**Terry Wright**

**What this does for me?**

Has the following things which might be useful in the future.

* GP search of sum stats. But GP usually has to be tuned. Does it work OK? You need to be able to bet on it. All I can say is that it gives better than a random choice. But would even simple brute force do better?
* If you can simulate the data generator then you can get parameter estimates – in an almost auto way
* There is going to be an absolutely massive list of statistics for time series or other forms of dependent data; this thing could chug its way through them
* Surely GP is the worst thing you would do if you have no idea which to choose.
* I had the idea of using dummy statistics which are just noise. So in the final elite list the positioning of these dummy variables would tell you where they end.

**Abstract**

Approximate Bayesian Computing can be used to find the posterior density without needing to know the likelihood function, but requires the ability to accurately simulate the data generator. In this paper the ABC approach is explained and then demonstrated with a simple example which is finding the posterior density with data from a logistic regression. Next ABC is applied to time series to fit ARIMA(1,d,0) processes. One of the main open problems of the ABC approach is that of choosing the best set summary statistics. In this paper a Genetic Programming (GP) approach is used to evolve the best choice of statistics in order to estimate ARIMA parameters. These algorithmically selected subsets are then used to with the ABC algorithm on a time series. The advantage of the procedure is that it could in principle be used to select effective subsets from an enormous number of possible statistics.

**Approximate Bayesian Computing (ABC)**

If the experimental result is **x\*** and parameters of a model are then the goal of a Bayesian analysis is to move from prior to the posterior distribution. Standardly this is accomplished using likelihood function for the model

ABC is a method which allows the approximate determination of the posterior distribution in situations where the likelihood is too complicated to specify. It works in its simplest form as a rejection algorithm. The basic idea is that if the simulated parameters are close to the true parameters then we would expect the simulated dataset to resemble the experimentally obtained data. Roughly speaking we repeatedly simulate **x** fromand then we select those values of which produced **x\*.**  This sample of would therefore be draws from As a result these could be used to make an empirical estimate of the likelihood. If the values of are initially drawn from the priors and then used to simulate **x**, then those which produce **x\*** must therefore be draws from the posterior density, **.** Hence the method can be used to make estimates of posterior density without explicitly calculating the likelihood – as a result this is known as a likelihood-free method.

As it stands this approach cannot be used with continuous data due to that fact that it would be impossible to exactly simulate x\*. Therefore a distance metric (usually Euclidean) is used along with a tolerance , to decide whether the simulated data is sufficiently close to the experimental results. In this way the estimation of the likelihood and posterior density becomes approximate.

This approach can be impractical as the dimension of **x\*** increases. Due to the increase in the space of possible **x**, it becomes less and less likely that one will be found sufficiently close to **x\*.**  Statistics can be used to reduce the dimensionality of the data. Ideally **sufficient statistics** would be used. A sufficient statistic contains as much information about a parameter as the data themselves and so do not introduce any more approximations. The use of sufficient statistics would make the process computationally much more efficient.

This produces a more efficient estimate for the posterior density.

Sufficient statistics can be very hard to find for non-standard distributions and so to make the process more tractable summary statistics are chosen which it is believed contain information about the parameters. This introduces another level of approximation and distortion into the estimation of the posterior density. The ABC approach can then be summarised as:

For the approach to provide a practical estimate of the true posterior we need summary statistics which contain information about the parameter, and a large enough set of acceptable samples (within the tolerance), such that the estimated distribution does not have significant sampling error. Finally we must ensure that the tolerance is small enough so that the acceptable **x** are close enough to **x\*** to have been generated by approximately the same.

**The ABC Rejection Algorithm (ABC-REJ)**

To obtain k approximate samples from the posterior distribution.

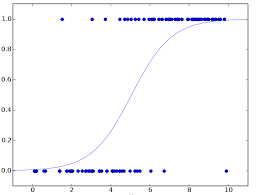
1. Simulate parameters from the prior density
2. Generate a simulated data set **x** from the model using parameters
3. Calculate a vector of summary statistics for the simulated data set S(**x**) and for the actual experimental data set S(**x\***).
4. If then accept as an observation from the posterior, is chosen as the tolerance.
5. Repeat until a sample size k has been obtained or a maximum number of iterations have taken place.

**A Simple Logistic Regression Example of ABC *(logistic.sample.R)***

The R “abc” package has been used to demonstrate a simple example of estimating the posterior distribution. The context is fitting a logistic regression where an explanatory variable X and each value X causes a different probability in an independent Bernoulli trial.

The initial prior for are both taken to be In each run through the 100 values of **X** will produce a response vector **Y** of 100 bits long depending on whether the trial was successful or not.

Logistic regressions take the form of sigmoid graphs whose shape is controlled by . Hence we need summary statistics which are related to the shape of the regression.



There are many possible statistics that could be used to characterise the shape of the curve the following were arbitrarily chosen:

1. The difference between the mean values of x when the trial was a success and the mean value of x when the trial was a failure. This should give some indication of the ‘slope’ of the logistic curve.
2. The proportion of successes in the first quarter of Y. This would give an indication of when the logistic curve starts to rise.

Each of these summaries should provide information about beta1 and beta2. To confirm this plots of summary statistics vs parameters were made with 50000 simulation. The plots show that statistic1 has weak information about both , and that statistic2 appears to have weak information about but it has no information about .

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

The ABC method was used with 50000 simulations and a tolerance of 0.01 which resulted in 500 samples drawn from the approximate posterior. This produced distributions with large amounts of probability density around the true values of the parameters.

|  |  |
| --- | --- |
|  |  |

This shows that even though the simple plots of summary statistics appeared to contain only weak information about the parameters the ABC method was able to extract and express that information in the posterior distributions.

The procedure was repeated with with 50000 simulations. The plots below shows the effect of reducing tolerance. With a tolerance of 0.9, virtually all of the samples are selected from our uniform prior. Hence the procedure has added no new information into the estimation of the posterior. As the tolerance is reduced to 0.01 probability density accumulate around the (0.75, 1) interval. For a tolerance of 0.001 there are now only 100 accepted samples and some sampling error is starting to become visible in the estimated distribution

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

This shows that a balance must be stuck with a small tolerance but also with a reasonable (100+) number of samples. An arbitrary rule of thumb that is commonly cited is to have a tolerance of 1%.

**A Time Series Example of ABC *(time.series.sample.R)***

The ABC method was then used to produce posterior estimates for the parameters behind the generation of time series. To demonstrate the approach, 20 points were generated from an ARIMA (1, d, 0) model with a residual standard deviation of r, where d, AR1 and r have priors

There are many statistics that can be used to describe time series. We could simple use the mean and standard deviation of the data, or we could model the trend of the data with a linear model – we could estimate an intercept and gradient. We could look at the correlation between the points through the autocorrelation function, which measure the correlation between points on the time series separated by a lag k. We could use the partial autocorrelation function to measure direct correlation between points within the time series. We could measure statistics between the first differences between successive members or second difference etc. To illustrate the technique the following statistics were arbitrarily chosen:

1. Mean of the first 5 terms in the autocorrelation function.
2. Mean of the first 5 terms of the partial autocorrelation.
3. Difference between the maximum and minimum values of the time series.

Plots of statistics vs parameters shows that the first statistic has a weak relationship with r. The first and second statistics show weak information about AR1 and all of the three statistics have weak information about d. However these statistics have only very vague information about the target parameters.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

ABC-REJ was run with 100000 simulations with a tolerance of 0.001. The approximate posteriors are shown below for .

|  |  |  |
| --- | --- | --- |
|  |  |  |

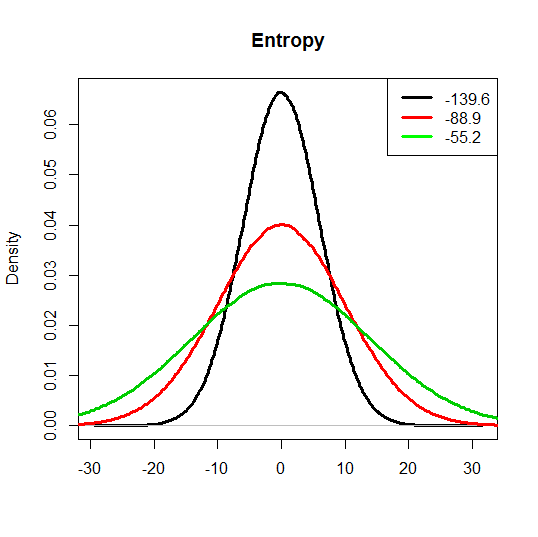
**Selection of Summary Statistics**

Clearly then the goal must be to choose a set of summary statistics which captures as much of the information in the sample about the parameters as possible. A naïve approach might be just to pack together a long list of summary statistics each of which contains some information about the parameters. The problem with this is that every time a statistic is added to the list as well as adding some signal to the group it also adds noise. As statistics are added it becomes harder and harder to find new information about the parameter, however each one will add noise. This noise will add up and eventually drown out the signal. Hence long collections of summary statistics do not work. What we need is a short number of information rich summary statistics. There is no established method at arriving at this. The general approach has been to consult the relevant literature and to build up a list of candidate statistics and then choose a subset of them that intuitively appear to be related to the parameters. Simple plots of statistics against parameters as used above can aid in the process of selection of good candidates. If a summary statistic has no information about the parameter then the posterior will remain the same as the prior.

**Algorithmic Selection of Summary Statistics Using Multi-Objective Optimisation**

With infinite computing power and a brute force approach we would work through each possible subset and look at posterior distributions and assess those which seem to carry the most information. Entropy is a measure of the amount of ‘ignorance’ within a distribution. A uniform distribution with a wide support has a high entropy. As the variance of a normal distribution increases so does the entropy. As the distribution becomes more informative then the entropy decreases. So ‘pointed’ distributions have low entropy.

An approximate method for finding the entropy of a vector of data is to first of all build a histogram and then calculate the entropy by:



We could use entropy as a way to distinguish between the posterior distributions generated by different subsets of statistics. The best combination of statistics would produce the lowest entropy. All that is needed for an algorithmic approach is to search through all possible subsets of statistics and then to choose the one with the lowest entropy. Unfortunately there can be a very large number of different subsets. There are for example 21699 different subsets of length up to 5 when selecting between 20 summary statistics. There are 760000 when choosing between groups from within 40 statistics. Hence in the case of there being a wide choice of possible statistics working through each combination might not be computationally viable. Genetic Programming (GP) and Evolutionary Algorithms (EA) have recently have become popular through their ability to search through enormous spaces and to locate credible solutions – in some cases discover new solutions to existing problems. This suggests that a GP approach could be used to select good subsets of statistics.

**Conflict of Interest**

It seems that groups of weak statistics can be used by the ABC approach to concentrate information about a parameter into the posterior distribution, but that also noise can increase as the length increases. Hence what we really want to do is to minimise the entropy and at the same time minimise the number of statistics in the list. In fact one might argue that if we have found the distribution with the minimum entropy then the length is irrelevant because it already has the most information. Unfortunately the ABC method as well as the calculation of entropy (particularly the calculation of entropy!) are approximate and so should be used as guiding principles. In general short vectors have less noise than long. In general longer vectors might have more information. Since these factors could be at odds with each other, thus the need for Multi-Objective Optimisation. The solutions will be trade-offs of the 2 objectives. There are lots of ways of defining ‘best’ but the predominant one is the **Pareto front.**

**Pareto Optimisation/Pareto Front**

Imagine you have 2 candidates A and B. A **Pareto dominates** B if A is at least as good as B in all regards and superior in at least one. Neither A nor B dominates the other. They could be identical in all regards, or A could be superior in some regards whereas B is superior in other regards. In this situation A and B are said to be non-dominated. The set of such solutions is called the Pareto non-dominated front (Pareto front). 2 very popular algorithms for enumerating the Pareto front are Non-Dominated Sorting Genetic Algorithm II (NSGA-II) and Strength Pareto Evolutionary Algorithm 2 (SPEA2). Both are archive algorithms which hold an archive of the best individuals discovered so far that use a genetic search to find the Pareto front.

**Genetic Programming**

Genetic Programming (GP) is a very popular technique for locating solutions in a large space of solutions using the evolutionary algorithm. Genetic Programming is generally reserved for any situation where the solution can be represented with a tree. In the case of this project each solution is actually a vector of integers. Each integer corresponding to a statistic. GP uses genetic operators to evolve a set of good solutions by simulating an evolutionary process. Good solutions are more likely to have children and pass on their ‘genes’ to their children. Poor solutions are removed from the gene-pool. The score that each solution is given involves the length of the vector of statistics as well as the entropy. This should lead to an information gradient that will steer the process as a whole towards the better subsets.

**Tournament Selection**

Small random groups are selected from the population and are ranked by their score. The best is used as a parent. This way it is possible for even poor solutions to have offspring which maintains the diversity of the population. It is believed that diversity is the key to the GP success.

**Genetic Operators**

Child solutions are produced by a process called **cross-over** where the contents of each parent are randomly allocated into the children e.g. (1,2,10) and (11, 16) could produce children (1,2,10,11) and (16). The point is that there is a chance the components that cause a particular subset of have a good score can start to accumulate and produce better solutions.

There is also a random chance that a particular subset will **mutate** e.g. (1,2,3) goes to (10,2,3). This means that in principle every subset is reachable from the current population.

**Time Series Example of Multi-Objective Optimisation ABC Example (MO\_ABC1.clj, RABC.java, mo.sample.R)**

The goal this example, is to find r, AR1 and d from an ARIMA(1,d,0) process – but this time to identify the ‘best’ set of summary statistics through the SPEA2 algorithm by minimising the entropy and the length of the list of statistics. That is to create a Pareto front of best candidates which can individually be examined and a choice made. Then this choice of summary statistics will be used as before to find the posterior distributions for r, AR1 and d.

Traditionally much of the genetic programming literature has used LISP and LISP dialects such as Clojure. This is because code and data in LISP take the same form – the form of a list. So (1,2,3) might represent data – or is could be executable code. This means that it is much easier to implement a system that can evolve computer code itself. This project has leveraged existing GP code written in Clojure in particular the Darwin codebase written by J Hudson and under the terms of the MIT licence. The Gorilla worksheet ***MO\_ABC1.clj*** uses this code and also contains the evolutionary operators such as cross-over and mutation in addition to the plumbing for the SPEA2 algorithm. The scoring of each subset takes place in R via the script ***mo.sample.R*** using ABC-REJ. To simplify the process of linking Clojure to R involved a Java class ***RABC.java*** . The ***MO\_ABC1.clj*** code calls ***RABC.java*** Score which then calls mo.score in ***mo.sample.R*** the subset under examination by the GP is then passed through to the R code. 50 ABC estimates of the posterior are performed for randomly chosen parameters and for each the empirical entropy is calculated. The mean entropy is returned via ***RABC.java*** to ***MO\_ABC1.clj***.

**Summary Statistics**

The following list of summary statistics could contain information about the parameters are:

1. Mean of the time series
2. Intercept of the time series (from linear model)
3. Gradient of the time series (from linear model)
4. Residual variance (from linear model)
5. Autocorrelation function (ACF)[2]
6. ACF[3]
7. ACF[4]
8. ACF[5]
9. Mean of the ACF
10. Standard deviation of the ACF
11. Partial Autocorrelation function (PACF)[1]
12. PACF[2]
13. PACF[3]
14. PACF[4]
15. Mean of PACF
16. Standard deviation of the PACF
17. Maximum of the time series – minimum of the time series

**Pareto Fronts**

Pareto fronts were evolved to choose the best subsets for r, AR1 and d. The GP algorithm was run for 100 generations for each parameter. In each case the populations appeared to collapse onto a small number of highly represented subsets.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **r** | **AR1** | **d** |
|  |  |  |  |
| **Subsets on the Pareto Front** | In this case the Pareto Front collapsed to a single point, so there is a genuine optimum subset.  (4) | (4,9,11,12), (9, 11, 12), (11, 12), (12) | (5 15 16), (11,16), (4) |

The ABC process was then performed with 100000 simulations and a tolerance of 0.005 with the subsets chosen by the GP. The actual parameters were r=1.79, AR1=-0.094, d=1, using subsets (4), (4,9,11,12) and (5,15,16).

|  |  |  |
| --- | --- | --- |
|  |  |  |

In each case the histograms were made out of 500 samples and so sampling error is likely to be small. Hence the spread in the distributions is most likely caused by the distorting effect of the summary statistics. The distributions do appear to have reduced spread compared with the estimates previously made using the previous choices of summary statistics. The MAP estimates are close to the true parameter values. Beyond constructing a long list of statistics the procedure is automatic.

**Bibliography**

1. Beaumont, M.A., Zhang, W., and Balding, D. J. 2002. Approximate Bayesian Computing in Population Genetics. Genetics, 162(4):2025-2035.
2. Lee, P.M., Bayesian Statistics an Introduction. Wiley
3. Turner, B.M., and Zandt, T.V. 2012. A Tutorial on approximate Bayesian Computing, Journal of Mathematical Psychology, 56(2):69-80.
4. Csillery, K., Lemaire, O., Francois, O., Blum, M.G.B., Approximate Bayesian Computing (ABC) in R: A Vignette. <https://cran.r-project.org/web/packages/abc/vignettes/abcvignette.pdf>
5. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D., Genetic Programming, An Introduction. Morgan Kaufmann Publishers Inc