1 Introduction

1.1 Scientific Motivation

Often in Biology or Psychology, complex stochastic processes are preferred over simple data generating processes because they enable us to model the salient or subtle features of a system, such as those that appear in genetics. However, the stochastic processes that appear in genetics research typically possess computationally intractable likelihood functions, which poses a great challenge for likelihood based statistical inference methods, ie, the Bayesian and Frequentist approaches. It may be possible to approach these problems in a strictly non-parametric or semi-parametric fashion, but recent work on approximate Bayesian computation have allowed researchers to shoehorn the Bayesian methodology into these inflexible studies. The most important concept in the Bayesian framework is the posterior distribution, and by Bayes Theorem, we know that this distribution is proportional to the likelihood of the data times the prior probability. Therefore, inference on models with computationally intractable likelihoods has received considerable attention in Bayesian literature, dating back to the mid 1980s.

1.2 Levels of Intractability

The likelihood function may be intractable at various levels. The most difficult scenario in mentioned above in which the data are generated by a complex stochastic process. We will now detail the other levels at which a likelihood function may be computationally or mathematically unworkable. This investigation is first attributed in Murray et al. [6].

1. A simple level of intractability is introduced by the presence of a constant in the unknown part

   $$\mathrm{P}(D|\theta) = f(D|\theta)/C$$

   Where f is the known part of the probability mass function or probability density function and C is a constant that is difficult to calculate. Strictly speaking, this problem has been solved previously by Markov Chain Monte Carlo methods. Modern sampling methods such as the Metropolis-Hastings algorithm only require that the likelihood is known up to a constant or proportionality. Thus, this situation poses no problem.

2. A moderate level of intractability occurs when the likelihood function is known to depend on an unknown function of θ:

   $$\mathrm{P}(D|\theta) = f(D|\theta)/C(\theta)$$

In this case, Murray claims that numerical integration methods may be applied to deal with the function $C(\theta)$, however numerical integration methods, especially those in high dimension, are usually ad hoc methods that lack a degree of generality. Additionally, standard MCMC algorithms do not apply to this situation, because it is not true that the likelihood functions are known up to a constant of integration. This situation may be treated by applying methods from the variational Bayes framework or the ABC framework.

3. Diggle and Gratton [5] consider statistical inference in a frequentist setting when the model is only known to be some stochastic data generating mechanism. In this setting, data may be generated conditional on some parameters, but nothing is known about the likelihoods of the data. In this case, no part of the model $\mathrm{P}(D|\theta)$ is available. Yet it will still be possible to generate data from the posterior distribution $\mathrm{P}(\theta|D)$. This is the natural setting for ABC algorithms, as it poses the greatest problem to genetic researchers.

2 Overview of ABC Methods

2.1 Basic ABC sampler

Suppose that we want to compute the posterior probability distribution of a univariate or multivariate parameter θ. A parameter value θ_i is sampled from its prior probability distribution to simulate a data set y_i for $i = 1, \ldots, n$, where n is the number or samples. At this point, a collection of summary statistics (which ideally are sufficient statistics) are computed from the data: $S(y_i)$. These summary statistics are then compared to the summary statistics of the actual data set, $S(y_0)$ using some measure of distance d. It is common to use the Euclidean distance measure, but it is possible that other measures of distance are appropriate. This is especially true if the data generated are realizations of a stochastic process over some continuous or discrete time. If the value $d(S(y_0), S(y_i))$ is judged to be sufficiently small, that is, it is below some threshold ϵ, then the value θ_i is accepted as a value from the posterior distribution. The list of accepted θ_i then form a chain (sample) from the approximate posterior distribution. The estimation of the posterior probability distribution may be improved by the use of regression techniques, see [7].

2.2 Example of ABC rejection

For convenience suppose we take a very simple model, the Weibull distribution. So we create a set of observed data, say a sample of size 20 from `rweibull` with shape parameter 2 and scale parameter 5. For summary statistics, let's take mean and standard deviation. And now we need a mechanism by which we can generate new data. For simplicity, we can take a weibull distribution, but in general we need not consider a closed form distribution. Once this is done, implementing the ABC rejection algorithm is simple: draw a large number of samples from the prior (say uniform for both of the parameters). Finally, the samples from the uniform distributions are accepted as realizations from the posterior distribution if the corresponding sample from the Weibull distribution admits a mean that is close to 2 and a standard deviation that is close to 5.

2.3 MCMC with ABC (Metropolis-Hastings)

In practice, using simulations from the prior distribution is inefficient because this does not account for the data at the proposal stage and thus can lead to proposed values located in low posterior probability regions. One solution to this problem is proposed by Marjoram et al. [4] which introduces an ABC-MCMC algorithm. This solutions is attractive for the same reason that MCMC is an attractive class of algorithms in any other setting: because correlating observations will lead to more time spent in regions of high posterior probability.

The algorithm proceeds as follows : Suppose we currently have a value θ which is assumed to have arose from the target posterior probability distribution. We propose a value θ' from a density $q(\theta, \theta')$. Now, simulate a data set D' based on the value θ' from our intractable likelihood function or stochastic process. If the D and D' are judged to be sufficiently similar, then we accept the move to the new value θ' with probability

$$\min\left(1, \frac{\pi(\theta')q(\theta', \theta)}{\pi(\theta)q(\theta, \theta')}\right)$$

The initialization of the MCMC sampler can be bypassed since the Markov chain forgets its initial state. The computational cost of the initialization is thus negligible. But we will need to run the MCMC longer to achieve convergence and omit the burn-in from the output, which also carried some computational cost. In the algorithm described above, we only need to tune a few parameters: the error tolerance ϵ, the summary statistics S, and the distance function d. In some respects, the error tolerance is the easiest aspect to calibrate, because it is known that as ϵ goes to zero, the ABC algorithm becomes exact. That is, ABC will generate draws from the target posterior distribution.

The chain of draws from the approximate posterior distribution must be evaluated for convergence. Convergence diagnostics are important because MCMC algorithms may suffer if the proposed distribution q is poorly chosen. While all MCMC algorithms are in danger of getting stuck in low probability regions, ABC MCMC algorithms are particularly susceptible because of the two criteria that the proposed valued θ' must meet: not only must it meet the acceptance probability of the standard Metropolis-Hastings sampler, but it must also generate data that is sufficiently close to the observed data. Therefore, the rejection rates for ABC MCMC samplers can be extremely high. Too make matters worse, according to Turner and Van Zandt, [3], the MCMC chains cannot be parallelized. See Maria Rizzo's book *Statistical Computing with R* for more information on parallel chain diagnostics.

2.4 Particle Filtering

Sequential Monte Carlo sampling differs from the MCMC approach by its use of a "particle filter". That is, rather than drawing candidates θ' one at a time from a data generating mechanism, these algorithms work over a large pool of candidates, which are called particles. Typically, this class of methods works over the set of particles simultaneously. The particles are filtered and perturbed at each stage of the algorithm through a process that brings the pool of candidates closer and closer to a sample drawn from the target posterior density.

These algorithms begin by generating a pool of N candidate values for θ. Usually this pool is obtained by sampling from the prior distribution. Then, over subsequent iterations, particles are chosen randomly from this pool, but the probability that a particle is chosen depends on a weight that is assigned to each particle. During the first iteration, it is likely that all particles are weighted equally, however, after the first iteration, the weights will change according to the specific algorithm being employed. The process of perturbing and filtering particles requires that we choose something called a "transition kernel". The transition kernel serves the same purpose as the proposal density in the ABC Metropolis-Hasting algorithm detailed above. To specify the transition kernel, we need to choose a distribution for a random variable η that will be added to each particle to move the particle around the parameter space. For example, if a particle θ' is from the pool of candidates and perturbed by adding a Gaussian random variable $\eta \sim N(0, \sigma^2)$ to it, then the new proposed value will be $\theta'' = \theta' + \eta$. The transition kernel then describes the distribution of the random variables θ'' given the value of θ': A Gaussian distribution with mean θ' and variance σ^2.

Sometimes these algorithms require that we specify a transition kernel that takes use back to θ' from θ''. If the distribution of θ'' given θ' is a "forward" transition kernel, then then the distribution of θ' given θ'' is said to be a "backward" transition kernel. If the forward transition kernel is Gaussian, as in the example above, then because $\theta' = \theta'' - \eta$, an obvious choice for the backward transition kernel is again a Gaussian transition kernel with mean θ'' and variance σ^2. In general, the forward and backward kernels need not be symmetric or equal in distribution, however, in practive they frequently are as this adds a degree of computational and mathematical ease.

2.5 Particle Filtering: Population Monte Carlo Sampling

It may be the case that the most popular type of particle filtering among Biologists and Psychologists is the method of ABC population Monte Carlo sampling (ABC PMC). In this realization of particle filtering methodology, ABC PMC does not require specifying both forward and backward transition kernels, and instead the PMC algorithm use a single adaptive transition kernel, which we may denote $q(*|\theta')$ that depends on the variance of the accepted particles in the previous iteration of the algorithm. This algorithm was inspired by population Monte Carlo methods developed for standard Bayesian estimation by Cappe et al. [2].

Specifically, given the weight $w_{i,t-1}$ for the particle $\theta_{i,t-1}$ on iteration $t-1$, the new weight on iteration t is computed by

$$w_{i,t} = \frac{\pi(\theta_{i,t})}{\sum_{j=1}^{N} w_{j,t-1} q(\theta_{j,t-1}|\theta_{i,t}, \sigma_{t-1})},$$

where $q(*|\theta_{i,t}, \sigma_{t-1})$ is a Gaussian kernel with mean $\theta_{i,t}$ and standard deviation σ_{t-1}. The variance σ_t^2 is given by

$$\sigma_t^2 = \frac{2}{N} \sum_{i=1}^{N} \left(\theta_{i,t} - \sum_{j=1}^{N} \frac{\theta_{j,t}}{N} \right)^2 . = 2\text{Var}(\theta_{1:N,t})$$

One serious problem with many sampling schemes is the speed with which the posterior estimates can be obtained. This speed is dictated by the particle acceptance rate. Very low acceptance rates, which may arise when proposal distributions or transition kernels are incorrectly selected, result in a tremendous amount of wasted computation time. The importance of the ABC PMC method is that it optimizes the acceptance probability. The happens because the weights serve to minimize the Kullback-Leibler distance between the target posterior density and the proposed distribution. It has been shown by Beaumont [1], that as ϵ goes to 0, that the ABC PMC algorithm produces exact posterior samples. Lastly, I would like to note that ABC PMC is again, just one method of implementing particle filtering in the ABC context. Other algorithms exist for whittling down a large sample to a target sampling including the methods of Sequential Monte Carlo Sampling and Partial Rejection Control. However, it is known to scientists that the ABC PMC algorithm consistently provides good results in scientific models and requires the user to specify only a small number of tuning parameters to get going.

2.6 Example of ABC PMC: Exponential Data

I'd like to detail a toy example of ABC PMC, in which we know a lot more about the data than what can be realistically assumed. The exponential distribution has the probability density function

$$f(y|\lambda) = \lambda \exp(-\lambda y)$$

if $y \geq 0$. Where λ is called the rate parameter of the distribution, and the mean of a random variable $Y \sim Exp(\lambda)$ is known to be $1/\lambda$. The Gamma distribution has probability density function

$$f(y|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \exp(-\beta y)$$

where the hyperparameters α and β are called the shape and rate parameters. It is known that the Gamma distribution is the conjugate prior for the Exponential likelihood model. The posterior distribution for λ is given by

$$\lambda|Y, \alpha, \beta \sim Gamma(\alpha_0 + n, \beta_0 + \sum_{i=1}^{n} Y_i)$$

Suppose that the hyperparameters are both fixed at 0.1. We will use this posterior distribution to evaluate the accuracy of the ABC PMC algorithm. Now, here is the problem that we wish to address: How do we generate simulated data X that is sufficiently close to Y when Y is continuous and perhaps the proposal distribution is very far from the posterior distribution? The solution to the first problem is to choose a "good" distance function and representative sufficient statistics. The solution to the second problem is to gradually reduce the error tolerance ϵ so that we move efficiently from the prior density to the target posterior density. Therefore, we must balance computational efficiency with the accuracy of the posterior estimates, and to do so we will specify a set of monotonic decreasing values of ϵ over which the ABC PMC algorithm will iterate. To explore the influence of the choice of distance function on the accuracy of the estimated posterior samples, we will explore three different distance functions:

1. $d_1(X, Y) = |\bar{X} - \bar{Y}|$
2. $d_2(X, Y) = |\text{median}(X) - \text{median}(Y)|$
3. $d_3(X, Y) = |\text{IQR}(X) - \text{IQR}(Y)|$

Note that the exponential distribution is not a symmetric distribution. So the first two distance functions may not provide critical information about the skewness or variability of the data. It is important to note that we are not limited to using just one summary statistic. One can consider the example given during my class presentation of the ABC Metropolist-Hastings algorithm, which incorportated two summary statistics of the data.

It is known that computational difficulties can arise when the threshold ϵ is set to be too small. To generate the data, I consider a population of $N = 500$ particles from an exponential distribution with $\lambda = 0.1$ and I chose a decreasing set of tolerances $\epsilon = \{3, 1, 10^{-1}, 10^{-3}, 10^{-4}, 10^{-5}\}$

The algorithm then will proceed as follows:

```
Data: Given prior data $Y$ and distribution $Y \sim Exp(\theta)$,
      $\epsilon = \{3, 1, 10^{-1}, 10^{-3}, 10^{-4}, 10^{-5}\}$, one of the three distance measures above $d$,
      and prior $Gamma(0.1, 0.1)$ :
Result: We will use ABC PMC to filter a sample
initialization;
for $i$ in $1 : 500$ do
    while $d(X, Y) > 3$ do
        Sample $\theta' \sim Gamma(0.1, 0.1)$
        Generate data $X \sim Exp(\theta')$
        Calculate $d(X, Y)$
    end
    Set $\theta_{i,1} \leftarrow \theta'$
    Set $w_{i,1} \leftarrow \frac{1}{500}$
end
for $t$ in $2 : 6$ do
    for $i$ in $1 : 500$ do
        while $d(X, Y) > \epsilon_t$ do
            Sample $\theta' \sim \theta_{1:500,t-1}$ with probabilities $w_{1:500,t-1}$
            Set $\theta'' \sim N(\theta', \sigma^2_{t-1})$
            Generate $X \sim Exp(\theta'')$
            Calculate $d(X, Y)$
        end
        Set $\theta_{i,t} \leftarrow \theta''$
        Set $w_{i,t} \leftarrow \frac{\pi(\theta_{i,t})}{\sum_{j=1}^{500} w_{j,t-1} q_f(\theta_{j,t-1}|\theta_{i,t},\sigma_{t-1})}$
    end
    Set $\sigma^2_t \leftarrow 2 * \mathrm{Var}(\theta_{1:500,t})$
end
```

Algorithm 1: The ABC PMC for the Exponential Data

Unfortunately, it would seem that this algorithm is still too demanding to run on my laptop (takes more than 8 hours of computation time), so I'm going to have to omit any further results and leave the example as is.

References

[1] Beaumont, M. A.
Approximate Bayesian computation in evolution and ecology
Annual Review of Ecology, Evolution, and Systematics, 41. p. 379 - 406. 2010.

[2] Cappe O., Guilin A., Marin J. M., Robert C. P.
Population Monte Carlo
Journal of Computational and Graphical Statistics 13, p. 907. 2004.

[3] Brandon M. Turner and Trisha Van Zandt
A tutorial on approximate Bayesian Computation
Journal of Mathematical Psychology, 56. p. 69 - 85.

[4] Marjoram P, Molitor J, Plagnol V, Tavare S
Markov Chain Monte Carlo without Likelihoods 2003.
Proceedings from the National Academy of Sciences. 100. p. 324 - 328.

[5] P.J. Diggle and R.J. Gratton
Monte Carlo Methods of inference for implicit statistical models
Journal of the Royal Statistical Society. B 46. p. 193 - 212.

[6] Iain Murray, Zoubin Ghahrammi, David J.C. MacKay
MCMC for doubly-intractable distributions
`http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.63.2970&rep=rep1&type=pdf`

[7] K Csilcery, L Lemaire, O Francois, MGB Blum
Approximate Bayesian Computation (ABC) in R: A Vignette
`https://cran.r-project.org/web/packages/abc/vignettes/abcvignette.pdf`