GPS-ABC: Gaussian Process Surrogate Approximate Bayesian Computation

Edward Meeds
Informatics Institute
University of Amsterdam
tmeeds@gmail.com

Max Welling
Informatics Institute
University of Amsterdam
welling.max@gmail.com

Abstract

Scientists often express their understanding of the world through a computationally demanding simulation program. Analyzing the posterior distribution of the parameters given observations (the inverse problem) can be extremely challenging. The Approximate Bayesian Computation (ABC) framework is the standard statistical tool to handle these likelihood free problems, but they require a very large number of simulations. In this work we develop two new ABC sampling algorithms that significantly reduce the number of simulations necessary for posterior inference. Both algorithms use confidence estimates for the accept probability in the Metropolis Hastings step to adaptively choose the number of necessary simulations. Our GPS-ABC algorithm stores the information obtained from every simulation in a Gaussian process which acts as a surrogate function for the simulated statistics. Experiments on a challenging realistic biological problem illustrate the potential of these algorithms.

1 Introduction

The morphogenesis of complex biological systems, the birth of neutrons stars, and weather forecasting are all natural phenomena whose understanding relies deeply upon the interaction between simulations of their underlying processes and their naturally observed data. Hypotheses that posit the generation of observations evolve after critical evaluation of the match between simulation and observation.

This hypothesis–simulation–evaluation cycle is the foundation of *simulation-based modeling*. For all but the most trivial phenomena, this cycle is grossly inefficient. A typical simulator is a complex computer program with a potentially large number of interacting parameters that not only drive the simulation program, but are also often the variables of scientific interest because they play a role in explaining the natural phenomena. Choosing an optimal value setting for these parameters can be immensely expensive.

While a single, optimal parameter setting may be useful to scientists, often they are more interested in the *distribution* of parameter settings that provide accurate simulation results [20, 3, 18]. The interaction between parameters can provide insight regarding not only the properties of the simulation program, but more importantly, the underlying phenomena of interest. The main challenges that we address in this paper are 1) simulation-based modeling in a *likelihood-free* setting (we do not have a model in the typical machine learning sense, and therefore we do not have a standard likelihood function), and 2) running simulations is very expensive.

The first challenge is partly addressed by the *approximate Bayesian computation* (ABC) approach to sampling in likelihood-free scenarios [25, 5]. The ABC approach will be described in a later section, but in short it uses the distance between simulated and observed data as a proxy for the likelihood term in the parameter posterior. ABC provides the necessary framework to make progress in simulation-based modeling, but it is a very inefficient approach, even on simple problems.

The second challenge is approached with a surrogate model in mind. This means that every simulation (parameters and result) is stored and used to maintain a surrogate of the mapping from parameters to result. By carefully constructing an approximate Markov chain Monte Carlo (MCMC) sampler, we are able to sample from the parameter distribution in such a way that our sampling error is controlled and decreases over time. The main advantage of this approach is that by accepting some bias, we are able to very efficiently sample from the approximate posterior because parameters can sometimes be accepted within MCMC with high confidence by relying on the surrogate and thus avoiding expensive simulations.

In this paper we present a procedure for approximate Bayesian inference using a Gaussian process surrogate for expensive simulation-based models. In our approach, simulations that are run over the course of the inference procedure are incorporated into a GP model, gradually improving the surrogate over time. Using MCMC with a Metropolis-Hastings (MH) accept/reject rule, our method uses the surrogate and its uncertainty to make all MH decisions. The key insight is that the uncertainty in the acceptance probability is used to determine if simulations are required to confidently proceed to the MH step. If the uncertainty is high, and we are likely to make an error, then more simulations are run.

2 Approximate Bayesian Computation

The current state-of-the-art for simulation-based model inference is *likelihood-free* or *approximate Bayesian computation* methods [23, 15]. In this section we briefly introduce likelihood-free inference with a focus on MCMC inference procedures as our modeling approach will extend naturally from this work.

In likelihood-free sampling, we do not have a model in the typical sense. Instead, we have access to a simulator that generates pseudo-data that, given an accurate model of the world, look like the observations. The goal is to infer the parameters of the simulator which produce accurate pseudo-data. Importantly, we do not have access to a tractable likelihood function. We now describe in detail the likelihood-free set-up and in particular MCMC sampling techniques.

One of the primary goals of Bayesian inference is to infer the posterior distribution of latent variables $\boldsymbol{\theta}$ using its prior $\pi(\boldsymbol{\theta})$ and its data likelihood $\pi(\mathbf{y}|\boldsymbol{\theta})$:

$$\pi(\boldsymbol{\theta}|\mathbf{y}) = \frac{\pi(\boldsymbol{\theta})\pi(\mathbf{y}|\boldsymbol{\theta})}{\int \pi(\boldsymbol{\theta})\pi(\mathbf{y}|\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{1}$$

where $\boldsymbol{\theta}$ is a vector of latent parameters and $\mathbf{y}$ is the observed data set. In simulation-based modeling, we do not have access to the likelihood $\pi(\mathbf{y}|\boldsymbol{\theta})$. Instead our model of the world consists of a simulator that generates samples $\mathbf{x} \overset{\text{sim}}{\sim} \pi(\mathbf{x}|\boldsymbol{\theta})$ (where we indicate that the simulator was run with parameters $\boldsymbol{\theta}$ and returns pseudo-data $\mathbf{x}$). Simulation results $\mathbf{x}$ are then compared with the observations $\mathbf{y}$ through a distribution $\pi_\epsilon(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$, which measures how similar $\mathbf{x}$ is to $\mathbf{y}$. The distribution is parameterized by ϵ which controls the acceptable discrepancy between $\mathbf{x}$ and $\mathbf{y}$. We can thus approximately infer the posterior distribution as

$$\pi_\epsilon(\boldsymbol{\theta}|\mathbf{y}) = \frac{\pi(\boldsymbol{\theta})}{\pi(\mathbf{y})} \int \pi_\epsilon(\mathbf{y}|\mathbf{x})\pi(\mathbf{x}|\boldsymbol{\theta})d\mathbf{x} \tag{2}$$

This approximate posterior is only equal to the true posterior for $\pi_{\epsilon=0}(\mathbf{y}|\mathbf{x}) = \delta(\mathbf{y}, \mathbf{x})$ where $\delta(\cdot)$ is the delta-function. For the exact posterior, one could apply a rejection sampling procedure that would repeatedly sample $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$, run a simulation $\mathbf{x} \overset{\text{sim}}{\sim} \pi(\mathbf{x}|\boldsymbol{\theta})$, then accept $\boldsymbol{\theta}$ only if it equals $\mathbf{y}$. For continuous data, ϵ acts as a slack variable because equality cannot be achieved. However, we prefer small ϵ because this will improve our approximation to the true posterior. Unfortunately, there remains an unavoidable trade-off between approximation bias (large for large ϵ) and rejection rate (large for small ϵ).

2.1 Marginal and Pseudo-Marginal ABC

Instead of the rejection sampler described in the previous section (which is hopeless in high dimensions), we now describe two MCMC procedures, the marginal sampler and the pseudo-marginal sampler [2]. At every iteration of MCMC we propose a new $\boldsymbol{\theta}' \sim q(\boldsymbol{\theta}'|\boldsymbol{\theta})$. Next we generate S samples $\mathbf{x}'_s \overset{\text{sim}}{\sim} \pi(\mathbf{x}'|\boldsymbol{\theta}'),\ s = 1..S$ from the simulator. From these samples we approximate the marginal likelihood as follows,

$$\pi_\epsilon(\mathbf{y}|\boldsymbol{\theta}') = \int \pi_\epsilon(\mathbf{y}|\mathbf{x})\pi(\mathbf{x}|\boldsymbol{\theta}')d\mathbf{x} \approx \frac{1}{S}\sum_{s=1}^{S} \pi_\epsilon(\mathbf{y}|\mathbf{x}^{(s)}, \boldsymbol{\theta}') \tag{3}$$

We accept the proposed parameter $\boldsymbol{\theta}'$ with probability equal to,

$$\alpha(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \min\left(1, \frac{\pi(\boldsymbol{\theta}')\sum_s \pi_\epsilon(\mathbf{y}|\mathbf{x}'^{(s)}, \boldsymbol{\theta}')q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})\sum_s \pi_\epsilon(\mathbf{y}|\mathbf{x}^{(s)}, \boldsymbol{\theta})q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right) \tag{4}$$

where the estimate of the marginal likelihood in the denominator (based on $\{\mathbf{x}_s, \boldsymbol{\theta}\}$) is carried over from the previous iteration. It can be shown that this algorithm is an instance of the more general pseudo-marginal procedure [2] because the estimate of the marginal likelihood is unbiased. From this we can immediately infer that this Markov chain converges to the posterior $\pi_\epsilon(\boldsymbol{\theta}|\mathbf{y})$. Interestingly, there is an alternative view of this sampler [22] that interprets it as a Markov chain over the extended state $\{\boldsymbol{\theta}, \mathbf{x}_1, ..., \mathbf{x}_S\}$ which also leads to the conclusion that the samples $\boldsymbol{\theta}'$ will asymptotically follow the distribution $\pi_\epsilon(\boldsymbol{\theta}|\mathbf{y})$.

Unfortunately, it is well known that pseudo-marginal samplers can suffer from slow mixing. In particular, when through a "lucky" draw our marginal likelihood estimate in Eqn. 3 attains a large value, then it is very difficult for the sampler to mix away from that state. To avoid this behavior it is sometimes beneficial to re-estimate the denominator (as well as the numerator) in every iteration. This procedure is more expensive and does not guarantee convergence to $\pi_\epsilon(\boldsymbol{\theta}|\mathbf{y})$ (unless $S \to \infty$), but can result in much better mixing. We will call this the marginal LF MCMC method [2].

While for the pseudo-marginal approach we can interpret the fluctuations induced by estimating the $\pi(\mathbf{y}|\boldsymbol{\theta})$ from a finite sample set as part of randomly proposing a new state, this is no longer true for the approximate marginal MCMC. For the latter it is instructive to study the uncertainty in the acceptance probability $\alpha(\boldsymbol{\theta}^{'}|\boldsymbol{\theta})$ due to these fluctuations: repeatedly estimating $\pi(\mathbf{y}|\boldsymbol{\theta})$ with S samples will produce a distribution over α. Clearly for small S the distribution will be wider while for very large S it will approach a delta peak. Our approach uses this distribution directly to determine the confidence in the MH accept step, allowing it to implicitly set S based on the local uncertainty. This will be discussed further in next sections.

Besides the marginal and pseudo-marginal approaches to ABC, there is a large body of work using sequential Monte Carlo sampling to approach this problem [21, 4, 26].

3 The Synthetic Likelihood

We next discuss the approximation introduced by Wood [29], who models the simulated pseudo-data $\{\mathbf{x}_1, .., \mathbf{x}_S\}$ at $\boldsymbol{\theta}$ using a normal distribution, i.e. $\pi(\mathbf{x}|\theta) \approx \mathcal{N}(\mathbf{x}|\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}})$. We will later replace this with a Gaussian process. The procedure is very simple: we draw S samples from our simulator and compute first and second order statistics,

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}} = \frac{1}{S}\sum_s \mathbf{x}^{(s)} \quad (5)$$

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} = \frac{1}{S-1}\sum_s \left(\mathbf{x}^{(s)} - \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}\right)\left(\mathbf{x}^{(s)} - \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}\right)^T \quad (6)$$

Using estimators $\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}$ we set $\pi(\mathbf{x}|\theta) = \mathcal{N}\left(\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}\right)$. Moreover, if we use a Gaussian kernel, then

$$\pi_\epsilon(\mathbf{y}|\mathbf{x}) = K_\epsilon\left(\mathbf{y}, \mathbf{x}\right) = \frac{1}{(2\pi\epsilon)^{J/2}} e^{-\frac{1}{2\epsilon^2}(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})} \quad (7)$$

where J is the dimension of $\mathbf{y}$, we can then analytically integrate over $\mathbf{x}$ in Eqn 2 giving the *synthetic-ABC* likelihood:

$$\pi(\mathbf{y}|\boldsymbol{\theta}) = \int K_\epsilon\left(\mathbf{y}, \mathbf{x}\right)\mathcal{N}\left(\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}\right) d\mathbf{x} \quad (8)$$

$$= \mathcal{N}\left(\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} + \epsilon^2 \boldsymbol{I}\right) \quad (9)$$

which has the satisfying result that likelihood of $\mathbf{y}|\boldsymbol{\theta}$ is the density under a Gaussian model at each simulation parameter setting $\boldsymbol{\theta}$.

This approximation has two advantages. First, we can take the limit $\epsilon \to 0$.[1] This implies that the bias introduced by the need to use a distribution $\pi_\epsilon(\mathbf{y}|\boldsymbol{\theta})$ to measure the similarity between simulations $\mathbf{x}$ and observations $\mathbf{y}$ is now

[1]We may not always want to do this as ϵ^2 acts both as a prior over and a smoother of the likelihood. In practice, smoothing the likelihood may be necessary for mixing, and likewise, we may have access to a prior which could make the sampler more robust.

Algorithm 1 Synthetic-likelihood ABC MH step

```
inputs: $q, \boldsymbol{\theta}, \pi(\mathbf{x}|\boldsymbol{\theta}), S, \epsilon, \mathbf{y}$
$\boldsymbol{\theta}^{'} \sim q(\boldsymbol{\theta}^{'}|\boldsymbol{\theta})$
for $s = 1 : S$ do
    $\mathbf{x}^{'(s)} \overset{\text{sim}}{\sim} \pi(\mathbf{x}|\boldsymbol{\theta}^{'}), \mathbf{x}^{(s)} \overset{\text{sim}}{\sim} \pi(\mathbf{x}|\boldsymbol{\theta})$
end for
Set $\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}'}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}'}, \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}$ using Eqns 5 and 6.
Set $\alpha$ using Eqn 10
if $\mathcal{U}(0,1) \leq \alpha$ then
    return $\boldsymbol{\theta}^{'}$
end if
return $\boldsymbol{\theta}$
```

removed. But this is traded off with the bias introduced by modeling the simulations from $\pi(\mathbf{x}|\boldsymbol{\theta})$ with a normal distribution. The second advantage was the main motivation in [29], namely that this procedure is more robust for extremely irregular probability distributions as encountered in chaotic or near chaotic simulation dynamics.

A marginal sampler based on a Gaussian approximation (Algorithm 1) has the following acceptance probability:

$$\alpha(\boldsymbol{\theta}^{'}|\boldsymbol{\theta}) = \min\left(1, \frac{\pi(\boldsymbol{\theta}^{'})\mathcal{N}\left(\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}'}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}'} + \epsilon^2\boldsymbol{I}\right) q(\boldsymbol{\theta}|\boldsymbol{\theta}^{'})}{\pi(\boldsymbol{\theta})\mathcal{N}\left(\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} + \epsilon^2\boldsymbol{I}\right) q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right) \quad (10)$$

As with the marginal sampler of Section 2, the fact that we estimate first and second order statistics from a finite sample set introduces uncertainty in the accept probability $\alpha(\boldsymbol{\theta}^{'}|\boldsymbol{\theta})$: another run of S simulations would have resulted in different values for these statistics and hence of the accept probability. See Figure 1 for an illustration. In the following section we will analyze the distribution over $\alpha(\boldsymbol{\theta}^{'}|\boldsymbol{\theta})$ and develop a method to decide how many simulations S we need in order to be sufficiently confident that we are making the correct accept/reject decision. Random acceptance probability distributions have been studied in general [16] and for the specific case of Gaussian log-energy estimates [9].

3.1 MCMC with a Random Acceptance Probability

We now make explicit the role of randomness in the MCMC sampler with synthetic (normal) likelihoods. At each iteration of the MCMC sampler, we compute estimators $\{\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}'}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}'}\}$ as before using Eqns 5 and 6. To estimate the distribution over accept probabilities we would need M sets of S simulations, which would be too expensive. Instead, we use our Gaussian assumption to derive that the variance of the mean is $1/S$ times the variance in the sample $\{\mathbf{x}_1, ..., \mathbf{x}_S\}$,

$$\boldsymbol{\mu}_{\boldsymbol{\theta}} \sim \mathcal{N}\left(\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}/S\right) \quad (11)$$

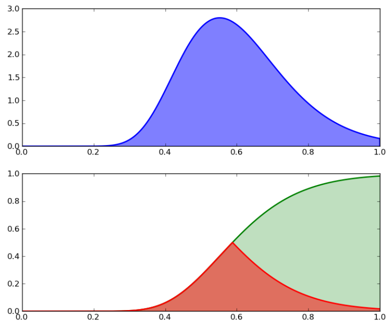


Figure 1: An example of $p(\alpha)$, the distribution over acceptance probabilities (**top**) and its CDF shown folded at its median (**bottom**).

and similarly for $\boldsymbol{\mu}_{\boldsymbol{\theta}'}$. This shortcut is important because it allows us to avoid a significant number of expensive simulations and replace them with samples from a normal distribution.

Given our M samples of $(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\mu}_{\boldsymbol{\theta}'})$, we can compute M samples of $\alpha(\boldsymbol{\theta}^{'}|\boldsymbol{\theta})$ by inserting them into the expression for the randomized MH accept probability:

$$\alpha^{(m)} = \min\left(1, \frac{\pi(\boldsymbol{\theta}^{'})\mathcal{N}\left(\mathbf{y}|\boldsymbol{\mu}_{\boldsymbol{\theta}'}^{(m)}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}'} + \epsilon^2 \boldsymbol{I}\right) q(\boldsymbol{\theta}|\boldsymbol{\theta}^{'})}{\pi(\boldsymbol{\theta})\mathcal{N}\left(\mathbf{y}|\boldsymbol{\mu}_{\boldsymbol{\theta}}^{(m)}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} + \epsilon^2 \boldsymbol{I}\right) q(\boldsymbol{\theta}^{'}|\boldsymbol{\theta})}\right) \tag{12}$$

We now derive a procedure to estimate the probability of making an error in an accept/reject decision ($\mathcal{E}(\alpha)$, the *Metropolis-Hastings error*) and a threshold τ for actually making the decision. The error of making an incorrect decision can either be measured conditioned on $u \sim \mathcal{U}(0,1)$ (the uniformly distributed draw used in the MH decision), or unconditionally, by integrating over $\mathcal{U}(0,1)$. First we start with the conditional error which trivially extends to the unconditional error by averaging.

If $u \leq \tau$, then we accept the MH proposal and move to the proposed state. The probability of making an error in this case is $P(\alpha < u)$ (i.e. the probability we should actually reject):

$$P(\alpha < u) = \frac{1}{M}\sum_m \left[\alpha^{(m)} < u\right] \tag{13}$$

Similarly, if $u > \tau$ then we reject, and the error is $P(\alpha > u)$ (i.e. the probability we should actually accept):

$$P(\alpha > u) = \frac{1}{M}\sum_m \left[\alpha^{(m)} \geq u\right] \tag{14}$$

The total conditional error is therefore:

$$\mathcal{E}_u(\alpha) = [u \leq \tau]\, P(\alpha < u) + [u > \tau]\, P(\alpha \geq u) \tag{15}$$

and the total unconditional error is:

$$\mathcal{E}(\alpha) = \int \mathcal{E}_u(\alpha)\mathcal{U}(0,1)du \tag{16}$$

which can again be estimated by Monte Carlo or grid values of u. The analytic calculation of $\mathcal{E}(\alpha)$ is the area under the cumulative distribution function of $p(\alpha)$ *folded* at τ (see Figure 1). This integral is also known as the *mean absolute deviation* [30] which is minimized at the median of $p(\alpha)$ (the value of α where the CDF equals $1/2$), justifying our decision threshold $\tau = \text{median}(\alpha)$ (also determined by samples $\alpha^{(m)}$).

With this in hand, we now have the necessary tools to construct an adaptive synthetic-likelihood MCMC algorithm that uses $\mathcal{E}(\alpha)$ as a guide for running simulations (Algorithm 2). At the start of each MH step, S_0 simulations are run for both $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{'}$; estimators are computed; then M $\alpha^{(m)}$ are sampled. Based on these samples, the median and $\mathcal{E}(\alpha)$ is computed. Note that this phase of the algorithm is very cheap; here we are sampling from J bivariate Gaussian distributions to compute Monte Carlo estimates of τ and $\mathcal{E}(\alpha)$, so M can be set high without a computational hit, though in practice $M < 100$ should be fine. While $\mathcal{E}(\alpha) > \xi$, ΔS new simulations are run and the estimators updated, along with new draws of $\alpha^{(m)}$, etc. The user-defined error threshold ξ is a knob which controls both the accuracy and computational cost of the MCMC. New simulations can be run at either $\boldsymbol{\theta}$ or $\boldsymbol{\theta}^{'}$; we run simulations at both locations, though selecting one over the other based on the higher individual mean uncertainty could result in fewer simulations. As S increases, the uncertainty around $p(\alpha)$ decreases such that $\mathcal{E}(\alpha) < \xi$; once this occurs, the MH is now *confident* and it proceeds using the usual acceptance test, with τ as the acceptance threshold.

In many cases, the actual number of simulations required at each step can be quite small, for example when one parameter setting is clearly better than another (where the median is at or close to 0 or 1). Nevertheless, there remains a serious drawback to this algorithm for expensive simulations: all simulations are discarded after each MH step; a great waste considering the result is a single binary decision. Using a Gaussian process surrogate, described in the next section, we will remember all simulations and use them to gradually improve the surrogate and as a consequence, eventually eliminate the need to run simulations.

4 Gaussian Process Surrogate ABC

As mentioned in the introduction, in many scientific disciplines simulations can be extremely expensive. The algorithms up till now all have the downside that at each MCMC update a minimum number of simulations needs to be conducted. This seems unavoidable unless we store the information of previous simulations and use them to make accept/reject decisions in the future. In particular, we can *learn* the mean and covariance $\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ of the synthetic likelihood as a function of $\boldsymbol{\theta}$ and as such avoid hav-

Algorithm 2 Adaptive Synthetic-likelihood ABC MH step

```
inputs: $q, \boldsymbol{\theta}, \pi(\mathbf{x}|\boldsymbol{\theta}), S_0, \Delta S, \epsilon, \mathbf{y}, \xi$
$\boldsymbol{\theta}' \sim q(\boldsymbol{\theta}'|\boldsymbol{\theta})$
Init $c = 1, S = S_0$
repeat
    for $s = c : c + S$ do
        $\mathbf{x}'^{(s)} \overset{\text{sim}}{\sim} \pi(\mathbf{x}|\boldsymbol{\theta}'), \mathbf{x}^{(s)} \overset{\text{sim}}{\sim} \pi(\mathbf{x}|\boldsymbol{\theta})$
    end for
    Update $c = S, S = S + \Delta S$
    Set $\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}'}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}'}, \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}$ using Eqns 5 and 6.
    for $m = 1 : M$ do
        Sample $\boldsymbol{\mu}^{(m)}_{\boldsymbol{\theta}'}, \boldsymbol{\mu}^{(m)}_{\boldsymbol{\theta}}$ using Eqn 11
        Set $\alpha^{(m)}$ using Eqn 12
    end for
    Set $\tau = \text{median}(\alpha^{(m)})$
    Set $\mathcal{E}(\alpha)$ using Eqn 16
until $\mathcal{E}(\alpha) < \xi$
if $\mathcal{U}(0, 1) \leq \tau$ then
    return $\boldsymbol{\theta}'$
end if
return $\boldsymbol{\theta}$
```

ing to perform simulations to compute them. There is a very natural tool that provides exactly this functionality, namely the Gaussian process (GP). For our purposes, the GP will "store" the simulation runs $\boldsymbol{\theta}_n, \mathbf{x}_n$ for all simulations conducted during the MCMC run. We will use the GP as a "surrogate" function for the simulated statistics from which will be able to estimate the marginal likelihood values away from regions where actual simulations were run. Importantly, the GP provides us with uncertainty estimates of the marginal likelihood which will inform us of the need to conduct additional experiments in order to make confident accept/reject decisions. Going from the synthetic likelihood model to the GP represents a change from frequentist statistics in favor of (nonparametric) Bayesian statistics. Gaussian processes have recently also become a popular tool in the machine learning literature as surrogates of expensive regression surfaces, such as log-likelihoods [17]; optimization surfaces [8, 24]; simulations of physical systems [6]; emulation of computer codes [10]; and for accelerating ABC [28].

Similar in spirit to our own work, [28] uses GP surrogates to model the ABC log-likelihood surface in successive waves of inference, each eliminating regions of implausibility and producing more and more accurate models of the log-likelihood. There are two important differences. Space-filling design points are used by [28] to train their GP models, whereas we control the simulations with ξ, and we model all J simulation outputs versus a single log-likelihood, which is a much larger overhead for our approach, but has advantages, e.g. enabling posterior analysis and evaluation of the simulator.

Our surrogate model and algorithm follow directly from the synthetic-ABC approximation and randomized acceptance algorithm. The main difference between the two is that in this paper, we model the J statistics as J independent Gaussian processes (recall J is the dimensionality of $\mathbf{y}$). We note that it would be better to model the J statistics using a single joint Gaussian process. This can be done using "co-Kriging" or variants thereof [12, 7]. Although the independence assumption may lead to overconfidence (because it is assuming–falsely–independent evidence), it is also more robust in high-dimensions where the estimator of the full output covariance has high variance (it overfits). It may be that the mentioned multi-output GPs can provide an appropriate solution by tying the covariance structure across parameter space using a small number of kernel hyperparameters. For our experiments we found that independent GPs worked well enough to illustrate the algorithm's potential. For high-dimensional outputs, modeling the log-likelihood directly may be more appropriate [28].

For each statistic j, the surrogate provides the following conditional predictive distribution of the expected value of statistic j:

$$\mu_{\boldsymbol{\theta}j} \sim \mathcal{N}\left(\bar{\mu}_{\boldsymbol{\theta}j}, \sigma^2_{\boldsymbol{\theta}j}\right) \tag{17}$$

where the mean and covariance are determined by the set of N training inputs $\{\boldsymbol{\theta}_n\}$ and N training outputs $\{\mathbf{x}_n\}$ (using only statistic j). They are given by the usual expressions for the GP mean and covariance,

$$\bar{\mu}_{\boldsymbol{\theta}j} = \boldsymbol{k}_{\boldsymbol{\theta}\boldsymbol{\Theta}j}\left[\boldsymbol{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}j} + \sigma^2_j \boldsymbol{I}\right]^{-1} \mathbf{X}[:, j] \tag{18}$$

and

$$\sigma^2_{\boldsymbol{\theta}j} = k_{\boldsymbol{\theta}\boldsymbol{\theta}j} - \boldsymbol{k}_{\boldsymbol{\theta}\boldsymbol{\Theta}j}\left[\boldsymbol{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}j} + \sigma^2_j \boldsymbol{I}\right]^{-1} \boldsymbol{k}_{\boldsymbol{\theta}\boldsymbol{\Theta}j}$$

where $\boldsymbol{k}_{\boldsymbol{\theta}\boldsymbol{\Theta}j}$ is a 1 by N vector of kernel evaluations for the j'th Gaussian process between $\boldsymbol{\theta}$ and the input training set $\boldsymbol{\Theta}$, $\boldsymbol{K}_{\boldsymbol{\Theta}\boldsymbol{\Theta}j}$ is the jth kernel matrix evaluated on the training data; σ^2_j is the data noise term for the j'th statistic (used below in the acceptance ratio), $\mathbf{X}$ is the N by J training output data set and $\mathbf{X}[:, j]$ is column j from the training data, and $k_{\boldsymbol{\theta}\boldsymbol{\theta}j}$ is a single kernel evaluation at $\boldsymbol{\theta}$ for Gaussian process j.

The GPS-ABC algorithm is now run as follows (Algorithm 3). At each MH step, using Eqn 17, and for each j, M independent draws of $\boldsymbol{\mu}_{\boldsymbol{\theta}'}$ and $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ are sampled from their conditional predictive distribution. Note that this significantly different from the SL-ABC because there are now no default simulations to be run at each MH step; instead, the current surrogate model is used to predict both the expectation and uncertainty in simulation output. As before, Monte Carlo statistics are computed from acceptance probabilities $\alpha^{(m)}$ as follows,

$$\alpha^{(m)} = \min\left(1, \frac{\pi(\boldsymbol{\theta}') \prod_j \mathcal{N}\left(y_j | \mu^{(m)}_{\boldsymbol{\theta}'j}, \sigma^2_j + \epsilon^2\right) q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}) \prod_j \mathcal{N}\left(y_j | \mu^{(m)}_{\boldsymbol{\theta}j}, \sigma^2_j + \epsilon^2\right) q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right) \tag{19}$$

Algorithm 3 GPS-ABC MH step

```
inputs: $q, \boldsymbol{\theta}, \pi(\mathbf{x}|\boldsymbol{\theta}), S_0, \Delta S, \epsilon, \mathbf{y}, \xi$
$\boldsymbol{\theta}^{'} \sim q(\boldsymbol{\theta}^{'}|\boldsymbol{\theta})$
repeat
  for $m = 1 : M$ do
    Sample $\mu^{(m)}_{\boldsymbol{\theta}' j}, \mu^{(m)}_{\boldsymbol{\theta} j}$ using Eqn 17
    Set $\alpha^{(m)}$ using Eqn 19
  end for
  Set $\tau = \mathrm{median}(\alpha^{(m)})$
  Set $\mathcal{E}(\alpha)$ using Eqn 16
  if $\mathcal{E}(\alpha) > \xi$ then
    Acquire new training point.
  end if
until $\mathcal{E}(\alpha) < \xi$
if $\mathcal{U}(0,1) \leq \tau$ then
  return $\boldsymbol{\theta}^{'}$
end if
return $\boldsymbol{\theta}$
```

The error $\mathcal{E}(\alpha)$ and acceptance threshold τ are computed from the M samples; if $\mathcal{E}(\alpha) > \xi$, then a procedure for acquiring training points (i.e. simulations) is run, with the objective of reducing uncertainty for this specific MH step. Again, as with the adaptive synthetic likelihood algorithm, computing the M samples is very cheap. The procedure is then free to select any input location to run a simulation (whereas before we were forced to run at either $\boldsymbol{\theta}$ or $\boldsymbol{\theta}^{'}$), though the new simulation should be impactful for the current MH step. This means that we can choose locations other than $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{'}$, perhaps trying to limit the number of future simulation runs required in the vicinity. Analogous to acquisition functions for Bayesian optimization [8], actively acquiring points has the implicit goals of speeding up MCMC, reducing MCMC error, and limiting simulations. We have intentionally left vague the procedure for acquiring training points; for now we run simulations at $\boldsymbol{\theta}$ or $\boldsymbol{\theta}^{'}$, even though this is an inefficient use of our surrogate. Once a new training point is selected and run, the training input-output pair is added to all J Gaussian processes and the model hyperparameters may or may not be modified (with a small number of optimization steps or by sampling).

The key advantage of GPS-ABC is that with increasing frequency, we will not have to do any expensive simulations whatsoever during a MH step because the GP surrogate is sufficiently confident about the statistics' surface in that region of parameter space.

4.1 Theoretical Aspects of GPS-ABC

Two of the main contributions of GPS-ABC are *MCMC under uncertainty* and the introduction of *memory* into the Markov chain; we consider these steps as the only way to reduce the number of expensive simulations and as such a necessary ingredient to GPS-ABC. Nevertheless, they present major differences from typical Bayesian inference procedures.

We now address two major theoretical aspects of the GPS-ABC algorithm: the *approximate* and *adaptive* nature of GPS-ABC. Although we have postponed formal proofs for future work we have outlined their main arguments below.

GPS-ABC is *approximate* because at each MH-step there is some probability that the chain will make an error, and that this corresponds to an error in the stationary distribution of the Markov chain (i.e. it is an approximation to the stationary distribution). In [14], another approximate MCMC algorithm is presented and it provides a proof for an upper bound on the error in the stationary distribution. The main argument is that if the MH-step error is small and bounded (along with a few other mild conditions), then the error in stationary distribution is bounded as well. We feel GPS-ABC fits into this same proof framework.

GPS-ABC is also *adaptive* since the approximation to the stationary distribution changes as more training points are added to the Gaussian process (we are learning the surrogate as we run the MCMC). Two of the major requirements for a valid adaptive MCMC algorithm are *diminishing adaptation* and *ergodicity* [19]. GPS-ABC satisfies the former as the number of training points acquired over an MCMC run rapidly decreases over time. When the adaptation slows and becomes insignificant, the Markov chain resembles the algorithm in [14], which, as we stated above, provides a proof of bounded convergence to the stationary distribution (and hence ergodicity); therefore we believe that GPS-ABC satisfies the latter requirement.

5 Experiments

The main goal of our experiments is to show the correctness and computational gains of GPS-ABC. Our results indicate that correct posterior samples can be obtained by GPS-ABC with orders of magnitude fewer simulation calls than traditional ABC sampling procedures.

We perform experiments on two simulation problems: 1) a toy Bayesian inference problem (the exponential problem) and 2) inference of parameters in a chaotic ecological system (the blowfly problem). In the former, the true posterior distribution is known and therefore provides a useful test-case for determining the correctness and convergence properties of ABC algorithms. There is no ground truth in the latter problem, making inference much more difficult, but we can nevertheless assess it by the quality of raw simulation outputs and convergence to a set of chosen statistics.

Here is a brief description of the algorithms used in our experiments. We ran ϵ-tube rejection sampling (REJ); synthetic-likelihood (SL, both marginal and pseudo-marginal); and two adaptive ξ-ABC algorithms: adaptive

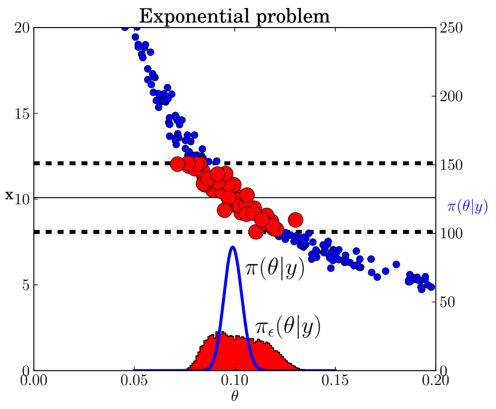


Figure 2: The exponential problem. Shown on the same y-axis are values from the simulator x and the density of the true and approximate posteriors ($\pi(\theta|y)$ and $\pi_\epsilon(\theta|y)$, respectively). The horizontal lines indicate y +/- ϵ. The approximate posterior was generated from ϵ-tube rejection sampling ($\epsilon = 2$).

synthetic likelihood (ASL) and Gaussian process surrogate ABC (GPS). Rejection sampling is a procedure where $\boldsymbol{\theta}$ is repeatedly drawn from the prior until all simulation statistics are within the ϵ-tube. This is repeated independently for each sample. Algorithms SL, ASL perform at least S simulations at each MH-step of a MCMC run; for marginal algorithms, this is $2S$ (the simulations at the current location are re-run), for ASL this can be higher, depending on ξ and due to randomness within the MH-step. Marginal versus pseudo-marginal results are indicated by the prefixes 'm' or 'p', respectively. Our MCMC algorithms are initialized with a single rejection sample, so initial simulation counts are affected by its ϵ value. GPS uses a small initial training set of size $S_0 = 50$. For the exponential problem, this training set was generated by rejection sampling. For the blowfly problem, a short MCMC run using SL was used to generate higher quality training points. GPS adapts its Gaussian process hyperparameters using MAP estimates found by conjugate gradient methods during initialization, and subsequently when the number of training points doubles (i.e. 100, 200, 400, etc). Finally, ASL and GPS adapt the number of simulations at each MH-step $\Delta S = 5$. Due to space constraints, some of the results are not shown in the main paper, but can be found in the supplementary material.

5.1 Inferring the parameter of an exponential distribution

In this illustrative problem we infer the rate of an exponential distribution under a gamma prior with shape α and rate β, having N observations at the true rate $\theta^\star$; this is the *exponential example* in [27]. Let $\mathbf{w}$ be a vector of N draws from an exponential distribution, i.e. $w_n \sim \text{Exp}(\theta)$. The posterior is a gamma distribution with shape $\alpha + N$ and rate $\beta + \sum w_n$. To use this problem with ABC, we use the exponential draws as the simulator and the mean of $\mathbf{w}$ as the statistic y and assume that N is known. The inference problem is therefore to sample from $p(\theta|y, \alpha, \beta, N)$.

For all runs we fixed $\alpha = \beta = 0.1$; a very broad prior over θ. Using the same random seed and $\theta^\star = 0.1$, we generated $y = 10.0867$, which induces the θ-MAP value 0.09916 (not quite 10 and 0.1 due to their random draw and to a small influence from the prior). An illustration of this problem is shown in Figure 2.

Figure 3 show the results of running REJ, SL, and GPS to generate 50000 samples; this is repeated 10 times per algorithm. These three algorithms were chosen because they give roughly the same final error in distribution. In Figure 3a, the convergence to the (known) target posterior distribution (error), *per sample*, is shown. Figure 3b shows the same convergence in error, but is overlaid with the convergence per *simulation call* instead of per sample. Rejection sampling, as expected, provides the best convergence per sample (each sample is an independent sample from the approximate posterior) but computationally performs very poorly when ϵ is set to a value that gives small error in distribution.

As GPS-ABC acquires training points it is also sampling from the approximate posterior. Once the surrogate has learned the statistic surfaces in a region of parameter space, it no longer makes any simulation calls in that region. Eventually, the surrogate learns the surface for all the regions of parameter space where the posterior density is high. For this problem (with $\xi = 0.2$), this occurs after approximately 1000 simulations. As the MCMC run progresses, GPS-ABC gathers samples with decreasing amounts of computation. Figure 3c shows how the Gaussian process adaptation levels off during MCMC runs, whereas traditional ABC algorithms require a constant number of simulation calls per sample. As ξ decreases, more simulation calls are required for the GPS-ABC to model the statistics surfaces with increased precision; they all, however, have adaptation curves similar to Figure 3c.

5.2 Chaotic Ecological Systems

Adult blowfly populations exhibit dynamic behavior for which several competing population models exist. In this experiment, we use observational data and a simulation model from Wood [29], based on their improvement upon previous population dynamics theory. Population dynamics are modeled using (discretized) differential equations that can produce chaotic behavior for some parameter settings. An example blowfly replicate series is shown in Figure 4a, along with times-series generated by a sample from $\pi(\boldsymbol{\theta}|\mathbf{y})$ using GPS-ABC.

In [29] there are several explanations of the population dynamics, corresponding to different simulations and parameters. We concentrate on the equation (1) in section 1.2.3 of the supplementary information, considered "a better al-

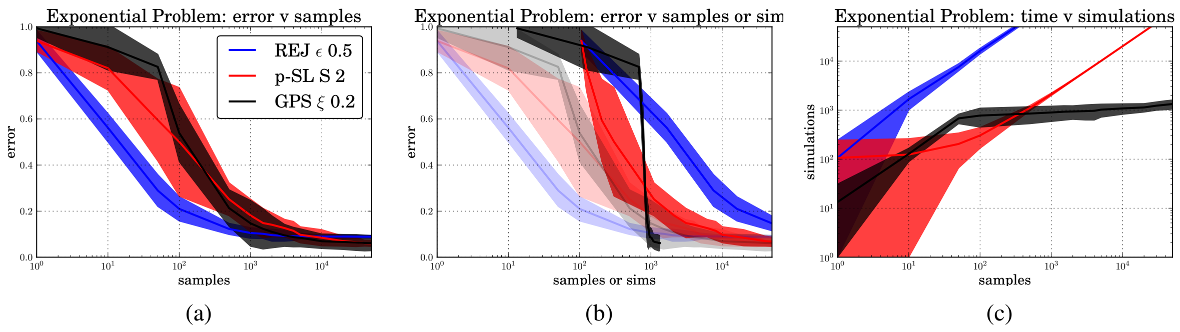


Figure 3: Convergence to target distribution for three algorithms: REJ ($\epsilon = 0.5$), p-SL ()$S = 2$), and GPS ($\xi = 0.2$). The three have roughly the same final error in distribution. In **(a)** the convergence to the target *per sample* is shown; in **(b)** this convergence is overlaid with convergence *per simulation*. Around sample 1000, GPS has essentially stopped adapting, has learned the statistic surfaces, and no longer requires simulations. In **(c)** the growth of simulations per time step (i.e. a sample) for the sample algorithms is shown; both REJ and SL use a fixed number of simulations per iteration, whereas GPS stops adding new training points around time 1000. Note that we start the plot at the 10th sample, which requires a different number of simulations, depending on the algorithm.

ternative model" by the author. The population dynamics equation generates $N_1, \ldots, N_T$ using the following update rule:

$$N_{t+1} = PN_{t-\tau} \exp(-N_{t-\tau}/N_0)e_t + N_t \exp(-\delta\epsilon_t)$$

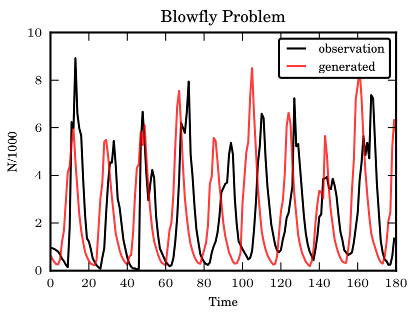
where $e_t \sim \mathcal{G}(1/\sigma_p^2, 1/\sigma_p^2)$ and $\epsilon_t \sim \mathcal{G}(1/\sigma_d^2, 1/\sigma_d^2)$ are sources of noise, and τ is an integer (not to be confused with the τ used as the MH acceptance threshold in our algorithms). In total, there are 6 parameters $\theta = \{\log P, \log\delta, \log N_0, \log\sigma_d, \log\sigma_p, \tau\}$. See [29] for further details about the significance of the parameters. We put Gaussian priors over all the parameters (with Gaussian proposal distributions), except for τ which has a Poisson prior (and a left/right increment proposal). Time-series generated with parameters from this prior distribution produce extremely varied results, some are chaotic, some are degenerate, etc. Modeling this simulator is *very* challenging.

As with any ABC problem the choice of statistics is important as it directly affects the quality of the results. It is also non-trivial and requires careful thought and sometimes trial and error. In total there are 10 statistics: the log of the mean of all 25% quantiles of $N/1000$ (4 statistics), the mean of the 25% quantiles of the first-order differences of $N/1000$ (4 statistics), and the maximal peaks of smoothed N, with 2 different thresholds (2 statistics). With these statistics it is possible to reproduce time-series that appear similar to the observations. Note that these statistics are different from Wood's, but they capture similar time-series features and are sufficient to produce credible population dynamics.

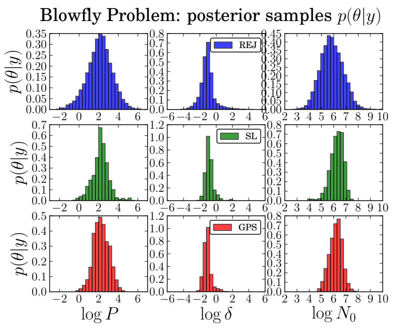
The first set of experiments for the blowfly problem is shown in Figure 4b. For these experiments we compared REJ ($\epsilon = 5$), corresponding to an acceptance rate of roughly 2%), p-SL ($S = 10$), and finally GPS ($\xi = 0.3$). For SL, a small $\epsilon = 0.5$ was required to improve mixing. This helped all algorithms, but was less important for GPS, though it can be important if the Gaussian processes are not properly calibrated. Each algorithm was run 5 times collecting 10K samples each. The GPS model for this problem consists of $J = 10$ independent Gaussian processes with $D = 6$ inputs each. The GPS was initialized with $S_0 = 50$ samples from a short SL-ABC MCMC run. In Figure 4b we show the posterior distributions for $\log P$, $\log\delta$, and $\log N_0$. Rejection sampling produces much broader posteriors than both SL and GPS, though they all share roughly the same mode. Between SL and GPS there is little difference in mode or shape, though GPS appears to have tighter confidence intervals. The real difference is the computational effort required: GPS used only 384 simulations to produce 10K, roughly 0.04 simulations per sample, whereas SL and REJ require 10 and 45 simulations for a single sample, respectively. Of course, GPS has much higher efficiency in practice, as the value 0.04 simulations per sample decreases over time and eventually reaches 0.

The second set of experiments examined the convergence properties of GPS compared to the other algorithms, focusing on the quality of the posterior predictive distributions *per sample* versus *simulation call*. For the blowfly data we do not have the ground-truth $\boldsymbol{\theta}^\star$, but we do have the statistics of the observations for which we can monitor convergence. We do this by evaluating the posterior predictive distribution $p(\mathbf{y}|\mathbf{y}^\star)$. Please note that we slightly change notation when discussing posterior predictive distributions by denoting $\mathbf{y}^\star$ the observed statistics, and $\mathbf{y}$ as statistics generated by the posterior samples $p(\boldsymbol{\theta}|\mathbf{y}^\star)$. Pairs $\mathbf{y}$ and $\boldsymbol{\theta}$ are generated, with the exception of GPS, during the MCMC run, and are exactly the quantities we require for $p(\mathbf{y}|\mathbf{y}^\star)$. Convergence will tell us the amount of computational effort required for an independent and (un)biased sample.

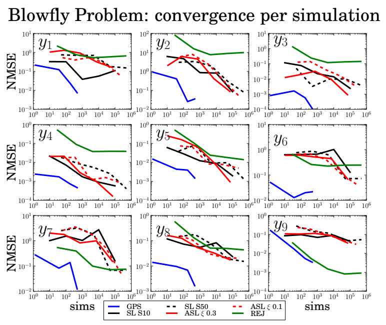
In Figure 4c we show the convergence in expected value of $\mathbf{y}$ to $\mathbf{y}^\star$ per simulation. This is calculated in an online fashion by averaging statistics generated at $\boldsymbol{\theta}$ (each *pos-*



(a)



(b)



(c)

Figure 4: **(a)** The blowfly population time-series. The observations are in black lines and a generated time-series is shown in red. Note that $\boldsymbol{\theta}$ is a sample from the GPS posterior ($\xi = 0.3$). See paper for description of the statistics. **(b)** Blowfly $p(\boldsymbol{\theta}|\mathbf{y})$ for different algorithms. The top (blue) row is REJ $\epsilon = 5$, the middle (green) is pseudo-marginal SL S10, and the bottom (red) is GPS $\xi = 0.3$. Their respective simulations per sample ratios were: 45, 10, and 0.04. I.e. REJ had an acceptance rate of 2.2%, and GPS only used 384 simulations for 10K samples. Similar distributions were observed for ASL and other pseudo-marginal SL. **(c)** Convergence to $\mathbf{y}^\star$ for different algorithms, using normalized mean-squared error. Each sub-plot shows the convergence to a single statistic as a function of simulation calls.

terior $\boldsymbol{\theta}$ sample is allowed one $\mathbf{y}$). We show the log-log plots of the normalized mean squared error (NMSE) versus number of simulations for the first 9 statistics. Algorithms run in this experiment were GPS ($\xi = 0.2$), p-SL ($S = 10, 50$), ASL ($\xi = 0.3, 0.1$), and REJ ($\epsilon = 5$). GPS not only converges systematically to $\mathbf{y}^\star$ but does so with dramatically less effort. For some statistics REJ performed reasonably well, but in general exhibited significant bias. Both SL and ASL converged to similar biases, usually outperforming REJ, but in some cases was worse. Parameter settings where more computation was required for SL and ASL did result in slightly improved convergence, but the gain comes with a significantly higher computational cost. In summary, GPS is able to model the statistic surfaces, enabling it to correctly sample from the approximate posterior target distribution with higher precision and with orders of magnitude fewer simulation calls than the other algorithms.

6 Discussion and Future Work

We have presented a promising framework for performing inference in expensive, simulation-based models. Our algorithms improve current state-of-the-art ABC methods in that they require many fewer calls to the simulator, sometimes orders of magnitude fewer.

Using GPs for surrogate modeling has an appealing elegance; as nonparametric Bayesian models, they naturally incorporate both model and pseudo-data uncertainty into the MCMC algorithms. However, there are several technical issues and modeling limitations with GPs used for surrogate modeling. Heteroskedatic noise is more likely the norm than the exception for complicated simulators. The blowfly simulator is a prime example of this. Improvements to our GPs may be achieved using an input-dependent noise model [11, 13], where the noise is an additional independent Gaussian process. Another limitation of our GP model is the output independence assumption. A more realistic assumption is a full covariance Gaussian process such as the convolution processes of [12, 7, 1]. One final limitation is GP calibration. We found that initializing the Gaussian process with rejection samples produced inferior results, as it tended to adapt its hyper-parameters optimally for those training points and had difficulty readapting. Despite these limitations, we feel that GPS-ABC deserves a place within the ABC toolbox.

Our GPS-ABC uses a random-walk proposal distribution which is inefficient for exploring the target distribution. Using GPs offers the opportunity to use other techniques to improve the mixing (and in turn computational cost). For example, in [17] a Hamiltonian Monte Carlo run on the GP surface is used to generate *independent* proposals. If applied to ABC, their algorithm would require the equivalent of a full simulation at the proposed location, whereas if we incorporated a similar technique, we would then test the GP uncertainty to determine if a simulation was required.

There has been a recent surge in interest in Bayesian optimization using Gaussian process (GP-BO) surrogates of the objective surface [8, 24]. GP-BO is often applied to problems where simulation or sometimes user feedback guides the surrogate's construction. What is most interesting about GP-BO is its use of model uncertainty to *actively* determine the next simulation location implemented through acquisition functions. These ideas can be generalized to our GPS-ABC algorithm to further reduce simulation costs, while at the same time maintaining control of the MCMC error.

Acknowledgements

MW acknowledges support from the CIFAR-NCAP program.

References

[1] Álvarez, M. and Lawrence, L.D. Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, 12:1425–1466, 2011.

[2] Andrieu, C. and Roberts, G. The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

[3] Bazin, Eric, Dawson, Kevin J, and Beaumont, Mark A. Likelihood-free inference of population structure and local adaptation in a bayesian hierarchical model. *Genetics*, 185(2):587–602, 2010.

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

[5] Beaumont, Mark A, Zhang, Wenyang, and Balding, David J. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

[6] Bilionis, I., Zabaras, N., Konomi, B.A., and Lin, G. Multi-output separable gaussian process: Towards an efficient fully bayesian paradigm for uncertainty quantification. *Journal of Computational Physics*, 241:212–239, 2013.

[7] Boyle, P. and Frean, M. Dependent gaussian processes. *Advances in Neural Information Processing Systems 17*, 2005.

[8] Brochu, E., Cora, Vlad M., and de Freitas, Nando. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical report, UBC Technical Report, 2010.

[9] Ceperley, D.M. and Dewing, M. The penalty method for random walks with uncertain energies. *Journal of Chemical Physics*, 110(20):9812–9820, 1999.

[10] Conti, S., Gosling, J.P., Oakley, J., and O'Hagan, A. Gaussian process emulation of dynamic computer codes. *Biometrika*, 96(3):663–676, 2009.

[11] Goldberg, P.W., Williams, C.K.I., and Bishop, C.M. Regression with input-dependent noise: A gaussian process treatment. *Advances in Neural Information Processing Systems 10*, 1998.

[12] Higdon, D. Space and space-time modeling using process convolutions. In Anderson, C.W., Barnett, V., Chatwin, P.C, and Abdel, E.-S.H. (eds.), *Quantitative Methods for Current Environmental Issues*, pp. 37–56. Springer London, 2002. ISBN 978-1-4471-1171-9.

[13] Kersting, K., Plagemann, C., Pfaff, P., and Burgard, W. Most likely heteroscedastic gaussian process regression. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 393–400. ACM, 2007.

[14] Korattikara, A., Chen, Y., and Welling, M. Austerity in mcmc land: Cutting the metropolis-hastings budget. *ICML*, 2014. To appear.

[15] Marin, J.-M., Pudlo, P., Robert, C.P., and Ryder, R.J. Approximate bayesian computational methods. *Statistics and Computing*, 22:1167–1180, 2012.

[16] Nicholls, G.K., Fox, C., and Watt, A.M. Coupled mcmc with a randomized acceptance probability. Technical report, arXiv:1205.6857v1, 2012.

[17] Rasmussen, C.E. Gaussian processes to speed up hybrid monte carlo for expensive bayesian integrals. *Bayesian Statistics*, 7:651–659, 2003.

[18] Ratmann, O., Jorgensen, O., Hinkley, T., Stumpf, M., Richardson, S., and Wiuf, C. Using likelihood-free inference to compare evolutionary dynamics of the protein networks of h. pylori and p. falciparum. *PLoS Computational Biology*, 3(11):e230, 2007.

[19] Roberts, G.O. and Rosenthal, J.S. Coupling and ergodicity of adaptive mcmc. *J. Appl. Prob.*, 44:458–475, 2007.

[20] Schafer, C.M. and Freeman, P.E. Likelihood-free inference in cosmology: Potential for the estimation of luminosity functions. In *Statistical Challenges in Modern Astronomy V*, pp. 3–19. Springer, 2012.

[21] Sisson, SA, Fan, Y, and Tanaka, Mark M. Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760, 2007.

[22] Sisson, S.A., Peters, G.W., Briers, M., and Fan, Y. A note on target distribution ambiguity of likelihood-free samplers. *Arxiv preprint arXiv:1005.5201v1 [stat.CO]*, 2010.

[23] Sisson, Scott A and Fan, Yanan. Likelihood-free markov chain monte carlo. *Arxiv preprint arXiv:1001.2058*, 2010.

[24] Snoek, J., Larochelle, H., and Adams, R.P. Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems 25*, 2012.

[25] Tavare, S., Balding, D.J., Griffiths, R.C., and Donnelly, P. Inferring coalescence times from dna sequence data. *Genetics*, 145(2):505–518, 1997.

[26] Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M.P.H. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009.

[27] Turner, Brandon M and Van Zandt, Trisha. A tutorial on approximate bayesian computation. *Journal of Mathematical Psychology*, 56(2):69–85, 2012.

[28] Wilkinson, R. Accelerating abc methods using gaussian processes. *AISTATS*, 2014.

[29] Wood, Simon N. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466 (7310):1102–1104, 2010.

[30] Xue, Jing-Hao and Titterington, D. Michael. The p-folded cumulative distribution function and the mean absolute deviation from the p-quantile. *Statistics and Probability Letters*, 81:1179–1182, 2011.