Adaptive ABC distance functions

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

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 es

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 $_i$ values determined in the rst 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 $_i$ values at each iteration to appropriate values. It is demonstrated that this algorithm pro-

2 Approximate Bayesian Computation

This section sets out the necessary background on ABC algorithms. Several review papers (e.g. Beaumont, 2010; Csillery 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 $\pi(y|\theta)$. Let $\pi(\theta)$ be the prior density and y_{obs} represent the observed data. It is assumed that $\pi(y|\theta)$ 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 $\pi(\theta|y)$. It requires several tuning choices: number of simulations N, a threshold $h \geq 0$, a function $S(y)$ mapping data to a vector of summary statistics, and a distance function $d(\cdot;\cdot)$.

Algorithm 1 ABC-rejection

1. Sample θ_i from $\pi(\theta)$ independently for $1 \leq i \leq N$.
2. Sample y_i from $\pi(y|\theta_i)$ independently for $1 \leq i \leq N$.
3. Calculate $s_i = S(y_i)$ for $1 \leq i \leq N$.
4. Calculate $d_i = d(s_i; s_{\text{obs}})$ (where $s_{\text{obs}} = S(y_{\text{obs}})$.)
5. Return $\{\theta_i | d_i \leq h\}$.

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 (Cappe et al., 2004), so it is referred to here as ABC-PMC. The algorithm involves a sequence of thresholds, $(h_t)_{t\ 1}$. Similarly to h in ABC-rejection, this can be speci ed 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 from ( ). Otherwise sample from importance density $q_t(\ )$ given in equation (2).
   (b) If ( ) = 0 reject and return to (a).
   (c) Sample y from $(y j\ _i)$ and calculate $s = S(y)$.
   (d) Accept if $d(s\ ; s_{\text{obs}})$ h_t.

   Denote the accepted parameters as $\ ^t_1; : : : ;\ ^t_N$.
3. Calculate w_i^t for 1 i N as follows. If $t = 1$ let $w_i^1 = 1$. Otherwise let $w_i^t = (\ ^t_i)=q_t(\ ^t_i)$.
4. Increment t and return to step 2.

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

$$q_t(\ ) = \sum_{i=1}^{N} w_i^{t\ 1} K_t(\ j\ ^{t\ 1}_i) = \sum_{i=1}^{N} w_i^{t\ 1}: \tag{2}$$

Drawing from this e ectively samples from the previous weighted population and perturbs

the result using kernel K_t

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

C2. The sets $A_t = \{\theta : \rho(s, s_{\text{obs}}) \leq h_t\}$ are Lebesgue measurable.

C3. $\pi(s_{\text{obs}}) > 0$.

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 s_{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 s_{obs} cannot be produced under the model of interest (in which case C2 generally also fails.) Alternatively, even if s_{obs} can occur under the model, the algorithm may converge on importance densities on 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 $h_t \to$

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:

$$d(x;y) = \left(\sum_{i=1}^{m} \{w_i(x_i - y_i)\}^2 \right)^{1/2}. \quad (3)$$

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 $d(x;y) = \sum_{i=1}^{m} w_i|x_i - y_i|$. 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 $w_i = 1/\tau_i$ where τ_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 τ_i is the empirical standard deviation of the ith summary statistic under the prior predictive distribution. Csillery 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 $_i$ 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 modi ed 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 modi cation.)

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

Initialisation

1. Let $t = 1$ and $h_1 = \ 1$.

Main loop

2. Repeat following steps until there are N acceptances.
   (a) If $t = 1$ sample from ( ). Otherwise sample from importance density $q_t(\ )$ given in equation (2).
   (b) If ( ) = 0 reject and return to (a).
   (c) Sample y from $(y j\ _i)$ and calculate $s\ = S(y\ )$.
   (d) Accept if $d(s\ ; s_{\text{obs}})$ h_t (if $t = 1$ always accept).
3. If $t = 1$:
   (a) Calculate ($_1$; $_2$;: : :), a vector of MADs for each summary statistic, calculated from all the simulations in step 2 (including those rejected).
   (b) De ne $d(\ ;\ )$ as the distance (3) using weights $(w_i)_{1\ \ i\ \ m}$ where $w_i = 1=\ _i$.

   Denote the accepted parameters as $\ ^t_1; : : : ;\ ^t_N$ and the corresponding distances as $d^t_1; : : : ; d^t_N$.
4. Calculate w^t_i for $1 \quad i \quad N$ as follows. If $t = 1$ let $w^1_i = 1$. Otherwise let $w^t_i = (\ ^t_i)=q_t(\ ^t_i)$.
5. Increment t, let h_t be the quantile of the d^t_i values and return to step 2.

3.2 Illustration

As an illustration, Figure 1 shows the di erence between using Euclidean and weighted Euclidean distance with $w_i = 1=\ _i$ within ABC-rejection. Here $_i$ 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 s_{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 $s = s_{obs}$. 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 de ning the acceptance rule is not known until *after* the simulations are performed. The solution implemented is to continue simulating until $M = dN= e$ simulations pass the acceptance rule of the previous iteration. Let A be the set of these simulations and B be the others. Next the new distance function is constructed (based on $A [B$) and the N with lowest distances (from A) are accepted. The tuning parameter has a similar interpretation to the corresponding parameter in Algorithm 3: the acceptance threshold in iteration t is the quantile of the realised distances from simulations in A.

Usings this approach means that, as well as adapting the distance function, another di erence 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 speci ed 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 s vectors to calculate scale estimates in step 3 of Algorithm 4 can be impractical. In practice storage is stopped after the rst 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 $d^t(\ ;\ )$

Initialisation

1. Let $t = 1$.

Main loop

2. Repeat following steps until there are $M = dN = e$ acceptances.

   (a) If $t = 1$ sample from ( ). Otherwise sample from importance density $q_t(\ )$ given in equation (2).

   (b) If ( ) = 0 reject and return to (a).

   (c) Sample y from $(y j\ _i)$ and calculate $s = S(y)$.

   (d) If $t = 1$ accept. Otherwise accept if $d^i(s\ ; s_{\text{obs}})$ h_i for all $i < t$.

   Denote the accepted parameters as $_1;:::;\ _M$ and the corresponding summary vectors as $s_1;:::;s_M$.

3. Calculate $(\ ^t_1;\ ^t_2;:::)$, a vector of MADs for each summary statistic, calculated from all the simulations in step 2 (including those rejected).

4. De ne $d^t(\ ;\ )$ as the distance (3) using weights $(w^t_i)_{1\ \ i\ \ m}$ where $w^t_i = 1 = \ ^t_i$.

5. Calculate $d_i = d^t(s_i\ ; s_{\text{obs}})$ for $1 \quad i \quad M$.

6. Let h_t be the Nth smallest d_i value.

7. Let $(\ ^t_i)_{1\ \ i\ \ N}$ be the $_i$ vectors with the smallest d_i values (breaking ties randomly).

8. Let $w^t_i = (\ ^t_i)=q_t(\ ^t_i)$ for $1 \quad i \quad N$.

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 de ned as $fsjd_i(s; s_{\text{obs}})$ h_i for all i

the rst iteration of Algorithm 4. The e ect is similar to making a short preliminary run of ABC-rejection to make these tuning choices. Both algorithms use $N = 2000$ and $= 1=2$.

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 rst 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 rst two iterations. After this, enough has been learnt about that the simulated summary statistics have a di erent 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 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 de ned by its quantile function:

$$A + B \quad 1 + c\frac{1 \quad \exp(\quad gz(x))}{1 + \exp(\quad gz(x))} \quad [1 + z(x)^2]^k z(x); \tag{4}$$

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 di cult. However simulation is straightforward: sample x 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 $(1250; 2500; : : : ;$

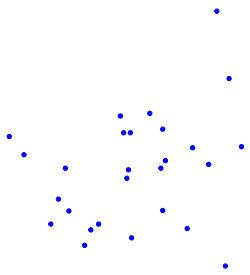

Figure 2: An illustration of ABC-PMC for a simple normal model using either Algorithm 2 (non-adaptive distance function) or Algorithm 4 (adaptive distance function). *Top row:* simulated summary statistics (including rejections) *Bottom row:* acceptance regions (note di erent scale to top row). In both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

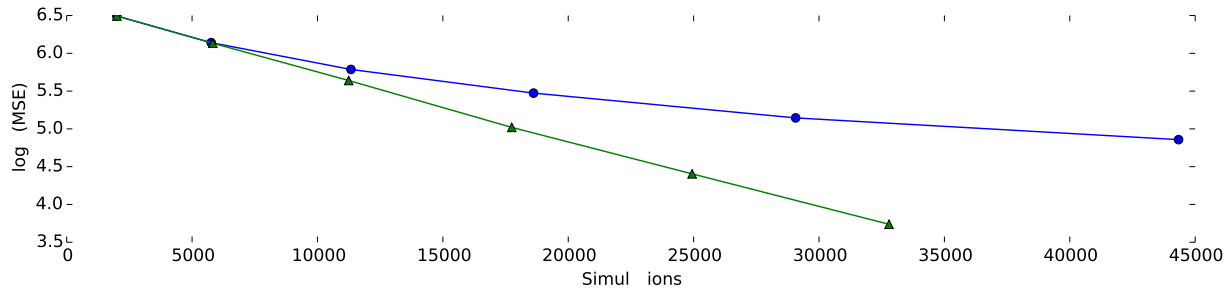

Figure 3: Mean squared error of the parameter for Algorithms 2 and 4 on a simple normal example.

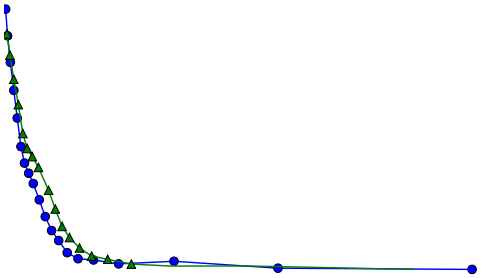

Figure 4: Mean squared error of each parameter from Algorithms 3 and 4 for the *g*-and-*k* example.

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

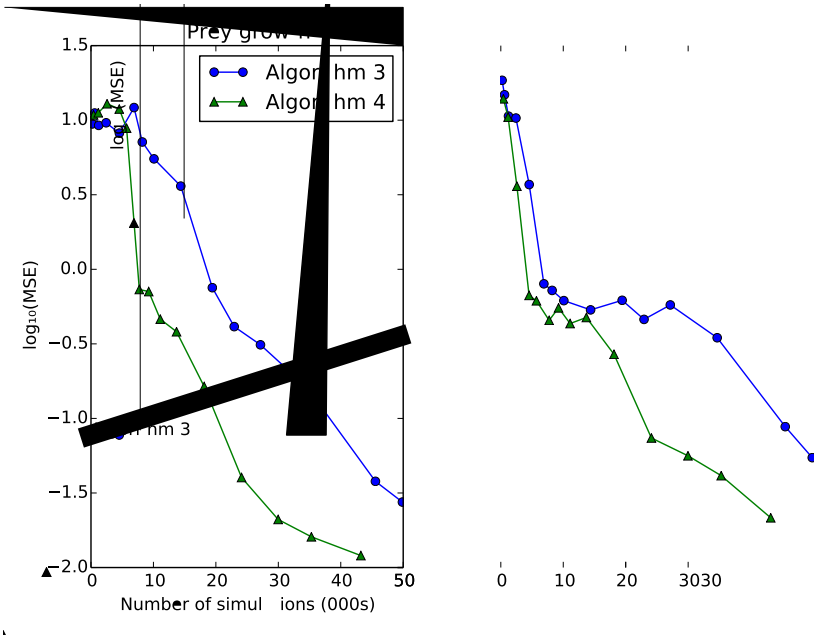

Figure 6: Mean squared error of each parameter from ABC-PMC output for Lotka-Volterra example.

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

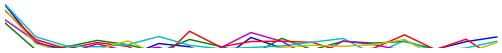

Figure 8: Observed dataset (black points) and samples of 20 simulated datasets (coloured lines) for the Lotka-Volterra example. The top row shows simulations from step 2 of the first iteration of Algorithm 3. The bottom row shows simulations from step 2 of the last iteration of Algorithm 4. These are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

from an importance density, (m ;) pairs are proposed, where m is a model indicator. This could be implemented in Algorithm 4 while leaving the other details unchanged. Drovandi

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 $d^t(\ ;\ )$. The ABC-PMC algorithms in this paper can be viewed as sequences of this form with speci c choices of how h_t and d^t are selected. Note ABC-rejection is a special case of Algorithm 5 with $g(\ ) = \ (\ )$, so this framework can also investigate its convergence as $h \ ! \ 0$.

Algorithm 5 ABC importance sampling

1. Sample $_i$ from density $g(\ )$ independently for $1 \quad i \quad N$.
2. Sample y_i from $(yj\ _i)$ independently for $1 \quad i \quad N$.
3. Calculate $s_i = S(y_i)$ for $1 \quad i \quad N$.
4. Calculate $d\cdot$

Theorem 1. *Under conditions C1-C5,* $\lim_{t \to \infty} \mathbb{E}_{\cdot,t}(\cdot|s_{obs}) = \pi(\cdot|s_{obs})$ *for almost every choice of* $(\cdot; s_{obs})$ *(with respect to the density* $\pi(\cdot; s)$*).*

The conditions are:

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

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

C3. $\pi(s_{obs}) > 0$.

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 $B_t = \{s : ||s - s_{obs}||_2 \leq r_t\}$ such that $A_t \subseteq B_t$ and $|A_t| \geq c|B_t|$, where $||\cdot||$ denotes the Euclidean norm and $c > 0$ is a constant.

Proof of $\pi(\theta|ABC, s_{obs})$

The third and fourth equalities follow by l'Hôpital's rule and the Lebesgue di erentiation 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., Cerou, F., and Guyader, A. (2015). New insights into approximate Bayesian computation. *Annales de l'Institut Henri Poincare (B) Probabilites 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.

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

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

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

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 appro