Approximate Bayesian Computation

Dom Hutchinson

December 29, 2020

Contents

1 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 $f_{\mathcal{G}}(x|\theta)$ (eg most IRL systems).

Proposition 1.1 - *Motivating Idea[1]*
Consider a set of observations[1][2] $\mathbf{y} := (y_1, \dots, y_n)$ 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 parameters $\boldsymbol{\theta}$ and general summary statistics $\mathbf{s}$, given the observed summary statistics $\mathbf{s}_{obs} := s(\mathbf{y})$

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

where $K_\epsilon(\cdot)$ is a kernel function with scaling parameter ϵ and $\|\cdot\|$ is a distance measure (e.g. Euclidean).[3]

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

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

[1] From a *Generative Model*

[2] Generally these observations are ordered in some way (by the variables of the system) so can be considered a sequence $\{y_t\}_{t \in T}$ where T specifies the variable values in each epoch.

[3] The likelihood $f(s|\theta)$ is the only one of these features which is not specified by the user, and thus what we need to "learn" it.

Monte-Carlo Algorithms can be used to sample from this posterior $p_\epsilon(\boldsymbol{\theta}|\mathbf{s}_{obs})$ without having to explicitly state the likelihood $f(\mathbf{s}|\theta)$. *Numerical Integration* methods can then be used to evaluate the integral[4].

Proposition 1.2 - *Setup of ABC*
To perform *ABC* we typically have/define the following features

- A set of observations from a *Generative Model* $\mathbf{y} := (y_1, \dots, y_n)$ where each y_i is high-dimensional.
- A map $s(\cdot)$ which maps the high-dimensional observed data to a lower dimension.
- A theorised model with posterior pdf $f_{\mathcal{T}}(\cdot|\boldsymbol{\theta}, \mathbf{x})$ where $\boldsymbol{\theta}$ are the parameters which we wish to fit to the *Generative Model* using ABC.
- A prior $\pi_0(\cdot)$ for the parameters $\boldsymbol{\theta}$.
- A kernel $K_\epsilon(\cdot)$ and a distance measure $\|\cdot\|$.

1.1 Decisions

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

- What theorised model $f(\cdot|\boldsymbol{\theta}, \mathbf{x})$ to use.
- 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 1.3 - *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* $K_\epsilon(\|\mathbf{s} - \mathbf{s}_{obs}\|) := \frac{3}{4\epsilon}\left(1 - \left(\frac{\|\mathbf{s}-\mathbf{s}_{obs}\|}{\epsilon}\right)^2\right)$ for $\|\mathbf{s} - \mathbf{s}_{obs}\| \leq \epsilon$
- *Gaussian Kernel* $K(\|\mathbf{s} - \mathbf{s}_{obs}\|) := \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\|\mathbf{s}-\mathbf{s}_{obs}\|^2}$

Proposition 1.4 - *Summary Statistics $s(\cdot)$*
See `SummaryStatisticSelection.pdf`

Proposition 1.5 - *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[5] are discarded.

[4] See *Monte-Carlo Integration* methods: *Uniform Sampling*, *Importance Sampling*
[5] M closest to s_{obs}.

2 ABC Algorithms

Proposition 2.1 - *ABC Algorithm - Rejection*
Consider the setup in `Proposition 1.2`. Here is a simple, online algorithm for ABC

i). Sample a set of parameters from the prior $\tilde{\boldsymbol{\theta}}_t \sim \pi_0(\boldsymbol{\theta})$.

ii). Sample from the theorised model using these sampled parameters

$$\mathbf{y}_t \sim f_{\mathcal{T}}(\mathbf{y}|\tilde{\boldsymbol{\theta}}_t)$$

iii). Calculate the summary statistic values for the sampled values $\mathbf{s}_t = \mathbf{s}(\mathbf{y}_t)$.

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

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

Our final sample contains a set of M summary statistics, along with the parameter values $\boldsymbol{\theta}$ and variable-space points $\mathbf{x}$, which produced them. This data can be used to approximate the posterior for the parameter values.

Remark 2.1 - *ABC-SMC*
The idea behind *ABC-Sequential Monte Carlo* is to initially use a kernel with a large acceptance range to produce a sample of parameters which v. roughly approximate of the posterior. And then to finese this sample to improve the posterior, by resampling and tightening the kernel.

Proposition 2.2 - *ABC Algorithm - SMC[2][3][4]*
Consider the setup in `Proposition 1.2`. Here is a *Sequential Monte-Carlo* algorithm for ABC

- *Initialisation* - Choose a set of *scaling parameters* $\{\epsilon_1, \dots, \epsilon_T\}$ which are increasingly tight[6]

  $$\{\epsilon_1, \dots, \epsilon_T\} \text{ where } \epsilon_1 > \dots > \epsilon_T$$

  Set N to be the number of sets of parameters to sample, choose some summary statistics $s(\cdot)$ and observe y_{obs} from the true model.

- *Initial Sampling Step* - $t = 1$.[7][8]

  i). Sample a set of parameters $\tilde{\boldsymbol{\theta}}_{1,i}$

  $$\tilde{\boldsymbol{\theta}}_1 \sim \pi_0(\boldsymbol{\theta})$$

  ii). Observe the theorised model $\mathcal{T}$ with these sampled parameters

  $$\mathbf{y}_{1,i} \sim f_{\mathcal{T}}(\mathbf{y}|\tilde{\boldsymbol{\theta}}_{1,i})$$

  iii). *If* $K_{\epsilon_1}(\|s(\mathbf{y}_{1,i}) - s(\mathbf{y}_{obs})\|)$:[9]
  - Store the sampled parameters $\tilde{\boldsymbol{\theta}}_{1,i}$ in set Θ_1.
  - Set $w_{1,i} = \frac{1}{N}$ and increment i.

[6] There are automatic methods where ϵ_{t+1} is calculated at the end of step t st convergence is encouraged.
[7] Here a set of possible parameters Θ_1 is generated.
[8] It is possible for the same set of parameters to appear in Θ_t multiple times.
[9] Sample parameters are accepted by the kernel.

  - *If* $i == N$: move to *Resampling Step*.

- *Resampling Step* - $t = 2, \dots, T$.[10]

  i). Sample $\tilde{\boldsymbol{\theta}}_{t,i}$ from set Θ_{t-1} with probability $w_{t-1,j}$.The weights form a KDE.

$$\mathbb{P}(\boldsymbol{\theta}_i^* = \tilde{\boldsymbol{\theta}}_j) = w_{t-1,j} \text{ for } \tilde{\boldsymbol{\theta}}_j \in \Theta_{t-1}$$

  ii). Perturb the sample parameters slightly using a *Pertubance Kernel* K^* to get a slightly different parameter set $\boldsymbol{\theta}_{t,i}^*$

$$\boldsymbol{\theta}_{t,i}^* \sim K^*(\theta|\tilde{\boldsymbol{\theta}}_{t,i})$$

  iii). *If* $\boldsymbol{\theta}_{t,i}^*$ *is impossible under the prior*:[11] return to i).

  iv). Observe the theorised model $\mathcal{T}$ with these perturbed parameters

$$\mathbf{y}_{t,i} \sim f_{\mathcal{T}}(\mathbf{y}|\boldsymbol{\theta}_{t,i}^*)$$

  v). *If* <u>*not*</u> $K_{\epsilon_t}(\|s(\mathbf{y}_{t,i}) - s(\mathbf{y}_{obs})\|)$:[12] return to i).

  vi). Add $\boldsymbol{\theta}_{t,i}^*$ to Θ_t and assign it weight $w_{t,i}$

$$\tilde{w}_{t,i} = \frac{\pi_0(\boldsymbol{\theta}_{t,i}^*)}{\sum_{j=1}^N w_{t-1,j}\mathbb{P}(K_t(\theta|\tilde{\boldsymbol{\theta}}_{t,i}) = \boldsymbol{\theta}_{t,i}^*)^{[13]}}$$

  vii). *If* $|\Theta_t| < N$: increment i and return to i).

  viii). Normalise weights

$$w_{t,i} = \frac{\tilde{w}_{t,i}}{\sum_{i=1}^N w_{t,i}}$$

  ix). Increment t.

Code for this can be found at https://stats.stackexchange.com/a/328384.

Proposition 2.3 - *ABC Algorithm - MCMC[5]*
Consider the setup in `Proposition 1.2`. Here is a *Markov Chain Monte-Carlo* algorithm for *ABC*

- *Initialisation* - Observe y_{obs} from the true model, choose some summary statistics $s(\cdot)$, define the length of the chain N[14], a theorised model T with pdf $f_{\mathcal{T}}(\cdot|\boldsymbol{\theta})$, define a prior for the priors $\pi_0(\boldsymbol{\theta})$ and a pertubance kernel $K^*(\cdot)$[15].

- *Starting Sample*

  i). Sample $\tilde{\boldsymbol{\theta}}_0 \sim \pi(\boldsymbol{\theta})$.

[10] Here, we determine weightings for the sets of parameters found in the previous step. With the weightings representing posterior probabilities.
[11] ie $\pi_0(\boldsymbol{\theta}_{t,i}^*) = 0$
[12] ie the observation is reject by the kernel
[13] Weighted sum of the probability this set of parameters $\boldsymbol{\theta}_{t,i}^*$ could have been observed under another one of the previous samples.
[14] This is used for the termination condition, but other termination conditions are common. Most are based around spotting convergence.
[15] e.g. Add some gaussian noise.

ii). Observe y from the theorised model $\mathcal{T}$ using these sampled parameter $\boldsymbol{\theta}_0$.

$$y \sim f_{\mathcal{T}}(y|\tilde{\boldsymbol{\theta}}_0)$$

iii). *If* $K_\epsilon(\|s(y) - s(y_{obs})\|)$: move to *MCMC Step.*
Else : return to i).

- *MCMC Step.*

  i). For $t \in [1, N]$:

  (a) Perturb the previous parameter value $\tilde{\boldsymbol{\theta}}_{t-1}$ to get a new value $\boldsymbol{\theta}^*$.

  $$\boldsymbol{\theta}^* = K^*(\tilde{\boldsymbol{\theta}}_{t-1})$$

  (b) Observe y from the theorised model $\mathcal{T}$ using these perturbed parameters $\boldsymbol{\theta}^*$.

  $$y \sim f_{\mathcal{T}}(y|\tilde{\boldsymbol{\theta}}_0)$$

  (c) *If* $K_\epsilon(\|s(y) - s(y_{obs})\|)$: Set $\tilde{\boldsymbol{\theta}}_t = \boldsymbol{\theta}^*$.
  Else : Set $\tilde{\boldsymbol{\theta}}_t = \tilde{\boldsymbol{\theta}}_{t-1}$.

- Produce a posterior from the accepted parameter values $\{\tilde{\boldsymbol{\theta}}_0, \dots, \tilde{\boldsymbol{\theta}}_N\}$

3 Semi-Automatic ABC

Definition 3.1 - *Semi-Automatic ABC[6]*
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 3.1 - *Semi-Automatic ABC - Algorithm*

i). Perform a pilot run of ABC[16] 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.[17]

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

Remark 3.1 - *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.

[16] We need to define some arbitrary summary-statistics for this.
[17] Potential methods inc. linear-regression, lasso analysis, cross-correlation analysis.

A *Liner Regression* approach is to learn the pick a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$[18] 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 $\mathcal{D}$[19] for the following

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

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 3.2 - *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 model comparison procedures[20] to determine which of the generate summary statistics are sufficient.

[18] This transformation can actually map to any dimension but we prefer for it to be a lower dimension that the simulated data $\mathbf{y}$.
[19] Generally least-square-error.
[20] e.g. BIC, sufficiency

References

[1] Mark A. Beaumont. Approximate bayesian computation. *Annual Review of Statistics and Its Application*, 6(1):379–403, 2019.

[2] Richard G. Everitt and Paulina A. Rowińska. Delayed acceptance abc-smc. *Journal of Computational and Graphical Statistics*, 0(0):1–12, 2020.

[3] Patrick Laub. Approximate bayesian computation: Introduction insurance examples.

[4] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael P.H Stumpf. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of The Royal Society Interface*, 6(31):187–202, 2009.

[5] Florian Hartig, Marshal says:, Florian Hartig says:, Matias Francisco Hernandez Arellano says:, and Nori Says:. A simple approximate bayesian computation mcmc (abc-mcmc) in r, Jan 2014.

[6] Paul Fearnhead and Dennis Prangle. 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, 2012.