Approximate Bayesian Computation

Dom Hutchinson

December 5, 2020

Intro to ABC

Definition 1.1 - *Approximate Bayesian Computation (ABC)*
Approximate Bayesian Computation (ABC) is a family of computational methods for estimating the posterior of model parameters for *Generative Models*. *Generative Models* are models which can be simulated from but we do not have an explicit definition for their posterior $f(x|\theta)$ (i.e. most IRL systems).

Proposition 1.1 - *Motivating Idea*
Consider a set of observations $\mathbf{y} := (y_1, \dots, y_n)$ where each $y_i \in \mathbb{R}^m$ is high dimensional. Let $s(\cdot) : \mathbb{R}^m \to \mathbb{R}^p$ be a mapping (known as a *Summary Statistic*) from the observed data to some lower dimension p.

ABC aims to infer the joint distribution of parameter θ and general summary statistics $\mathbf{s}$, given the observed summary statistics $\mathbf{s}_{obs} := s(\mathbf{y})$

$$p_\epsilon(\theta, \mathbf{s}|\mathbf{s}_{obs}) \propto \pi_0(\theta) f(s|\theta) K_\epsilon(\|\mathbf{s} - \mathbf{s}_{obs}\|)$$

where $\pi_0(\theta)$ is the prior for parameter θ, $f(\mathbf{s}|\theta)$ is the likelihood of the summary statistics, $K_\epsilon(\cdot)$ is a kernel function scaling parameter ϵ and $\|\cdot\|$ is a distance measure (e.g. Euclidean).[1]

From this joint distribution the posterior for parameter θ, given the observed summary statistics $\mathbf{s}_{obs} := s(\mathbf{y})$, can be calculated as

$$p_\epsilon(\theta|\mathbf{s}_{obs}) = \int p_\epsilon(\theta, \mathbf{s}|\mathbf{s}_{obs}) d\mathbf{s}$$

Monte-Carlo Algorithms can be used to sample from this posterior $p_\epsilon(\theta|\mathbf{s}_{obs})$ without having to explicitly state the likelihood $f(\mathbf{s}|\theta)$. *Numerical Integration* methods can then be used to evaluate the integral[2].

Proposition 1.2 - *Setup of ABC*
Consider having the following:

- A set of observations $\mathbf{y} := (y_1, \dots, y_n)$ where each y_i is high-dimensional.
- A map $s(\cdot)$ which maps the high-dimensional observed data to a lower dimension.
- A parameter θ which we wish to find the posterior for.
- A prior $\pi_0(\theta)$ for the parameter θ.

[1] $f(s|\theta)$ is the only one of these features which is not specified by the user, and thus what we need to "learn".
[2] See *Monte-Carlo Integration* methods: *Uniform Sampling*, *Importance Sampling*

- A kernel $K_\epsilon(\cdot)$ and a distance measure $\|\cdot\|$.

Proposition 1.3 - *ABC Algorithm - Simple, Online*
Consider the setup in `Proposition 1.2`. Here is a simple, online algorithm for ABC

i). Sample a set of parameters from the prior $\theta_t \sim \pi_0(\theta)$.

ii). Simulate summary statistic values $\mathbf{s}_t$ from the implicit likelihood[3] $f(\mathbf{s}|\theta_t)$ for the summary statistics given the sample parameter value.

iii). Reject the sample summary statistic value $\mathbf{s}_t$ with probability $K_\epsilon(\|\mathbf{s}_t - \mathbf{s}_{obs})$ where $\mathbf{s}_{obs} = s(\mathbf{y})$.

iv). Repeat steps i)-iii) until a total of M simulated values have been accepted.

Our final sample contains a set of summary statistics a long with the parameter values which produced them. This data can be used to approximate the posterior for the parameter values.

Decisions

Remark 1.1 - *Decisions*
When implementing ABC there are several decisions to make, including:

- What kernel $K_\epsilon(\cdot)$ to use.
- What summary statistics $s(\cdot)$ to use.
- Do we even need summary statistics?
- How long to sample for?

Proposition 1.4 - *Kernels $K_\epsilon(\cdot)$*
A *Kernel* is used to determine with what probability to accept a sample, given it is a certain distance away from observed data. Here are some common kernels

- *Uniform Kernel* $K_\epsilon(\|\mathbf{s}-\mathbf{s}_{obs}\|) := \mathbb{1}\{\|\mathbf{s}-\mathbf{s}_{obs}\| \leq \epsilon\}$ which accepts simulated values if they are within ϵ of observed data.
- *Epanechnikov Kernel* $K(\|\mathbf{s}-\mathbf{s}_{obs}\|) := \frac{3}{4}(1-\|\mathbf{s}-\mathbf{s}_{obs}\|^2)$ for $\|\mathbf{s}-\mathbf{s}_{obs}\| \in [0,1]$
- *Gaussian Kernel* $K(\|\mathbf{s}-\mathbf{s}_{obs}\|) := \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\|\mathbf{s}-\mathbf{s}_{obs}\|^2}$

Proposition 1.5 - *Summary Statistics $s(\cdot)$*
See `SummaryStatisticSelection.pdf`

Proposition 1.6 - *How long to sample for*
The algorithm given in `Proposition 1.3` runs the algorithm until a sufficiently large sample has been produced. This is not ideal as the algorithm will run for an unknown period of time and is dependent upon the kernel $K_\epsilon(\cdot)$ which has been defined.

Alternatively, all simulated values could be kept and then all but the best M[4] are discarded.

[3]Run the system with the sampled parameters
[4]M closest to $_{\text{obs}}$.

Semi-Automatic ABC

Definition 1.2 - *Semi-Automatic ABC*[5]
In *Semi-Automatic ABC* summary statistics are learnt from simulation, but the user still has to make choices around what transformation $\mathbf{f}(\cdot)$ of simulated data $\mathbf{y}$.

An application of *Semi-Automatic ABC* should perform better in a general setting than traditional *ABC*.

Proposition 1.7 - *Semi-Automatic ABC - Algorithm*

i). Perform a pilot run of ABCWe need to define some arbitrary summary-statistics for this. to determine a training-region of non-negligible posterior mass.

ii). for $t \in [1, M]$:

   (a) Simulate parameters $\boldsymbol{\theta}_t$ from our prior $\pi_0(\boldsymbol{\theta})$, with the prior truncated to the training-region determined in (i).

   (b) Simulate results $\mathbf{y}_t \sim f(\mathbf{y}|\boldsymbol{\theta}_t)$ using these parameters.

iii). Use simulated data and parameter values to estimate summary statistics.[6]

iv). Run ABC with these estimated-summary statistics.

Remark 1.2 - *Step iii)*
In step iii) we have simulated data $\mathcal{D} := \{(\boldsymbol{\theta}_1, \mathbf{y}_1), \dots, (\boldsymbol{\theta}_M, \mathbf{y}_M)\}$ where $\boldsymbol{\theta}_t \in \mathbb{R}^m,\ \mathbf{y}_t \in \mathbb{R}^n_t\ \forall\ t \in [1, M]$. We want to learn a transformation $f(\mathbf{y}_t)$ of the simulated data $\mathbf{y}_t$ st the parameters $\boldsymbol{\theta}_t$ can be learnt from the transformation.

A *Liner Regression* approach is to learn the pick a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$[7] which maps the simulated data $\mathbf{x}_t$ to the same dimension as the simulated parameters (ie to m dimensions) and then the parameters $\boldsymbol{\beta}_0, \boldsymbol{\theta}_1$ which give the least total-error across the whole data set $\mathcal{D}$[8] for the following

$$\begin{array}{rcl} \boldsymbol{\theta}_t &=& \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 \cdot \mathbf{f}(\mathbf{y}_t) + \varepsilon_t \\ \Leftrightarrow [\boldsymbol{\theta}_t]_i &=& [\boldsymbol{\beta}_0]_i + [\boldsymbol{\beta}_1]_i \cdot [\mathbf{f}(\mathbf{y}_t)]_i + [\varepsilon_t]_i \end{array}$$

Our estimate for the model parameters $\boldsymbol{\theta}$, given some data $\mathbf{y}$, is thus the fitted value

$$\hat{\boldsymbol{\theta}} = \mathbb{E}[\boldsymbol{\theta}|\mathbf{y}] = \hat{\boldsymbol{\beta}}_0 + \hat{\boldsymbol{\beta}}_1 \mathbf{f}(\boldsymbol{y})$$

The constant terms $\hat{\boldsymbol{\beta}}_0$ can be ignored as ABC only uses the distance between summary statistics (not their absolute value). This means our m summary statistics are the different dimensions of $\hat{\boldsymbol{\beta}}_1 \mathbf{f}(\cdot)$

Remark 1.3 - *Choosing transformation* $\mathbf{f}(\cdot)$
In `Remark 1.2` the user has to define how to transform the simulated results (ie define $\mathbf{f}(\cdot)$) and this choice will affect the set of summary statistics generated. It is easy to run this stage multiple times, using different transformations on the same data $\mathcal{D}$ and then using standard

[5]Fearnhead, P., Prangle, D. (2012). Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 74(3), 419-474.*

[6]Potential methods inc. linear-regression, lasso analysis, cross-correlation analysis.

[7]This transformation can actually map to any dimension but we prefer for it to be a lower dimension that the simulated data $\mathbf{y}$.

[8]Generally least-square-error.

model comparison procedures[9] to determine which of the generate summary statistics are sufficient.

[9]e.g. BIC, sufficiency