Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

HYPERPARAMETER OPTIMIZATION FOR APPROXIMATE BAYESIAN COMPUTATION

Prashant Singh
Andreas Hellander

Division of Scientific Computing
Department of Information Technology
Uppsala University
Box 337, SE-751 05 Uppsala, SWEDEN

ABSTRACT

Approximate Bayesian computation is a popular methodology for simulation-based parameter inference in scenarios where the likelihood function is either analytically intractable or computationally infeasible. The likelihood of simulator parameters fitting given data is approximated by iteratively simulating samples that are generated according to a specified prior distribution. The convergence speed and the quality of inference are highly sensitive to the choice of hyperparameters such as the chosen summary statistic, the value of the acceptance threshold and the distance function. The choice is typically left to the domain expert as summary statistics vary across disciplines and threshold values are problem-specific. This work explores automated hyperparameter optimization for approximate Bayesian computation using Bayesian optimization as an alternative to time consuming manual selection. The problem setting assumes availability of a fast low-fidelity simulator, which is used during the optimization process. The optimized hyperparameters are then used to perform inference using the high-fidelity simulator.

1 INTRODUCTION

Simulation models enable rich analysis and understanding of complex phenomena across disciplines. Simulation-driven analysis often involves fitting the model to observed experimental data in order to ensure model accuracy. This gives rise to the parameter inference problem wherein simulation model parameters are to be inferred using an observed data set. A straight-forward approach to accomplish parameter inference is to derive the likelihood function describing the probability of the observed data, given the parameters (Severini 2000). However, deriving an analytical form of the likelihood function for complex simulators is not always possible. Even when the likelihood function can be formulated, its computational cost may be prohibitive, such as in the case of models of stochastic gene regulation encountered in systems biology. Therefore, likelihood-free parameter inference approaches have gained mass popularity (Bazin et al. 2010; Gutmann and Corander 2016; Gutmann et al. 2018). Approximate Bayesian computation (ABC) (Sunnåker et al. 2013; Lintusaari et al. 2017) has emerged as the most popular likelihood-free approach and has been shown to perform reliable inference in a variety of applications and domains (Chan et al. 2014; Ugarte et al. 2014; Christopher et al. 2017).

The problem setting in a typical parameter inference scenario includes a simulation model f with control parameters $\boldsymbol{\theta}$ and an observed data set $\mathbf{y}_{obs}$. The model $f\colon (\boldsymbol{\theta}, V) \rightarrow \mathbf{y}$ maps $\boldsymbol{\theta}$ and V to responses $\mathbf{y}$. The random variables V capture the stochasticity within the simulator and are responsible for random fluctuation in responses $\mathbf{y}$ over multiple simulation runs for fixed $\boldsymbol{\theta}$. This allows definition of a random variable $\mathbf{y}_\theta$ corresponding to the simulation model f. The task is to infer the values of $\boldsymbol{\theta}$ that result in simulated responses being equal to the observed data set, $\mathbf{y}_\theta = \mathbf{y}_{obs}$.

Parameter inference within the ABC framework proceeds as follows. Let $p(\boldsymbol{\theta})$ be the prior distribution on parameters $\boldsymbol{\theta}$. The posterior $p(\boldsymbol{\theta} \mid \mathbf{y}_{obs})$ can be estimated by repeatedly performing simulations $f(\boldsymbol{\theta}')$,

where $\boldsymbol{\theta}' \sim p(\boldsymbol{\theta})$, and accepting parameter values that result in $\mathbf{y}_\theta = \mathbf{y}_{obs}$. In practice, the probability $P(\mathbf{y}_\theta = \mathbf{y}_{obs})$ is vanishingly small, therefore the equality condition is replaced by the tolerance condition $d(\mathbf{y}_\theta, \mathbf{y}_{obs}) \leq \tau$, where d is a measure of discrepancy, such as a distance function operating over summary statistics, and τ is the tolerance bound. The user typically specifies the desired number of accepted samples N, and the inferred parameters are typically reported as the mean of the parameter values corresponding to the accepted samples.

The ABC framework described above consists of several variables or hyperparameters, namely the tolerance bound τ, the desired number of accepted samples N, the choice of the prior $p(\boldsymbol{\theta})$ and the distance function d (along with associated summary statistics). Each of these hyperparameters influences the quality and accuracy of inferred parameters. Selection of optimal hyperparameters is a challenging problem-specific task and is crucial for effective inference. The selection is typically performed by a domain expert, making use of problem-specific information and past experience with similar problems.

With increasing complexity of inference problems, manual hyperparameter optimization invariably becomes impractical and sub-optimal choices are often made. Computationally expensive simulators also prohibit the use of optimization algorithms to arrive at optimal hyperparameters. However, certain applications and simulators allow the possibility of varying the fidelity of the simulations, thereby enabling access to low-fidelity but relatively computationally cheap simulations. Such low-fidelity simulations can be used in an optimization framework to achieve optimal ABC configuration.

This paper explores an automated approach towards optimization of ABC hyperparameters using Bayesian optimization for problems with access to computationally cheap simulations. The paper is organized as follows. Section 2 discusses ABC hyperparameters and their effect on parameter inference quality. Section 3 explains the Bayesian hyperparameter optimization approach, which is empirically evaluated in Section 4. Section 5 presents a brief discussion of the significance of the proposed approach. Finally, Section 6 concludes the paper.

2 APPROXIMATE BAYESIAN COMPUTATION HYPERPARAMETERS

The popular ABC rejection sampling algorithm (Pritchard et al. 1999) consists of the prior function $p(\boldsymbol{\theta})$, the acceptance tolerance bound τ, and the combination of the distance function and summary statistic(s) as hyperparameters. The more recent ABC-Sequential Monte Carlo (ABC-SMC) and ABC-Particle Monte Carlo (ABC-PMC) (Beaumont et al. 2009) add the choice of multilevel tolerance bound transition as a hyperparameter. An effective ABC parameter inference run achieves accurate characterization of parameters $\boldsymbol{\theta}$ using as few evaluations of the simulator f as possible. The following text discusses the effect of each hyperparameter towards effective ABC inference.

2.1 The Prior Function

The prior $p(\boldsymbol{\theta})$ represents prior knowledge regarding the distribution of the parameters $\boldsymbol{\theta}$ to be estimated. In absence of any information, a uniform distribution is often assumed for generation of candidate $\boldsymbol{\theta}'s$ in each rejection sampling iteration. The prior function should ideally generate candidate parameter combinations $\boldsymbol{\theta}'$ that result in accepted samples with as small distances $d(\mathbf{y}_{\theta'}, \mathbf{y}_{obs})$ as possible. Typically the practitioner has little to no information about the form of the prior function, and is limited to assuming a uniform distribution.

2.2 The Tolerance Bound

The acceptance tolerance bound τ is crucial towards achieving accurate ABC inference. Smaller τ values imply a tighter acceptance threshold and hence, higher confidence and lower variance between accepted samples. However, too low a value of τ may lead to an exceedingly large number of ABC sampling iterations required to achieve N accepted samples, leading to impractically lengthy inference times. Therefore, a good value of τ must balance low variance of accepted samples while maintaining a practical acceptance ratio.

A reliable and effective approach followed by ABC-SMC and ABC-PMC towards achieving this balance has been to start with a relative loose τ and reduce its value in subsequent stages (Beaumont et al. 2009). This is achieved by explicitly specifying the stage-wise values of τ, or specifying a transition function that dictates the rules for reduction of τ in each stage. ABC-SMC and ABC-PMC have proven to be robust and more efficient as compared to rejection sampling (Lintusaari et al. 2017). Nevertheless, the practitioner must specify a initial value τ_0 and the transition function, or subsequent values of τ. Selection of a good value of τ has always been challenging as it depends upon the complexity of the problem at hand, and varies across applications and disciplines.

2.3 Summary Statistics

Summary statistics are metrics that capture specific behavior(s) in the simulated response $\mathbf{y}_\theta$. The acceptance or rejection in ABC sampling iterations is based on distance values computed over summary statistics of the simulated responses and the data set. It is imperative that the summary statistic has the ability to capture minute variations and meaningful patterns within the responses and data samples. Using a large number of summary statistics in rejection sampling is also not straight-forward. As the data dimensionality (number of summary statistics) increases, the rate of inference error decay worsens (Prangle 2015). Therefore, careful qualitative and quantitative selection of summary statistics is important. There have been efforts towards enabling automated summary statistic selection (Prangle et al. 2014; Prangle 2015) with some success. However, existing methods either require a certain amount of pre-processing simulations, or lack interpretability, or require certain domain knowledge for specification of auxiliary likelihoods. For a detailed explanation and comparison, the reader is referred to (Prangle 2015).

2.4 The Distance Function

The distance function must be able to relay (dis)similarity between two samples to the sampling process effectively. The distance function must be sensitive enough to convey minute differences, while not being over-sensitive towards noise. Popular distance functions include Euclidean, Manhattan and Chebyshev distance measures, among others. The applicability of each measure varies with problem dimensionality and complexity.

2.5 Tolerance Bound Transition

In case of multi-level ABC approaches such as ABC-SMC and ABC-PMC, the transition of the tolerance bound τ must be specified. The specification is performed either explicitly by listing the tolerance for each stage or level, or by specifying a transition function that dictates the change in τ between successive stages. The transition function must lower the tolerance sufficiently to balance inference speed, acceptance ratio and inference quality.

Each of the hyperparameters described above contributes significantly towards effective inference, and must be chosen carefully. Typically, domain experts set the hyperparameters based on past experience, problem specific knowledge and considerations on computational expense. There have been efforts in the recent past towards automated selection of individual hyperparameters such as the tolerance bound τ and the summary statistics. The ABC-SMC and ABC-PMC approaches simplify the problem of selecting the tolerance bound τ by varying τ in stages. Summary statistic selection methods (Prangle 2015) enable the practitioner to conduct preliminary exploration to identify well-performing summary statistics. Semi-automatic ABC variants have been proposed that make use of intelligent summary statistic selection and distance computation to alleviate some of the hyperparameter optimization burden on the practitioner (Thornton 2009; Fearnhead and Prangle 2012).

This work considers a more holistic, unified view of selecting the optimal *combination* of hyperparameters instead of optimizing each hyperparameter in isolation. The approach presented herein allows the practitioner to obtain an optimal combination of parameters for satisfying a specified goal (e.g., maximize inference

quality, maximize inference speed while maintaining set accuracy, etc.). To the best of authors' knowledge, this is the first work in literature that attempts to automatically optimize hyperparameters of ABC within a unified framework. This is achieved by using Bayesian optimization to infer the optimal combination of hyperparameters of the ABC rejection sampler. The following text briefly discusses Bayesian optimization and formally defines the hyperparameter optimization problem.

3 BAYESIAN HYPERPARAMETER OPTIMIZATION

Bayesian optimization is a model driven optimization method that involves iterative learning of the relationship between the hyperparameters and the objective function (Jones 2001). Model training begins with the construction of a probabilistic model using an *initial design* of training points. As model uncertainty is high to begin with, it is important to explore the hyperparameter space in a space-filling manner. Statistical designs including Latin hypercube sampling and factorial designs (Singh 2016) are popular choices as initial designs.

The probabilistic model trained over the initial design is used in conjunction with an iterative sampling algorithm. The sampling algorithm typically uses model predictions and uncertainty estimates to select additional training points with the goal of exploiting regions in the hyperparameter space likely to lead to the optima. This is typically achieved by generating a large number of Monte-Carlo candidate sample points, and estimating the value of a sampling criterion (see Sec. 3.2) for each candidate sample and subsequently selecting the candidate with the best estimated criterion value. Alternatively, the sampling algorithm may also use certain heuristics to generate new points (e.g., distance based approaches that generate new samples in unexplored spaces based on calculated distances between existing samples). The model is refined and updated using the selected samples, and the cycle of sampling and model refinement ensues until certain stopping criteria such as computational budget, acceptable optima, etc. are met. The probabilistic model types used in the context of Bayesian optimization include Gaussian process models and the related family of Kriging models (Kleijnen 2009). This work uses Gaussian processes within the Bayesian optimization framework. A detailed explanation of Gaussian processes can be found in (Rasmussen and Williams 2006).

3.1 Gaussian Processes

A Gaussian process (GP) generalizes the multivariate Gaussian distribution to an infinite-dimensional stochastic process with any finite combination of variables being jointly-Gaussian (Rasmussen and Williams 2006). A Gaussian process $g(\mathbf{x}) \sim \mathscr{G}\mathscr{P}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ is completely described by its mean m and covariance k functions. The mean function is used to encode prior domain-specific knowledge. In absence of any prior information (as is often the case), $m(\mathbf{x}) = 0$ is a popular choice without loss of generality. The covariance function k describes the variation of the process from the mean function and controls the expressive capability of the model. Popular covariance functions include the squared exponential (SE) kernel (Snoek et al. 2012) and the Matérn family of functions (Rasmussen and Williams 2006). This work uses the SE kernel for all described experiments as it is a good general purpose choice (Shahriari et al. 2016). The SE kernel is defined as $k(\mathbf{x}_i, \mathbf{x}_j) = exp\Big(-\frac{1}{2\theta^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\Big)$, where θ is a hyperparameter controlling the width of the kernel.

Let $\mathscr{D}_n = (X, \mathbf{y})$ be a training set consisting of n points and $\mathbf{K}$ be the kernel matrix encoding the pairwise covariances between data points in X,

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \dots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}.$$

Let $y_{n+1} = \hat{y}(\mathbf{x}_{n+1})$ be the value predicted by the GP model $\hat{y}$ for a new point $\mathbf{x}_{n+1}$. As $\mathbf{y}$ and y_{n+1} are jointly Gaussian,

$$\begin{bmatrix} \mathbf{y} \\ y_{n+1} \end{bmatrix} \sim \mathcal{N}\Big(0, \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^\intercal & k(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}) \end{bmatrix}\Big),$$

where $\mathbf{k} = [k(\mathbf{x}_{n+1}, \mathbf{x}_1,), \ldots k(\mathbf{x}_{n+1}, \mathbf{x}_n,)]$. The posterior distribution is computed using the Sherman-Morrison-Woodbury formula (Rasmussen and Williams 2006) as $P(y_{n+1}|\mathscr{T}, \mathbf{x}_{n+1}) = \mathcal{N}(\mu_n(\mathbf{x}_{n+1}), \sigma_n^2(\mathbf{x}_{n+1}))$, where,

$$\mu_n(\mathbf{x}_{n+1}) = \mathbf{k}^\intercal \mathbf{K}^{-1} \mathbf{y},$$
$$\sigma_n^2(\mathbf{x}_{n+1}) = k(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}) - \mathbf{k}^\intercal \mathbf{K}^{-1} \mathbf{k}.$$

The mean and variance of prediction provided by the GP model is utilized by the sampling algorithm in the search for the optima.

3.2 Sampling Algorithms

In order to drive the search for optima effectively, the sampling algorithm (also known as an acquisition function) must make appropriate trade-offs between space-filling exploration of the hyperparameter space, and exploitation of regions likely to contain the optima. The sampling scheme makes use of information provided by the probabilistic model in determining the trade off. The variance of prediction estimated by the GP model will be higher in regions far away from existing points. Such regions are good candidates for exploration. The regions containing smaller predicted values may potentially contain the optima (for a minimization problem); Such regions are good candidates for exploitation.

Given a large number of Monte-Carlo candidate points in the hyperparameter space, the sampling algorithm can rank each candidate based on specified exploration and exploitation metrics. The top ranked candidates can then be selected to be evaluated to obtain objective function values, and supplemented to the training set. The model is then re-trained with the updated training set.

Popular choices for sampling algorithms include the expected improvement (EI) criterion (Jones 2001), the probability of improvement (PI) criterion (Snoek et al. 2012), and the upper/lower confidence bounds (UCB/LCB) (Shahriari et al. 2016). The probability of improvement criterion is biased towards exploitation and is ideal for scenarios where exploration is primarily performed via the initial design. In such cases one often wants to solve the optimization problem using very few evaluations of the objective function. The expected improvement criterion and the upper and lower confidence bounds are more liberal towards exploration (Shahriari et al. 2016), and maintain a healthy balance with exploitation. This work uses the expected improvement criterion as it is a good general purpose choice.

The expected improvement criterion measures the amount of improvement that a candidate point $\mathbf{x}_{n+1}$ brings over the current known optima t. This is made possible by defining an improvement function as,

$$I(\mathbf{x}_{n+1}, \nu, \theta) = (\nu - t)\mathbb{I}(\nu > t),$$

where ν is obtained using the GP model as $\nu = g(\mathbf{x}_{n+1})$ and θ describes the GP model hyperparameters. Computing the expectation over the improvement function leads to the definition of the EI criterion as follows. As the random variable ν is normally distributed,

$$\begin{aligned} \alpha_{EI}(\mathbf{x}_{n+1}, \mathscr{D}_n) &= \mathbb{E}(I(\mathbf{x}_{n+1}, \nu, \theta)), \\ &= (\mu_n(\mathbf{x}_{n+1}) - t)\Phi\Big(\frac{\mu_n(\mathbf{x}_{n+1}) - t)}{\sigma_n(\mathbf{x}_{n+1})}\Big) + \sigma_n(\mathbf{x}_{n+1})\phi\Big(\frac{\mu_n(\mathbf{x}_{n+1}) - t)}{\sigma_n(\mathbf{x}_{n+1})}\Big), \end{aligned}$$

where ϕ is the standard normal probability density function and Φ is the standard normal cumulative distribution function. The current known optima t is adaptively set in each iteration as $t = max_{i=1:n}(y_i)$. The prediction mean $\mu_n(\mathbf{x}_{n+1})$ and variance $\sigma^2(\mathbf{x}_{n+1})$ for the candidate point $\mathbf{x}_{n+1}$ is provided by the GP model as described in Section 3.1.

3.3 Problem Formulation

Consider a parameter inference problem of estimating parameters $\boldsymbol{\theta}$, to be solved using an ABC parameter inference algorithm. Let $S = \{s_1, ..., s_k\}$ be a pool of k candidate summary statistics, $D = \{d_1, ..., d_l\}$ be a pool l distance functions, $P = \{p_1(\boldsymbol{\theta}), ..., p_r(\boldsymbol{\theta})\}$ be a pool of r candidate prior functions and τ be the tolerance bound, $\tau_{min} \leq \tau \leq \tau_{max}$ within bound constraints τ_{min} and τ_{max}. The ABC inference problem is then parameterized by $\mathscr{P} = (p(\boldsymbol{\theta}), \tau, s, d)$, where $p(\boldsymbol{\theta}) \in P, s \in S, d \in D$. The task is to identify optimal values of hyperparameters in $\mathscr{P}$.

The choice of the objective function $h(\mathscr{P})$ is crucial towards accurate estimation of the hyperparameters. Let f be a computationally inexpensive simulation model to be used as part of the ABC hyperparameter optimization process. The model f may be a model based on a coarse(r) mesh in the context of mesh-based simulators, or a model involving a relatively small number of trajectories in the context of stochastic biochemical reaction networks (Abel et al. 2016), or even a fast surrogate model of a simulator (Singh and Hellander 2017).

The structure of the objective functions considered in this work consist of an atomic instance of an ABC run for performing parameter inference on the simulator f. The ABC run is parameterized by $\mathscr{P}$ as discussed above. The observables from such a run include n_{total} - the total number of evaluations of the simulator f performed to achieve n_{acc} accepted samples, the vector of values of the distance function $\mathbf{d}$ for the accepted samples, and posterior samples corresponding to accepted distances. This work evaluates n_{total} and $\mathbf{d}$ as choices for the objective function $h(\mathscr{P})$. The suitability and motivation behind the choice of the two quantities is discussed below.

3.3.1 The Trial Count

The trial count n_{total} and the desired number of accepted samples n_{acc} defines the acceptance ratio $\frac{n_{acc}}{n_{total}}$. Setting too tight a tolerance bound τ or selecting summary statistics that result in unexpectedly high distances may lead to impractically high number of trials required to attain the specified number of accepted samples. The practitioner may want to maintain the acceptance ratio within a desired range keeping computational considerations in mind. In such cases, the objective function $h(\mathscr{P})$ can be formulated in terms of the trial count as $(-n_{total})$. Minimizing this objective will prefer ABC runs that achieve the specified n_{acc} quickly.

3.3.2 The Accepted Distances

The distance values of the accepted samples point towards how well the ABC run performed, i.e., the inference quality. Smaller distances are desirable and imply that the rejection sampler was able to identify the 'good' regions where estimated parameters are likely very close to true parameters $\boldsymbol{\theta}$. Let $d_\mu = mean(\mathbf{d})$ be the mean distance of accepted samples in a particular ABC instance. Minimizing d_μ would imply preferring ABC configurations that achieve smaller distances. Going further, let $\mathbf{d}'$ be a vector containing the final $\left\lfloor \frac{length(\mathbf{d})}{2} \right\rceil$ elements from $\mathbf{d}$. Let $d'_\mu = mean(\mathbf{d}')$ be the mean distance achieved by the final half of accepted samples during rejection sampling. The quantity d'_μ can be used as a heuristic to gauge how well the rejection sampler was able to understand the true distribution of parameters $\boldsymbol{\theta}$. Therefore, the quantities d_μ and d'_μ can be used as objective functions to influence the inference quality of an ABC run. Additionally, smaller values or ranges corresponding to the tolerance bound can be chosen to enhance the quality of inference.

3.4 The Optimization Problem

Having discussed the candidate objective functions, it is now possible to formally define the optimization problem. With $\mathscr{P} = (p(\boldsymbol{\theta}), \tau, s, d)$, the optimization problem can be defined as,

$$\begin{aligned} &\text{min. } h(\mathscr{P}), \\ &\text{s. t. } p(\boldsymbol{\theta}) \in \{p_1(\boldsymbol{\theta}), ..., p_r(\boldsymbol{\theta})\}, \\ &\quad \tau_{min} \leq \tau \leq \tau_{max}, \\ &\quad s \in \{s_1, ..., s_k\}, \\ &\quad d \in \{d_1, ..., d_l\}. \end{aligned}$$

The following section evaluates the proposed approach on the problem of parameter inference of a genetic oscillator, modeled as a stochastic biochemical reaction network.

4 EXPERIMENTS

A complex biochemical reaction network with oscillatory behavior due to (Vilar et al. 2002) is considered for the purpose of experiments. The network involves 9 species undergoing 18 reactions parameterized by 15 reaction constants or variables. Let $\mathscr{S} = \{D_A, D_A^*, M_A, D_R, D_R^*, M_R, C, A, R\}$ be the set of species with initial copy numbers $\{1,0,0,1,0,0,10,10,10\}$ respectively. The model involves the following reactions,

$$\begin{aligned} sD_A &: D_A^* \xrightarrow{\theta_A} D_A, & sA_3 &: D_R^* \xrightarrow{\theta_A} D_R^*, A, \\ sD_A^* &: D_A, A \xrightarrow{\gamma_A} D_A^*, & aA &: A \xrightarrow{\delta_A} \phi, \\ sD_R &: D_R^* \xrightarrow{\theta_R} D_R, & sC &: A, R \xrightarrow{\gamma_C} C, \\ sD_R^* &: D_R, A \xrightarrow{\gamma_R} D_R^*, & sMR_1 &: D_R^* \xrightarrow{\alpha_{R^*}} D_R^*, M_R, \\ sMA_1 &: D_A^* \xrightarrow{\alpha_A^*} D_A^*, M_A, & sMR_2 &: D_R \xrightarrow{\alpha_R} D_R, M_R, \\ sMA_2 &: D_A \xrightarrow{\alpha_A} D_A, M_A, & aMR &: M_R \xrightarrow{\delta_{MR}} \phi, \\ aM_A &: M_A \xrightarrow{\delta_{MA}} \phi, & sR_1 &: M_R \xrightarrow{\beta_R} M_R, R, \\ sA_1 &: M_A \xrightarrow{\beta_A} A, M_A, & aR &: R \xrightarrow{\delta_R} \phi, \\ sA_2 &: D_A^* \xrightarrow{\theta_A} D_A^*, A, & sR_2 &: C \xrightarrow{\delta_A} R. \end{aligned}$$

Parameter inference is to performed within the following search space,

$$\begin{aligned} \alpha_A &\in [30,70], & \delta_A &\in [0,2], \\ \alpha_A^* &\in [200,600], & \delta_R &\in [0,0.5], \\ \alpha_R &\in [0,1], & \gamma_A &\in [0.5,1.5], \\ \alpha_R^* &\in [30,70], & \gamma_R &\in [0.5,1.5], \\ \beta_A &\in [30,70], & \gamma_C &\in [1,3], \\ \beta_R &\in [1,10], & \theta_a &\in [30,70], \\ \delta_{MA} &\in [1,12], & \theta_r &\in [80,120]. \\ \delta_{MR} &\in [0,1], \end{aligned} \tag{1}$$

4.1 Experimental Settings

The simulation model is implemented in StochSS (Drawert et al. 2016) and evaluated in Python using GillesPy (Abel et al. 2016). The low-fidelity simulator used to find optimal ABC hyperparameters consists of 10 trajectories of Specie A spanning 20 time steps. The high-fidelity simulation model used for validation consists of 1000 trajectories of Specie A spanning 20 time steps. The true parameters were fixed as $\{50, 500, 0.01, 50, 50, 5, 10, 0.5, 1, 0.2, 1, 1, 2, 50, 100\}$ to generate the observed data. The observed data set used during the optimization process consists of 10 and 1000 samples spanning 20 time steps.

The ABC hyperparameters optimized using the proposed approach include the tolerance bound, the distance function and the summary statistic. The respective choices for each of the hyperparameters are as follows,

$$\tau \in [0.05, 1],$$
$$S = \{\text{Burstiness, Standard Deviation, Absolute Energy}\},$$
$$D = \{\text{Euclidean, Manhattan, Chebyshev}\}.$$

The burstiness summary statistic aims at capturing short but intense spans of activity (Goh and Barabási 2008). The absolute energy summary statistic is simply the sum over squared values of the time series in question. The distance values are normalized to the range $[0, 1]$. Thus, 0.05 indicates accepting top 5% estimated values to be accepted. The set of candidate summary statistics and distance functions are discrete variables, while the tolerance bound τ is a continuous variable. For the purpose of comparison, an ABC configuration is selected manually that the authors had been investigating for the considered test problem. The manually selected hyperparameters include the burstiness summary statistic (Goh and Barabási 2008), the Euclidean distance function and setting $\tau = 0.05$ (top 5% distance values accepted) to achieve a high level of confidence in inferred parameters. The Python Bayesian optimization library GPyOpt (GPyOpt Authors, The. 2016) is used for the experiments in this paper. The initial design chosen is a Latin hypercube of 10 points, and the total budget assigned to Bayesian optimization is 25 evaluations of the objective function. This corresponds to a total of 25 separate ABC runs using the computationally cheap simulator.

4.2 Results

Table 1 lists the mean squared error (MSE) between the inferred parameters and true parameters, and execution times are reported in Table 2. It can be seen that the consistently best performing formulation of the objective function is $\mathbf{d}'_\mu$, which corresponds to minimizing the mean distances of accepted samples in the final half of ABC rejection sampling. It is interesting to note that the ABC configuration found using Bayesian optimization of $\mathbf{d}'_\mu$ is also relatively fast in terms of execution time using the full-fidelity model. Interestingly, it is also observed that minimizing the trial count also achieves reasonably accurate parameter inference. However, the execution times of ABC runs with configurations minimizing the trial count are comparatively longer as opposed to intuition. Distance-based objective functions achieve faster execution times as compared to minimization of the trial count. This is due to the relatively limited objective function evaluation budget of 25 evaluations assigned to Bayesian optimization. The 25 hyperparameter combinations explored by the Bayesian optimization framework simply did not cover the hyperparameter space sufficiently. A separate validation run with 80 allowed evaluations confirmed that with greater computational budget, optimization of the objective function $-n_{total}$ resulted in higher τ values being selected. However, values in Table 2 serve as caveats to be taken note of while using the proposed approach.

The optimal hyperparameters found using Bayesian optimization are listed in Table 3. The running times of the optimization process are of the order of a few minutes, which is a small fraction of the time taken by a representative ABC inference run using the full-fidelity model. It can be seen from Table 3 that the distance-based objective functions find Manhattan distance to be most effective, while the trial count objective function leads to the selection of Euclidean distance. The distance-based objective function

Table 1: Mean Squared Error (MSE) estimates of the inferred parameters as compared to true parameters for various ABC hyperparameter optimization methods. BO indicates Bayesian Optimization.

Method	$n_{acc} = 50$	$n_{acc} = 100$	$n_{acc} = 1000$
BO: $\mathbf{d}_\mu$	793.3489	675.3714	801.2964
BO: $\mathbf{d}'_\mu$	**414.4038**	**565.2935**	**763.8218**
BO: $-n_{total}$	536.0395	663.3679	900.7100
Manual Selection	1279.5372	1138.6242	975.3599

Table 2: Execution time in seconds of various ABC configurations with the full-fidelity simulator found using Bayesian optimization (BO). Best results are indicated in bold.

Method	$n_{acc} = 50$	$n_{acc} = 100$	$n_{acc} = 1000$
BO: $\mathbf{d}_\mu$	2327.6710	**4015.8512**	8056.3441
BO: $\mathbf{d}'_\mu$	**2072.7013**	4465.5739	**7122.8508**
BO: $-n_{total}$	2674.8738	5137.2667	10053.5686
Manual Selection	5856.9458	8213.4848	17516.2279

formulations prefer absolute energy as a summary statistic, while minimization of the trial count leads to selection of burstiness as a summary statistic. The values of τ found using the optimization approach are particularly revealing in the context of inference quality. The values obtained using Bayesian optimization range between ~ 0.45 and ~ 0.88, corresponding to accepting roughly 45% and 88% of lowest distances in the considered summary statistics. Results indicate that even with such high tolerance bounds, one can achieve better inference quality with well-chosen summary statistics as compared to the combination of a very tight tolerance bound (0.05), and a manually chosen sub-optimal summary statistic (burstiness). The practitioner may also take the tolerance bound out of the optimization loop, and concentrate only on searching for the ideal combination of summary statistic(s) and the distance function. Thereafter, an approach like ABC-SMC or ABC-PMC can be followed where τ is adaptively varied.

Table 3: Optimal ABC hyperparameters found using Bayesian optimization (BO) and the time taken for the optimization process.

Method	τ	s	d	time (s)
BO: $\mathbf{d}_\mu$	0.4521	Absolute Energy	Manhattan	158.0984
BO: $\mathbf{d}'_\mu$	0.6091	Absolute Energy	Manhattan	710.4243
BO: $-n_{total}$	0.8810	Burstiness	Euclidean	481.7147
Manual Selection	0.05	Burstiness	Euclidean	-

5 DISCUSSION

The parameter inference quality of results in Table 1 must be evaluated keeping in mind the magnitude of parameter values describing the search space in Eq. (1). The parameters α_A, α_A^* and θ_r have comparatively high magnitudes describing the search space. Thus, the squared error of these three parameters amplifies the MSE values seen in Table 1. For instance, the true and closest inferred parameters are,

$$\boldsymbol{\theta}_{true} = \{50, 500, 0.01, 50, 50, 5, 10, 0.5, 1, 0.2, 1, 1, 2, 50, 100\},$$
$$\boldsymbol{\theta}_{inferred} = \{49.83, 421.38, 0.46, 53.67, 46.82, 5.64, 7.49, 0.47, 1.09, 0.25, 1.08, 1.06, 1.90, 51.99, 99.88\}.$$

It can be seen that the majority of inferred parameters are within 10% of true values except for the second parameter α_A^*, which is off by $\sim 16\%$. However, that corresponds to a magnitude of approximately 80 units, which severely punishes the MSE. This also shows that α_A^* is hard to infer in general. In order to

minimize the inference error, the objective function evaluation budget of the Bayesian optimizer can be increased. Additionally, the tolerance bound τ can be made tighter.

As shown by the results, the inference quality is greatly influenced by the choice of summary statistics. The pool of summary statistics can be curated using approaches mentioned in (Prangle 2015). The curated pool S will be more effective towards achieving high quality parameter inference. The optimization-based approach presented in this work is intended to supplement existing summary selection methods as opposed to replacing them. The experiments presented herein concerned the ABC rejection sampling algorithm (Pritchard et al. 1999), but the proposed approach is equally applicable to ABC-SMC, ABC-PMC and related likelihood-free parameter inference methods. The basic rejection sampling algorithm was chosen in the interest of simplicity and ease of exposition.

The selection of the kernel function for GP regression and the sampling algorithm should also take the nature of the problem into consideration. The ABC hyperparameters considered in this work consisted of a mix of discrete (S, D) and continuous variables (τ), which has practical consequences. The Bayesian optimization pipeline was modified to treat all variables as continuous, and thereafter constrain the search using EI to include only feasible discrete values. Alternatively, sampling algorithms more suited to discrete spaces such as the knowledge gradient (KG) (Frazier et al. 2008) can be used. Similarly, various choices for the GP kernel function such as polynomial kernels can be explored for mixed-space problems.

Future work includes exploring multi-objective formulations that simultaneously optimize multiple metrics, such as mean distances of accepted samples and the trial count. This will allow the practitioner greater flexibility towards searching for optimal ABC hyperparameters. The effect of varying the fidelity of the simulator used during the optimization process will also be studied in future.

6 CONCLUSION

This paper presents a novel approach for optimization of approximate Bayesian computation hyperparameters for parameter inference problems with access to variable-fidelity simulations. The low fidelity computationally cheap simulator is used to formulate objective functions enabling hyperparameter optimization within practical time frames. Three objective function formulations are explored in this work including variants that minimize the mean distance of accepted samples, and the trial count. The proposed approach is demonstrated on the problem of inferring parameters of a complex biochemical reaction network from systems biology. Empirical results validate the efficacy of the proposed approach towards providing insight into well-performing hyperparameter values. The optimal hyperparameters can eventually be used to perform inference using the full-fidelity simulator with improved inference accuracy.

ACKNOWLEDGMENTS

This work was funded by the Göran Gustafsson foundation and the eSSENCE strategic collaboration on eScience.

REFERENCES

Abel, J. H., B. Drawert, A. Hellander, and L. R. Petzold. 2016. "GillesPy: A Python Package for Stochastic Model Building and Simulation". *IEEE Life Sciences Letters* 2(3):35–38.

Bazin, E., K. J. Dawson, and M. A. Beaumont. 2010. "Likelihood-Free Inference of Population Structure and Local Adaptation in a Bayesian Hierarchical Model". *Genetics* 185(2):587–602.

Beaumont, M. A., J.-M. Cornuet, J.-M. Marin, and C. P. Robert. 2009. "Adaptive Approximate Bayesian Computation". *Biometrika* 96(4):983–990.

Chan, Y. L., D. Schanzenbach, and M. J. Hickerson. 2014. "Detecting Concerted Demographic Response Across Community Assemblages using Hierarchical Approximate Bayesian Computation". *Molecular Biology and Evolution* 31(9):2501–2515.

Christopher, J., C. Lapointe, N. Wimer, T. Hayden, I. Grooms, G. B. Rieker, and P. E. Hamlington. 2017. "Parameter Estimation for a Turbulent Buoyant Jet Using Approximate Bayesian Computation". In *55th AIAA Aerospace Sciences Meeting*, 2025–2043.

Drawert, B., A. Hellander, B. Bales, D. Banerjee, G. Bellesia, B. J. Daigle, G. Douglas, M. Gu, A. Gupta, S. Hellander, C. Horuk, D. Nath, A. Takkar, S. Wu, P. Lötstedt, C. Krintz, and L. R. Petzold. 2016, December. "Stochastic Simulation Service: Bridging the Gap between the Computational Expert and the Biologist". *PLoS Computational Biology* 12(12):e1005220.

Fearnhead, P., and D. Prangle. 2012. "Constructing Summary Statistics for Approximate Bayesian Computation: Semi-Automatic Approximate Bayesian Computation". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74(3):419–474.

Frazier, P. I., W. B. Powell, and S. Dayanik. 2008. "A Knowledge-Gradient Policy for Sequential Information Collection". *SIAM Journal on Control and Optimization* 47(5):2410–2439.

Goh, K. I., and A. L. Barabási. 2008. "Burstiness and Memory in Complex Systems". *EPL (Europhysics Letters)* 81(4):48002.

GPyOpt Authors, The. 2016. "GPyOpt: A Bayesian Optimization Framework in Python". http://github.com/SheffieldML/GPyOpt.

Gutmann, M. U., and J. Corander. 2016, January. "Bayesian Optimization for Likelihood-Free Inference of Simulator-Based Statistical Models". *Journal of Machine Learning Research* 17(1):4256–4302.

Gutmann, M. U., R. Dutta, S. Kaski, and J. Corander. 2018. "Likelihood-Free Inference via Classification". *Statistics and Computing* 28(2):411–425.

Jones, D. R. 2001. "A Taxonomy of Global Optimization Methods Based on Response Surfaces". *Journal of Global Optimization* 21(4):345–383.

Kleijnen, J. P. 2009. "Kriging Metamodeling in Simulation: A Review". *European Journal of Operational Research* 192(3):707–716.

Lintusaari, J., M. U. Gutmann, R. Dutta, S. Kaski, and J. Corander. 2017. "Fundamentals and Recent Developments in Approximate Bayesian Computation". *Systematic Biology* 66(1):e66–e82.

Prangle, D. 2015. "Summary Statistics in Approximate Bayesian Computation". *arXiv preprint arXiv:1512.05633*.

Prangle, D., P. Fearnhead, M. P. Cox, P. J. Biggs, and N. P. French. 2014. "Semi-Automatic Selection of Summary Statistics for ABC Model Choice". *Statistical Applications in Genetics and Molecular Biology* 13(1):67–82.

Pritchard, J. K., M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman. 1999. "Population Growth of Human Y Chromosomes: A Study of Y Chromosome Microsatellites.". *Molecular Biology and Evolution* 16(12):1791–1798.

Rasmussen, C. E., and C. K. Williams. 2006. *Gaussian Processes for Machine Learning*, Volume 1. MIT Press Cambridge.

Severini, T. A. 2000. *Likelihood Methods in Statistics*. Oxford University Press.

Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. 2016. "Taking the Human Out of the Loop: A Review of Bayesian Optimization". *Proceedings of the IEEE* 104(1):148–175.

Singh, P. 2016. *Design of Experiments for Model-Based Optimization*. Ph. D. thesis, Ghent University.

Singh, P., and A. Hellander. 2017. "Surrogate Assisted Model Reduction for Stochastic Biochemical Reaction Networks". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan et al., 1773–1783. Piscataway, New Jersey: IEEE.

Snoek, J., H. Larochelle, and R. P. Adams. 2012. "Practical Bayesian Optimization of Machine Learning Algorithms". In *Advances in Neural Information Processing Systems*, 2951–2959.

Sunnåker, M., A. G. Busetto, E. Numminen, J. Corander, M. Foll, and C. Dessimoz. 2013. "Approximate Bayesian Computation". *PLoS Computational Biology* 9(1):e1002803.

Thornton, K. R. 2009. "Automating Approximate Bayesian Computation by Local Linear Regression". *BMC Genetics* 10(1):35–39.

Ugarte, M. D., A. Adin, T. Goicoa, and A. F. Militino. 2014. "On Fitting Spatio-Temporal Disease Mapping Models using Approximate Bayesian Inference". *Statistical Methods in Medical Research* 23(6):507–530.

Vilar, J. M., H. Y. Kueh, N. Barkai, and S. Leibler. 2002. "Mechanisms of Noise-Resistance in Genetic Oscillators". *Proceedings of the National Academy of Sciences* 99(9):5988–5992.

AUTHOR BIOGRAPHIES

PRASHANT SINGH is a post-doctoral researcher in the Division of Scientific Computing, Department of Information Technology at Uppsala University. His research interests include model-based optimization, surrogate modeling, applied machine learning and scientific computing. His e-mail address is prashant.singh@it.uu.se.

ANDREAS HELLANDER is an Associate Professor of Scientific Computing in the Division of Scientific Computing, Department of Information Technology at Uppsala University. His research interests are in the fields of systems biology, distributed computing and stochastic simulations. His e-mail address is andreas.hellander@it.uu.se.