximate Bayesian Inference for Piecewise-Linear Stiffness Systems*, in: Nonlinear Structures & Systems, Volume 1: Proceedings of the 40th IMAC, A Conference and Exposition on Structural Dynamics, Cham, Springer International Publishing, 2022.

[30] E. Numminen, L. Cheng, M. Gyllenber and J. Corander, *Estimating the transmission dynamics of Streptococcus pneumoniae from strain prevalence data*, Biometrics **69** (3), 748-757, 2013.

[31] T. Pospisil and A.B. Lee, *RFCDE: Random forests for conditional density estimation*, arXiv: 1804.05753 [stat.ML].

[32] T.P. Prescott and R.E. Baker, *Multifidelity approximate Bayesian computation with sequential Monte Carlo parameter sampling*, SIAM-ASA J. Uncertain. **9** (2), 788-817, 2021.

[33] J.K. Pritchard, M.T. Seielstad, A. Perez-Lezaun and M.W. Feldman, *Population growth of human Y chromosomes: a study of Y chromosome microsatellites*, Mol. Biol. Evol. **16** (12), 1971-1798, 1999.

[34] B. Ripley, B. Venables, D.M. Bates, K. Hornik, A. Gebhardt, D. Firth and M.B. Ripley, Package: "MASS", R package version: 7.3-58.2, 2022.

[35] M. Rosenblatt, *Conditional probability density and regression estimators*, in: P.R. Krishnaiah (ed.) Multivariate Analysis II, 25–31, Academic Press, New York, 1969.

[36] B.W. Silverman, *Density Estimation for Statistics and Data Snalysis*, CRC Press, 1986.

[37] U. Simola, J. Cisewski-Kehe, M.U. Gutmann and J. Corander, *Adaptive approximate Bayesian computation tolerance selection*, Bayesian Anal. **16** (2), 397-423, 2021.

[38] U. Simola, B. Pelssers, D. Barge, J. Conrad and J. Corander, *Machine learning accelerated likelihood-free event reconstruction in dark matter direct detection*, J. Instrum. **14** (3), P03004, 2019.

[39] S.A. Sisson, Y. Fan and M. Beaumont, *Handbook of Approximate Bayesian Computation*, CRC Press, 2018.

[40] S.A. Sisson, Y. Fan and M.M. Tanaka, *Sequential Monte Carlo without Likelihoods*, Proc. Natl. Acad. Sci. USA **104** (6), 1760-1765, 2007.
[41] M.L. Stein, C. Zhiyi and W.J. Leah, *Approximating likelihoods for large spatial data sets*, J. R. Stat. Soc. Ser. B. Stat. Methodol. **66** (2), 275-296, 2004.
[42] M.S. Stigler, *Thomas Bayes's Bayesian inference*, J. Roy. Statist. Soc. Ser. A **145** (2), 250-258, 1982.
[43] R. Swinburne, *Bayes' theorem*, Rev. Philos. Fr. Etrang **194** (2), 250-251, 2004.
[44] S. Tavaré, D.J. Balding, R.C. Griffiths and P. Donnelly, *Inferring coalescence times from DNA sequence data*, Genetics **145** (2), 505-518, 1997.
[45] G.R. Terrell, *The maximal smoothing principle in density estimation*, J. Amer. Statist. Assoc. **85** (410), 470-477, 1990.
[46] T. Toni, D. Welch, N. Strelkowa, A. Ipsen and M.P. Stumpf, *Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems*, J. R. Soc. Interface **6** (31), 187-202, 2009.
[47] B.M. Turner and T. van Zandt, *Hierarchical approximate Bayesian computation*, Psychometrika **79** (2), 185-209, 2014.
[48] L. Wasserman, *All of Nonparametric Statistics*, Springer Science & Business Media, 2006.
[49] S. Watson, *Sequential methods in approximate Bayesian computation*, PhD thesis, University of Bristol, 2018.
[50] A. Weyant, C. Schafer and W.M. Wood-Vasey, *Likelihood-free cosmological inference with type Ia supernovae: approximate Bayesian computation for a complete treatment of uncertainty*, Astrophys. J. **764** (2), 116, 2013.