School of Mathematical
and Physical Sciences

University of Reading

Department of Mathematics and Statistics

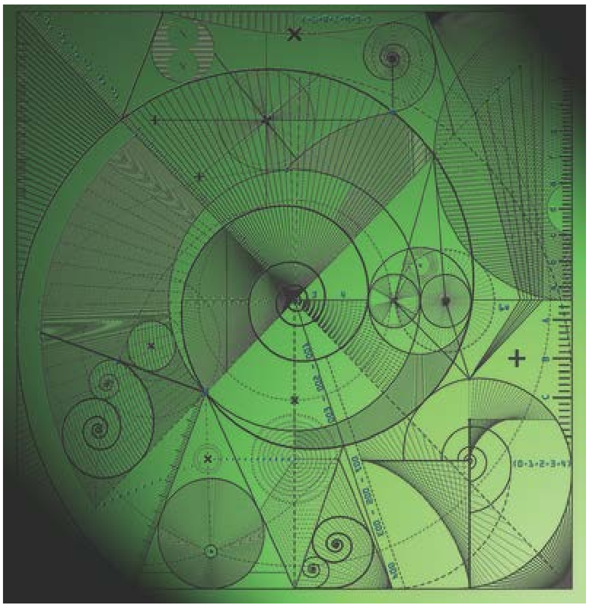
Preprint MPS-2015-09

3 July 2015

Adapting the ABC distance function

by

Dennis Prangle



Adapting the ABC distance function

Dennis Prangle[*]

Abstract

Approximate Bayesian computation performs approximate inference for models where likelihood computations are expensive or impossible. Instead simulations from the model are performed for various parameter values and accepted if they are close enough to the observations. There has been much progress on deciding which summary statistics of the data should be used to judge closeness, but less work on how to weight them. Typically weights are chosen at the start of the algorithm which normalise the summary statistics to vary on similar scales. However these may not be appropriate in iterative ABC algorithms, where the distribution from which the parameters are proposed is updated. This can substantially alter the resulting distribution of summary statistics, so that different weights are needed for normalisation. This paper presents an iterative ABC algorithm which adaptively updates its weights, without requiring any extra simulations to do so, and demonstrates improved results on test applications.

Keywords: likelihood-free inference, population Monte Carlo, quantile distributions, Lotka-Volterra

1 Introduction

Approximate Bayesian computation (ABC) is a family of approximate inference methods which can be used when the likelihood function is expensive or impossible to compute but

[*] University of Reading. Email d.b.prangle@gmail.com

simulation from the model is straightforward. The simplest algorithm is a form of rejection sampling. Here parameter values are simulated from the prior distribution and corresponding datasets are simulated. Each simulation is converted to a vector of summary statistics $\boldsymbol{s} = (s_1, s_2, \ldots, s_m)$ and a distance between this and the summary statistics of the observed data, $\boldsymbol{s}_{\text{obs}}$, is calculated. Parameters producing distances below some threshold are accepted and form a sample from an approximation to the posterior distribution.

The choice of summary statistics has long been recognised as being crucial to the quality of the approximation (Beaumont et al., 2002), but there has been less work on the role of the distance function. A popular distance function is weighted Euclidean distance:

$$d(\boldsymbol{s}, \boldsymbol{s}_{\text{obs}}) = \left[\sum_{i=1}^{m} \left(\frac{s_i - s_{\text{obs},i}}{\sigma_i} \right)^2 \right]^{1/2} \tag{1}$$

where σ_i is an estimate of the prior predictive standard deviation of the ith summary statistic. In ABC rejection sampling a convenient estimate is the empirical standard deviation of the simulated s_i values. Scaling by σ_i in (1) normalises the summaries so that they vary over roughly the same scale, preventing the distance being dominated by the most variable summary.

This paper concerns the choice of distance in more efficient iterative ABC algorithms, in particular those of Toni et al. (2009), Sisson et al. (2009) and Beaumont et al. (2009). The first iteration of these algorithms is the ABC rejection sampling algorithm outlined above. The sample of accepted parameters is used to construct an importance density. An ABC version of importance sampling is then performed. This is similar to ABC rejection sampling, except parameters are sampled from the importance density rather than the prior, and the output sample is weighted appropriately to take this change into account. The idea is to concentrate computational resources on performing simulations for parameter values likely to produce good matches. The output of this step is used to produce a new importance density and perform another iteration, and so on. In each iteration the acceptance threshold is reduced, resulting in increasingly accurate approximations. Full details of this algorithm are reviewed later.

Weighted Euclidean distance is commonly used in this algorithm with σ_i values determined in the first iteration. However there is no guarantee that these will normalise the summary statistics produced in later iterations, as these are no longer drawn from the prior predictive. This paper proposes a variant iterative ABC algorithm which updates the σ_i values at each iteration to appropriate values. It is demonstrated that this algorithm provides substantial advantages in applications. Also, it does not require any extra simulations to be performed. Therefore even when a non-adaptive distance performs adequately, there is no major penalty in using the new approach. (Some additional calculations are required – calculating more σ_i values and more expensive distance calculations – but these form a negligible part of the overall computational cost.)

The proposed algorithm has some similarities to the iterative ABC methods of Sedki et al. (2012) and Bonassi and West (2015). These postpone deciding some elements of the tuning of iteration t until during that iteration. The new algorithm also uses this strategy but for different tuning decisions: the distance function and the acceptance threshold.

The remainder of the paper is structured as follows. Section 2 reviews ABC algorithms. This includes some novel material on the convergence of iterative ABC methods. Section 3 discusses weighting summary statistics in a particular ABC distance function. Section 4 details the proposed algorithm. Several examples are given in Section 5. Section 6 summarises the work and discusses potential extensions. Finally Appendix A contains technical material on convergence of ABC algorithms. Computer code to implement the methods of this paper in the Julia programming language (Bezanson et al., 2012) is available at `https://github.com/dennisprangle/ABCDistances.jl`.

2 Approximate Bayesian Computation

This section sets out the necessary background on ABC algorithms. Several review papers (e.g. Beaumont, 2010; Csilléry et al., 2010; Marin et al., 2012) give detailed descriptions of other aspects of ABC, including tuning choices and further algorithms. Sections 2.1 and 2.2 review ABC versions of rejection sampling and PMC. Section 2.3 contains novel material on the convergence of ABC algorithms.

2.1 ABC rejection sampling

Consider Bayesian inference for parameter vector θ under a model with density $\pi(\boldsymbol{y}|\theta)$. Let $\pi(\theta)$ be the prior density and $\boldsymbol{y}_{\text{obs}}$ represent the observed data. It is assumed that $\pi(\boldsymbol{y}|\theta)$ cannot easily be evaluated but that it is straightforward to sample from the model. ABC rejection sampling (Algorithm 1) exploits this to sample from an approximation to the posterior density $\pi(\theta|\boldsymbol{y})$. It requires several tuning choices: number of simulations N, a threshold $h \geq 0$, a function $S(\boldsymbol{y})$ mapping data to a vector of summary statistics, and a distance function $d(\cdot, \cdot)$.

Algorithm 1 ABC-rejection

```
1. Sample $\theta_i^*$ from $\pi(\theta)$ independently for $1 \leq i \leq N$.
2. Sample $\boldsymbol{y}_i^*$ from $\pi(\boldsymbol{y}|\theta_i^*)$ independently for $1 \leq i \leq N$.
3. Calculate $\boldsymbol{s}_i^* = S(\boldsymbol{y}_i^*)$ for $1 \leq i \leq N$.
4. Calculate $d_i^* = d(\boldsymbol{s}_i^*, \boldsymbol{s}_{\text{obs}})$ (where $\boldsymbol{s}_{\text{obs}} = S(\boldsymbol{y}_{\text{obs}})$.)
5. Return $\{\theta_i^* | d_i^* \leq h\}$.
```

The threshold h may be specified in advance. Alternatively it can be calculated following step 4. For example a common choice is to specify an integer k and take h to be the kth smallest of the d_i^* values (Biau et al., 2015).

2.2 ABC-PMC

Algorithm 2 is an iterative ABC algorithm taken from Toni et al. (2009). Very similar algorithms were also proposed by Sisson et al. (2009) and Beaumont et al. (2009). The latter note that this approach is an ABC version of population Monte Carlo (Cappé et al., 2004), so it is referred to here as ABC-PMC. The algorithm involves a sequence of thresholds, $(h_t)_{t \geq 1}$. Similarly to h in ABC-rejection, this can be specified in advance or during the algorithm, as discussed below.

Algorithm 2 ABC-PMC

```
**Initialisation**
1. Let $t = 1$.
**Main loop**
2. Repeat following steps until there are $N$ acceptances.
    (a) If $t = 1$ sample $\theta^*$ from $\pi(\theta)$. Otherwise sample $\theta^*$ from importance density $q_t(\theta)$ given in equation (2).
    (b) If $\pi(\theta^*) = 0$ reject and return to (a).
    (c) Sample $\boldsymbol{y}^*$ from $\pi(\boldsymbol{y}|\theta_i^*)$ and calculate $\boldsymbol{s}^* = S(y^*)$.
    (d) Accept if $d(\boldsymbol{s}^*, \boldsymbol{s}_{\text{obs}}) \leq h_t$.
   Denote the accepted parameters as $\theta_1^t, \ldots, \theta_N^t$.
3. Calculate $w_i^t$ for $1 \leq i \leq N$ as follows. If $t = 1$ let $w_i^1 = 1$. Otherwise let $w_i^t = \pi(\theta_i^t)/q_t(\theta_i^t)$.
4. Increment $t$ and return to step 2.
```

When $t > 1$ the algorithm samples parameters from the following importance density

$$q_t(\theta) = \sum_{i=1}^{N} w_i^{t-1} K_t(\theta|\theta_i^{t-1}) / \sum_{i=1}^{N} w_i^{t-1}. \tag{2}$$

Drawing from this effectively samples from the previous weighted population and perturbs the result using kernel K_t. Beaumont et al. (2009) show that a good choice of the latter is

$$K_t(\theta|\theta') = \phi(\theta', 2\Sigma_{t-1}),$$

where ϕ is the density of a normal distribution and Σ_{t-1} is the empirical variance matrix of $(\theta_i^{t-1})_{1\leq i\leq N}$ calculated using weights $(w_i^{t-1})_{1\leq i\leq N}$

As mentioned above, the schedule of thresholds can be specified in advance. However it is hard to do this well. A popular alternative (Drovandi and Pettitt, 2011a) is to choose h_t at the end of iteration $t-1$ as the α quantile of the accepted distances (Details will be shown in Algorithm 3 in the next section.) This leaves h_1 and α as tuning choices. A simple default for h_1 is ∞, in which case all simulations are accepted when $t = 1$. Alternative updating rules for h_t have been proposed such as choosing it to reduce an estimate of effective sample size by a prespecified proportion (Del Moral et al., 2012) or using properties of the predicted ABC acceptance rate (Silk et al., 2013).

A practical implementation of Algorithm 2 requires a condition for when to terminate. In this paper the total number of datasets to simulate is specified as a tuning parameter and the algorithm stops once a further simulation is required.

Several variations on Algorithm 2 have been proposed which are briefly discussed in Section 6. Some of these are ABC versions of sequential Monte Carlo (SMC). The phrase "iterative ABC" will be used to cover ABC-PMC and ABC-SMC.

2.3 Convergence of ABC-PMC

Conditions C1-C5 ensure that Algorithm 2 converges on the posterior density in an appropriate sense as the number of iterations tends to infinity. This follows from Theorem 1 which is described in Appendix A. Although only finite computational budgets are available in practice, such convergence at least guarantees that the target distribution become arbitrarily accurate as computational resources are increased.

C1. $\theta \in \mathbb{R}^n$, $\boldsymbol{s} \in \mathbb{R}^m$ for some m, n and these random variables have density $\pi(\theta, \boldsymbol{s})$ with respect to Lebesgue measure.

C2. The sets $A_t = \{\boldsymbol{s} | d(\boldsymbol{s}, \boldsymbol{s}_{\text{obs}}) \leq h_t\}$ are Lebesgue measurable.

C3. $\pi(\boldsymbol{s}_{\text{obs}}) > 0$.

C4. $\lim_{t \to \infty} |A_t| = 0$ (where $| \cdot |$ represents Lebesgue measure.)

C5. The sets A_t have *bounded eccentricity.*

Bounded eccentricity is defined in Appendix A. Roughly speaking, it requires that under any projection of A_t to a lower dimensional space the measure still converges to zero.

Condition C1 is quite strong, ruling out discrete parameters and summary statistics, but makes proof of Theorem 1 straightforward. Condition C2 is a mild technical requirement. The other conditions provide insight into conditions required for convergence. Condition C3 requires that it must be possible to simulate $\boldsymbol{s}_{\text{obs}}$ under the model. Condition C4 requires that the acceptance regions A_t shrink to zero measure. For most distance functions this corresponds to $\lim_{t \to \infty} h_t = 0$. It is possible for this to fail in some situations, for example if datasets close to $\boldsymbol{s}_{\text{obs}}$ cannot be produced under the model of interest (in which case C2 generally also fails.) Alternatively, even if $\boldsymbol{s}_{\text{obs}}$ can occur under the model, the algorithm may converge on importance densities on θ under which it is impossible. This corresponds to concentrating on the wrong mode of the ABC target distribution in an early iteration. Finally, condition C5 prevents A_t converging to a set where some but not all summary statistics are perfectly matched.

Conditions C4 and C5 can be used to check which distance functions are sensible to use in ABC-PMC, usually by investigating whether they hold when $h_t \to 0$. For example it is straightforward to show this is the case when $d(\cdot, \cdot)$ is a metric induced by a norm.

3 Weighted Euclidean distance in ABC

This paper concentrates on using weighted Euclidean distance in ABC. Section 3.1 discusses this distance and how to choose its weights. Section 3.2 illustrates its usefulness in a simple example.

3.1 Definition and usage

Consider the following distance:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left[\sum_{i=1}^{m} \{w_i(x_i - y_i)\}^2 \right]^{1/2}. \tag{3}$$

If $w_i = 1$ for all i, this is is *Euclidean distance.* Otherwise it is a form of *weighted Euclidean distance.*

Many other distance functions can be used in ABC, as discussed in Section 2.3, for example weighted L_1 distance $d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{m} w_i |x_i - y_i|$. To the author's knowledge the only published comparison of distance functions is by McKinley et al. (2009). This did not find any distances which provide a significant improvement over (3). Owen et al. (2015) report the same conclusion but not the details. This finding is also supported in unpublished work by the author of this paper and by others (Sisson, personal communication). Therefore the paper focuses on Euclidean distance and the choice of weights to use with it.

Summary statistics used in ABC may vary on substantially different scales. In the extreme case Euclidean distance will be dominated by the most variable. To avoid this, weighted Euclidean distance is generally used. This usually takes $w_i = 1/\sigma_i$ where σ_i is an estimate of the scale of the ith summary statistic. (Using this choice in weighted Euclidean distance gives the distance function (1) discussed in the introduction.)

A popular choice (e.g. Beaumont et al., 2002) of σ_i is the empirical standard deviation of the ith summary statistic under the prior predictive distribution. Csilléry et al. (2012) suggest using median absolute deviation (MAD) instead since it is more robust to large outliers. MAD is used throughout this paper. For many ABC algorithms these σ_i values can be calculated without requiring any extra simulations. For example this can be done between steps 3 and 4 of ABC-rejection. ABC-PMC can be modified similarly, resulting in Algorithm 3, which also updates h_t adaptively. (n.b. All of the ABC-PMC convergence discussion in Section 2.3 also applies to this modification.)

Algorithm 3 ABC-PMC with adaptive h_t and $d(\cdot, \cdot)$

```
    Initialisation
1. Let $t = 1$ and $h_1 = \infty$.
    Main loop
2. Repeat following steps until there are $N$ acceptances.
    (a) If $t = 1$ sample $\theta^*$ from $\pi(\theta)$. Otherwise sample $\theta^*$ from importance density $q_t(\theta)$ given in equation (2).
    (b) If $\pi(\theta^*) = 0$ reject and return to (a).
    (c) Sample $\boldsymbol{y}^*$ from $\pi(\boldsymbol{y}|\theta_i^*)$ and calculate $\boldsymbol{s}^* = S(y^*)$.
    (d) Accept if $d(\boldsymbol{s}^*, \boldsymbol{s}_{\text{obs}}) \leq h_t$ (if $t = 1$ always accept).
3. If $t = 1$:
    (a) Calculate $(\sigma_1, \sigma_2, \ldots)$, a vector of MADs for each summary statistic, calculated from all the simulations in step 2 (including those rejected).
    (b) Define $d(\cdot, \cdot)$ as the distance (3) using weights $(w_i)_{1 \leq i \leq m}$ where $w_i = 1/\sigma_i$.
   Denote the accepted parameters as $\theta_1^t, \ldots, \theta_N^t$ and the corresponding distances as $d_1^t, \ldots, d_N^t$.
4. Calculate $w_i^t$ for $1 \leq i \leq N$ as follows. If $t = 1$ let $w_i^1 = 1$. Otherwise let $w_i^t = \pi(\theta_i^t)/q_t(\theta_i^t)$.
5. Increment $t$, let $h_t$ be the $\alpha$ quantile of the $d_i^t$ values and return to step 2.
```

3.2 Illustration

As an illustration, Figure 1 shows the difference between using Euclidean and weighted Euclidean distance with $w_i = 1/\sigma_i$ within ABC-rejection. Here σ_i is calculated using MAD.

For both distances the acceptance threshold is tuned to accept half the simulations. In this example Euclidean distance mainly rejects simulations where s_1 is far from its observed value: it is dominated by this summary. Weighted Euclidean distance also rejects simulations where s_2 is far from its observed value and is less stringent about s_1.

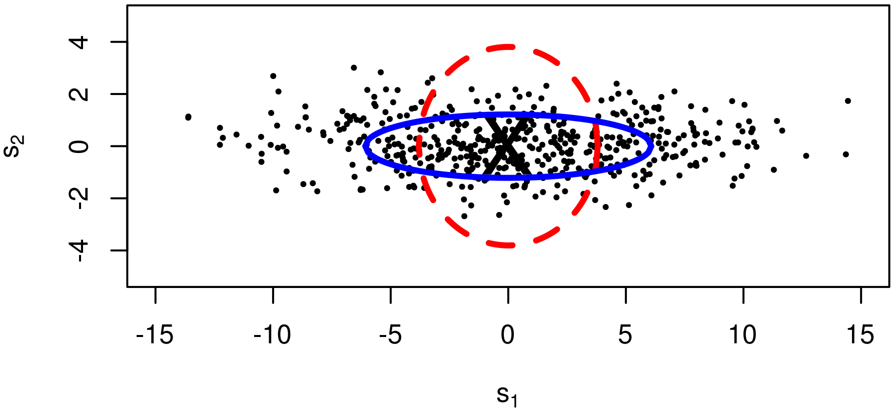


Figure 1: An illustration of distance functions in ABC rejection sampling. The points show simulated summary statistics s_1 and s_2. The observed summary statistics are taken to be $(0, 0)$ (black cross). Acceptance regions are shown for two distance functions, Euclidean (red dashed circle) and Mahalanobis (blue solid ellipse). These show the sets within which summaries are accepted. The acceptance thresholds have been tuned so that each region contains half the points.

Which of these distances is preferable depends on the relationship between the summaries and the parameters. For example if s_1 were the only informative summary, then Euclidean distance would preferable. In practice, this relationship may not be known. Weighted Euclidean distance is then a sensible choice as both summary statistics contribute to the acceptance decision.

This heuristic argument supports the use of weighted Euclidean distance in ABC more generally. One particular case is when low dimensional informative summary statistics have been selected, for example by the methods reviewed in Blum et al. (2013). In this situation all summaries are known to be informative and should contribute to the acceptance decision.

Note that in Figure 1 the observed summaries $\boldsymbol{s}_{\text{obs}}$ lie close to the centre of the set of simulations. When some observed summaries are hard to match by model simulations this is not the case. ABC distances could now be dominated by the summaries which are hardest to match. How to weight summaries in this situation is discussed in Section 6.

4 Methods: Sequential ABC with an adaptive distance

The previous section discussed normalising ABC summary statistics using estimates of their scale under the prior predictive distribution. This prevents any summary statistic dominating the acceptance decision in ABC-rejection or the first iteration of Algorithm 3, where the simulations are generated from the prior predictive. However in later iterations of Algorithm 3 the simulations may be generated from a very different distribution so that this scaling is no longer appropriate. This section presents a version of ABC-PMC which avoids this problem by updating the distance function at each iteration. Normalisation is now based on the distribution of summary statistics generated in the current iteration. The proposed algorithm is presented in Section 4.1.

An approach along these lines has the danger that the summary statistic acceptance regions at each iteration no longer form a nested sequence of subsets converging on the point $\boldsymbol{s} = \boldsymbol{s}_{\text{obs}}$. To avoid this, the proposed algorithm only accepts a simulated dataset at iteration t if it also meets the acceptance criteria of *every previous iteration*. This can be viewed as sometimes modifying the ith distance function to take into account information from previous iterations. Section 4.2 discusses convergence in more depth.

4.1 Proposed algorithm

Algorithm 4 is the proposed algorithm. An overview is as follows. Iteration t draws parameters from the current importance distribution and simulates corresponding datasets. These are used to construct the tth distance function. The best N simulations are accepted and used to construct the next importance distribution.

A complication is deciding how many simulations to perform in each iteration. This should continue until N are accepted. However the distance function defining the acceptance rule is not known until *after* the simulations are performed. The solution implemented is to continue simulating until $M = \lceil N/\alpha \rceil$ simulations pass the acceptance rule of the previous iteration. Let $\mathcal{A}$ be the set of these simulations and $\mathcal{B}$ be the others. Next the new distance function is constructed (based on $\mathcal{A} \cup \mathcal{B}$) and the N with lowest distances (from $\mathcal{A}$) are accepted. The tuning parameter α has a similar interpretation to the corresponding parameter in Algorithm 3: the acceptance threshold in iteration t is the α quantile of the realised distances from simulations in $\mathcal{A}$.

Usings this approach means that, as well as adapting the distance function, another difference with Algorithm 3 is that selection of h_t is delayed from the end of iteration $t-1$ to part-way through iteration t (and therefore h_1 does not need to be specified as a tuning choice.) If desired, this novelty can be used without adapting the distance function. This variant algorithm was tried on the examples of this paper, but the results are omitted as performance is closely comparable to Algorithm 3.

Storing all simulated $\boldsymbol{s}^*$ vectors to calculate scale estimates in step 3 of Algorithm 4 can be impractical. In practice storage is stopped after the first few thousand simulations, and scale estimation is done using this subset. The remaining details of Algorithm 4 – the choice of perturbation kernel K_t and the rule to terminate the algorithm – are implemented as described earlier for ABC-PMC.

4.2 Convergence

This section shows that conditions for the convergence of Algorithm 4 in practice are essentially those described in Section 2.3 for standard ABC-PMC plus one extra requirement: $e_t = \frac{\max_i w_i^t}{\min_i w_i^t}$ is bounded above.

In more detail, conditions ensuring convergence of Algorithm 4 can be taken from The-

Algorithm 4 ABC-PMC with adaptive h_t and $d^t(\cdot, \cdot)$

```
    Initialisation
1. Let $t = 1$.
    Main loop
2. Repeat following steps until there are $M = \lceil N/\alpha \rceil$ acceptances.
    (a) If $t = 1$ sample $\theta^*$ from $\pi(\theta)$. Otherwise sample $\theta^*$ from importance density $q_t(\theta)$ given in equation (2).
    (b) If $\pi(\theta^*) = 0$ reject and return to (a).
    (c) Sample $\boldsymbol{y}^*$ from $\pi(\boldsymbol{y}|\theta_i^*)$ and calculate $\boldsymbol{s}^* = S(y^*)$.
    (d) If $t = 1$ accept. Otherwise accept if $d^i(\boldsymbol{s}^*, \boldsymbol{s}_{\text{obs}}) \leq h_i$ for all $i < t$.
    Denote the accepted parameters as $\theta_1^*, \ldots, \theta_M^*$ and the corresponding summary vectors as $\boldsymbol{s}_1^*, \ldots, \boldsymbol{s}_M^*$.
3. Calculate $(\sigma_1^t, \sigma_2^t, \ldots)$, a vector of MADs for each summary statistic, calculated from all the simulations in step 2 (including those rejected).
4. Define $d^t(\cdot, \cdot)$ as the distance (3) using weights $(w_i^t)_{1 \leq i \leq m}$ where $w_i^t = 1/\sigma_i^t$.
5. Calculate $d_i^* = d^t(\boldsymbol{s}_i^*, \boldsymbol{s}_{\text{obs}})$ for $1 \leq i \leq M$.
6. Let $h_t$ be the $N$th smallest $d_i^*$ value.
7. Let $(\theta_i^t)_{1 \leq i \leq N}$ be the $\theta_i^*$ vectors with the smallest $d_i^*$ values (breaking ties randomly).
8. Let $w_i^t = \pi(\theta_i^t)/q_t(\theta_i^t)$ for $1 \leq i \leq N$.
9. Increment $t$ and return to step 2.
```

orem 1 in Appendix A. These are the same as those given for other ABC-PMC algorithms in Section 2.3 with the exception that the acceptance region A_t is now defined as $\{\boldsymbol{s} | d_i(\boldsymbol{s}, \boldsymbol{s}_{\text{obs}}) \leq h_i \text{ for all } i \leq t\}$. Two conditions behave differently under this change: C4 and C5.

Condition C4 states that $\lim_{t\to\infty} |A_t| = 0$ i.e. Lebesgue measure tends to zero. The definition of A_t ensures $|A_t|$ is decreasing in t. However it may not converge to zero. Reasons for this are the same as why condition C4 can fail for standard ABC-PMC, as described in Section 2.3.

Condition C5 is bounded eccentricity (defined in Appendix A) of the A_t sets. Under distance (3) this can easily be seen to correspond to e_t having an upper bound. This is not guaranteed by Algorithm 4, but it can be imposed, for example by updating w_i^t to $w_i^t + \delta \max_i w_i^t$ after step 4 for some small $\delta > 0$. However this was not found to be necessary in any of the examples of this paper.

5 Examples

This section presents three examples comparing the proposed algorithm with existing ABC-PMC algorithms: a simple illustrative normal model, the g-and-k distribution and the Lotka-Volterra model.

5.1 Normal distribution

Suppose there is a single parameter θ with prior distribution $N(0, 10^2)$. Let $s_1 \sim N(\theta, 0.1^2)$ and $s_2 \sim N(0, 1^2)$ independently. These are respectively informative and uninformative summary statistics. Let $s_{\text{obs},1} = s_{\text{obs},2} = 0$.

Figures 2 and 3 illustrate the behaviour of ABC-PMC for this example using Algorithms 2 and 4. For ease of comparison the algorithms use the same random seed, and the distance function and first threshold value h_1 for Algorithm 2 are specified to be those produced in the first iteration of Algorithm 4. The effect is similar to making a short preliminary run of ABC-rejection to make these tuning choices. Both algorithms use $N = 2000$ and $\alpha = 1/2$.

Under the prior predictive distribution the MAD for s_1 is in the order of 100 while that for s_2 is in the order of 1. Therefore the first acceptance region in Figure 2 is a wide ellipse. Under Algorithm 2 (left panel) the subsequent acceptance regions are smaller ellipses with the same shape and centre. The acceptance regions for Algorithm 4 (right panel) are similar for the first two iterations. After this, enough has been learnt about θ that the simulated summary statistics have a different distribution, with a reduced MAD for s_1. Hence s_1 is given a larger weight, while the MAD and weight of s_2 remain roughly unchanged. Thus the acceptance regions change shape to become narrower ellipses, which results in a more accurate estimation of θ under Algorithm 4, as shown by the comparison of mean squared errors (MSEs) in Figure 3.

5.2 g-and-k distribution

The g-and-k distribution is a popular test of ABC methods. It is defined by its quantile function:

$$A + B\left[1 + c\frac{1-\exp(-gz(x))}{1+\exp(-gz(x))}\right][1+z(x)^2]^k z(x), \tag{4}$$

where $z(x)$ is the quantile function of the standard normal distribution. Following the literature (Rayner and MacGillivray, 2002), $c = 0.8$ is used throughout. This leaves (A, B, g, k) as unknown parameters.

The g-and-k distribution does not have a closed form density function making likelihood-based inference difficult. However simulation is straightforward: sample $x \sim \text{Unif}(0,1)$ and substitute into (4). The following example is taken from Drovandi and Pettitt (2011b). Suppose a dataset is 10,000 independent identically distributed draws from the g-and-k distribution and the summary statistics are a subset of the order statistics: those with indices $(1250, 2500, \ldots, 8750)$. (As in Fearnhead and Prangle, 2012, a fast method is used

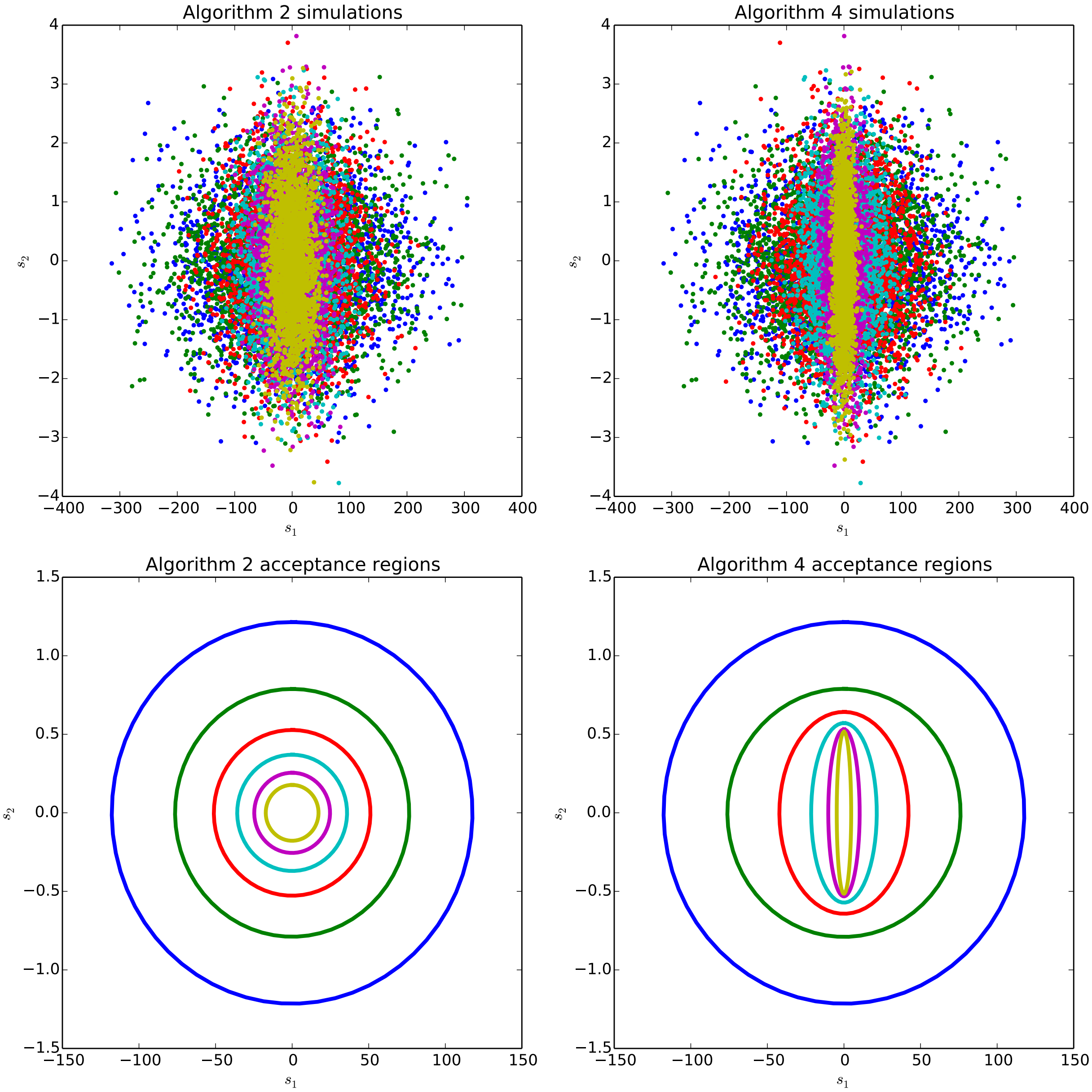


Figure 2: An illustration of ABC-PMC for a simple normal model using either Algorithm 2 (non-adaptive distance function) or Algorithm 4 (adaptive distance function). *Top row:* simulated summary statistics (including rejections) *Bottom row:* acceptance regions (note different scale to top row). In both rows colour indicates the iteration of the algorithm.

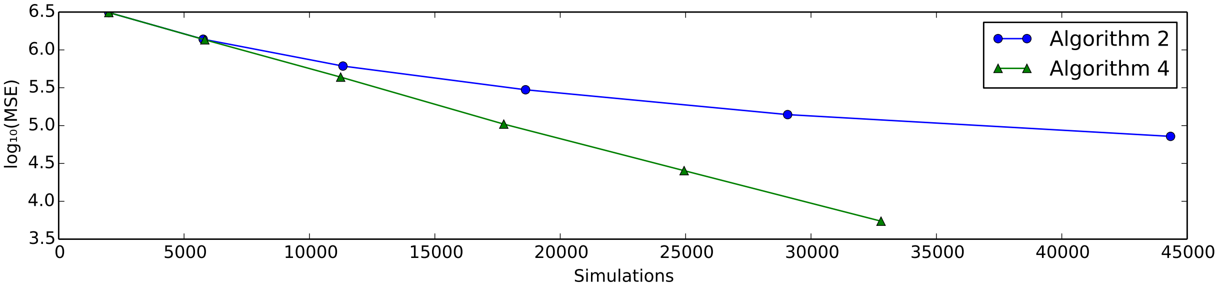


Figure 3: Mean squared error of the parameter for Algorithms 2 and 4 on a simple normal example.

to simulate these order statistics without sampling an entire dataset.) The parameters are taken to have independent Unif$(0, 10)$ priors.

To use as observations, 100 datasets were simulated from the prior predictive distribution. Each was analysed using Algorithms 3 and 4. Each analysis used a total of 10^6 simulations and tuning parameters $N = 1000$ and $\alpha = 1/3$.

Table 1 shows root mean squared errors for the output of both algorithms, averaged over all the observed datasets. These show that Algorithm 4 is more accurate overall for every parameter.

	A	B	g	k
Algorithm 3	0.468	0.544	1.048	0.172
Algorithm 4	0.079	0.364	0.584	0.135

Table 1: Root mean squared errors of each parameter in the g-and-k example, averaged over analyses of 100 simulated datasets.

Figures 4 and 5 show more detail for a particular observed dataset, which has been simulated under parameter values $(3, 1, 1.5, 0.5)$. Figure 4 shows the estimated MSE of each parameter for each iteration of both algorithms. Algorithm 4 performs better throughout for the g and k parameters. Also, after roughly $150,000$ simulations Algorithm 4 has a significant advantage for these parameters and B, and similar performance to Algorithm 3 for A.

Figure 5 shows some of the distance function weights produced by the algorithms. Algorithm 3 places low weights on the most extreme order statistics, as they are highly variable in the prior predictive distribution. This is because the prior places significant weight upon parameter values producing very heavy tails. However by the last iteration of Algorithm 4, such parameter values have been ruled out. The algorithm therefore assigns larger weights which provide access to the informational content of these statistics.

5.3 Lotka-Volterra model

The Lotka-Volterra model describes two interacting populations. In its original ecological setting the populations represent predators and prey. However it is also a simple example of biochemical reaction dynamics of the kind studied in systems biology. This section concentrates on a stochastic Markov jump process version of this model with state $(X_1, X_2) \in \mathbb{Z}^2$ representing prey and predator population sizes. Three transitions are possible:

$$\begin{aligned}
&(X_1, X_2) \to (X_1 + 1, X_2) && \text{(prey growth)} \\
&(X_1, X_2) \to (X_1 - 1, X_2 + 1) && \text{(predation)} \\
&(X_1, X_2) \to (X_1, X_2 - 1) && \text{(predator death)}
\end{aligned}$$

These have hazard rates $\theta_1 X_1$, $\theta_2 X_1 X_2$ and $\theta_3 X_2$ respectively. Simulation is straightforward by the Gillespie method. Following either a transition at time t, or initiation at $t = 0$, the time to the next transition is exponentially distributed with rate equal to the sum of the hazard rates at time t. The type of the next transition has a multinomial distribution with probabilities proportional to the hazard rates. For more background see for example Owen et al. (2015), from which the following specific inference problem is taken.

The initial conditions are taken to be $X_1 = 50, X_2 = 100$. A dataset is formed of observations at times 0, 2, 4, $\ldots$, 32. Both X_1 and X_2 are observed plus independent $N(0, \sigma^2)$ errors, where σ is fixed at $\exp(2.3)$. The unknown parameters are taken to be $\log(\theta_1), \log(\theta_2)$ and $\log(\theta_3)$. These are given independent $\text{Unif}(-6, 2)$ priors. The vector of all 34 noisy observations is used as the ABC summary statistics.

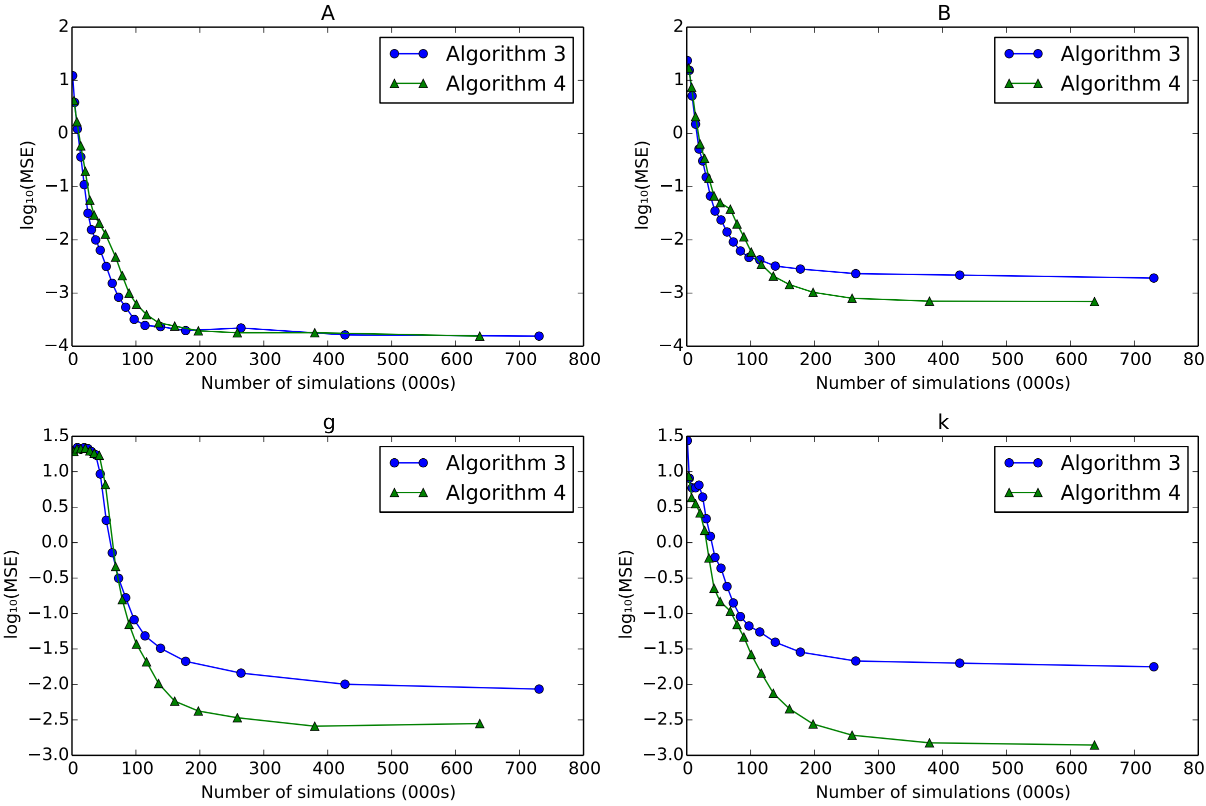


Figure 4: Mean squared error of each parameter from Algorithms 3 and 4 for the *g*-and-*k* example.

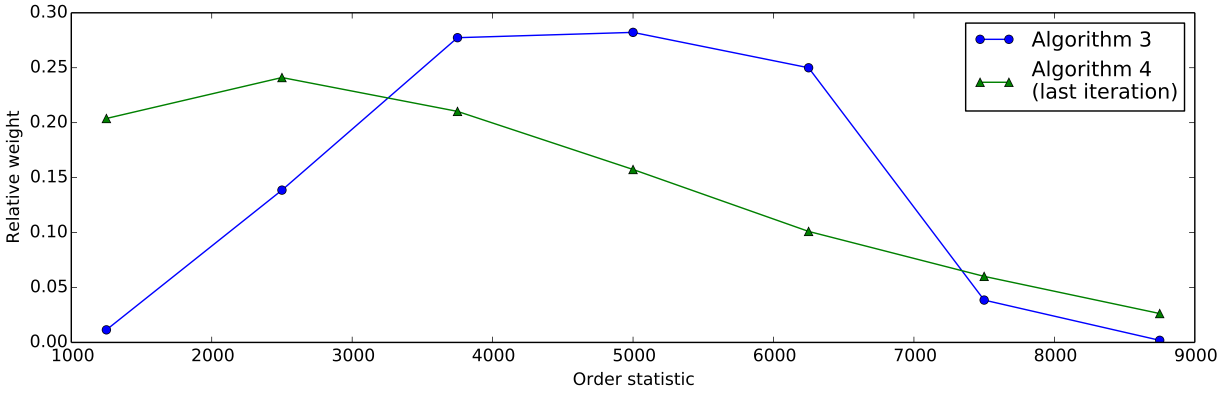


Figure 5: Summary statistic weights used in Algorithms 3 and 4 for the *g*-and-*k* example, rescaled to sum to 1.

A single simulated dataset is analysed (shown in Figure 8.) This is generated from the model with $\theta_1 = 1, \theta_2 = 0.005, \theta_3 = 0.6$. ABC analysis was performed using Algorithms 3 and 4. A total of $50,000$ simulations were used by both algorithms. The tuning parameters are $N = 200$ and $\alpha = 1/2$. Any Lotka-Volterra simulation reaching $100,000$ transitions is terminated and automatically rejected. This avoids extremely long simulations, such as exponential prey growth if predators die out. These incomplete simulations are excluded from the MAD calculations, but this should have little effect as they are rare.

Figure 6 shows the MSEs resulting from the two analyses. Algorithm 4 has smaller errors for all parameters after roughly 10,000 simulations. Figure 7 shows the weights produced in Algorithm 3 and the final weights produced in Algorithm 4, which are clearly very different. Figure 8 explains this by showing a sample of simulated datasets on which these weights are based. Under the prior predictive distribution, at least one population usually quickly becomes extinct, illustrating that the prior distribution concentrates on the wrong system dynamics and so is unsuitable for choosing distance weights for later iterations of the algorithm.

6 Discussion

This paper has presented an ABC-PMC algorithm with an adaptive distance function. Compared to standard ABC-PMC, the algorithm requires no extra simulations and has similar convergence properties. Several examples have been shown where the new algorithm improves performance. This is because in each example the scale of the summary statistics varies significantly between prior and posterior predictive distributions. This section discussions several possibilities to extend this work.

Several variations on ABC-PMC have been proposed in the literature. The adaptive distance function idea introduced here can be used in most of these. This is particularly simple for ABC model choice algorithms (e.g. Toni et al., 2009). Here, instead of proposing θ^* values

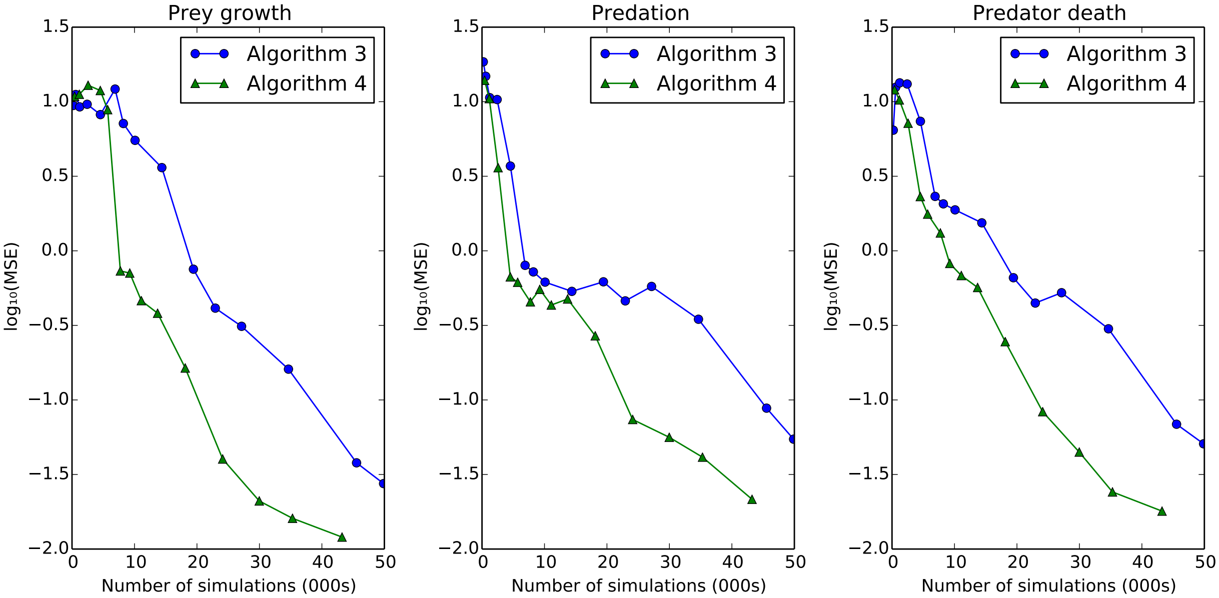


Figure 6: Mean squared error of each parameter from ABC-PMC output for Lotka-Volterra example.

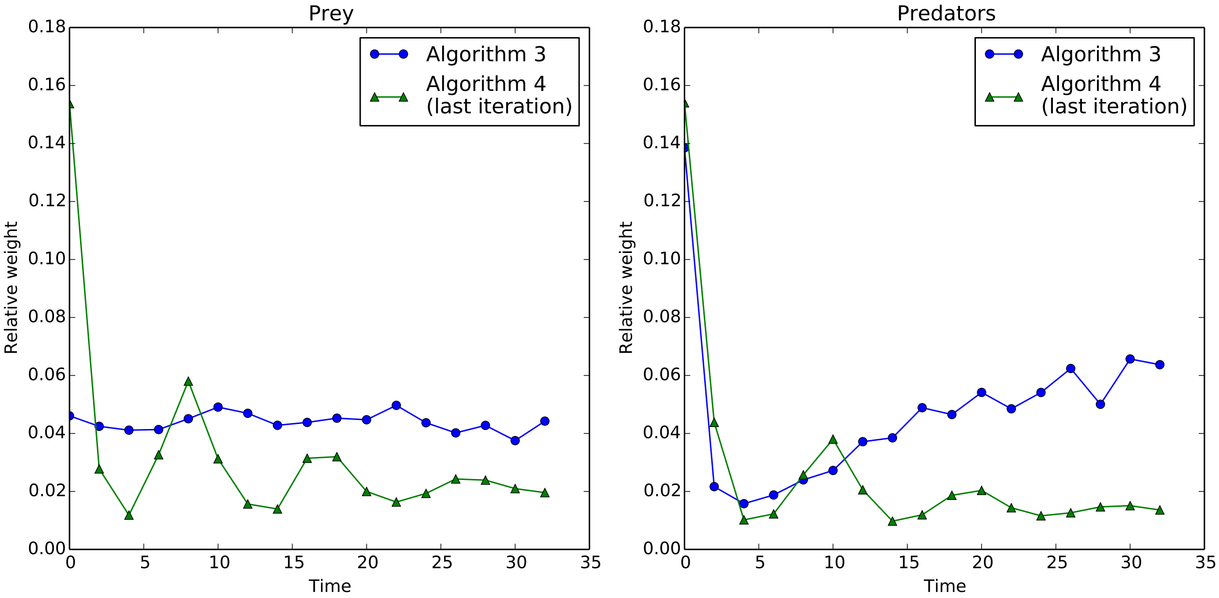


Figure 7: Summary statistic weights used in ABC-PMC for Lotka-Volterra example, rescaled to sum to 1.

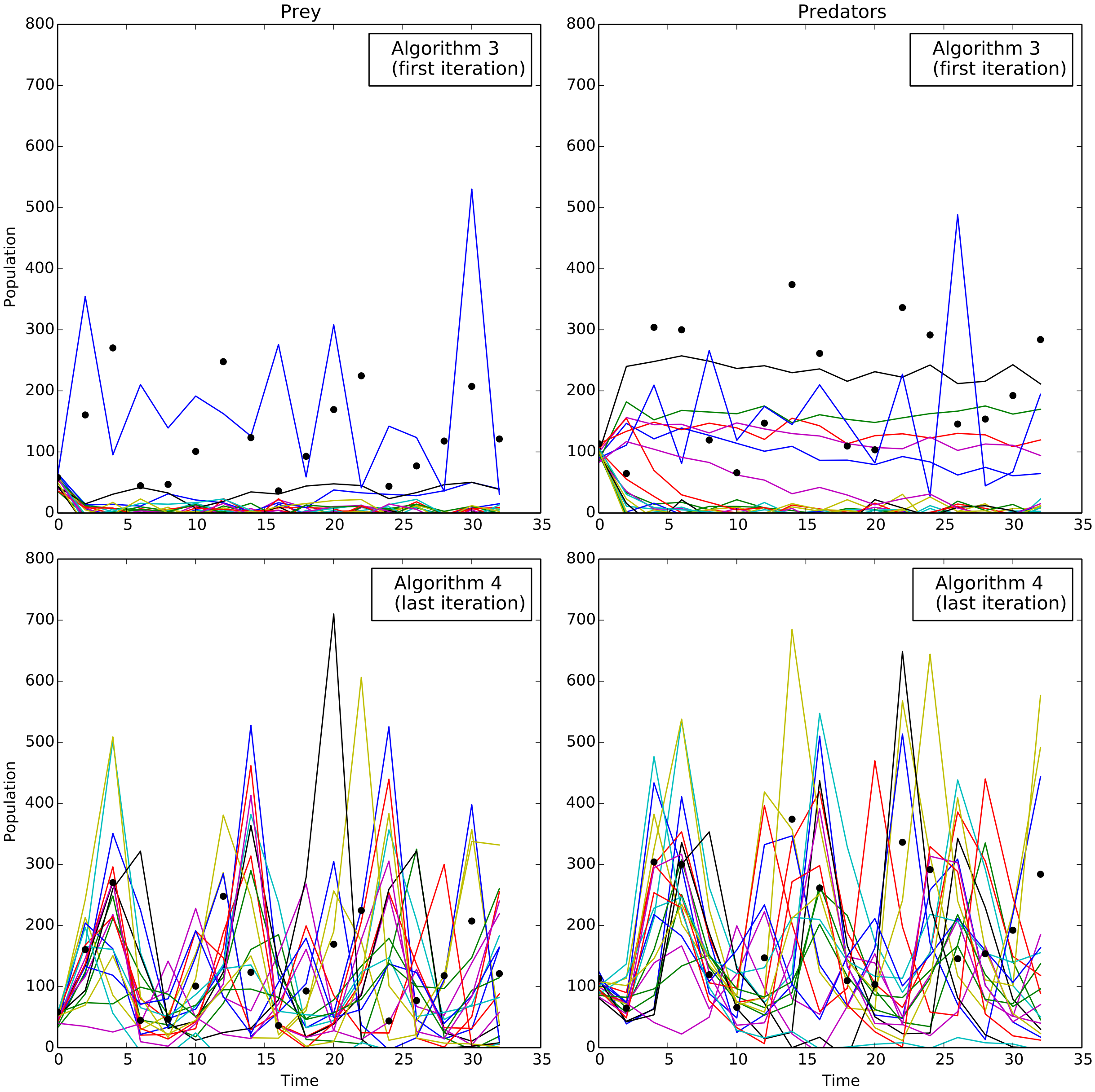


Figure 8: Observed dataset (black points) and samples of 20 simulated datasets (coloured lines) for the Lotka-Volterra example. The top row shows simulations from step 2 of the first iteration of Algorithm 3. The bottom row shows simulations from step 2 of the last iteration of Algorithm 4. These are representative examples of the simulations used to select the weights shown in Figure 7.

from an importance density, (m^*, θ^*) pairs are proposed, where m^* is a model indicator. This could be implemented in Algorithm 4 while leaving the other details unchanged. Drovandi and Pettitt (2011a), Del Moral et al. (2012) and Lenormand et al. (2013) propose ABC-SMC algorithms which update the population of $(\theta, \boldsymbol{s})$ pairs between iterations in different ways to ABC-PMC. In all of these it seems possible to first simulate the required new summary statistic vectors, then use these to update the distance function and threshold values, and finally make acceptance decisions to produce the new population. Note that some of these variations would require additional convergence results to those given in Appendix A.

Several aspects of Algorithm 4 could be modified. One natural alternative is to use distance functions $d^t(\boldsymbol{x}, \boldsymbol{y}) = \left[(\boldsymbol{x} - \boldsymbol{y})^T W^t (\boldsymbol{x} - \boldsymbol{y})\right]^{1/2}$ where W^t is an estimate of the precision matrix from all the simulations in step 2 at iteration t. Exploratory work with this choice found it did not improve performance for the examples in this paper. The weighted Euclidean distance function (3) is preferred for this reason, and also because its weights are easier to interpret and there are more potential numerical difficulties in estimating a precision matrix.

Another reason it may be desirable to modify the distance function (3) is if some summary statistic, say s_i, has an observed value far from most simulated values. In this case $|s_{\text{obs},i} - s_i|$ can be much larger than σ_i, and so s_i can dominate the ABC distance used in this paper. It is tempting to downweight s_i so that the others summaries can also contribute. Finding a good way to do this without ignoring s_i altogether is left for future work.

Algorithm 4 updates the distance function at each iteration. There may be scope for similarly updating other tuning choices. It is particularly appealing to try to improve the choice of summary statistics as the algorithm progresses (as suggested by Barnes et al., 2012.) Summary statistics could be selected at the same time as the distance function based on the same simulations, for example by a modification of the regression method of Fearnhead and Prangle (2012). Further work would be required to ensure the convergence of such an algorithm.

Acknowledgements Thanks to Michael Stumpf and Scott Sisson for helpful discussions. This work was completed while the author was supported by a Richard Rado postdoctoral fellowship from the University of Reading.

A Convergence of ABC-PMC algorithms

Algorithm 5 is an ABC importance sampling algorithm. This appendix considers a sequence of these algorithms. Denote the acceptance threshold and distance function in the tth element of this sequence as h_t and $d^t(\cdot,\cdot)$. The ABC-PMC algorithms in this paper can be viewed as sequences of this form with specific choices of how h_t and d^t are selected. Note ABC-rejection is a special case of Algorithm 5 with $g(\theta) = \pi(\theta)$, so this framework can also investigate its convergence as $h \to 0$.

Algorithm 5 ABC importance sampling

```
1. Sample $\theta_i^*$ from density $g(\theta)$ independently for $1 \leq i \leq N$.
2. Sample $\boldsymbol{y}_i^*$ from $\pi(\boldsymbol{y}|\theta_i^*)$ independently for $1 \leq i \leq N$.
3. Calculate $\boldsymbol{s}_i^* = S(\boldsymbol{y}_i^*)$ for $1 \leq i \leq N$.
4. Calculate $d_i^* = d(\boldsymbol{s}_i^*, \boldsymbol{s}_{\text{obs}})$.
5. Calculate $w_i^* = \pi(\theta_i^*)/g(\theta_i^*)$ (where $\pi(\theta)$ is the prior density)
6. Return $\{(\theta_i^*, w_i^*) | d_i^* \leq h\}$.
```

The output of importance sampling is a weighted sample $(\theta_i, w_i)_{1 \leq i \leq P}$ for some value of P. A Monte Carlo estimate of $E[h(\theta)|\boldsymbol{s}_{\text{obs}}]$ for an arbitrary function $h(\cdot)$ is then $\frac{\sum_{i=1}^P h(\theta)_i w_i}{\sum_{i=1}^P w_i}$. For large P this asymptotically equals (as shown in Prangle, 2011 for example) the expectation under the following density:

$$\pi_{\text{ABC},t}(\theta|\boldsymbol{s}_{\text{obs}}) \propto \int \pi(\boldsymbol{s}|\theta)\pi(\theta)\mathbb{1}[d_t(\boldsymbol{s}, \boldsymbol{s}_{\text{obs}}) \leq h_t] d\boldsymbol{s},$$

known as the ABC posterior.

Theorem 1. *Under conditions C1-C5,* $\lim_{t\to\infty} \pi_{ABC,t}(\theta|\boldsymbol{s}_{obs}) = \pi(\theta|\boldsymbol{s}_{obs})$ *for almost every choice of* $(\theta, \boldsymbol{s}_{obs})$ *(with respect to the density* $\pi(\theta, \boldsymbol{s})$*).*

The conditions are:

C1. $\theta \in \mathbb{R}^n$, $\boldsymbol{s} \in \mathbb{R}^m$ for some m, n and these random variables have density $\pi(\theta, \boldsymbol{s})$ with respect to Lebesgue measure.

C2. The sets $A_t = \{\boldsymbol{s}|d_t(\boldsymbol{s}, \boldsymbol{s}_{\text{obs}}) \leq h_t\}$ are Lebesgue measurable.

C3. $\pi(\boldsymbol{s}_{\text{obs}}) > 0$.

C4. $\lim_{t\to\infty} |A_t| = 0$ (where $|\cdot|$ represents Lebesgue measure.)

C5. The sets A_t have *bounded eccentricity*.

The definition of bounded eccentricity is that for any A_t, there exists a set $B_t = \{\boldsymbol{s} \mid ||\boldsymbol{s} - \boldsymbol{s}_{\text{obs}}||_2 \leq r_t\}$ such that $A_t \subseteq B_t$ and $|A_t| \geq c|B_t|$, where $||.||$ denotes the Euclidean norm and $c > 0$ is a constant.

Proof. Observe that:

$$\begin{aligned}
\lim_{t\to\infty} \pi_{\text{ABC}}(\theta|\boldsymbol{s}_{\text{obs}}) &= \lim_{t\to\infty} \frac{\int \pi(\theta, \boldsymbol{s})\mathbb{1}(\boldsymbol{s} \in A_t)d\boldsymbol{s}}{\int \pi(\theta, \boldsymbol{s})\mathbb{1}(\boldsymbol{s} \in A_t)d\boldsymbol{s}d\theta} \\
&= \lim_{t\to\infty} \frac{\int_{\boldsymbol{s}\in A_t} \pi(\theta, \boldsymbol{s})d\boldsymbol{s}}{\int_{\boldsymbol{s}\in A_t} \pi(\boldsymbol{s})d\boldsymbol{s}} \\
&= \frac{\lim_{t\to\infty} \frac{1}{|A_t|} \int_{\boldsymbol{s}\in A_t} \pi(\theta, \boldsymbol{s})d\boldsymbol{s}}{\lim_{t\to\infty} \frac{1}{|A_t|} \int_{\boldsymbol{s}\in A_t} \pi(\boldsymbol{s})d\boldsymbol{s}} \\
&= \frac{\pi(\theta, \boldsymbol{s}_{\text{obs}})}{\pi(\boldsymbol{s}_{\text{obs}})} \quad \text{almost everywhere} \\
&= \pi(\theta|\boldsymbol{s}_{\text{obs}}).
\end{aligned}$$

The third and fourth equalities follow by l'Hôpital's rule and the Lebesgue differentiation theorem respectively. The latter theorem requires conditions C4 and C5. For more details of it see Stein and Shakarchi (2009) for example.

References

Barnes, C. P., Filippi, S., and Stumpf, M. P. H. (2012). Contribution to the discussion of Fearnhead and Prangle (2012). *Journal of the Royal Statistical Society: Series B*, 74:453.

Beaumont, M. A. (2010). Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution and Systematics*, 41:379–406.

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. (2009). Adaptive approximate Bayesian computation. *Biometrika*, pages 2025–2035.

Beaumont, M. A., Zhang, W., and Balding, D. J. (2002). Approximate Bayesian computation in population genetics. *Genetics*, 162:2025–2035.

Bezanson, J., Karpinski, S., Shah, V. B., and Edelman, A. (2012). Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*.

Biau, G., Cérou, F., and Guyader, A. (2015). New insights into approximate Bayesian computation. *Annales de l'Institut Henri Poincaré (B) Probabilités et Statistiques*, 51(1):376–403.

Blum, M. G. B., Nunes, M. A., Prangle, D., and Sisson, S. A. (2013). A comparative review of dimension reduction methods in approximate Bayesian computation. *Statistical Science*, 28:189–208.

Bonassi, F. V. and West, M. (2015). Sequential Monte Carlo with adaptive weights for approximate Bayesian computation. *Bayesian Analysis*, 10(1):171–187.

Cappé, O., Guillin, A., Marin, J.-M., and Robert, C. P. (2004). Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4).

Csilléry, K., Blum, M. G. B., Gaggiotti, O., and François, O. (2010). Approximate Bayesian computation in practice. *Trends in Ecology & Evolution*, 25:410–418.

Csilléry, K., François, O., and Blum, M. G. B. (2012). abc: an R package for approximate Bayesian computation (ABC). *Methods in Ecology and Evolution*, 3:475–479.

Del Moral, P., Doucet, A., and Jasra, A. (2012). An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020.

Drovandi, C. C. and Pettitt, A. N. (2011a). Estimation of parameters for macroparasite population evolution using approximate Bayesian computation. *Biometrics*, 67(1):225–233.

Drovandi, C. C. and Pettitt, A. N. (2011b). Likelihood-free Bayesian estimation of multivariate quantile distributions. *Computational Statistics & Data Analysis*, 55(9):2541–2556.

Fearnhead, P. and Prangle, D. (2012). Constructing summary statistics for approximate Bayesian computation: Semi-automatic ABC. *Journal of the Royal Statistical Society, Series B*, 74:419–474.

Lenormand, M., Jabot, F., and Deffuant, G. (2013). Adaptive approximate Bayesian computation for complex models. *Computational Statistics*, 28(6):2777–2796.

Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180.

McKinley, T., Cook, A. R., and Deardon, R. (2009). Inference in epidemic models without likelihoods. *The International Journal of Biostatistics*, 5(1).

Owen, J., Wilkinson, D. J., and Gillespie, C. S. (2015). Likelihood free inference for Markov processes: a comparison. *Statistical applications in genetics and molecular biology*, 14(2):189–209.

Prangle, D. (2011). *Summary statistics and sequential methods for approximate Bayesian computation*. PhD thesis, Lancaster University.

Rayner, G. D. and MacGillivray, H. L. (2002). Numerical maximum likelihood estimation for the g-and-k and generalized g-and-h distributions. *Statistics and Computing*, 12(1):57–75.

Sedki, M., Pudlo, P., Marin, J.-M., Robert, C. P., and Cornuet, J.-M. (2012). Efficient learning in ABC algorithms. *arXiv preprint arXiv:1210.1388*.

Silk, D., Filippi, S., and Stumpf, M. P. H. (2013). Optimizing threshold-schedules for sequential approximate Bayesian computation: applications to molecular systems. *Statistical applications in genetics and molecular biology*, 12(5):603–618.

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2009). Correction: Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 106(39):16889–16890.

Stein, E. M. and Shakarchi, R. (2009). *Real analysis: measure theory, integration, and Hilbert spaces*. Princeton University Press.

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