Predictive Approximate Bayesian Computation via Saddle Points

Yingxiang Yang[*] Bo Dai[⋆] Negar Kiyavash[†] Niao He[*†]
{yyang172,kiyavash,niaohe} @illinois.edu
bohr.dai@gmail.com

Abstract

Approximate Bayesian computation (ABC) is an important methodology for Bayesian inference when the likelihood function is intractable. Sampling-based ABC algorithms such as rejection- and K2-ABC are inefficient when the parameters have high dimensions, while the regression-based algorithms such as K- and DR-ABC are hard to scale. In this paper, we introduce an optimization-based ABC framework that addresses these deficiencies. Leveraging a generative model for posterior and joint distribution matching, we show that ABC can be framed as saddle point problems, whose objectives can be accessed directly with samples. We present *the predictive ABC algorithm (P-ABC)*, and provide a probabilistically approximately correct (PAC) bound that guarantees its learning consistency. Numerical experiment shows that P-ABC outperforms both K2- and DR-ABC significantly.

1 Introduction

Approximate Bayesian computation (ABC) is an important methodology to perform Bayesian inference on complex models where likelihood functions are intractable. It is typically used in large-scale systems where the generative mechanism can be simulated with high accuracy, but a closed form expression for the likelihood function is not available. Such problems arise routinely in modern applications including population genetics [Excoffier, 2009, Drovandi and Pettitt, 2011], ecology and evolution [Csilléry et al., 2012, Huelsenbeck et al., 2001, Drummond and Rambaut, 2007], state space models [Martin et al., 2014], and image analysis [Kulkarni et al., 2014].

Formally, ABC aims to estimate the posterior distribution $p(\theta|y) \propto \pi(\theta)p(y|\theta)$. The word "approximate" refers to the fact that the joint distribution $\pi(\theta)p(y|\theta)$ is only available through fitting simulated data $\{(\theta_j, y_j)\}_{j=1}^N \sim p(y|\theta)\pi(\theta)$. Based on how the fitting is performed, existing ABC methods can be summarized into two main categories: sampling- and regression-based algorithms.

Sampling-based algorithms. A sampling-based algorithm directly approximates the likelihood function using simulated samples "similar" to the true observations according to certain choices of similarity measurement based on informative summary statistics, e.g., [Joyce and Marjoram, 2008, Nunes and Balding, 2010, Blum and François, 2010, Wegmann et al., 2009, Blum et al., 2013]. More recent representative algorithms include rejection ABC, indirect score ABC [Gleim and Pigorsch], K2-ABC [Park et al., 2016], distribution regression ABC (DR-ABC) [Mitrovic et al., 2016], expectation propagation ABC (EP-ABC) [Barthelmé and Chopin, 2011], random forest ABC [Raynal et al., 2016], Wasserstein ABC [Bernton et al., 2017], Copula ABC [Li et al., 2017], and ABC aided by neural network classifiers [Gutmann et al., 2014, 2016]. The aformentioned work can be viewed under a unified framework that approximates the posterior $p(\theta|y)$ with

[*] Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign.
[†] Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign.
[⋆] Google Brain.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

$$p_\epsilon(\theta|y) \propto \int_{\mathcal{Y}} K_\epsilon(s_x, s_y) p(x|\theta)\pi(\theta)\mathrm{d}x \approx \frac{\pi(\theta)}{N} \sum_{i=1}^{N} K_\epsilon(s_{x_i}, s_y), \tag{1}$$

where $\mathcal{Y}$ is the domain for the samples x and y, and the x_i's are drawn from the model $p(x_i|\theta)$. We use s_y to denote the summary statistics for y, and let $K_\epsilon(s_x, s_y)$ be an appropriate weighting kernel that measures similarity. For example, when $K_\epsilon(s_x, s_y) = \mathbf{1}\{s_x = s_y\}$ with $s_x = x$ and θ and y discrete, (1) recovers the true posterior asymptotically. When $K_\epsilon(s_x, s_y) = \mathbf{1}\{\rho(s_x, s_y) \leq \epsilon\}$ for some metric ρ, (1) reduces to rejection-ABC. When $K_\epsilon(s_x, s_y) = \exp(-\rho(s_x, s_y))/\epsilon)$, (1) reduces to soft-ABC [Park et al., 2016], which can also be regarded as a variation of the synthetic likelihood inference [Wood, 2010, Price et al., 2018] under the Bayesian setting. Finally, when $K_\epsilon(s_x, s_y)$ is the zero/one output of a neural network classifier, (1) reduces to ABC via classification [Gutmann et al., 2018].

Note that, most of the aforementioned algorithms require summary statistics and a smoothing kernel, which suffer from information loss when the summary statistics are insffucient, and introduce bias. To address the issue of insufficient summary statistics, Park et al. [2016] proposed K2-ABC, in which $K_\epsilon(s_x, s_y)$ is replaced by a smoothing kernel over the empirical maximum mean discrepancy (MMD) obtained from kernel embedding of the empirical distributions of the samples. When a characteristic kernel is selected, the kernel embedding of the distribution will be a sufficient statistics, therefore, without information loss. Meanwhile, Rodrigues et al. [2018] proposed recalibration techniques to debias the estimates of ABC algorithms. However, despite their simplicity and continuous improvements, sampling-based ABC algorithms suffer from bias caused by the weighting kernel K_ϵ, and the potential need of large amount of samples when the dimensions of θ and y are large.

Regression-based algorithms. Regression-based ABC algorithms establish regression relationships between the model parameter and the simulated data within an appropriate function space $\mathcal{F}$. Representative algorithms in this category include high-dimensional ABC [Nott et al., 2014], Kernel-ABC (K-ABC) [Blum et al., 2013], DR-ABC [Mitrovic et al., 2016]. In DR-ABC, the posterior is obtained by performing a distribution regression using the given samples. In contrary to the sampling-based algorithms, regression-based algorithms mitigate the bias introduced by the smoothing kernel. However, they do not provide an estimation for the posterior density. Meanwhile, it is often hard for such algorithms to scale. For example, the distribution regression involved in DR-ABC requires computing the inverse of an $N \times N$ kernel matrix, which has $\mathcal{O}(N^3)$ computation cost as the dataset scales.

Neither sampling- nor regression-based algorithms are satisfactory: while regression-based algorithms have better performances compared to the sampling-based algorithms, they are not scalable to high dimensions. Therefore, an important question is whether one can design an algorithm that can perform well on large datasets? In this paper, we propose an optimization-based ABC algorithm that can successfully address the deficiencies of both sampling- and regression-based algorithms. In particular, we show that ABC can be formulated under a unified optimization framework: finding the saddle point of a minimax optimization problem, which allows us to leverage powerful gradient-based optimization algorithms to solve ABC. More specifically, our contributions are three-fold: **First**, we start with a generative model for posterior approximation and show that the ABC problem can be formulated as a saddle point optimization through both joint distribution matching and posterior matching. This approach circumvents the difficulties associated with choosing sufficient summary statistics or computing kernel matrices, as needed in K2- and DR-ABC. More critically, the saddle point objectives can be evaluated based purely on samples, without assuming any implicit form of the likelihood. **Second**, we provide an efficient SGD-based algorithm for finding the saddle point, and provide a probabilistically approximately correct (PAC) bound guaranteeing the consistency of the solution to the problem. **Numerically**, we compare the proposed algorithm to K2- and DR-ABC. The experiment shows that our algorithm outperforms K2- and DR-ABC significantly and is close to optimal on the toy example dataset.

2 Approximate Bayesian Computation via Saddle Point Formulations

When the likelihood function is given, the true posterior $p(\theta|y)$ given observation y can be obtained by optimizing the *evidence lower bound* (ELBO) in the space $\mathcal{P}$ that contains all probability density functions [Zellner, 1988],

$$\min_{q \in \mathcal{P}} \ \mathrm{KL}(q||\pi) - \mathbb{E}_{\theta \sim q}[\log p(y|\theta)], \tag{2}$$

Divergence	Saddle point objective
χ^2 divergence	$\mathbb{E}_{(\theta,y)\sim p(y\mid\theta)\pi(\theta)}[u(\theta,y)] - \mathbb{E}_{\theta\sim f(y,\xi),\xi\sim p_0(\xi),y\sim p(y)}[u(\theta,y)+u^2(\theta,y)/4]$
Wasserstein distance	$\mathbb{E}_{(\theta,y)\sim p(y\mid\theta)\pi(\theta)}[u(\theta,y)] - \mathbb{E}_{\theta\sim f(y,\xi),\xi\sim p_0(\xi),y\sim p(y)}[u(\theta,y)]$
KL divergence	$\mathbb{E}_{(\theta,y)\sim p(y\mid\theta)\pi(\theta)}[u(\theta,y)] - \mathbb{E}_{\theta\sim f(y,\xi),\xi\sim p_0(\xi),y\sim p(y)}[1+\log(u(\theta,y))]$

Table 1: A list of divergences and their corresponding saddle point objective.

where KL denotes the Kullback-Leibler divergence: $\mathrm{KL}(q\|\pi) = \mathbb{E}_{\theta\sim q}[\log\frac{q(\theta)}{\pi(\theta)}]$. When dealing with an intractable likelihood, this conventional optimization approach cannot work without combining it with methods that fit $p(y|\theta)$ with samples. In this paper, we introduce a new class of optimization objective that allows the learner to directly leverage the samples from the likelihood $p(y|\theta)$, which is available under the ABC setting, for estimating the posterior. The method we propose does not merely find $\theta^* = \operatorname{argmax}_\theta p(\theta|y)$ for a given data point y, but rather finds a representation of θ generated from $p(\theta|y)$ using a generative model $\theta = f(y,\xi)$ for any data y, with $\xi \sim p_0(\xi)$ generated from an appropriately chosen distribution. This idea is inspired by the recent success of generative models in variational inference [Kingma and Welling, 2013]. Note that in the context of ABC, several recent work have also considered generative models, e.g., Tran et al. [2017]. However, the generative models in these work are typically used for the purpose of modeling the implicit likelihood in order to generate a synthetic observation y, rather than approximating the posterior distribution. In this section, we propose two models for approximate Bayesian computation, both of which can be reformulated as saddle point problems and can be solved by stochastic gradient algorithm.

2.1 Joint Distribution Matching

Given the availability of sampling from the joint distribution $p(y|\theta)\pi(\theta)$ and from the posterior $p(\theta|y)$ through $f(y,\xi)$, recall the joint distribution can be written as $p(y|\theta)\pi(\theta) = p(\theta|y)p(y)$, a natural idea for estimating the posterior is through matching the empirical joint distributions.

Let us denote D_ν as the f-divergence associated with some convex function ν, i.e., $D_\nu(p,q) = \int q(x)\nu\left(\frac{p(x)}{q(x)}\right)\mathrm{d}x$. We then have the following divergence minimization problem for ABC:

$$p(\theta|y) = \operatorname*{argmin}_{q(\theta|y)\in\mathcal{P}} D_\nu\left(p(y|\theta)\pi(\theta), q(\theta|y)p(y)\right), \tag{3}$$

in which the divergence integrates over θ and y. The above optimization problem is difficult to solve due to the nonlinearity of the objective in terms of the joint distributions. This nonlinearity makes gradient computation hard as the f-divergence cannot be computed directly through samples obtained from the joint distribution. To address this issue, we apply Fenchel duality and the interchangeability principle as introduced in Dai et al. [2017], which yield an equivalent saddle point reformulation:

$$\min_{f\in\mathcal{F}}\max_{u\in\mathcal{U}} \Phi(f,u) := \mathbb{E}_{(\theta,y)\sim p(y|\theta)\pi(\theta)}\left[u(\theta,y)\right] - \mathbb{E}_{\xi\sim p_0(\xi),y\sim p(y)}\left[\nu^*\left(u(f(y,\xi),y)\right)\right]. \tag{4}$$

Due to the space limitation, please refer to Appendix B for the details of the derivation. In this equivalent formulation, $\mathcal{U}$ is a function space that contains $u^*(\theta,y) = \nu'(\frac{p(y|\theta)p(\theta)}{p(\theta|y)p(y)})$ and ν^* is the Fenchel dual of ν. When $\mathcal{F}$ is experessive enough, a saddle point solution to (4) would recover the posterior distribution.

The class of f-divergence covers many common divergences, including the KL divergence, Pearson χ^2 divergence, Hellinger distance, and Jensen-Shannon divergence. Other than f-divergence, we can also employ other metrics to measure the distance between the joint distributions $p(y|\theta)p(\theta)$ and $p(\theta|y)p(y)$, e.g., the Wasserstein distance. If the training data come with labels, we can also choose the objective function to be the mean square error between θ^* and $\widehat{\theta}$. In Table 1, we provide some examples of divergences and their corresponding saddle point objectives. From a density ratio estimation perspective, the optimal solution of the dual variable, $u(\theta,y)$, is a discriminator that tells the true and synthetic joint distributions by computing their density ratios, which is related to the ratio matching in Mohamed and Lakshminarayanan [2016].

2.2 Posterior Distribution Matching

Another way to learn the posterior representation is by directly matching the posterior distributions. Specifically, we plug the generative model into the objective function defined in K-ABC,

$$\min_{f\in\mathcal{F}}\max_{h\in\mathcal{H}} \mathbb{E}_y\left[(\mathbb{E}_{\theta|y}[h(\theta)] - \mathbb{E}_{\xi\sim p_0(\xi)}[h(f(y,\xi))])^2\right] \tag{5}$$

Directly solving the optimization (5) is extremely difficult due to the inner conditional expectation. We reformualte it to an equivalent saddle point formulation by applying Fenchel duality and the interchangeability principle introduced in Dai et al. [2017],

$$\min_{f\in\mathcal{F}}\max_{h\in\mathcal{H},v\in\mathcal{V}} \mathbb{E}_{(\theta,y)\sim p(y|\theta)\pi(\theta)}\left[v(y)h(\theta)\right] - \mathbb{E}_{\xi\sim p_0(\xi),y\sim p(y)}\left[v(y)h(f(y,\xi))\right] - \frac{1}{4}\mathbb{E}_y\left[v^2(y)\right] \quad (6)$$

where $\mathcal{V}$ is the entire space of functions on $\mathcal{Y}$. We provide the details of the equivalence proof between (5) and (6) in Appendix B. The resulting saddle point objective (6) is much easier to solve than (5) and in particular stochastic gradient-based methods could be applied.

2.3 Discussion

The saddle point framework for ABC is closely related to both regression- and GAN-based ABC algorithms.

Regression-based ABC algorithms, such as K-ABC, aim to compute the conditional expectation of the posterior by finding its conditional kernel embedding $C(y) : \mathcal{Y} \to \mathcal{H}$ in an RKHS. With such parametrization, the objective (5) becomes

$$\min_{C:\mathcal{Y}\to\mathcal{H}} L(C) := \sup_{\|h\|_{\mathcal{H}}\leq 1} \mathbb{E}_y[(\mathbb{E}_{\theta|y}[h(\theta)] - \langle h, C(y)\rangle_{\mathcal{H}})^2].$$

This problem is further relaxed to a distribution regression problem by swapping the square operator with the inner expectation, which leads to minimizing $\mathbb{E}_{\theta,y}[\|K(\cdot,\theta) - C(y)\|^2]$, an upper bound of $L(C)$. Specifically, we have that

$$\begin{aligned}
&\sup_{\|h\|_{\mathcal{H}}\leq 1} \mathbb{E}_y\left[\left(\mathbb{E}_{\theta|y}\left[h(\theta)\right] - \langle h, C(y)\rangle_{\mathcal{H}}\right)^2\right] \leq \sup_{\|h\|_{\mathcal{H}}\leq 1} \mathbb{E}_y\left[\left(\mathbb{E}_{\theta|y}\left[\langle h, k(\cdot,\theta)\rangle\right] - \langle h, C(y)\rangle_{\mathcal{H}}\right)^2\right] \\
\leq\ & \sup_{\|h\|_{\mathcal{H}}\leq 1} \mathbb{E}_{\theta,y}\left[\langle h, k(\cdot,\theta) - C(y)\rangle^2\right] \leq \sup_{\|h\|_{\mathcal{H}}\leq 1} \|h\|^2_{\mathcal{H}}\, \mathbb{E}_{\theta,y}\left[\|k(\cdot,\theta) - C(y)\|^2\right] \\
=\ & \mathbb{E}_{\theta,Y}\left[\|k(\cdot,\theta) - C(y)\|^2\right].
\end{aligned}$$

In contrast, the proposed optimization framework for posterior matching does not restrict the testing function to an RKHS, and moreover, the saddle point objective (6) is an exact reformulation of (5), rather than an upper bound.

GAN-based ABC algorithms are algorithms that leverage the representation power of the neural networks to optimize the ELBO. One example is the use of variational autoencoder (VAE), where both q and p in (2) are represented by Gaussian distributions parameterized by neural networks. Better performances have been observed in Mescheder et al. [2017] by embedding the optimal value of $q(\theta|y)$ as the optimal solution of a real-valued discriminator network, equivalent to performing reparametrization. However, compared to the saddle point formulation, Mescheder et al. [2017] requires computing an additional layer of optimization due to the embedding performed. Meanwhile, when the underlying parameter is discrete, the saddle point formulation can be viewed as a special case of conditional GAN (CGAN) [Mirza and Osindero, 2014].

3 Algorithm and Theory

In this section, we analyze the learning rate of the proposed saddle point framework by providing a probabilistically approximately correct (PAC) learning bound in Theorem 1.

3.1 Theoretical Properties

We study the statistical error for solving solving the finite-sample approximation (i.e., empirical risk) of the saddle point problem (4), namely,

$$\min_{f\in\mathcal{F}}\max_{u\in\mathcal{U}} \widehat{\Phi}(f,u) := \frac{1}{N}\sum_{i=1}^{N}[u(\theta_i, y_i) - \nu^*(u(f(y_i,\xi_i), y_i))],$$

whose solution we denote as f_N^* and u_N^*. We give a high probability upper bound on

$$\epsilon_N = D_\nu(p(y|\theta)\pi(\theta), q_N^*(\theta|y)p(y)) - D_\nu(p(y|\theta)\pi(\theta), q^*(\theta|y)p(y)),$$

given $(\theta_i, \xi_i, y_i)_{i=1}^N \sim p(y|\theta)\pi(\theta)p_0(\xi)$, where we denote f^* and u^* as the solution to (4). By invoking the tail inequality in Antos et al. [2008] and the ϵ-net argument, we have the following theorem, the proof of which can be found in Appendix A.

Algorithm 1 Predictive ABC

```
**Input:** Maximum number of iterations $T$. Prior distribution $\pi(\theta)$, model $p(y|\theta)$. Step sizes $\{\eta_k^u\}_{k=1}^T$ and $\{\eta_k^f\}_{k=1}^T$, objective function $\Phi$.
**Initialize:** $f_1$, $u_1$.
**for** $k = 1$ **to** $T$ **do**
    Sample $\theta_k$ from $\pi(\theta)$, and generate $y_k \sim p(y|\theta_k)$. Sample $\xi_k$ from $p_0(\xi)$.
    $f_{k+1} \leftarrow \Pi_{\mathcal{F}}(f_k - \eta_k^f \nabla_f \Phi(f_k, u_k))$.
    $u_{k+1} \leftarrow \Pi_{\mathcal{U}}(u_k - \eta_k^u \nabla_u \Phi(f_k, u_k))$.
**end for**
**Output:** $\bar{f} = \frac{\sum_{k=1}^T \eta_k^f f_k}{\sum_{k=1}^T \eta_k^f}$, and $\bar{u} = \frac{\sum_{k=1}^T \eta_k^u u_k}{\sum_{k=1}^T \eta_k^u}$.
```

Theorem 1. Suppose $(\theta_i, \xi_i, y_i)_{i=1}^N$ is a β-mixing sequence [2] with $\beta_m \leq \bar{\beta} \exp(-bm^\kappa)$ for constants $\bar{\beta}$, b and κ, and suppose that function class $\mathcal{U}$ has a finite pseudo dimension D. [3] In addition, suppose that $u \in [-C_u, C_u]$ and the Fenchel dual satisfies $\nu^*(u) \leq C_\nu$. Then, with probability $1 - \delta$,

$$\epsilon_N \leq \sqrt{\frac{C_1(\max(C_1/b, 1))^{1/\kappa}}{C_2 N}},$$

where $C_1 = 0.5D \log N + \log(e/\delta) + [\log(2 \max(16e(D+1)C_2^{D/2}, \bar{\beta}))]_+$ and $C_2 = 1/(512(C_\nu + C_u)^2)$.

Theorem 1 states that learning is consistent at a rate of $\mathcal{O}(N^{-1/2} \log N)$ where N is the number of samples, when the empirical saddle point approximation can be exactly solved. From a theoretical perspective, global convergence of first-order methods such as stochastic gradient descent (SGD) can be achieved when the objective function $\Phi(f, u)$ is convex-concave. For example, when u belong to an RKHS and the conditional expectation operator with respect to $q(\theta|y)$ can be embedded within an RKHS; or when u and f belong to RKHSs where $\mathcal{U}$ has a linear reproducing kernel. More often than not, the objective function is not convex-concave, for which stochastic gradient descent (SGD) based algorithms are only guaranteed to converge towards a stationary point in certain restricted cases [Sinha et al., 2017, Li and Yuan, 2017, Kodali et al., 2018]. Below, we provide an SGD-based algorithm that works well in practice.

3.2 The Predictive-ABC Algorithm

We introduce the algorithm, Predictive ABC (P-ABC), for solving the empirical counterpart of (4), in Algorithm 1. Under the reparametrization that represents $\theta \sim p(\theta|y)$ as $\theta = f(y, \xi)$, the stochastic gradients of $\Phi(f, u)$ in (4) can be computed from the chain rule:

$$\nabla_f \Phi(f, u) = -\frac{\mathrm{d}\nu^*(x)}{\mathrm{d}x}\bigg|_{x=u(\theta,y), \theta=f(y,\xi)} \cdot \frac{\partial u(\theta, y)}{\partial \theta}\bigg|_{\theta=f(y,\xi)} \cdot \nabla_f f(y, \xi), \tag{7}$$

$$\nabla_u \Phi(f, u) = \nabla_u u(\theta, y) - \frac{\mathrm{d}\nu^*(x)}{\mathrm{d}x}\bigg|_{x=u(\theta,y), \theta=f(y,\xi)} \cdot \nabla_u u(f(y, \xi), y). \tag{8}$$

When $\mathcal{F}$ and $\mathcal{U}$ are RKHSs, we have $\nabla_f f(y, \xi) = K_{\mathcal{F}}((y, \xi), \cdot)$ and $\nabla_u u(\theta, y) = K_{\mathcal{U}}((\theta, y), \cdot)$ where $K_{\mathcal{F}}$ and $K_{\mathcal{U}}$ are the reproducing kernels of $\mathcal{F}$ and $\mathcal{U}$. Operations $\Pi_{\mathcal{F}}$ and $\Pi_{\mathcal{U}}$ Denote the projections onto the spaces $\mathcal{F}$ and $\mathcal{U}$, respectively. When f and u are represented by neural networks, ∇_f and ∇_u can be viewed as the gradients with respect to the coefficients representing those neural networks, which can be efficiently calculated through back propagation.

One advantage of Algorithm 1 over sampling- and regression-based ABC algorithms is that it is capable of estimating the conditional distribution of $p(\theta|y)$ for any given collection of training samples $(\theta_k, y_k)_{k=1}^N$, while most sampling- and regression-based ABC algorithms require, for each θ_k, at least

[2] A discrete time stochastic process is mixing if widely separated events are asymptotically independent. Here, β_m provides an upper bound on the dependency of two events separated by n intervals of time. See Meir [2000] for a detailed definition.

[3] Pseudo dimension, also known as the Pollard dimension, is a generalization of VC dimension to the function class (see chapter 11 of Anthony and Bartlett [2009]).

two y_k's drawn from $p(y|\theta_k)$ in order to compute summary statistics. In particular, K2- and DR-ABC both need large number of samples drawn from the conditional distribution corresponding to the same parameter in order to compute the maximum mean discrepancy (MMD) statistics accurately. In fact, MMD cannot be computed when only one y_k is available for each θ_k. This advantage is due to the saddle point formulation (4) of the original learning problem (3).

4 Numerical Experiment

We tested the performance of the proposed saddle point framework on ABC. For benchmarks, we chose K2- and DR-ABC as the representatives from sampling- and regression-based ABC algorithm categories, respectively.

4.1 Synthetic Dataset I: Superpoisition of Uniform Distributions

Consider a toy example where we observe a set of samples $Y^* := \{y_i^*\}_{i=1}^n$ where all y_i^*'s correspond to the same underlying parameter $\theta^* \in \mathbb{R}^p$. We assume the generating model is $y_i^* = \theta^* + u_i$ with $\{u_i\}_{i=1}^n$ being i.i.d. random vectors and each dimension of u_i as well as each dimension of θ^* is uniformly distributed on $[-0.5, 0.5]$. Denote the p coordinates of each observed sample y_i^* and θ^*, respectively, as $y_{i1}^*, \ldots, y_{ip}^*$ and $\theta_1^*, \ldots, \theta_p^*$, then the posterior can be written as

$$p(\theta^*|Y^*) \propto \prod_{j=1}^{p} \mathbb{1}\left[\max\{-0.5, \max_{i\in\{1,\ldots,n\}} y_{ij}^* - 0.5\} \leq \theta_j^* \leq \min\{0.5, \min_{i\in\{1,\ldots,n\}} y_{ij}^* + 0.5\}\right],$$

which is a uniform distribution whose boundary on the j-th dimension is defined by the values of the maximum and minimum values of the j-th coordinate among all y_i^*'s. Due to the fact that K2- and DR-ABC evaluate their performances using the mean square error, we use predictive ABC (P-ABC) to find the optimal minimum mean square error (MMSE) estimator for θ^*. We denote the optimal estimator by $\widehat{\theta}_{\text{opt}}$, which has a closed form solution with the j-th coordinate being

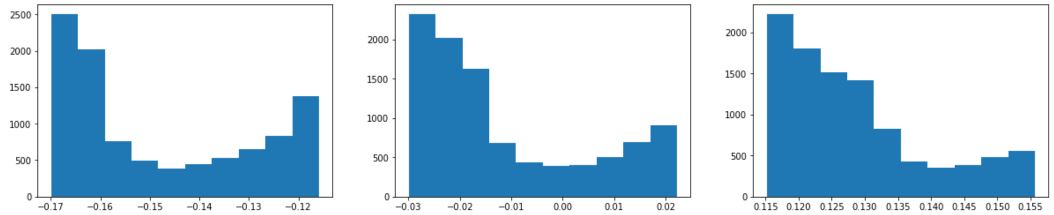
$$\widehat{\theta}_j = \begin{cases} \frac{1}{2} \cdot \max_{i\in\{1,\ldots,n\}} y_{ij}^*, & \min_{i\in\{1,\ldots,n\}} y_{ij}^* \geq 0 \\ \frac{1}{2} \cdot \min_{i\in\{1,\ldots,n\}} y_{ij}^*, & \max_{i\in\{1,\ldots,n\}} y_{ij}^* \leq 0 \\ \frac{1}{2}\left(\max_{i\in\{1,\ldots,n\}} y_{ij}^* + \min_{i\in\{1,\ldots,n\}} y_{ij}^*\right), & \min_{i\in\{1,\ldots,n\}} y_{ij}^* \leq 0 \leq \max_{i\in\{1,\ldots,n\}} y_{ij}^* \end{cases},$$

for all $j \in \{1, \ldots, p\}$. A sub-optimal estimator for this example is $\widehat{\theta}_{\text{ave}} = n^{-1}\sum_{i=1}^n (y_{i1}^*, \ldots, y_{ip}^*)$, which exploits the information that the expectation of the noise is a zero vector. We include these two closed-form estimators in our benchmarks in addition to K2- and DR-ABC.

Scalar case. We first examine the case where $n = 1$ and θ^* is a scalar. That is, there is only one observation y^* that is a scalar. From the expression of $\widehat{\theta}_j$, we have $\widehat{\theta}_{\text{opt}} = y^*/2$. We tested the performance of P-ABC, when the neural networks representing f and u in (4) were trained on 1000 samples. Each neural network contained two fully connected layers of size 8 with exponential linear unit (ELU) activation functions, and the final output layer for f was activated by the hyperbolic tangent. We chose ξ to be a one-dimensional uniform distribution on $[-1, 1]$ and used a learning rate of 10^{-4}. In 2×10^5 iterations, P-ABC achieved 0.0413 MSE on the training set and 0.0416 MSE on the test set. The optimal MSE corresponding to $\widehat{\theta}_{\text{opt}}$ was 0.0411 for the training set. Figure 1 shows the histogram of $f(y, \xi)$ for different values of y in $[-0.5, 0.5]$, using 10^4 trials of ξ. We see that the empirical probability distribution concentrates tightly around $y^*/2$, demonstrating that the output of P-ABC was nearly optimal. The training and testing errors are reported in Table 2, and the result shows that the performance of P-ABC was close to the theoretical optimum. By comparison, since there is only one observation available, K2- and DR-ABC do not output meaningful results as the computation of the MMD statistics requires at least two observations to form a non-trivial empirical distribution.

Performance under higher dimensions. We examined the performance of P-ABC when the dimension of θ^* was higher. For illustration purpose, we chose $\dim(\theta) = 16, 128, 256$, and we assumed that the set of observations, Y^*, contained 10 samples for each parameter value. Once again, we used neural networks to represent f and u in P-ABC, for which we trained with 1000 sets of samples.

To reduce the input dimension of the neural networks, for each input set of samples $Y := \{y_1, \ldots, y_n\}$, we set $f(Y, \xi) = \frac{1}{n}\sum_{i=1}^n f(y_i, \xi)$ and $u(\theta, Y) = \frac{1}{n}\sum_{i=1}^n u(\theta, y_i)$. More specifically, rather than taking the entire set of samples as the input, the neural network representing f took



$y^* = -0.25$ $\quad$ $y^* = 0$ $\quad$ $y^* = 0.25$

Figure 1: Histogram of $f(y, \xi)$ computed from 10^4 samples of $y \sim p(y)$ and $\xi \sim p_0(\xi)$. The distribution shows that the estimated θ concentrates closely around $\widehat{\theta}_{\text{opt}} = y^*/2$, suggesting that the P-ABC estimate is near optimal.

MSE	P-ABC [test,train]	K2-ABC	DR-ABC	$\widehat{\theta}_{\text{opt}}$	$\widehat{\theta}_{\text{ave}}$
$\dim(\theta^*) = 1$	[0.009,0.010]	0.011	0.083	0.003	0.008
$\dim(\theta^*) = 16$	[0.182, 0.155]	1.283	1.143	0.050	0.134
$\dim(\theta^*) = 128$	[2.749,1.793]	21.478	10.730	0.409	1.064
$\dim(\theta^*) = 256$	[4.266,1.399]	41.830	21.324	0.818	2.119

Table 2: MSE for estimating θ^* with different dimensions using K2-, DR- and P-ABC. For K2- and DR-ABC, we set $\epsilon = 0.01$ when computing MMD. For P-ABC, the hidden layer sizes are 8,32,128,256 for different values of $\dim(\theta)$, and the dimension of ξ's are 1,4,4,4, respectively.

each sample individually, and used their average as the final value of $f(Y, \xi)$. Under this setting, for 2×10^5 iterations, the obtained results are shown in Table 2. We can see that P-ABC outperformed both K2- and DR-ABC in all four cases, and when the dimension of θ^* was small, the performance of P-ABC was close to that of $\widehat{\theta}_{\text{ave}}$.

4.2 Synthetic Dataset II: Gaussian Mixtures

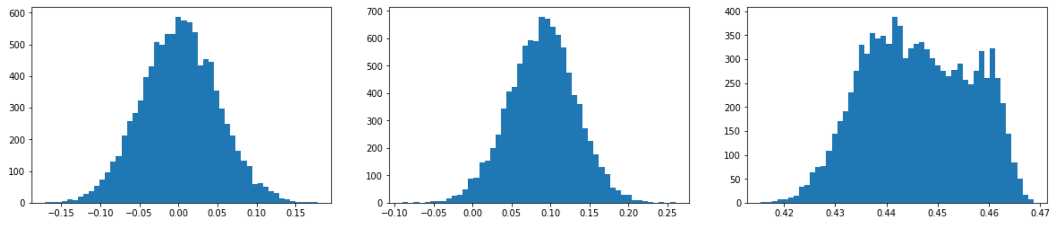
Consider a model where the underlying parameter $\theta^* \in \mathbb{R}$ is uniformly distributed on $[-0.5, 0.5]$, and the observation y^* was sampled from a Gaussian mixture: $y^* = (0.5 + \theta^*)\mathcal{N}(-1, 1) + (0.5 - \theta^*)\mathcal{N}(1, 1)$. In this example, we compared the performances between K2-, DR-, EP-, and the proposed P-ABC. For P-ABC, we adopted the same network structures for the neural networks representing f and g as in the previous example, and trained them with 4000 sets of samples. Each set of samples contained 250 samples corresponding to the same θ^* and the same amount of samples were used for the benchmarks. P-ABC achieved an MSE of 0.004, and EP-ABC achiieved an MSE of 0.06. [4] Note that the implementation of EP-ABC requires Cholesky factorization for each iteration, which is computationally expensive and particularly sensitive to initialization. In fact, the run time of EP-ABC was significantly longer than P-ABC. While P-ABC took less than 5 minutes to average its performance over 1000 sets of test samples, EP-ABC took 10 minutes to average its performance over 100 sets of samples. K2- and DR-ABC, by comparison, were unable to run 100 trials within 1 hour. This experiment demonstrated the efficiency of implementing the P-ABC algorithm.

Although P-ABC has demonstratee superior numerical performances over the benchmarks, we would also like to point out that it suffers from some of the defficiencies of the other existing ABC algorithms. One such defficiency is that the algorithm is prone to mismatched priors. To see this, we plotted the histogram of $f(y, \xi)$ for $\theta^* = 0$ when P-ABC was trained on a mismatched prior. In particular, we applied the transformation $\widetilde{\theta} = (\theta + a)/(2a + 1)$, and used $\widetilde{\theta}$ as the sampled parameter from the prior. This transformation introduced bias between the true prior and the prior used for training, and as can be seen in Figure 2, the range of the estimated parameter by P-ABC shifted away from θ^* as a increased.

4.3 Ecological Dynamic System

Time series observations are an important application scenario for ABC. We compared the performances of K2-, DR- and P-ABC over the example of an ecological dynamic system studied in Park

[4] Per implementation of the code made available online by Barthelmé and Chopin [2011].



$\theta^* = 0, a = 0$ $\theta^* = 0, a = 1$ $\theta^* = 0, a = 10$

Figure 2: Impact of improper prior on P-ABC. Consider finding uniformly distributed $\theta \sim \mathcal{U}[-0.5, 0.5]$ from $y = (0.5+\theta)\mathcal{N}(-1,1) + (0.5-\theta)\mathcal{N}(-1,1)$. Improper priors are obtained by $\widetilde{\theta} = (\theta + a)/(2a+1)$ with $a = 1, 10$. We see that training on improper prior injects bias into the output of P-ABC.

et al. [2016], whose population dynamics follow the relationship

$$y_{t+1} = P y_{t-\tau} \exp\left(-\frac{y_{t-\tau}}{y_0}\right) e_t + y_t \exp(-\delta \epsilon_t).$$

Let $Y = (y_1, \ldots, y_t)$ denote the set of samples that contains the population size data up to time t. The noise $e_t \sim \Gamma(\sigma_p^{-2}, \sigma_p^2)$, $\epsilon_t \sim \Gamma(\sigma_d^{-2}, \sigma_d^2)$, while $\theta = (P, y_0, \sigma_d^2, \sigma_p^2, \tau, \delta)$. Similar to Park et al. [2016], we sample each dimension of $\log\theta$ from a uniform distribution on $[-5, 2]$, and set $\tau = \lceil\tau\rceil$.

For P-ABC, we implemented a recurrent neural network (RNN) with LSTM cells to capture the dynamics of the underlying time series. The output of the LSTM cell is then plugged into a fully connected layer along with θ or ξ. The structures of the neural networks representing f and u are shown in Figure 4 in Appendix C. When training, we set the size of each sample set Y to 30, and we used 1000 sets of samples to train the algorithms. For P-ABC, we set $\dim(\xi) = 4$, the size of the LSTM cell to 32 and the size of the fully connected layer to 16. For K2- and DR-ABC, the samples within each set Y were regarded as i.i.d.. The obtained result is shown in Figure 3, with the verticle axis denoting the MSE of the estimated parameter. P-ABC outperformed K2-ABC and DR-ABC on all aspects: the MSE was 12.9 for P-ABC, 24.7 for K2-ABC, and 16.4 for DR-ABC. In addition, P-ABC had the lowest average, quartile, and better performance on outliers.

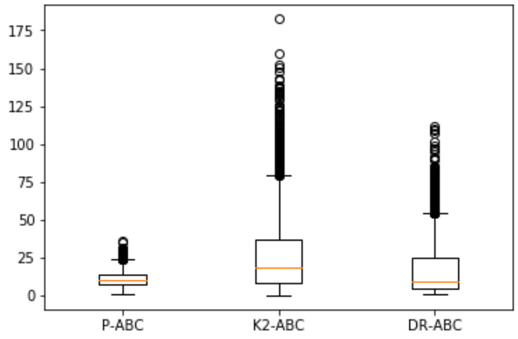


Figure 3: Statistics of MSEs for P-, K2- and DR-ABC trained on 1000 sequences of length 30.

5 Conclusion

In this paper, we presented a unifying optimization framework for ABC, named Predictive-ABC, under which we showed that ABC can be formulated as a saddle point problem for different objective functions. We presented a high-probability error bound that decays at the speed of $\mathcal{O}(N^{-1/2}\log N)$ with N being the number of samples and we presented a stochastic-gradient-descent-based algorithm, P-ABC, to find the solution. In practice, P-ABC significantly outperforms K2- and DR-ABC, representatives for the state-of-the-art sampling- and regression-based algorithms, respectively.

Acknowledgement

This work was supported in part by MURI grant ARMY W911NF-15-1-0479, ONR grant W911NF-15-1-0479, NSF CCF-1755829 and NSF CMMI-1761699.

References

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71 (1):89–129, 2008.

Simon Barthelmé and Nicolas Chopin. ABC-EP: Expectation propagation for likelihoodfree Bayesian computation. In *ICML*, pages 289–296, 2011.

Peter L Bartlett, Nick Harvey, Chris Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *arXiv preprint arXiv:1703.02930*, 2017.

Espen Bernton, Pierre E Jacob, Mathieu Gerber, and Christian P Robert. Inference in generative models using the Wasserstein distance. *arXiv preprint arXiv:1701.05146*, 2017.

Michael GB Blum and Olivier François. Non-linear regression models for approximate Bayesian computation. *Statistics and Computing*, 20(1):63–73, 2010.

Michael GB Blum, Maria Antonieta Nunes, Dennis Prangle, Scott A Sisson, et al. A comparative review of dimension reduction methods in approximate Bayesian computation. *Statistical Science*, 28(2):189–208, 2013.

Katalin Csilléry, Olivier François, and Michael G B Blum. Abc: An R package for approximate Bayesian computation (ABC). *Methods in Ecology and Evolution*, 3(3):475–479, 2012. ISSN 2041210X. doi: 10.1111/j.2041-210X.2011.00179.x.

Bo Dai, Niao He, Yunpeng Pan, Byron Boots, and Le Song. Learning from conditional distributions via dual embeddings. In *Artificial Intelligence and Statistics*, pages 1458–1467, 2017.

C. C. Drovandi and A. N. Pettitt. Estimation of parameters for macroparasite population evolution using approximate Bayesian computation. *Biometrics*, 67(1):225–233, 2011. ISSN 0006341X. doi: 10.1111/j.1541-0420.2010.01410.x.

Alexei J Drummond and Andrew Rambaut. Beast: Bayesian evolutionary analysis by sampling trees. *BMC evolutionary biology*, 7(1):214, 2007.

Christoph Leuenberger Daniel Wegmann Laurent Excoffier. Bayesian computation and model selection in population genetics. *Genetics*, page 18, 2009. doi: 10.1534/genetics.109.102509. URL `http://arxiv.org/abs/0901.2231`.

Alexander Gleim and Christian Pigorsch. Approximate Bayesian computation with indirect summary statistics.

Michael U Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Statistical inference of intractable generative models via classification. *arXiv preprint arXiv:1407.4981*, 2014.

Michael U Gutmann, Jukka Corander, et al. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 2016.

Michael U Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Likelihood-free inference via classification. *Statistics and Computing*, 28(2):411–425, 2018.

David Haussler. Sphere packing numbers for subsets of the boolean n-cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 69(2):217–232, 1995.

John P Huelsenbeck, Fredrik Ronquist, Rasmus Nielsen, and Jonathan P Bollback. Bayesian inference of phylogeny and its impact on evolutionary biology. *science*, 294(5550):2310–2314, 2001.

Paul Joyce and Paul Marjoram. Approximately sufficient statistics and Bayesian computation. *Statistical applications in genetics and molecular biology*, 7(1), 2008.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Naveen Kodali, James Hays, Jacob Abernethy, and Zsolt Kira. On convergence and stability of GANs. 2018.

T Kulkarni, Ilker Yildirim, Pushmeet Kohli, W Freiwald, and Joshua B Tenenbaum. Deep generative vision as approximate Bayesian computation. In *NIPS 2014 ABC Workshop*, 2014.

Jingjing Li, David J Nott, Yanan Fan, and Scott A Sisson. Extending approximate Bayesian computation methods to high dimensions via a Gaussian copula model. *Computational Statistics & Data Analysis*, 106:77–89, 2017.

Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with ReLU activation. In *Advances in Neural Information Processing Systems*, pages 597–607, 2017.

Gael M. Martin, Brendan P. M. McCabe, Worapree Maneesoonthorn, and Christian P. Robert. Approximate Bayesian computation in state space models. *arXiv:1409.8363*, pages 1–38, 2014. URL `http://arxiv.org/abs/1409.8363`.

Ron Meir. Nonparametric time series prediction through adaptive model selection. *Machine learning*, 39(1):5–34, 2000.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*, 2017.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Jovana Mitrovic, Dino Sejdinovic, and Yee Whye Teh. DR-ABC: Approximate Bayesian computation with kernel-based distribution regression. 2016.

Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

David J Nott, Y Fan, L Marshall, and SA Sisson. Approximate Bayesian computation and Bayes linear analysis: toward high-dimensional ABC. *Journal of Computational and Graphical Statistics*, 23(1):65–86, 2014.

Matthew A Nunes and David J Balding. On optimal selection of summary statistics for approximate Bayesian computation. *Statistical applications in genetics and molecular biology*, 9(1), 2010.

Mijung Park, Wittawat Jitkrittum, and Dino Sejdinovic. K2-ABC: Approximate Bayesian computation with kernel embeddings. 2016.

Leah F Price, Christopher C Drovandi, Anthony Lee, and David J Nott. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11, 2018.

Louis Raynal, Jean-Michel Marin, Pierre Pudlo, Mathieu Ribatet, Christian P Robert, and Arnaud Estoup. ABC random forests for Bayesian parameter inference. *arXiv preprint arXiv:1605.05537*, 2016.

GS Rodrigues, Dennis Prangle, and Scott A Sisson. Recalibration: A post-processing method for approximate Bayesian computation. *Computational Statistics & Data Analysis*, 126:53–66, 2018.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Dustin Tran, Rajesh Ranganath, and David M Blei. Hierarchical implicit models and likelihood-free variational inference. *arXiv preprint arXiv:1702.08896*, 2017.

Daniel Wegmann, Christoph Leuenberger, and Laurent Excoffier. Efficient approximate Bayesian computation coupled with Markov chain Monte Carlo without likelihood. *Genetics*, 182(4): 1207–1218, 2009.

Simon N Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466 (7310):1102, 2010.

Arnold Zellner. Optimal information processing and Bayes's theorem. *The American Statistician*, 42 (4):278–280, 1988.