Approximate Bayesian Computing

Brian Sheridan

August 3, 2023

Abstract

Bayesian statistics asserts that probability is a degree of belief, mathematically described by Bayes' Theorem. This equation requires a likelihood function, which describes the probability of data given some parameters, a function whose form is often analytically intractable or expensive to compute. Approximate Bayesian Computing bypasses the need for this function by generating synthetic data using given parameters in order to construct a posterior distribution. This is explored in the cases of discrete and continuous experiment output spaces.

1 Introduction

In mathematics and science, we are interested in making predictions on future events and wish to describe the *probability* of certain outcomes to happen, meaning how likely the outcomes of an event are to happen. In general, when an event is subject to random chance, there are two interpretations of what probability is, the *objectivist* and the *subjectivist* views. The frequentist version of objectivism assigns a number to probability, while the Bayesian version of subjectivism, first formulated by Thomas Bayes in 1763, assigns to a probability a distribution, or a degree of belief, using Bayes' Theorem

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}, \tag{1}$$

where, for example, $P(A|B)$ denotes the conditional probability of A given B. Bayes' theorem can also be rewritten as

$$p(\theta|D, M) = \frac{\mathcal{L}(D|\,\theta)\,\pi(\theta)}{\mathcal{Z}}, \tag{2}$$

where p is the posterior distribution of the parameters given the data according to a specific model M, $\mathcal{L}$ is the likelihood of the data given some parameters, π states the prior beliefs for the parameters and $\mathcal{Z} = P(D|M)$ is the probability of the data given the model and called the *evidence*, which is an overall normalisation factor.

1.1 Approximate Bayesian Computing

Often times it is difficult or impossible to find a tractable expression for the likelihood function in equation (2), which describes the probability of data, or outcomes, given a set of parameters, θ. It may also become impractical to compute it computationally. We then turn to methods of *simulation based inference* in order to obtain the posterior distribution in equation (2). Although we may not be able to analytically derive the function $\mathcal{L}(D|\,\theta)$ we may be able to sample values $d \sim \mathcal{L}(\cdot|\,\theta)$. In this manner, *Approximate Bayesian Computing* (ABC) bypasses the requirement of deriving a likelihood function and simulates the data generating process in order to obtain a posterior distribution [1]. The simplest version of Approximate Bayesian Computing is *Rejection ABC*, where data generated by parameters chosen at random from the prior distributions are compared to the observed data, and accepted or rejected up to a certain tolerance. Either the data itself or *summary statistics* of the data, which compress the data into a single or small number of values to avoid the curse of dimensionality, are compared with observed data via a distance metric. This distance is compared to a certain acceptance tolerance, which then determines whether the parameters which generated the synthetic data are accepted or rejected. The algorithm is described as follows, assuming the use of summary statistics.

Rejection ABC Algorithm

1. Sample parameters, θ, from the predetermined prior distributions.
2. Using these parameters, generate data, x_{gen}, based on the given model.
3. Calculate the summary statistic(s) for the generated and observed data, $y_{gen} = S(x_{gen})$ and $y_{obs} = S(x_{obs})$.
4. Compare the summary statistic(s) using a given distance metric and accept the parameters θ if the distance is less than or equal to a given threshold, $\rho(y_{gen}, y_{obs}) \leq \epsilon$, otherwise reject the parameters.
5. Repeat the procedure for the desired number of samples. The accepted parameters will approximate the posterior distribution.

The posterior distribution obtained by ABC is approximate since we accept samples up to the tolerance set by ϵ. Care must be taken to choose the tolerance ϵ neither too small nor too large. If it is too small very few samples will be accepted, while if it is too large the accuracy of the posterior diminishes. We can investigate ABC methods in more detail with the aid of numerical methods, which is the aim of this

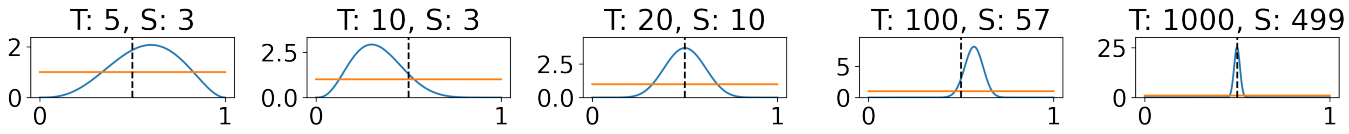

Figure 1: The analytical posterior distribution for a fair coin being flipped. The orange line shows the uniform flat prior for the bias of the coin while the dashed vertical is the true value of $\theta = 0.5$. The number of trials and number of successes (e.g. heads) is shown for each plot. The posterior distribution peaks and narrows at the true value as more data is given.

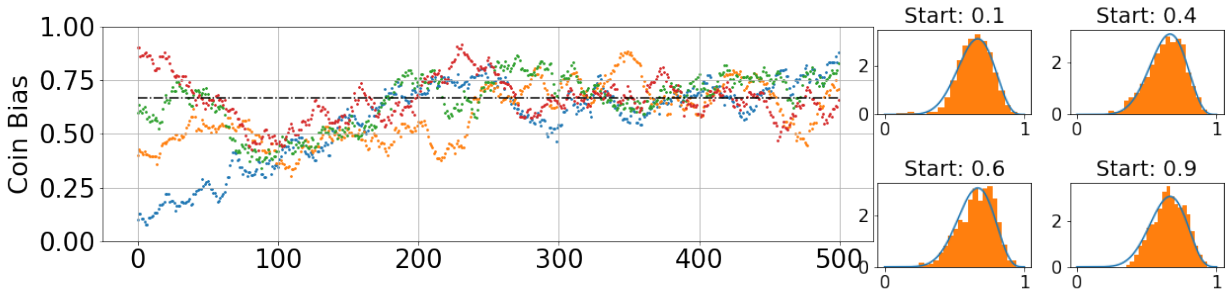

Figure 2: The Metropolis-Hastings algorithm run for the coin flipping experiment with observed data of 12 trials and 8 successes. A manually chosen "scale" of 0.025 from SCIPY.STATS.NORM is used in the proposal Gaussian. The chains of length 5000 converge to the same region and oscillate around the true value. Only the first 500 chain samples are shown. The obtained posteriors from the different initial points are very close to the analytical result.

report. In general, the results of an experiment in Bayesian analyses may be discrete or continuous and both cases are examined in this report.

2 Discrete Bayesian Statistics

Discrete Bayesian statistics involve experiments for which the possible outputs take discrete values. This leads to the output space of the distance function being discrete. We start with the simple and instructive example of flipping a coin, then move onto an interesting case of state switching.

2.1 Coin Flipping

The possible outcomes of a coin flipping experiment are heads or tails, which we can numerically denote by 1 and 0 - a discrete output space with two outcomes. For a fair coin, the probability of heads is $\theta = 0.5$, which becomes different for a biased coin. If we did not know *a-priori* the bias of the coin, we could perform a coin flipping experiment, flipping the coin n times and noting the resulting number of "successes", meaning heads. This would allow us to construct a posterior distribution for the bias of the coin. An experiment with n trials and k successes with success probability p is described exactly by a *binomial distribution*, which we can use for the likelihood function in equation (2). The probability mass function of the binomial distribution is given as

$$\Pr(k; n, p) = \binom{n}{k} p^k (1-p)^{n-k}, \tag{3}$$

describing the expected number of successes given a specific value for the bias of the coin and the number of experiments.

The prior distribution can be chosen as the beta distribution, with probability density function given by

$$\beta(\theta; a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \theta^{a-1}(1-\theta)^{b-1} = B(a, b)\, \theta^{a-1}(1-\theta)^{b-1}, \tag{4}$$

where a and b are the *hyperparameters*, which are the parameters of the prior distribution, as a distinction to the underlying model parameter θ. We choose this distribution since it is the conjugate to the binomial distribution and it is also capable of describing different distributions by varying the two parameters. In the context of Bayesian statistics, given a certain likelihood function, a prior distribution can be chosen from the distributions in the same family as the likelihood, a so-called conjugate prior, which results in the posterior distribution having the same form as the prior distribution [2]. In this example, the resulting posterior is therefore in the form of a beta distribution. The added benefit is that choosing $a = b = 1$ for our prior beta distribution results in a flat uniform distribution. To obtain the analytical form of the posterior, the normalising evidence must be computed, the marginalisation of the likelihood, namely

$$\mathcal{Z} = P(D) = \int d^N \theta \mathcal{L}(D|\,\theta) P(\theta). \tag{5}$$

which, upon insertion of equations (3) and (4) into equation (5), returns the posterior as

$$p(\theta; k, n, a, b) = \frac{\theta^{k+a-1}(1-\theta)^{n-k+b-1}}{B(k+a,\, n-k+b)}. \tag{6}$$

Figure 1 shows analytical solutions in the case of a fair coin being flipped. Although an analytically tractable solution

has been found, we generally are not able to find this and must *sample* from the posterior in order to construct the full posterior distribution.

2.1.1 Posterior Sampling - Conventional MCMC

The Metropolis-Hastings algorithm is a popular and powerful algorithm to sample from the posterior distribution. An example of Markov-Chain-Monte-Carlo sampling (MCMC), it takes an initial state, proposes new states from a specific distribution and accepts or rejects the proposed states based on a certain acceptance criterion. In this way, a chain is created which approximates the posterior distribution in the case of Bayesian statistics.

The Metropolis-Hastings algorithm is applied to the case of the coin flipping experiment. A new state is proposed from a Gaussian distribution, centred on the current state with a manually chosen variance. The ratio between the unnormalised posterior distribution values for the new and current states, given observed data of 12 trials and 8 successes, is calculated and the transition accepted if the ratio is larger than a randomly chosen number between 0 and 1. Figure 2 shows the chain and resulting posteriors for different initial states for the bias of the coin. We can see that each chain converges towards the same region and oscillates around the true value. There exists, however, a *burn-in* time for the chains, which is the time it takes to reach the region of high probability density. Instead of discarding the first few steps of the chain, we can combat this by using *simulated annealing*, which is an optimisation algorithm, inspired by metallurgy, which uses a temperature dependent acceptance probability for proposed states to optimise a function and place the initial state in a region of high-density, in a sense a way of doing maximum a-posteriori (MAP) [3]. This is particularly useful for high-dimensional or multi-modal distributions.

Simulated Annealing Algorithm

1. Set initial state θ_0 with energy E.
2. Propose a new state θ_n with energy E_n randomly, using a Gaussian centred on the current state.
3. Accept or reject the new state θ_n if the temperature dependent probability
$$P(E, E_n, T) = \begin{cases} 1, & (E_n < E) \\ \exp\left(-(E_n - E)/T\right), & (E_n > E) \end{cases}$$
is greater than or less than a random number drawn from a uniform distribution between 0 and 1.
4. Repeat for desired number of iterations, decreasing the temperature each time until it reaches a minimum at the last iteration and the process "cools" to the optimum value.

In this example, the energy is the value of the negative of the likelihood function, or equivalently the negative of the posterior distribution, since we use flat priors. We also use a linearly decreasing temperature function over iterations steps. We therefore locate the maximum by minimising the negative. Performing the Metropolis-Hastings MCMC with simulated annealing allows us to construct the posterior distribution without discarding any initial steps. From the sampling we obtain 0.65 ± 0.12 for the value of theta, using the initial simulated annealing estimate of $\theta = 0.69$, a good agreement with the true value of 0.67. Figure 4 shows the resultant posterior distribution.

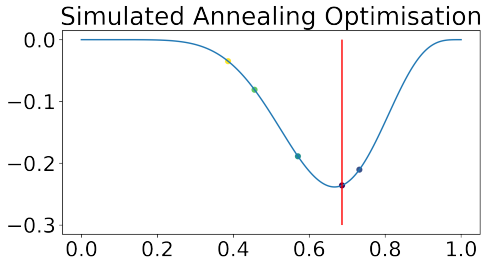

Figure 3: Simulated Annealing optimises the negative of the posterior which results in finding the maximum. The brighter colours are selected earlier iteration steps while the red line indicates the set end value of $\theta = 0.69$.

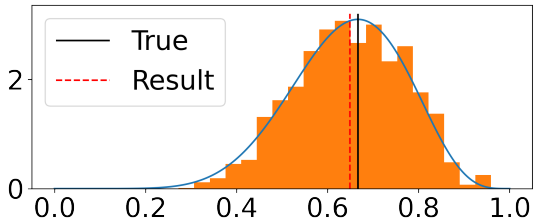

Figure 4: The MCMC sampling of the posterior distribution using the simulated annealing initial guess. The posterior result is $\theta = 0.65 \pm 0.12$.

2.1.2 Posterior Sampling - ABC

The ABC rejection algorithm, as described in the introduction, can be used to obtain the posterior distribution of the coin flipping experiment. Firstly, an appropriate distance function needs to be defined for the algorithm. In the case of the coin flipping experiment, the normalised absolute value of the difference of the sums of each array of data is used, i.e. the sequence of ones and zeros, denoting heads or tails, respectively. Namely,

$$\rho(x_{obs}, x_{gen}) = \left(\left|\sum x_{obs} - \sum x_{gen}\right|\right)/n, \qquad (7)$$

where n is the number of coin flips. In this sense, the summary statistic is simply the sum of the coin values for heads or tails. This is equivalent to directly comparing the data, since there are no negative values for the possible outcomes, and is therefore a sufficient summary statistic. Figure 5 shows the result of this ABC procedure. Firstly, we note that higher coin biases favour lower distances, which agrees with the observed data of 8 successes from 12 trials. The two cases of a perfect and imperfect threshold are shown, with $\epsilon = 0$ and $\epsilon = 0.2$ respectively. A perfect acceptance threshold is only possible in the discrete case and yields the correct

posterior distribution. The posterior becomes flatter for a higher value of the acceptance threshold, which makes sense, since if we keep increasing the acceptance threshold, we will end up accepting all proposed values. Since we are drawing from a flat prior, we will end up with a flat, uniform posterior. The case of $\epsilon = 0$ yields a posterior value $\theta = 0.64 \pm 0.12$, and for the case of $\epsilon = 0.2$ a value of $\theta = 0.64 \pm 0.16$.

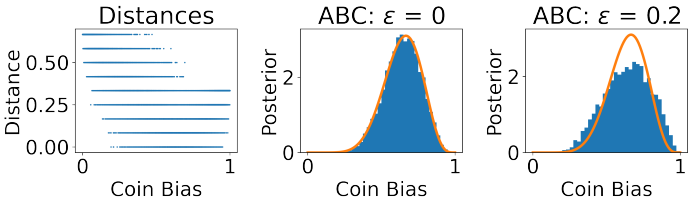

Figure 5: Using ABC, the distance is calculated for each prior sample. The posteriors for perfect and imperfect thresholds are shown, with $\theta = 0.64 \pm 0.12$ for the perfect case and $\theta = 0.64 \pm 0.16$ for the imperfect case. In both cases, 5000 samples are taken.

2.2 Hidden Markov Models

Another interesting case of discrete experiments is *Hidden Markov Models* (HMM) where the underlying parameter is unobservable [4]. An observable process is however influenced by the unobserved parameter, and inference is made about the parameter based on the observed process. An example is of a dynamic bistable HMM, used in biological mathematics, where the system is composed of two states, `A` and `B`, whereby the state has a certain switching probability, θ, and a certain probability of correct measurement, γ. In this sense, states may be measured incorrectly and thus *noise* is introduced into the bistable system. The underlying laws governing state switching may be very complicated, especially in the context of biology and genetic sequencing, making it difficult to obtain a likelihood function for the process. However, analysing the system through the lens of ABC bypasses the need for a likelihood function. We assume the true parameters of the system to be $\theta = 0.25$ and $\gamma = 0.8$. Although an exact solution can be found using the *Viterbi* algorithm [5], a synthetic sequence of states is generated as observed data using $\theta = 0.25$ with which we can perform ABC to discern the transition probability θ without knowing it beforehand, assuming we know that $\gamma = 0.8$.

Although the switching probability is not directly observable, the resulting sequence is, and we can choose the *switching frequency*, ω, as our summary statistic, which is the number of times the state switches in a given sequence. So given the sequence `AABBABB`, we get $\omega = 3$. Therefore, the distance metric can be calculated as

$$\rho(x_{obs}, x_{gen}) = |\,\omega_{obs} - \omega_{gen}\,|. \tag{8}$$

For convenience, we start each sequence with `A`, while the sequences are of length 100, with 5000 trials for the ABC investigation. We set the acceptance threshold as $\epsilon = 0$ in search of the true posterior for this discrete system. Figure 6 shows the results of the ABC, with $\theta = 0.12 \pm 0.07$ for the case of $\gamma = 0.8$, which is a sharp departure from the known $\theta = 0.25$.

The reason for this is that the summary statistic used is insufficient, meaning that the switching frequency does not capture all of the necessary detail about the system and thus leads to inaccurate determination of the posterior. The imperfect probability of correct measurement introduces noise, or an extra degree of freedom, into the system, which isn't captured by the switching frequency. In the same figure, setting $\gamma = 1$ results in $\theta = 0.26 \pm 0.04$ which is a great match with theory, and for contrast setting $\gamma = 0.1$ gives a wildly incorrect value of $\theta = 0.77 \pm 0.06$.

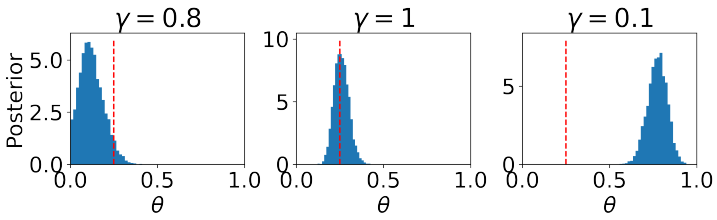

Figure 6: The posterior distributions of the switching probability θ for the bistable HMM for different values of γ, the probability of correct measurement. The correct posterior is obtained by setting the measurements to be perfect, while imperfect measurements introduce an extra degree of freedom giving an incorrect posterior. The red lines show the true switching probability of 0.25.

3 Continuous Bayesian Statistics

We can now shift our attention to continuous output spaces, where experiment outcomes may take values from a continuous range. In the context of ABC, the output space of the distance metric is continuous. Consider the sterile case of an observed data point of $y_{obs} = 2$ and the data generating function of

$$\mathcal{L}(\theta) = \mathcal{N}(\sqrt{\theta}, 0.25), \tag{9}$$

namely a Gaussian with mean $\mu = \sqrt{\theta}$ and constant variance $\sigma^2 = 0.25$. The observed y_{obs} suggests a true $\theta = 4$. With ABC we can investigate the true value of θ which gave the observed data, assuming we didn't know it beforehand. To perform this, we assume a uniform prior distribution for θ from 0 to 10. Data are generated (randomly sampled) from equation (9), $y \sim \mathcal{L}(\theta)$, and the resulting statistic y_{gen} compared to observation y_{obs} simply using the absolute distance, $\rho(x_{obs}, x_{gen}) = |\,y_{obs} - y_{gen}\,|$. Taking a threshold of $\epsilon = 0.1$, the results are shown in Figure 7 which gives a posterior mean of 4.19 and standard deviation of 1.03 for the value of θ. Using the error propagation formula

$$\mu = \theta^n \;\Rightarrow\; \Delta\mu = n\theta^{n-1}\Delta\theta, \tag{10}$$

we find, for $n = 1/2$, the mean to be $\mu = 2.05 \pm 0.25$, quite close to the true value of 2.

Note that an acceptance threshold of zero would not have given a posterior distribution since the probability of obtaining an exact number in the continuous case is zero, $p(y) = 0$.

Rather we need a finite slice of ϵ since $p(y)\,dy \neq 0$. However, how would the results change if another threshold value of ϵ had been chosen? We can repeat the above investigation of calculating the posterior mean for a certain ϵ, then take the difference with the true value of $\theta = 4$ and repeat for a wide range of thresholds. The results are presented in Figure 8. Firstly, we note that smaller values of ϵ give better agreements with observation, as expected. The data eventually saturate to 1 at about $\epsilon = 2$. This occurs since all prior samples are accepted at this threshold, so further increasing ϵ would not change anything. Since the prior samples are uniform over a domain from 0 to 10, the expected value is thus 5, which is a difference of 1 away from the true value of 4, resulting in the given saturated difference value.

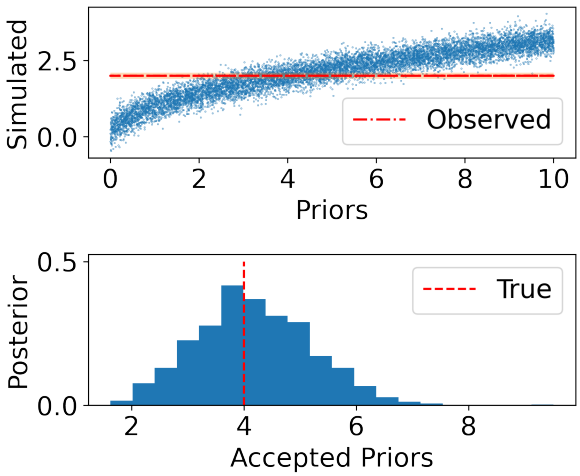

Figure 7: The posterior distribution for θ gives 4.19 ± 1.03, with 10000 prior samples taken and a threshold of $\epsilon = 0.1$. There is strong agreement with the true value of $\theta = 4$.

Before saturation, the difference grows beyond 1 at a hump around $\epsilon = 1.16$. This occurs due to the square-root growth of θ as opposed to the linear growth of ϵ. At the minimum end of the prior range, the "spine" of the simulated data lies at $\sqrt{0} = 0$, a distance of 2 from the observed value. At the maximum of the prior range, the spine rests at $\sqrt{10} = 3.16$, a distance of 1.16 from the observed value. Therefore, by the time $\epsilon = 1.16$, all of the prior samples θ_i with $y_i > 4$ are consumed, while there are still unaccepted samples for smaller θ. This shifts the mean of the posterior to a higher value of θ. As ϵ is further increased, more samples of lower θ are accepted, reducing the posterior mean and the difference with observation until saturation. During the simulations, it was noticed that the approximate posterior mean obtained by ABC was always higher than the observed 4, and this reasoning explains why. The samples of θ which give a higher y are consumed quicker than those smaller, therefore always skewing the approximate posterior to higher values. If instead of $\mu \sim \sqrt{\theta}$ we had $\mu \sim f(\theta) = 0.1\theta + 0.04\theta^2$, samples of lower θ would be consumed quicker since $f(0) = 0$ and $f(10) = 5$ compared to the observed $y_{obs} = 2$, thus always skewing the distribution to be lower. In this sense, it would be recommended to always try to represent ABC data linearly, such that we do not run into these complications, for example plotting exponential data on a logarithm plot to linearise it. This may also necessitate sampling the priors from a corresponding non-uniform distribution.

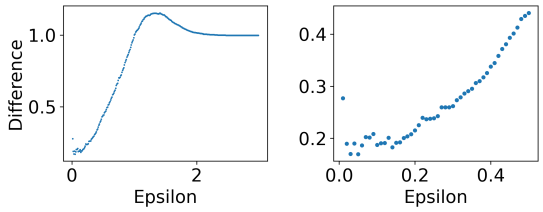

Figure 8: The ABC posterior is generated and the corresponding mean calculated for each threshold value and the difference taken with the true observed value of $\theta = 4$. The difference between the posterior θ and true θ grows as a function of the threshold until it peaks, then reduces slightly and saturates.

4 Conclusions

Approximate Bayesian Computing has shown to be a viable method of sampling the posterior distribution within a Bayesian framework, in both cases of a discrete or continuous output space. There exist, however, limitations. A weakness with ABC lies in the fact that finite acceptance tolerances lead to *approximate* posterior distributions, $p(\theta|\,\rho(y_{obs}, y_{gen}) \leq \epsilon)$ rather than $p(\theta|\,y)$. In this way bias is introduced into the problem. This bias may however be interpreted as "noise" in the summary statistics, leading to *noisy ABC*. Also, summary statistics are often necessary to avoid the curse of dimensionality, however, insufficient summary statistics reduce the precision of the likelihood, leading to incorrect posteriors. Although low-dimensional, "minimal" summary statistics are optimum, their existence is not guaranteed. It may be difficult to find sufficient summary statistics [6].

As a variation of rejection ABC, *regression adjustment* can be used as a more sophisticated version. After choosing prior samples, θ^i and computing associated summary statistics, a weighting for each prior sample is computed from a statistical kernel $K_h(||s^i - s_{obs}||)$, with $K_h(||\cdot||) = K(||\cdot||/h)$, with h the bandwidth parameter. A regression model is fitted to adjust the parameters θ^i to produce a weighted sample of adjusted values. The adjustment can either be homoscedastic or heteroscedastic. In the former case, the residuals around the conditional expectation do not depend on the summary statistics, while they do in the latter. The weighted samples approximate the posterior, with the benefit of shrinking the posterior, in contrast to simple rejection ABC, which is desired since the credibility intervals, which are often wide with rejection ABC, become narrower. [7]

Circumventing these limitations and implementing the improvements may make the already powerful ABC method of posterior sampling even more powerful.

References

[1] Simon Tavaré, David J Balding, Robert C Griffiths, and Peter Donnelly. Inferring coalescence times from dna sequence data. *Genetics*, 145(2):505–518, 1997.

[2] Luis E Padilla, Luis O Tellez, Luis A Escamilla, and Jose Alberto Vazquez. Cosmological parameter inference with bayesian statistics. *Universe*, 7(7):213, 2021.

[3] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[4] Mikael Sunnåker, Alberto Giovanni Busetto, Elina Numminen, Jukka Corander, Matthieu Foll, and Christophe Dessimoz. Approximate bayesian computation. *PLoS computational biology*, 9(1):e1002803, 2013.

[5] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[6] Scott A Sisson, Yanan Fan, and Mark Beaumont. *Handbook of approximate Bayesian computation*. CRC Press, 2018.

[7] Michael GB Blum. Regression approaches for abc. *Handbook of approximate Bayesian computation*, pages 71–85, 2018.

Code

This report made heavy use of numerical methods and computational algorithms in order to investigate Bayesian statistics and Approximate Bayesian Computing. It would, however, not be suitable to show the code for all algorithms used, especially those for Metropolis-Hastings MCMC which are already well known. Therefore, the Python code for two selected algorithms of relevance is shown here, namely for Simulated Annealing and for the ABC method applied to the coin flipping experiment.

Listing 1: The simulated annealing algorithm used to find the maximum of the posterior mode for the coin flipping experiment by minimising the negative of the posterior.

```
1
2  def SimulatedAnnealing(function, init_state,
       maxSteps, Tm=1, Tc=0.05, Gauss_var=0.5, a=1,
       b=1, data=[12,8]):
3
4      i = 0
5      #Define a linearly decreasing temperature
           function.
6      def temp(k):
7          return Tm*(1 - k/maxSteps) + Tc
8
9      #Set the state value.
10     s = init_state
11     new_states = []
12
13     #Loop through iterations.
14     while i < maxSteps:
15         #Calculate temperature.
16         T = temp(i)
17
18         #Propose new state.
19         snew = np.random.normal(s, Gauss_var)
20         #Make sure the state, a probability, is
               actually a possible value.
21         if snew < 0 or snew > 1:
22             snew = np.random.normal(s, Gauss_var)
23
24         #Calculate energy difference.
25         Ediff = function(s, a, b, data) -
               function(snew, a, b, data)
26
27         #Random acceptance probability.
28         u = np.random.uniform()
29
30         #Accept the proposal up to the
               probability.
31         if np.exp(-Ediff/T) < u:
32             s = snew
33             new_states.append(s)
34         i += 1
35     #Return the optimum state and the
           successfully accepted new states.
36     return [s, new_states]
```

Listing 2: The ABC algorithm used for the coin flipping experiment. The code continues to sample from the prior beta distribution until the generated data lies within the acceptance threshold. The accepted samples form the approximate posterior distribution.

```
1  #Define the distance function as the absolute
       difference.
2  def distance_func(x,y):
3      d = (1/len(x))*abs(sum(x)-sum(y))
4      return d
5
6  #ABC sampling of the posterior.
7  def ABC(data, trials, eps): #Assumes a uniform
       prior.
8      posterior = [] #Accepted samples.
9      n = len(data)
10
11     for i in range(0, trials):
12         #Set distance above the threshold.
13         distance = eps + 1
14         #Sample theta from uniform beta
               distribution while above threshold.
15         while distance > eps:
16             theta = np.random.beta(1, 1, size=1)
                   [0]
17             #Generate data.
18             x = np.random.binomial(1, theta, n)
19             #Calculate the distance.
20             distance = distance_func(x, data)
21         #The accepted sample is added to the
               posterior results.
22         posterior.append(theta)
23     return posterior
```