An introduction to Approximate Bayesian Computation

Antonietta Mira

Universita' della Svizzera italiana and University of Insubria

Thanks to Louis Raynal - Harvard School of Public Health for a preliminary version of the slides

May 8, 2020

Introduction

Let us consider an observed data $\mathbf{y} \in \mathcal{Y}$, whose generation process can be described by a statistical model parameterised by an unknown vector of parameters $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$

The likelihood of $\mathbf{y}$ is denoted $p(\mathbf{y} \mid \boldsymbol{\theta})$

The likelihood expression is unknown and we cannot approximate it easily

Sources of intractability

There are two main situations where the likelihood is intractable:

- **Latent variables:**
  Some data $\mathbf{u} \in \mathcal{U}$ are unobserved, so that

$$p(\mathbf{y} \mid \boldsymbol{\theta}) = \int p(\mathbf{y}, \mathbf{u} \mid \boldsymbol{\theta}) d\mathbf{u}$$

$\Rightarrow$ The high dimensionality of $\mathbf{u}$ prevents the computation of the integral

Sources of intractability

An example is the [Kingman, 1982]'s Kingman (1982) coalescent process

From present genetic data $\mathbf{y}$, it reconstructs the sample past in a time-backward perspective

$\mathbf{y}$ is observed, but its past gene history is unobserved $\mathbf{u}$

We need to integrate over all possible past gene histories
$\Rightarrow$ Too many possible past can lead to $\mathbf{y}$ (intractable likelihood)

Sources of intractability

- **Normalising constant:**
  The likelihood is written as

$$p(\mathbf{y} \mid \boldsymbol{\theta}) = \frac{\widetilde{p}(\mathbf{y} \mid \boldsymbol{\theta})}{Z(\boldsymbol{\theta})}$$

  where $\widetilde{p}(\mathbf{y} \mid \boldsymbol{\theta})$ is the unnormalised likelihood, and
  $Z(\boldsymbol{\theta}) = \int \widetilde{p}(\mathbf{y} \mid \boldsymbol{\theta}) d\mathbf{y}$ its normalising constant

$\Rightarrow$ The high dimensionality of $\mathbf{y}$ prevents the computation of the normalising constant

(For example: Exponential random graph models for network data)

ABC framework

Approximate Bayesian Computation (ABC) only relies on data simulations

Hypothesis: Even though the likelihood is intractable, for a given parameter value, we assume that generating pseudo-data according to the model is possible (we talk about generative model)

Bayesian framework: we set a prior distribution on $\boldsymbol{\theta}$, denoted $p(\boldsymbol{\theta})$

Objective

$$p(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{y} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \propto p(\mathbf{y} \mid \boldsymbol{\theta})p(\boldsymbol{\theta}).$$

Outline

1 Basics of ABC

2 Tuning in ABC

3 Regression adjustment methods

Outline

1 Basics of ABC

2 Tuning in ABC

3 Regression adjustment methods

Basics of ABC

Goal of ABC:
Recovering simulations from the posterior distribution $p(\boldsymbol{\theta} \mid \mathbf{y})$

Idea of ABC:
Simulating parameters from $p(\boldsymbol{\theta})$, and keeping the ones able to generate pseudo-data $\mathbf{x}$ similar to $\mathbf{y}$

Relies only on simulations $\Rightarrow$ very flexible!

Exact rejection-sampling algorithm

The origins of ABC

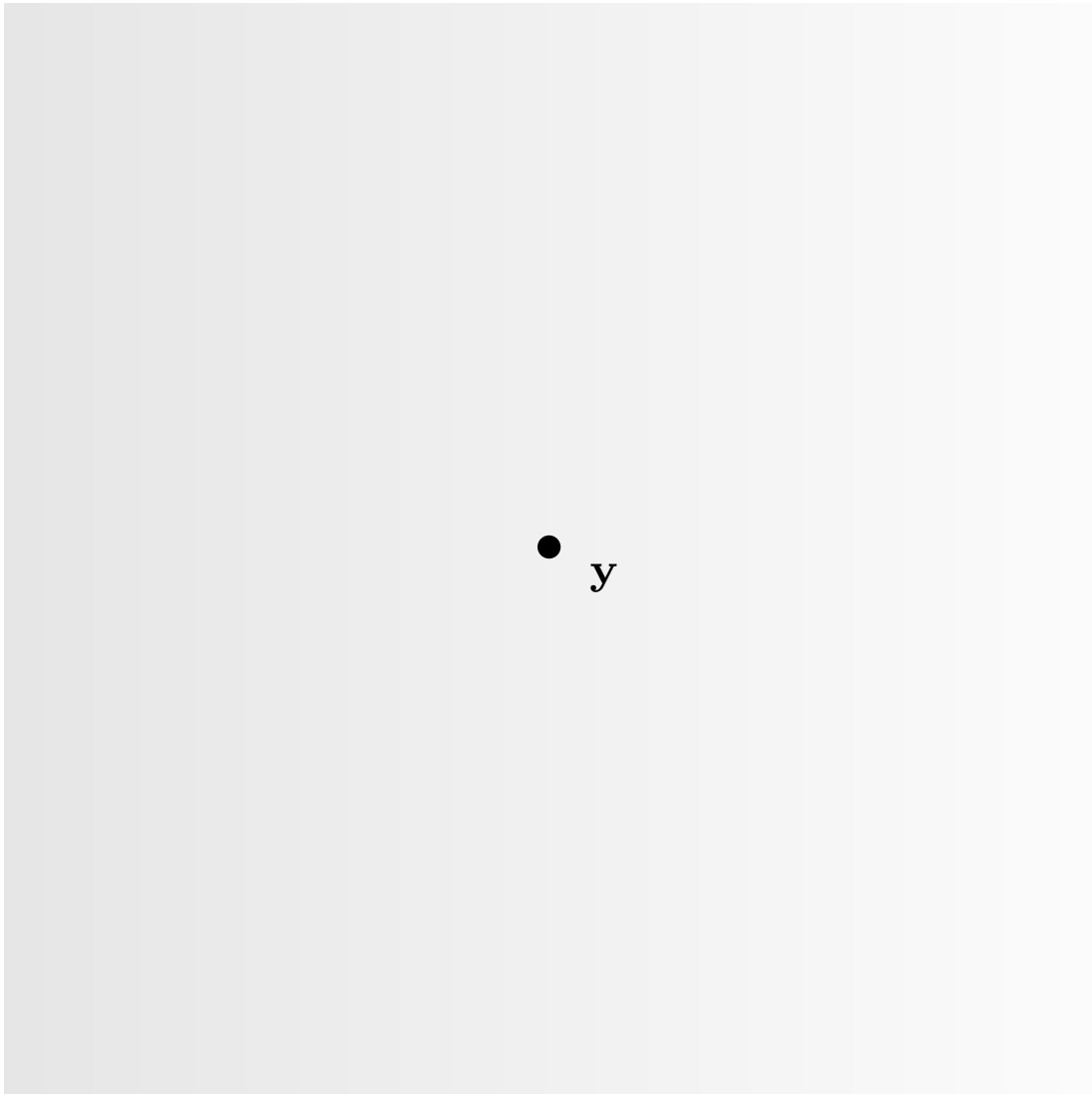
Algorithm 1: Exact rejection-sampling algorithm

for $i \leftarrow 1$ **to** N **do**
  repeat
    Simulate $\boldsymbol{\theta}^{(i)} \sim p(\cdot)$;
    Simulate $\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$;
  until $\mathbf{x}^{(i)} = \mathbf{y}$;
  Accept $(\boldsymbol{\theta}^{(i)}, \mathbf{x}^{(i)})$;
end

We retain parameters able to provide an exact match between $\mathbf{x}^{(i)}$ and $\mathbf{y}$

Exact rejection-sampling algorithm

$\boldsymbol{\theta}^{(i)} \sim p(.)$

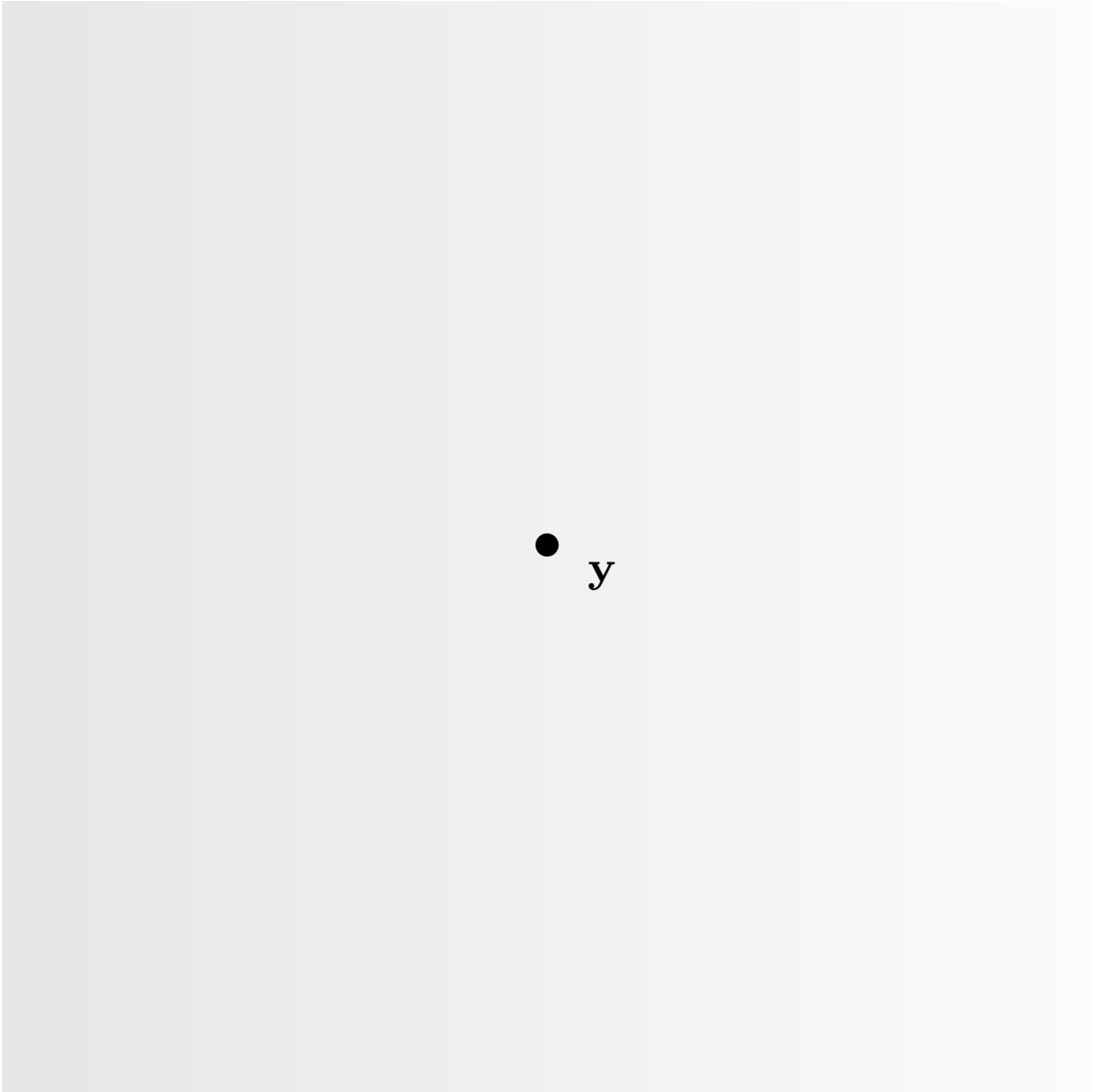


Exact rejection-sampling algorithm

$$\boldsymbol{\theta}^{(i)} \sim p(.)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

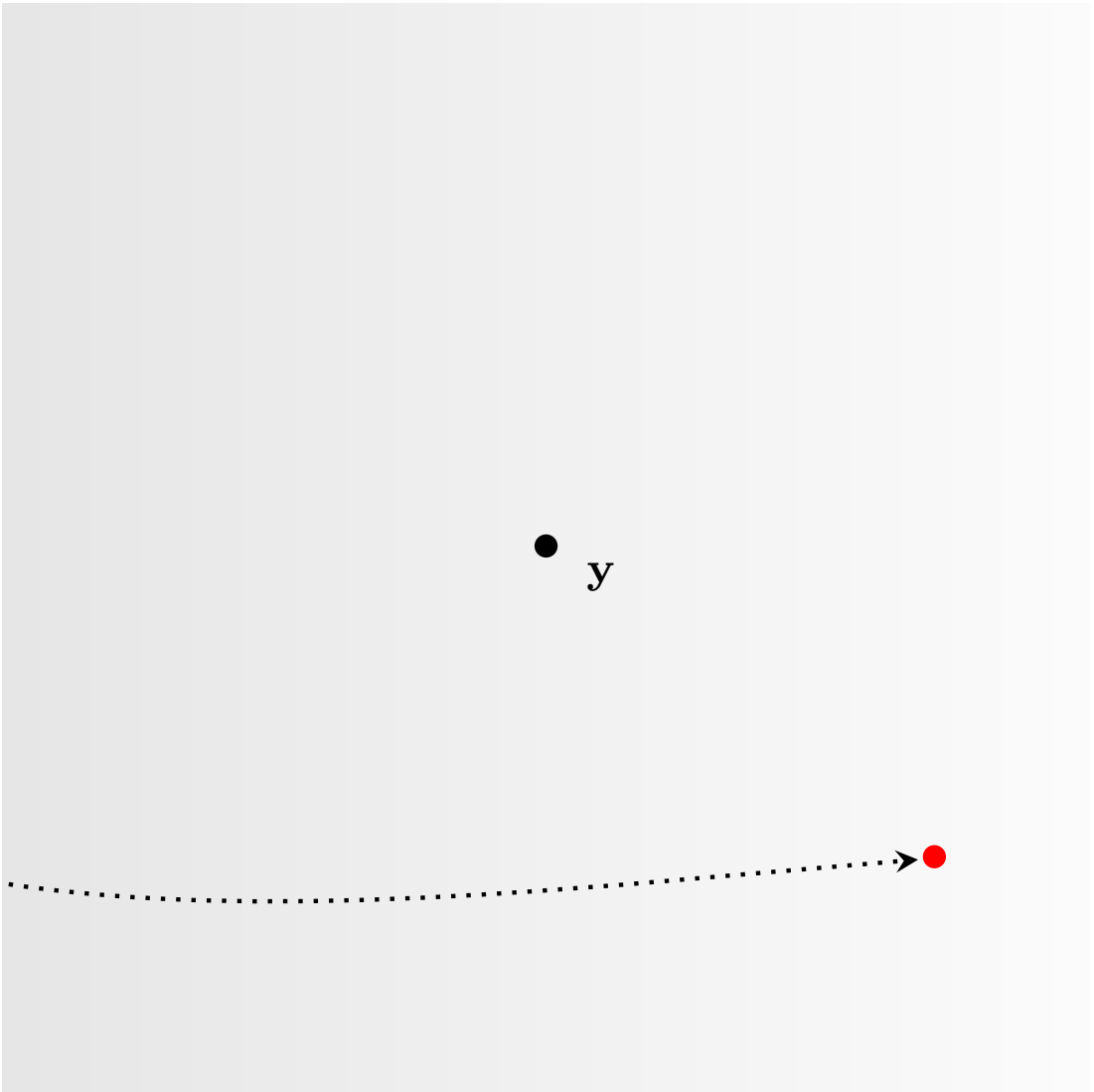


Exact rejection-sampling algorithm

$$\boldsymbol{\theta}^{(i)} \sim p(.)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

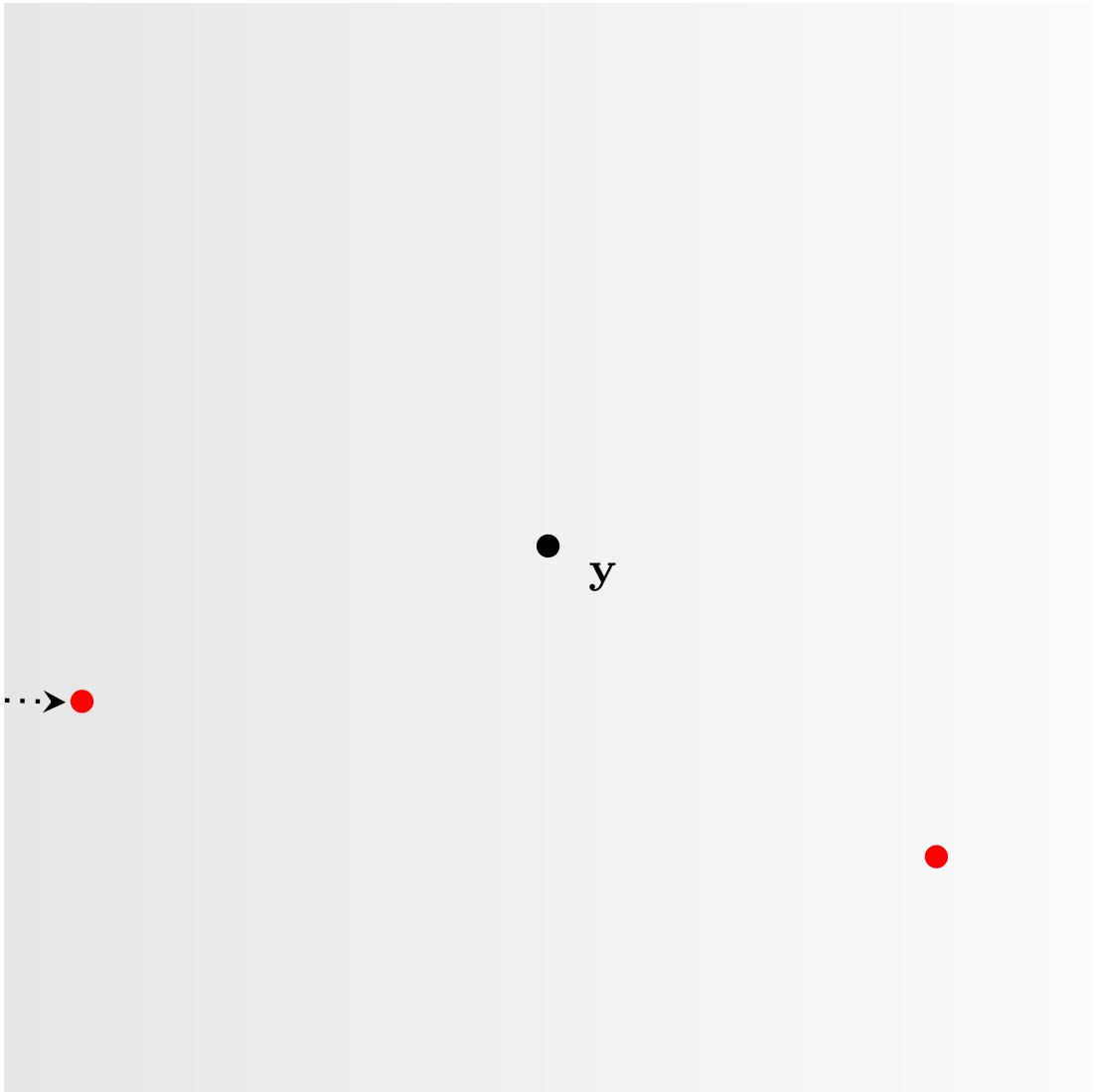


Exact rejection-sampling algorithm

$$\boldsymbol{\theta}^{(i)} \sim p(.)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

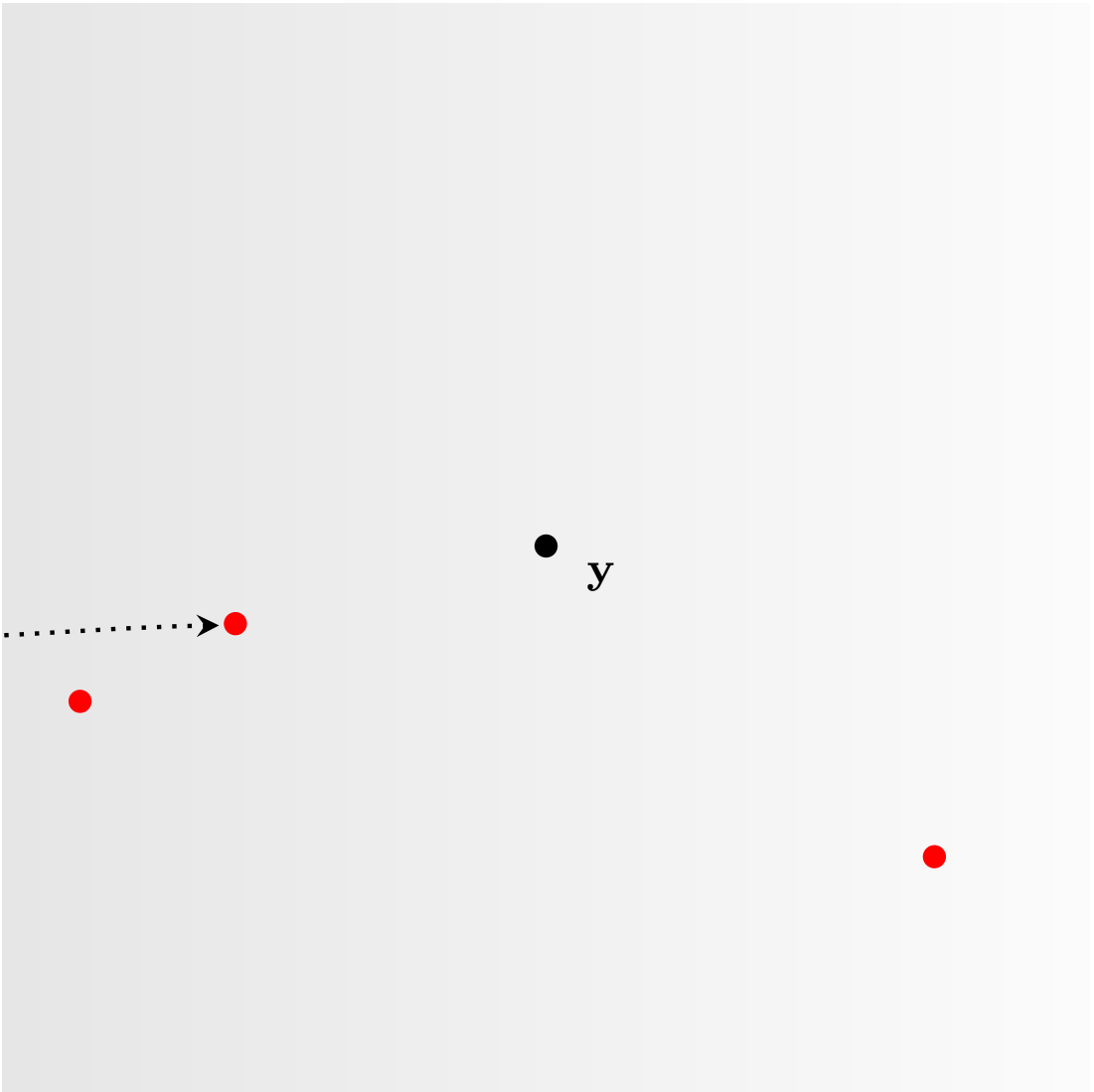


Exact rejection-sampling algorithm

$$\boldsymbol{\theta}^{(i)} \sim p(.)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

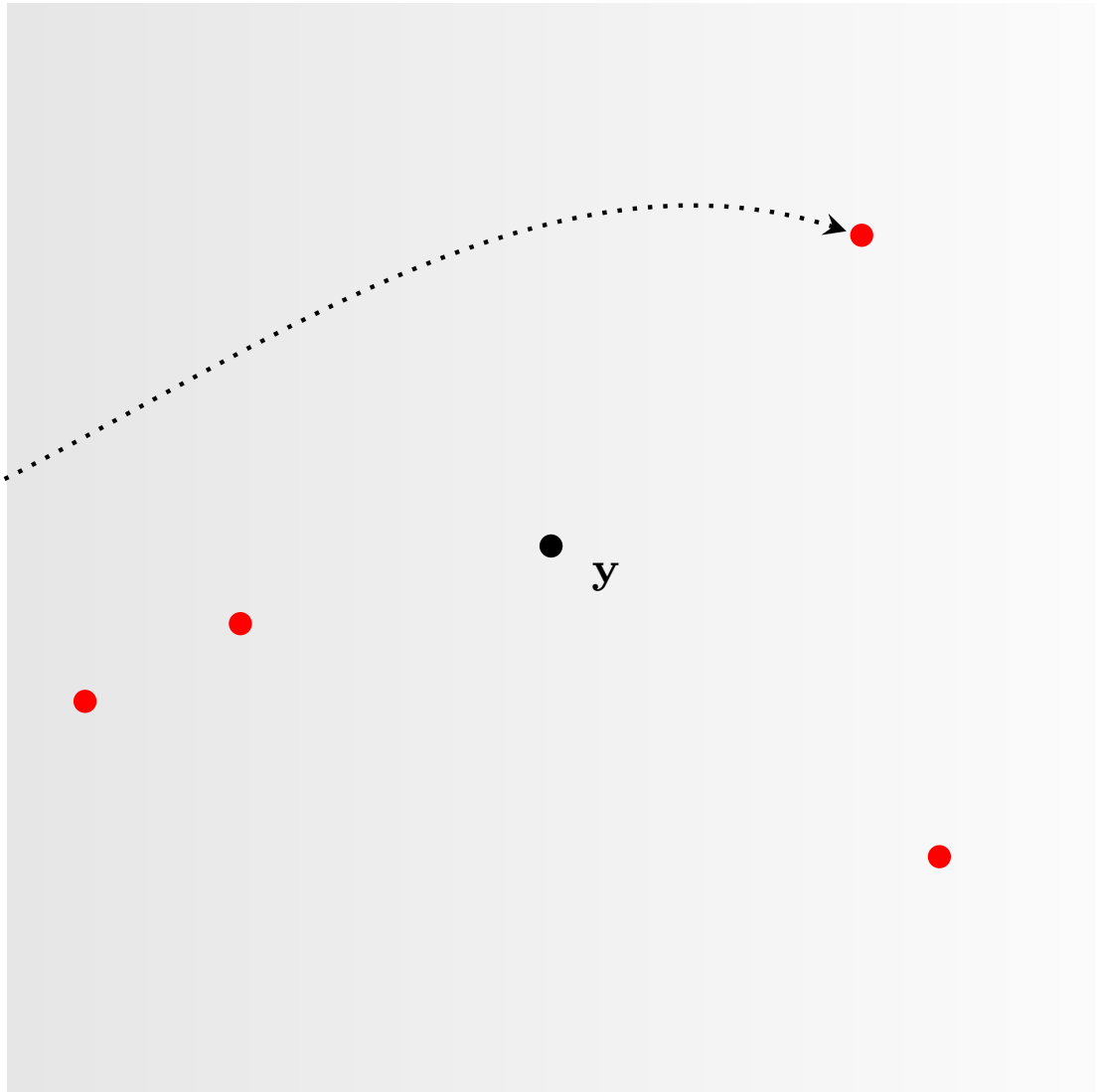


Exact rejection-sampling algorithm

$$\boldsymbol{\theta}^{(i)} \sim p(.)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

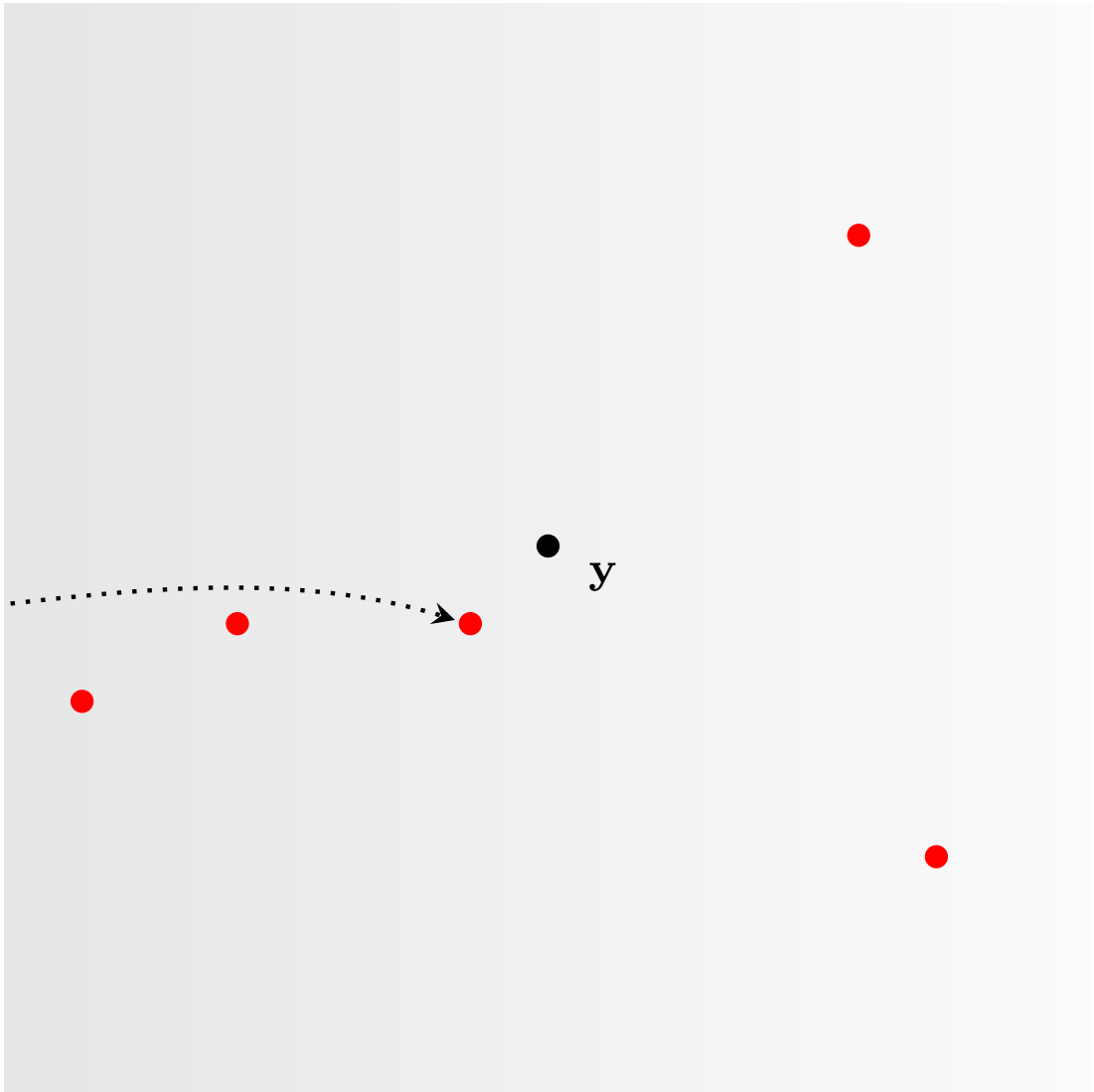


Exact rejection-sampling algorithm

$$\boldsymbol{\theta}^{(i)} \sim p(.)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

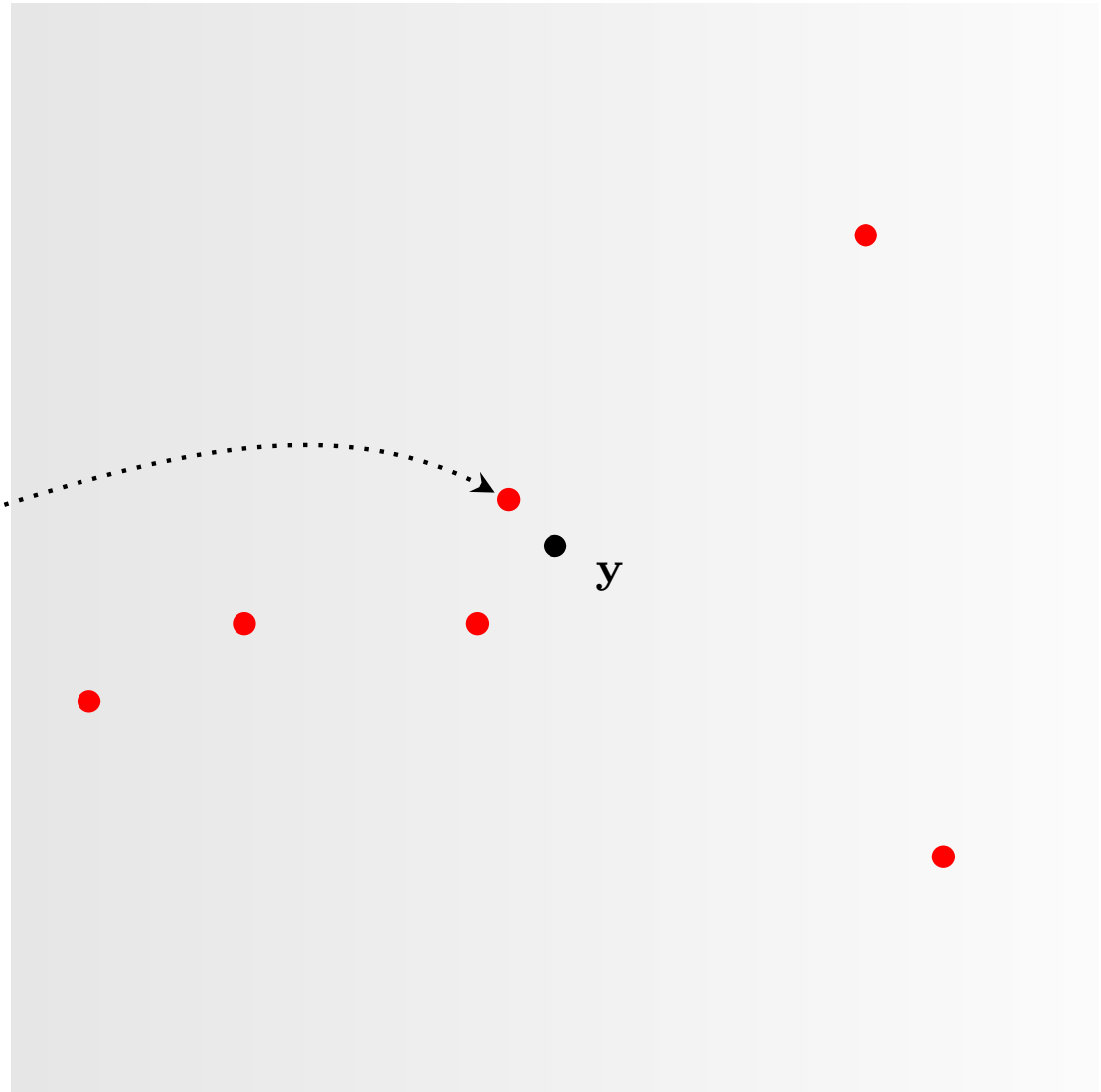


Exact rejection-sampling algorithm

$$\boldsymbol{\theta}^{(i)} \sim p(.)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

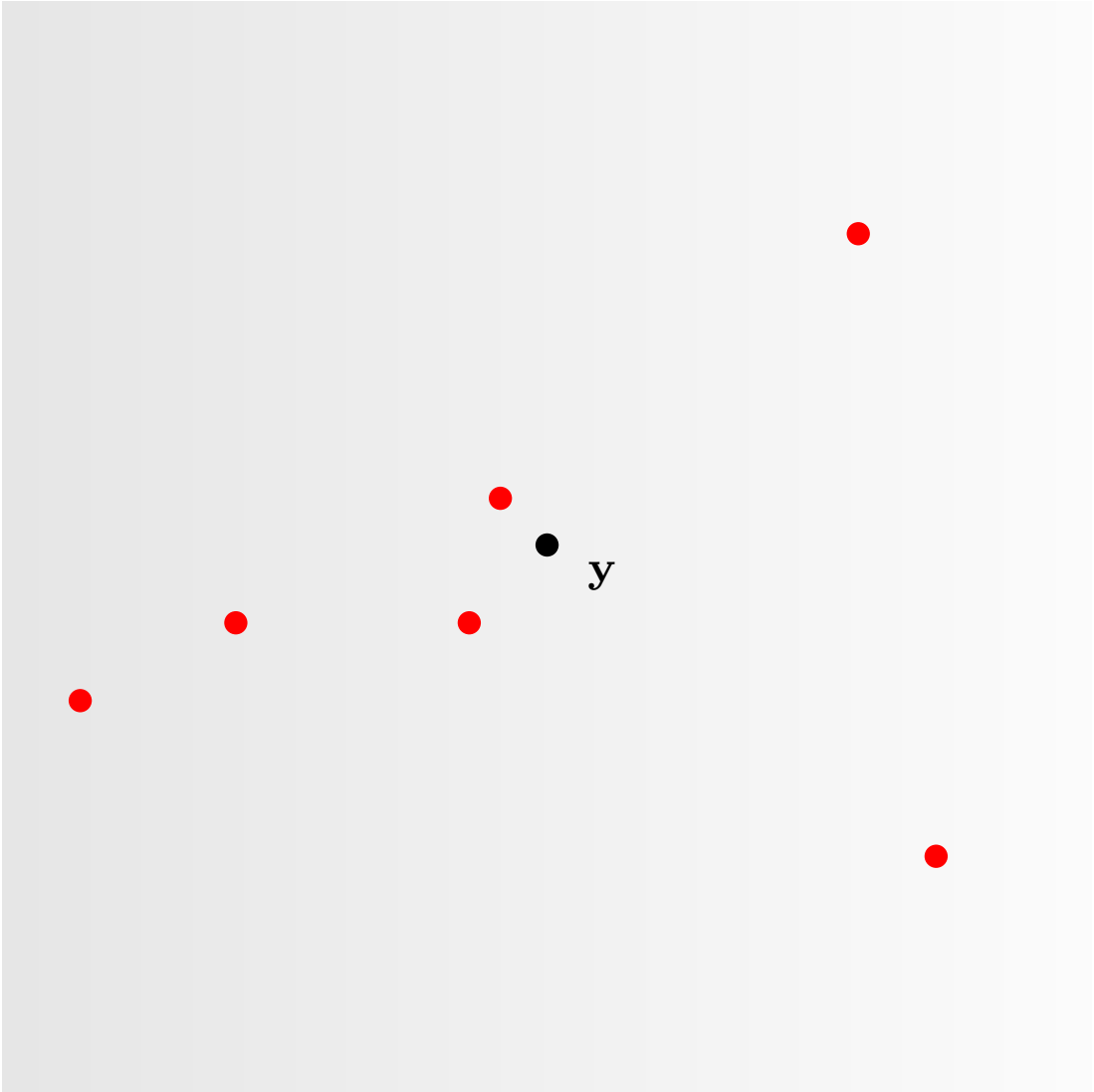


Exact rejection-sampling algorithm

$$\boldsymbol{\theta}^{(i)} \sim p(.)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$



Exact rejection-sampling algorithm

This sampling process recovers $(\boldsymbol{\theta}, \mathbf{x})$ values drawn from

$$\mathbb{1}\{\mathbf{x} = \mathbf{y}\}\, p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$$

When discarding the $\mathbf{x}$ values we obtain a N sample from the desired posterior

$$\int \mathbb{1}\{\mathbf{x} = \mathbf{y}\}\, p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\mathbf{x} = p(\mathbf{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$$

Unless $\mathbf{y}$ is discrete and low dimensional, this algorithm is inefficient as we rarely have the exact match $\mathbf{x} = \mathbf{y}$

Basics of ABC

Instead, we keep pseudo-data that are close enough to $\mathbf{y}$

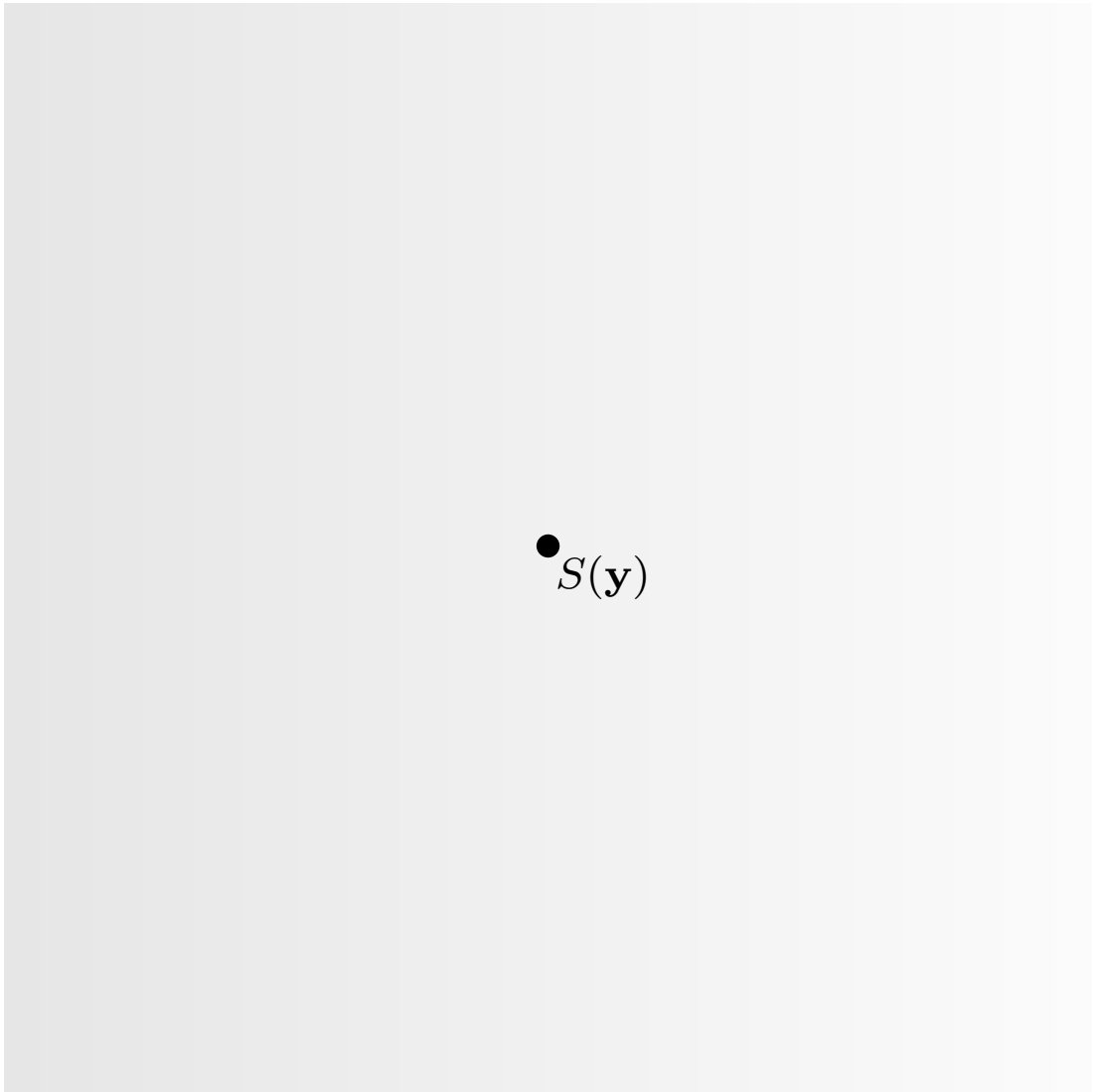
To facilitate the comparison between data, we project them on a lower dimensional space thanks to a set of d summary statistics $S(.)$

Basics of ABC

$$\boldsymbol{\theta}^{(i)} \sim p(\cdot)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$



Basics of ABC

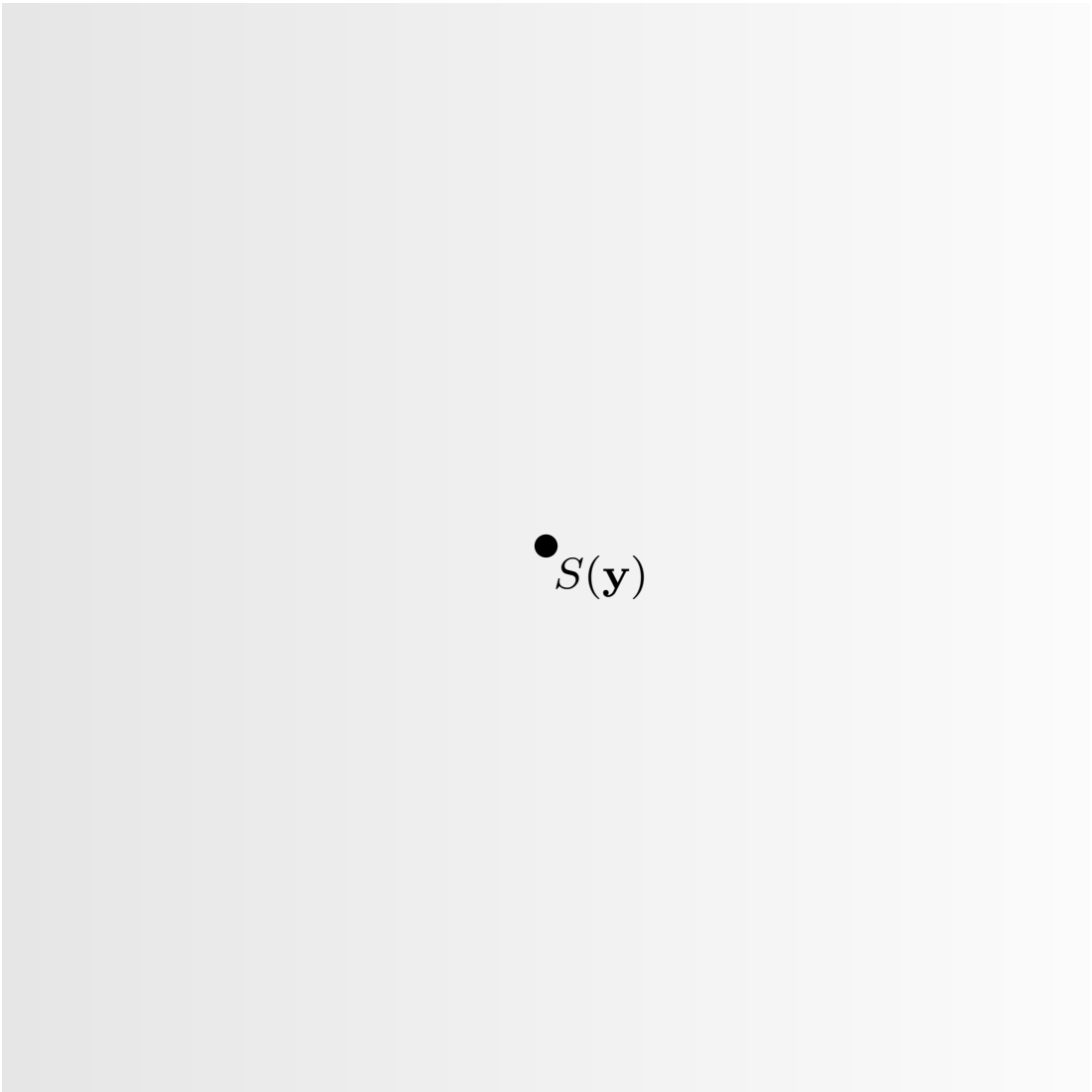
$$\boldsymbol{\theta}^{(i)} \sim p(\cdot)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

$$\downarrow$$

$$S(\mathbf{x}^{(i)})$$



Basics of ABC

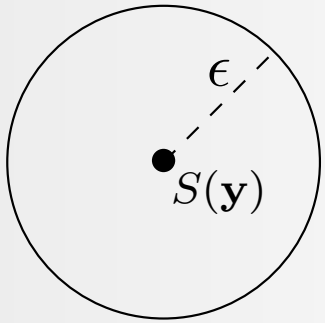
$$\boldsymbol{\theta}^{(i)} \sim p(\cdot)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

$$\downarrow$$

$$S(\mathbf{x}^{(i)})$$



Basics of ABC

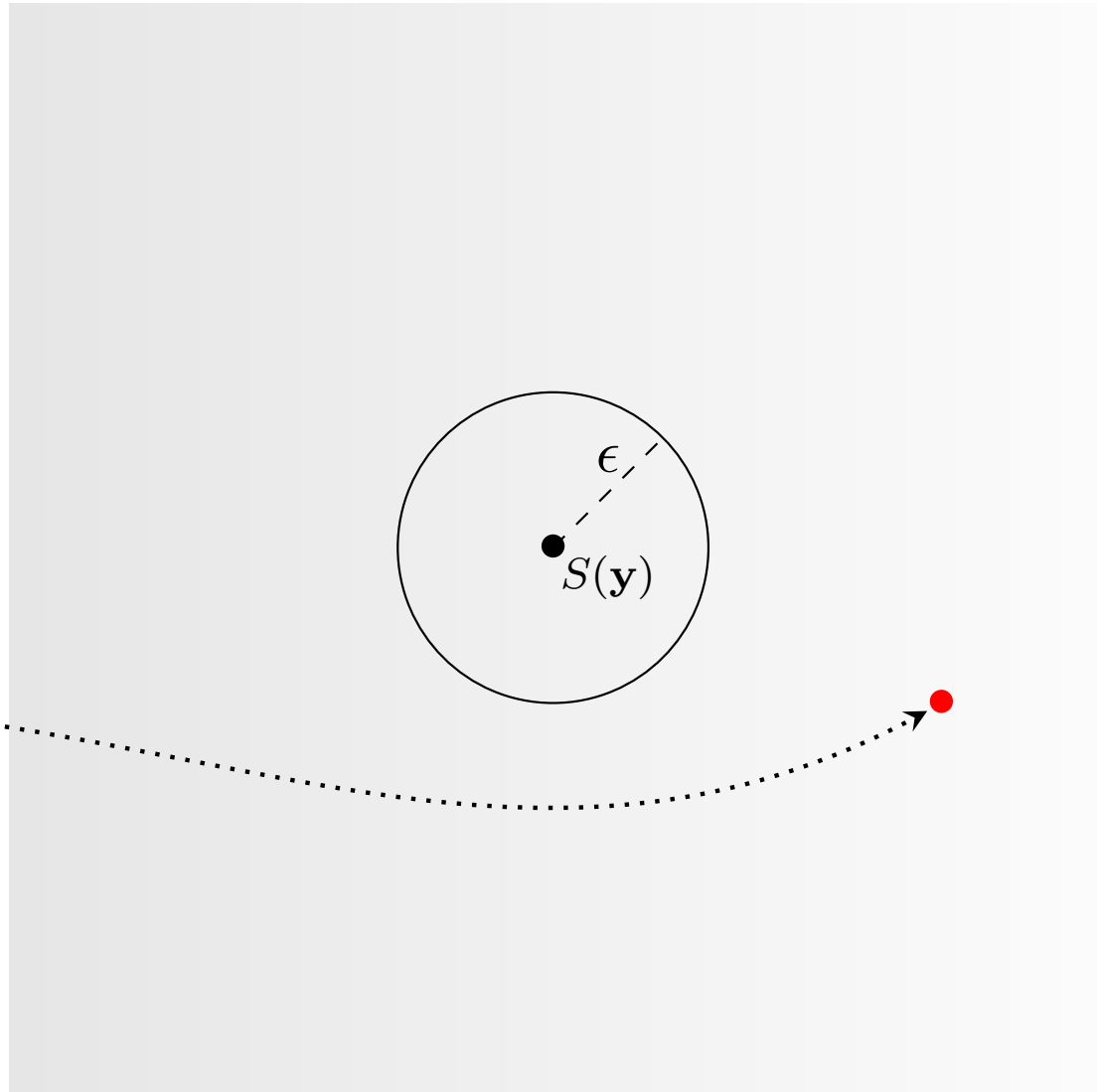
$$\boldsymbol{\theta}^{(i)} \sim p(\cdot)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

$$\downarrow$$

$$S(\mathbf{x}^{(i)})$$



Basics of ABC

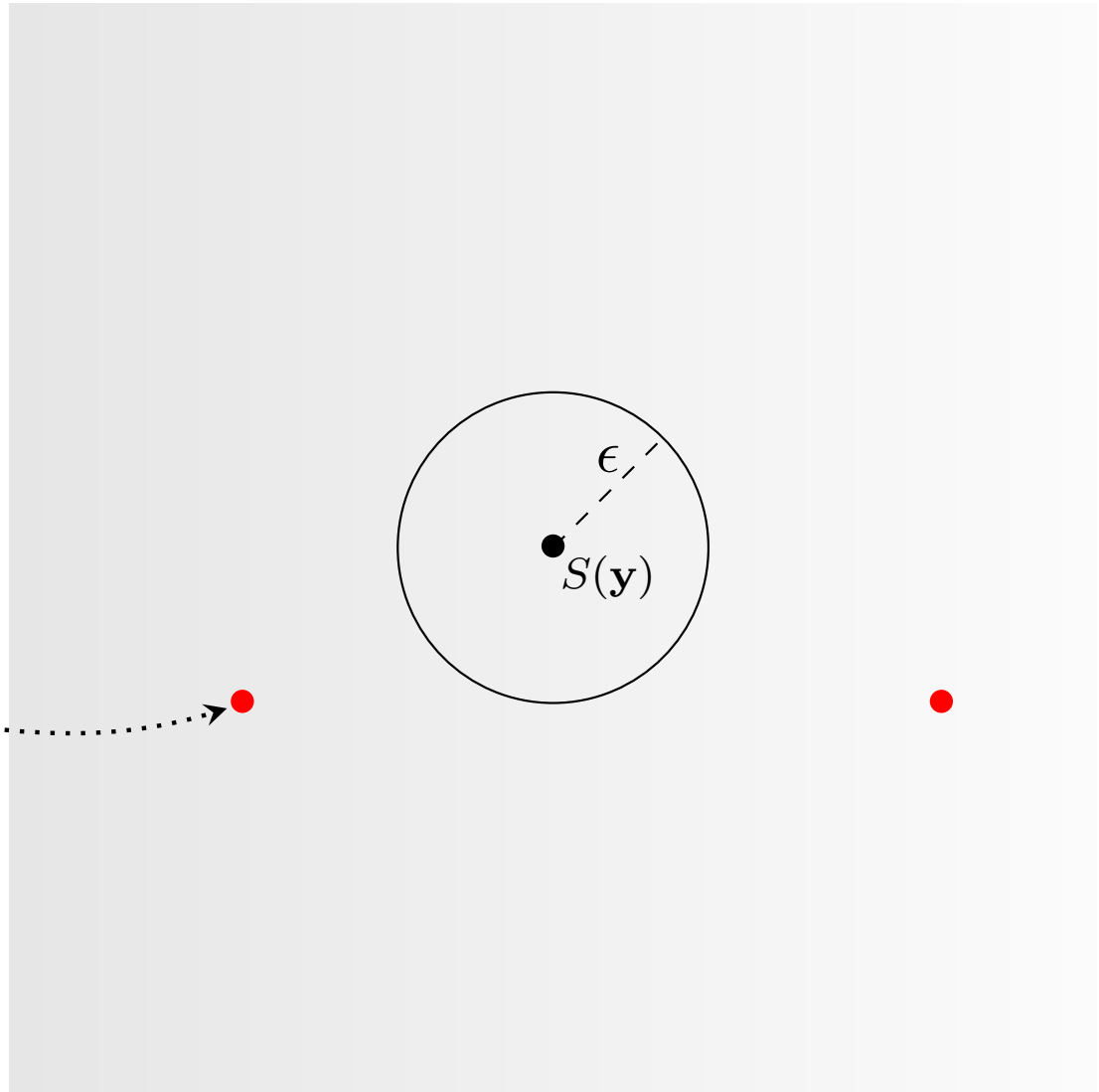
$$\boldsymbol{\theta}^{(i)} \sim p(\cdot)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

$$\downarrow$$

$$S(\mathbf{x}^{(i)})$$



Basics of ABC

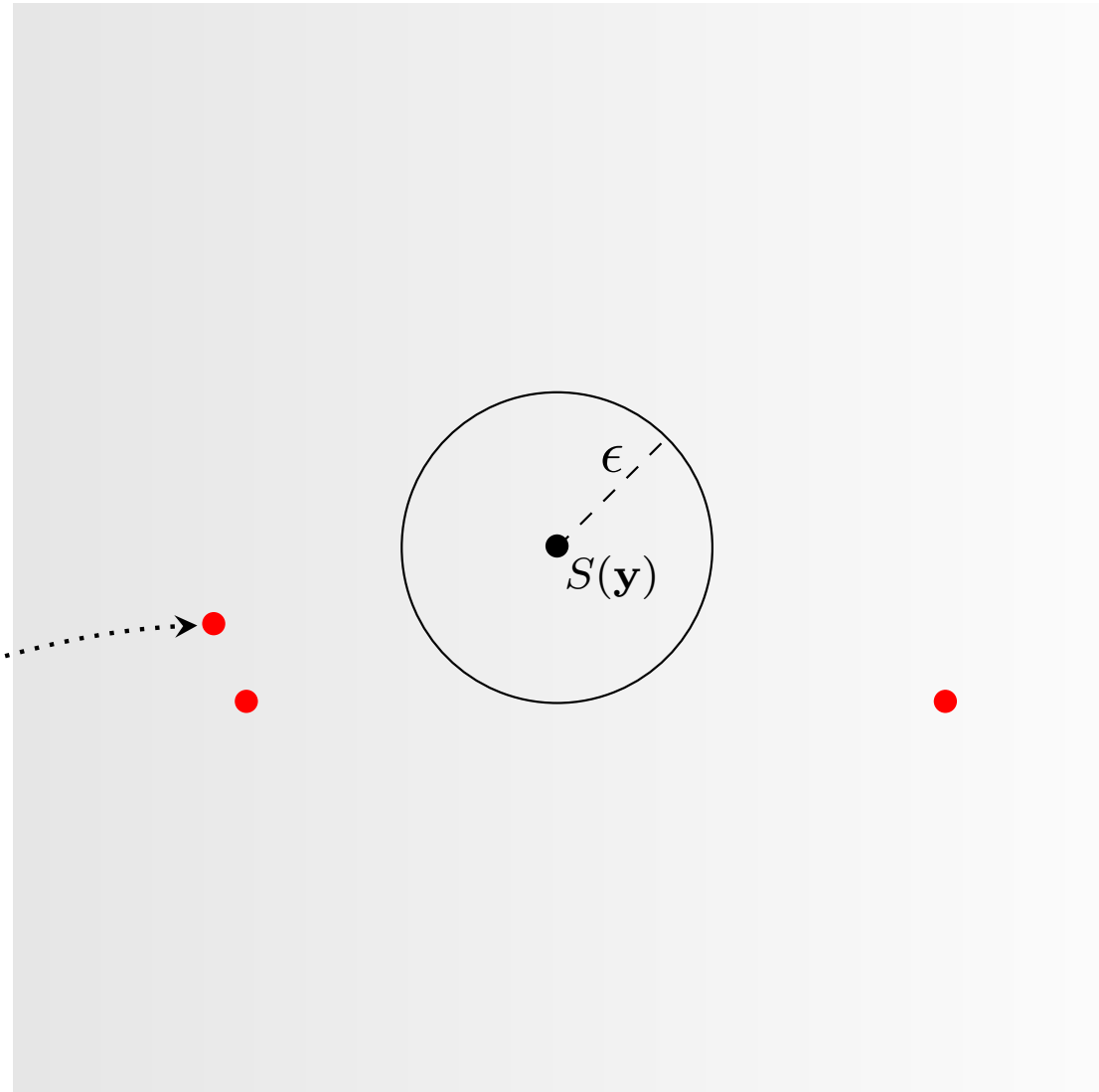
$$\boldsymbol{\theta}^{(i)} \sim p(\cdot)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

$$\downarrow$$

$$S(\mathbf{x}^{(i)})$$



Basics of ABC

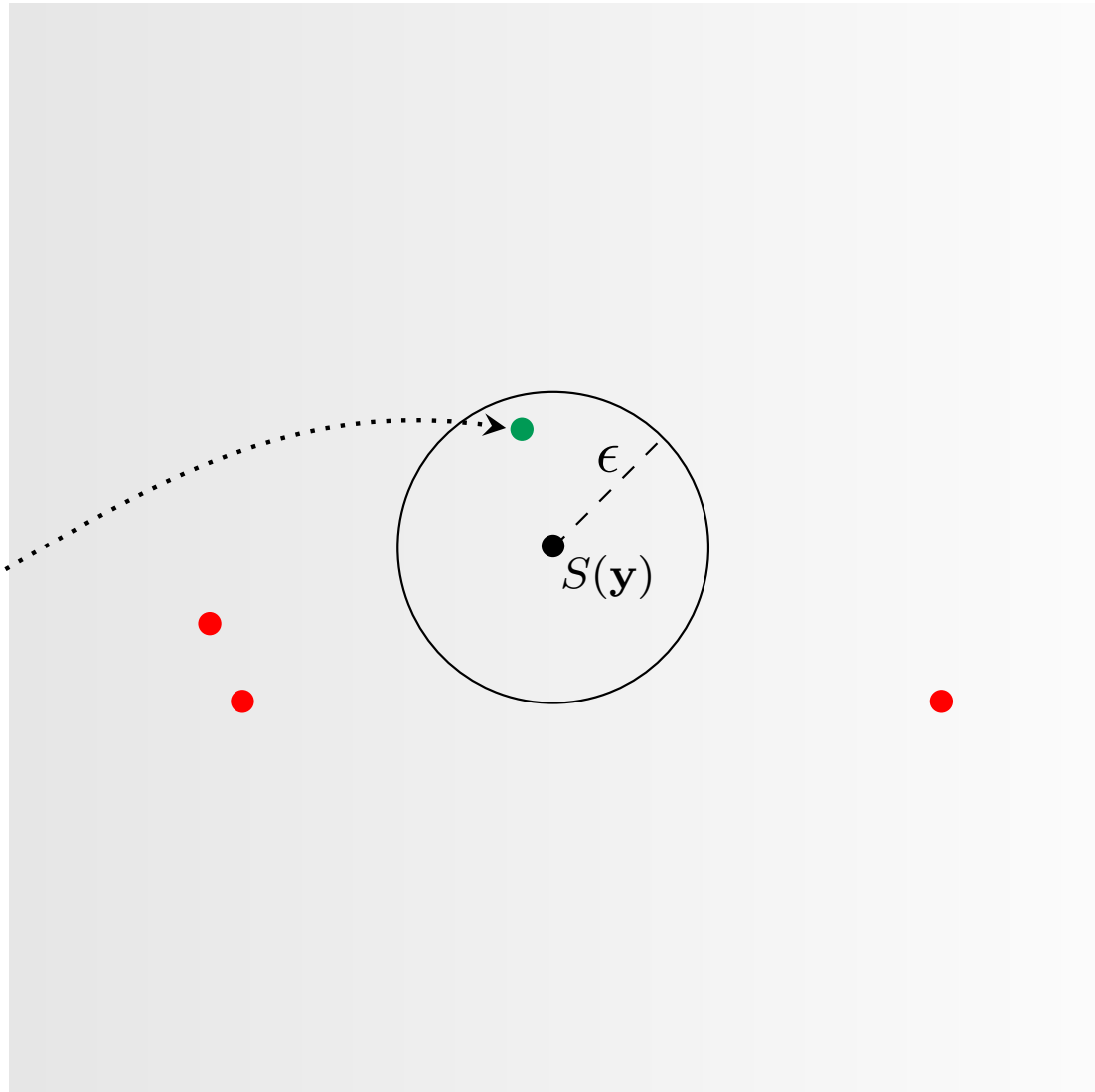
$$\boldsymbol{\theta}^{(i)} \sim p(\cdot)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

$$\downarrow$$

$$S(\mathbf{x}^{(i)})$$



Basics of ABC

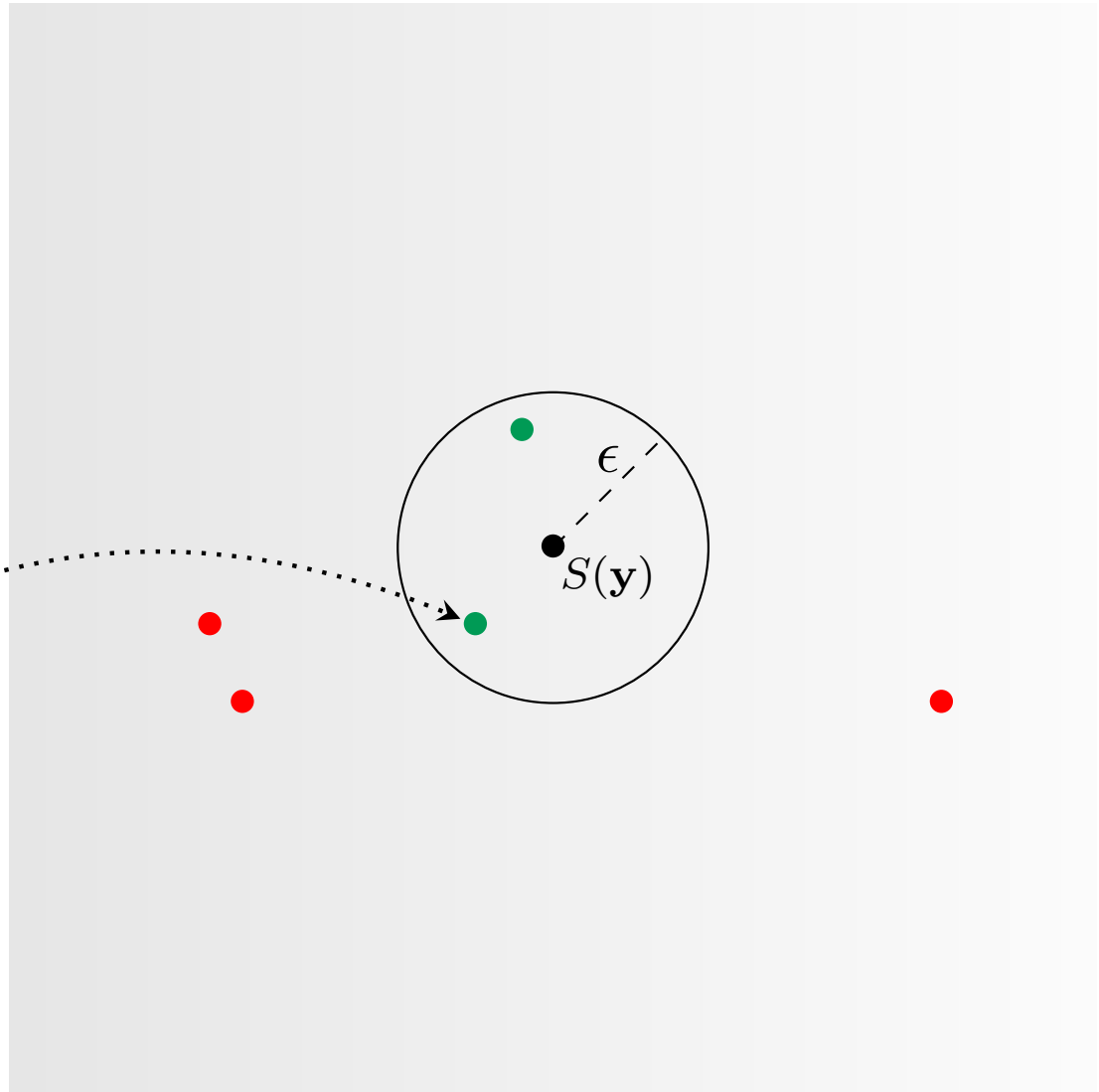
$$\boldsymbol{\theta}^{(i)} \sim p(\cdot)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

$$\downarrow$$

$$S(\mathbf{x}^{(i)})$$



Basics of ABC

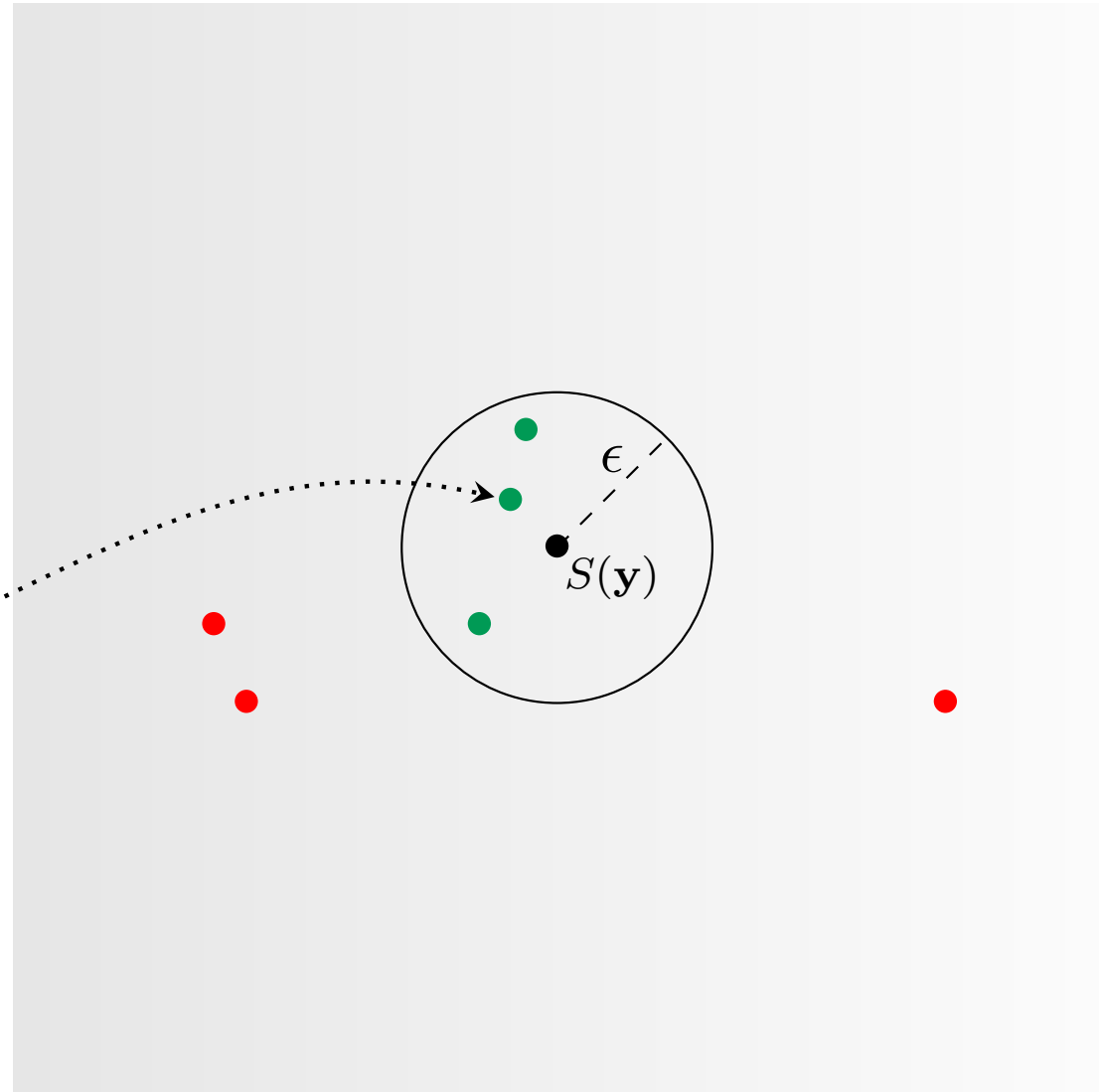
$$\boldsymbol{\theta}^{(i)} \sim p(\cdot)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

$$\downarrow$$

$$S(\mathbf{x}^{(i)})$$



Basics of ABC

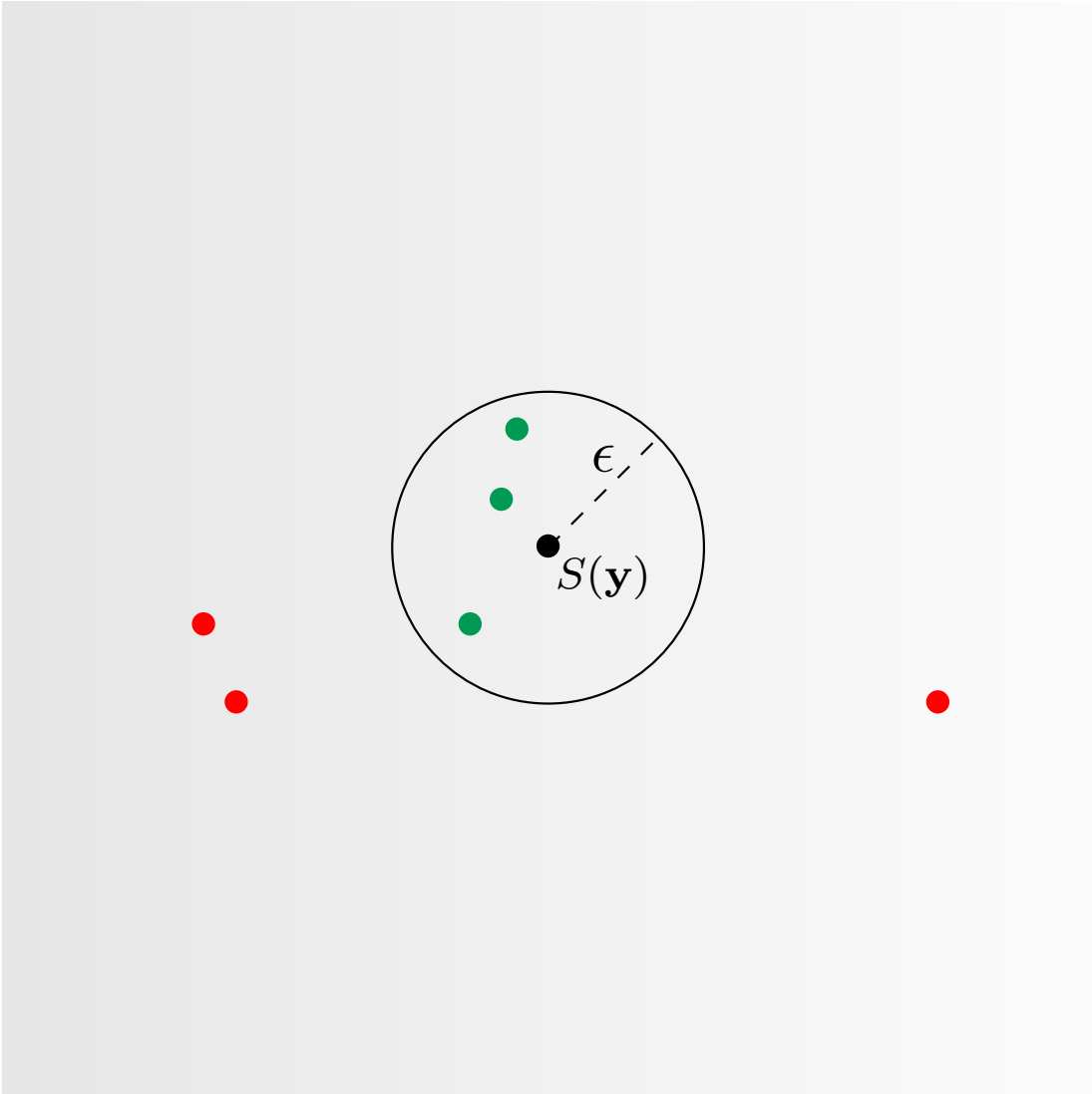
$$\boldsymbol{\theta}^{(i)} \sim p(\cdot)$$

$$\downarrow$$

$$\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$$

$$\downarrow$$

$$S(\mathbf{x}^{(i)})$$



Basics of ABC

(In the remaining I use $S(\mathbf{x}) = S_{\mathbf{x}}$)

Algorithm 2: Basic ABC rejection sampling [Pritchard et al., 1999]

for $i \leftarrow 1$ **to** N **do**
  repeat
    Simulate $\boldsymbol{\theta}^{(i)} \sim p(\cdot)$;
    Simulate $\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$;
    Compute $S_{\mathbf{x}^{(i)}}$;
  until $\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}}) \leq \epsilon$;
  Accept $(\boldsymbol{\theta}^{(i)}, S_{\mathbf{x}^{(i)}})$;
end

Seminal papers: [Tavaré et al., 1997], [Weiss and von Haeseler, 1998] [Pritchard et al., 1999]

The A of ABC

Two approximation aspects in ABC (outside the Monte Carlo approximation)

1. The summary statistics are rarely sufficient, thus the target posterior becomes $p(\boldsymbol{\theta} \mid S_{\mathbf{y}})$

2. The similarity bwn observed and simulated data is measured thanks to a distance $\rho(\cdot, \cdot)$ and an acceptance threshold ϵ, leading to an approximated posterior $p_{\rho,\epsilon}(\boldsymbol{\theta} \mid S_{\mathbf{y}})$

Basics of ABC

The sampling procedure draws $(\boldsymbol{\theta}, S_{\mathbf{x}})$ values from the joint posterior

$$p_{\rho,\epsilon}(\boldsymbol{\theta}, S_{\mathbf{x}} \mid S_{\mathbf{y}}) \propto \mathbb{1}\left\{\rho(S_{\mathbf{x}}, S_{\mathbf{y}}) \leq \epsilon\right\} p(S_{\mathbf{x}} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$$

When discarding $S_{\mathbf{x}}$ values, it returns N parameter values drawn from the approximated distribution

$$p_{\rho,\epsilon}(\boldsymbol{\theta} \mid S_{\mathbf{y}}) \propto \int \mathbb{1}\left\{\rho(S_{\mathbf{x}}, S_{\mathbf{y}}) \leq \epsilon\right\} p(S_{\mathbf{x}} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) dS_{\mathbf{x}}$$

ABC can be seen as providing an approximation of the likelihood which is used to perform regular Bayesian inference

$$p_{\rho,\epsilon}(\boldsymbol{\theta} \mid S_{\mathbf{y}}) \propto \underbrace{\int \mathbb{1}\left\{\rho(S_{\mathbf{x}}, S_{\mathbf{y}}) \leq \epsilon\right\} p(S_{\mathbf{x}} \mid \boldsymbol{\theta}) dS_{\mathbf{x}}}_{\mathbb{P}(\mathbb{1}\{\rho(S_{\mathbf{x}}, S_{\mathbf{y}}) \leq \epsilon\} \mid \boldsymbol{\theta})} p(\boldsymbol{\theta})$$

Basics of ABC

The idea behind ABC is that using relevant summary statistics with a small tolerance level should provide a good approximation of the posterior distribution:

$$p_{\rho,\epsilon}(\boldsymbol{\theta} \mid S_{\mathbf{y}}) \approx p(\boldsymbol{\theta} \mid \mathbf{y})$$

If $\epsilon \to 0$ and the summary statistics are sufficient, then ABC samples from the exact posterior distribution

If $\epsilon \to \infty$ we recover the prior distribution

$\Rightarrow$ ϵ reflects the tension between computational cost and accuracy

Hands on a toy example

Model:
$\mathbf{y} \mid \boldsymbol{\theta} \sim \mathcal{N}_1(\boldsymbol{\theta}, \sigma)$ with $\sigma^2 = 2$

Prior:
$\boldsymbol{\theta} \sim \mathcal{N}_1(\mu, \tau)$ with $\mu = 3$ and $\tau^2 = 10$

Observed data:
$\mathbf{y} = 8$

Distance:
$\rho(\mathbf{x}, \mathbf{y}) = |\mathbf{x} - \mathbf{y}|$

ABC in practice

In practice, the basic ABC algorithm can be extremely expensive in time for small ϵ
(the rate of acceptance is quasi-zero and a large number of simulations is required)

[Beaumont et al., 2002] introduced a more practical ABC version

ABC in practice

Algorithm 3: Weighted ABC sampler [Beaumont et al., 2002]

for $i \leftarrow 1$ **to** N **do**

  Simulate $\boldsymbol{\theta}^{(i)} \sim p(\cdot)$;

  Simulate $\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$;

  Compute $S_{\mathbf{x}^{(i)}}$ and $\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}})$;

  Assign to $(\boldsymbol{\theta}^{(i)}, S_{\mathbf{x}^{(i)}})$ a weight $w^{(i)} \propto K_\epsilon(\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}}))$;

end

$\Rightarrow$ The resulting weighed sample can be used to obtain a kernel approximation of the posterior distribution, i.e.

$$\hat{p}(\boldsymbol{\theta} \mid S_{\mathbf{y}}) = \frac{\sum_{i=1}^{N} K_b(\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}) K_\epsilon(\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}}))}{\sum_{i=1}^{N} K_\epsilon(\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}}))},$$

where K_b is a density-estimation kernel with bandwidth b, usually different from K_ϵ

ABC seen as a k-nearest neighbours

A common choice is the Epanechnikov (quadratic) kernel

If we consider an uniform kernel:

$$K_\epsilon(\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}})) \propto \mathbb{1}\{\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}}) \leq \epsilon\}$$

and ϵ is chosen equal to a certain quantile of the distances $\{\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}})\}_{i=1,\ldots,N}$ so that only k simulations have non-zero weights

The previous algorithm can be seen as a k-NN algorithm, where selecting ϵ implies choosing the number of neighbours to $S_{\mathbf{y}}$

See [Biau et al., 2015] for this vision of ABC

ABC seen as a k-nearest neighbours

ABC can be reformulated as in the following pseudo-code

Algorithm 4: ABC in practice

for $i \leftarrow 1$ **to** N **do**
  Simulate $\boldsymbol{\theta}^{(i)} \sim p(\cdot)$;
  Simulate $\mathbf{x}^{(i)} \sim p(\cdot \mid \boldsymbol{\theta}^{(i)})$ and compute $S_{\mathbf{x}^{(i)}}$;
  Calculate $\rho^{(i)} = \rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}})$;
end
Order the distances $\rho^{(1)}, \ldots, \rho^{(N)}$;
Accept the $(\boldsymbol{\theta}^{(i)}, S_{\mathbf{x}^{(i)}})$ that correspond to the k-smallest distances;

$\Rightarrow$ intuitive, simple to implement, easily parallelisable

ABC reference table

There is an ABC - Machine Learning perspective where a reference table is simulated

$$\begin{bmatrix} \boldsymbol{\theta}^{(1)} & S_{\mathbf{x}^{(1)},1} & S_{\mathbf{x}^{(1)},2} & \cdots & S_{\mathbf{x}^{(1)},d} \\ \boldsymbol{\theta}^{(2)} & S_{\mathbf{x}^{(2)},1} & S_{\mathbf{x}^{(2)},2} & \cdots & S_{\mathbf{x}^{(2)},d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\theta}^{(N)} & S_{\mathbf{x}^{(N)},1} & S_{\mathbf{x}^{(N)},2} & \cdots & S_{\mathbf{x}^{(N)},d} \end{bmatrix}$$

and a supervised machine learning algorithm is trained using this artificially simulated training data set (for example a k-NN algorithm)

Toy example from Richard Wilkinson (Tutorial on ABC, NIPS 2013)

Model:
$\mathbf{y} \mid \boldsymbol{\theta} \sim \mathcal{N}_1\left(2(\boldsymbol{\theta}+2)\boldsymbol{\theta}(\boldsymbol{\theta}-2), 0.1+\boldsymbol{\theta}^2\right)$

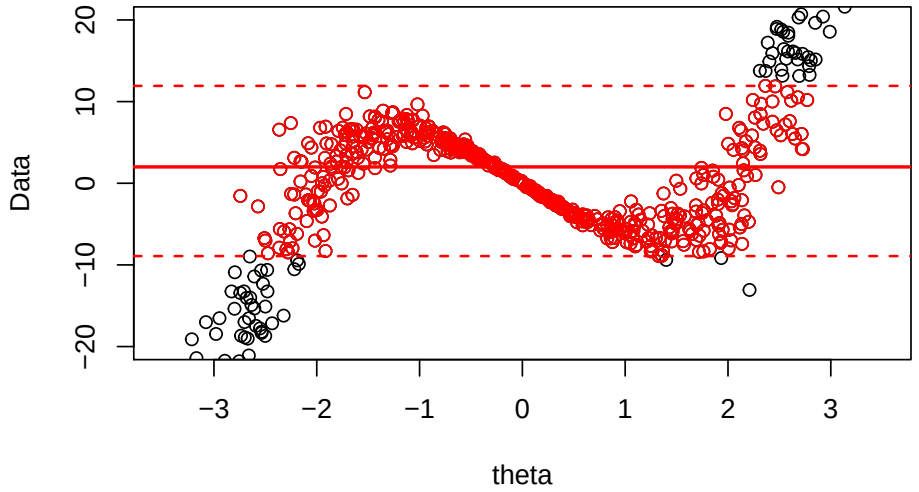
Prior:
$\boldsymbol{\theta} \sim \mathcal{U}_{[-10,10]}$

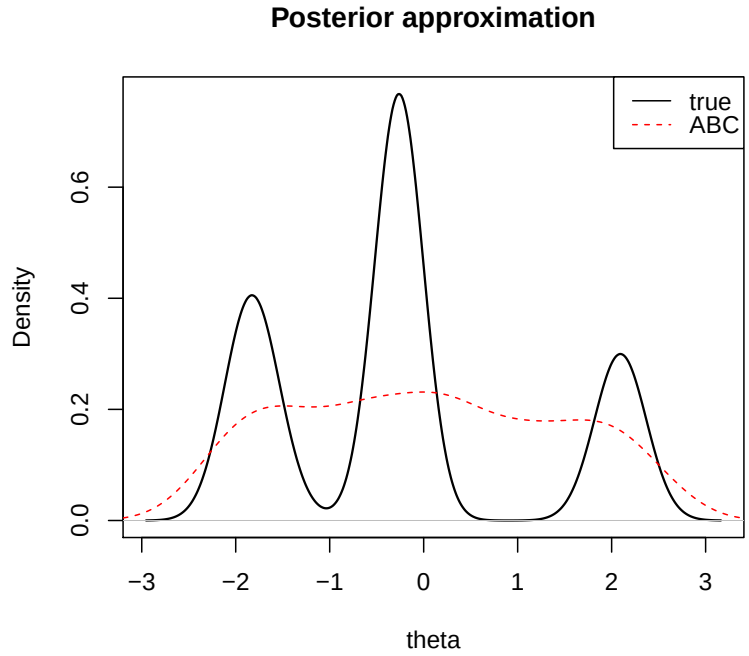
Observed data:
$\mathbf{y} = 2$

Distance:
$\rho(\mathbf{x}, \mathbf{y}) = |\mathbf{x} - \mathbf{y}|$

Number of simulations:
$N = 2000$

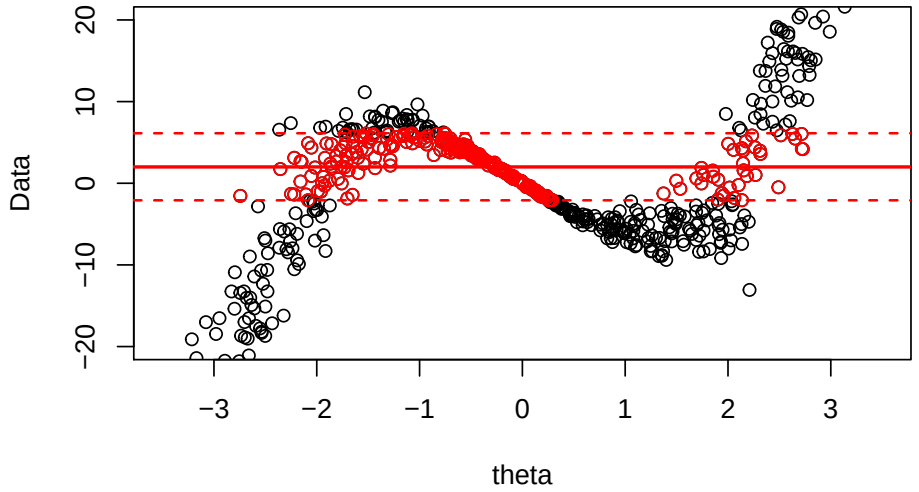
Toy example

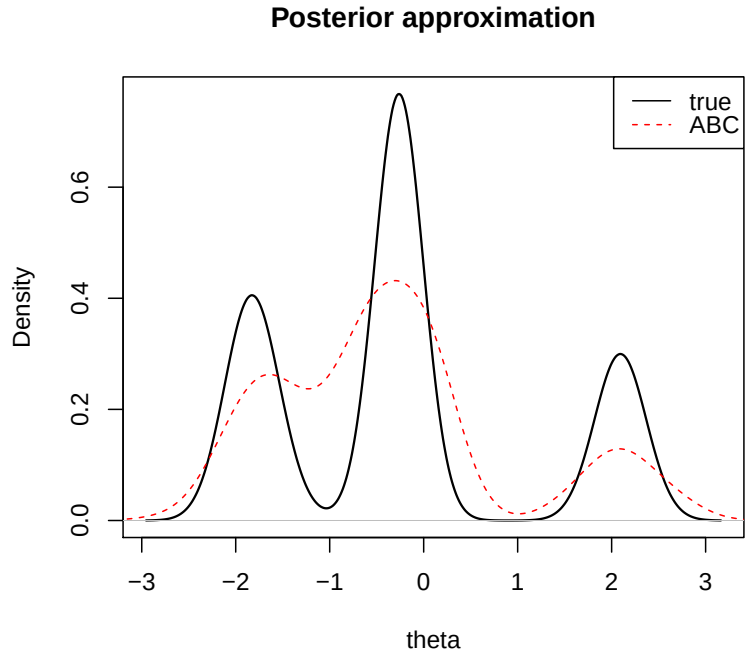




$$k = 500$$

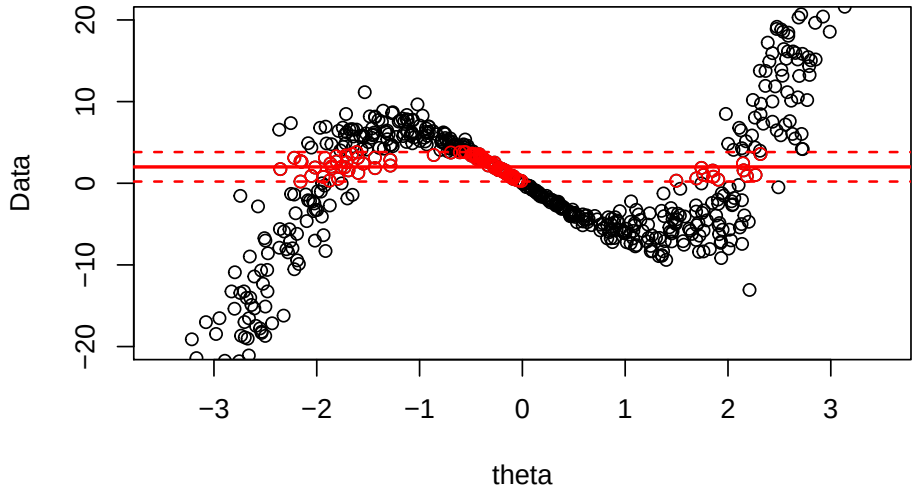
Toy example

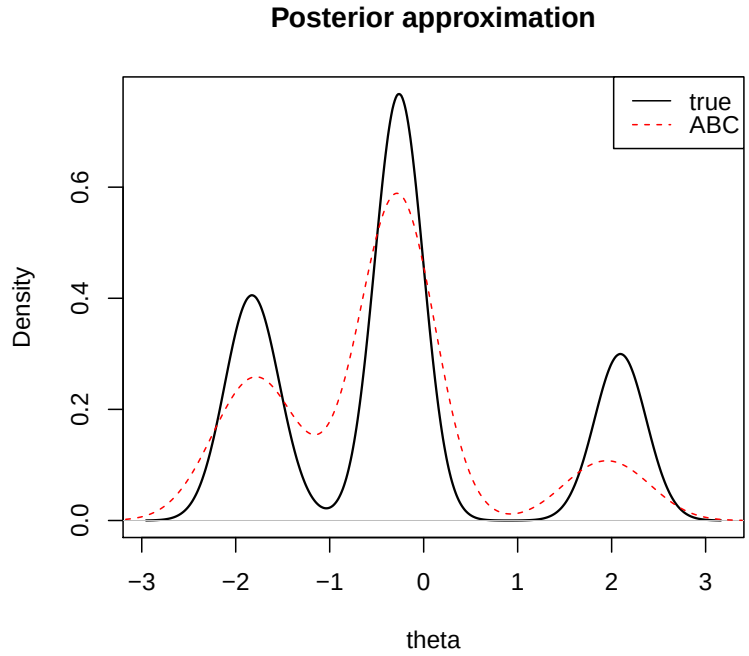




$$k = 250$$

Toy example





$$k = 100$$

Curse of dimensionality

When the number of summary statistics d increases, the ABC algorithm performances greatly deteriorate
$\Rightarrow$ It is difficult to have $\rho(S_{\mathbf{x}}, S_{\mathbf{y}}) \leq \epsilon$
$\Rightarrow$ The distance between (pseudo and real) data increases with the number of summary statistics

The number of summary statistics should be low, but still relevant!
[Fearnhead and Prangle, 2012]

Outline

1 Basics of ABC

2 Tuning in ABC

3 Regression adjustment methods

Tuning in ABC

The major drawbacks of ABC are the tuning aspects

ABC requires to choose

- a distance ρ
- a tolerance ϵ
- a set of summary statistics S

These choices impact the approximation of the posterior distribution

Tuning in ABC: distance ρ

Because different summary statistics can present different spread and correlation, the distance should include scaling weights

A common choice is the Euclidean distance, normalised by the empirical mean absolute deviation [Csilléry et al., 2012] or standard deviation [Beaumont et al., 2002]: suppose d summary statistics are available, then a possible choice of weighted distance can be:

$$\rho(S_{\mathbf{x}}, S_{\mathbf{y}}) = \left[\sum_{i=1}^{d} \left(\frac{S_{\mathbf{x},i} - S_{\mathbf{y},i}}{\sigma_i} \right)^2 \right]^{\frac{1}{2}}$$

The scaling weights are often deduced from a preliminary ABC run

For more information see [Prangle, 2017]

Tuning in ABC: summary statistics S

The choice of S is the most studied aspect of ABC and different strategies exist concerning the choice of S:

- Selection
- Projection
- Indirect inference
- Regularisation

For a large review about this subject see
[Beaumont, 2010], [Blum et al., 2013], [Prangle, 2018]

Tuning in ABC: summary statistics S

Selection techniques. Given a set of user-specified, knowledge domain driven summary statistics, we search for the best subset. Various approaches, e.g.

- based on entropy of the approximated posterior
  [Nunes and Balding, 2010]
- based on Akaike/Bayesian information criterion (AIC/BIC)
  [Sedki and Pudlo, 2012, Blum et al., 2013]
- based on the Kullback-Leibler divergence
  [Barnes et al., 2012, Filippi et al., 2012]

Highly interpretable (because summaries are proposed in advance with their interpretation in mind)
Problem: relevant summaries might not be in the pre-specified user defined set

Tuning in ABC: summary statistics S

Projection techniques. The relevant summary statistics might be combinations of existing ones

Use as summary statistics, e.g.

- partial least squares (PLS) regression approach [Wegmann et al., 2009]: extracts orthogonal components from a high-dimensional data set of predictor variables, but in addition, these components are chosen to appropriately explain the variability of the response variables by maximizing the covariance matrix of predictor and response variables.
  In ABC the **predictor** variables are raw summary statistics and the **response** variables are model parameters. The choice of the number of PLS components to include is usually based on a leave-one-out validation procedure.

Tuning in ABC: summary statistics S

Projection techniques. The relevant summary statistics might be combinations of existing ones

- approximations of the posterior expectation $\mathbb{E}(\boldsymbol{\theta} \mid \mathbf{y})$ [Fearnhead and Prangle, 2012]

Projection techniques are more automatized than selection but lack of interpretability

Tuning in ABC: summary statistics S

The projection method proposed by [Fearnhead and Prangle, 2012] consists in the following steps:

1. Simulate N parameter values and corresponding pseudo-data
2. Choose an arbitrary set of data transformations $f(\cdot)$
3. For each dimension of $\boldsymbol{\theta}$, train a linear regression model (e.g.) between $\boldsymbol{\theta}_j^{(i)}$'s and $f(\mathbf{x}^{(i)})$'s
4. Use the values $f(\mathbf{x}^{(1)})^\top \hat{\beta}_j, \ldots, f(\mathbf{x}^{(N)})^\top \hat{\beta}_j, f(\mathbf{y})^\top \hat{\beta}_j$ as summary statistics
5. Run a regular ABC method with this new set of summary statistics

Tuning in ABC: summary statistics S

Indirect inference summaries. Use a simpler model (with a tractable likelihood) to extract information regarding $\mathbf{y}$ and summary statistics for the intractable model

Choose a simpler auxiliary model parameterised by $\boldsymbol{\phi} \in \Phi$ with known likelihood $p_{\text{aux}}(\mathbf{y} \mid \boldsymbol{\phi})$

[Drovandi and Pettitt, 2011] propose to use as summary statistics the estimated parameter values associated to each ABC pseudo and observed data, i.e. $\hat{\boldsymbol{\phi}}_{\mathbf{x}(1)}, \ldots, \hat{\boldsymbol{\phi}}_{\mathbf{x}(N)}, \hat{\boldsymbol{\phi}}_{\mathbf{y}}$

Choice of summary statistics transformed into the choice of an auxiliary model

See also [Gleim and Pigorsch, 2013], [Drovandi et al., 2015], [Drovandi, 2018]

Tuning in ABC: summary statistics S

Regularisation. Use regularisation (L_1 or L_2) in regression adjustment approaches (presented below)

Tuning in ABC: threshold ϵ

The choice of ϵ depends on the considered ABC algorithm

For the basic ABC rejection algorithm,
ϵ impacts the trade-off between computational cost and accuracy

For the weighted ABC, ϵ impacts the bias-variance trade-off of the posterior density estimator

$$\hat{p}(\boldsymbol{\theta} \mid S_{\mathbf{y}}) = \frac{\sum_{i=1}^{N} K_b(\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}) K_\epsilon(\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}}))}{\sum_{i=1}^{N} K_\epsilon(\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}}))},$$

- large ϵ value leads to high bias but small variance for the estimator
- small ϵ value leads to low bias at the cost of a larger variance for the estimator

See [Blum, 2010]

Tuning in ABC: threshold ϵ

Some ABC techniques developed to reduce the influence of ϵ

These are regression adjustment methods and sequential ABC schemes, that we now present

Outline

1 Basics of ABC

2 Tuning in ABC

3 Regression adjustment methods

Regression adjustment methods

Given a set of simulated $(\boldsymbol{\theta}^{(i)}, S_{\mathbf{x}^{(i)}})$ weighted by $w^{(i)} \propto K_\epsilon(\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}})$

[Beaumont et al., 2002] propose to use post-hoc adjustment of the parameter values to weaken the effect of the discrepancy between $S_{\mathbf{x}^{(i)}}$ and $S_{\mathbf{y}}$

Idea:
Correct the $\boldsymbol{\theta}^{(i)}$ values depending on the discrepancy between $S_{\mathbf{x}^{(i)}}$ and $S_{\mathbf{y}}$ thanks to an assumed relationship between $\boldsymbol{\theta}$ and S

Regression adjustment methods

Let us denote $\mu(S_{\mathbf{x}}) = \mathbb{E}(\boldsymbol{\theta} \mid S_{\mathbf{x}})$
and assume that we have the relationship

$$\boldsymbol{\theta}^{(i)} = \mu(S_{\mathbf{x}^{(i)}}) + e^{(i)}$$

where $e^{(i)}$ i.i.d. residuals with zero mean and common variance

Methodology:

1. Estimate $\mu(\cdot)$ by $\hat{\mu}(\cdot)$ using a local linear regression model
2. Compute the empirical residuals $\hat{e}^{(i)} = \boldsymbol{\theta}^{(i)} - \hat{\mu}(S_{\mathbf{x}^{(i)}})$
3. Deduce the adjusted values

$$\boldsymbol{\theta}_c^{(i)} = \hat{\mu}(S_{\mathbf{y}}) + \hat{e}^{(i)} = \boldsymbol{\theta}^{(i)} + (\hat{\mu}(S_{\mathbf{y}}) - \hat{\mu}(S_{\mathbf{x}^{(i)}}))$$

Linear regression adjustment

We consider that $\boldsymbol{\theta}$ is unidimensional

[Beaumont et al., 2002] assume that the relationship between θ and S is

$$\theta^{(i)} = \underbrace{\alpha + (S_{\mathbf{x}^{(i)}} - S_{\mathbf{y}})^\top \beta}_{\mu(S_{\mathbf{x}^{(i)}})} + e^{(i)}, \quad i = 1, \ldots, N$$

Remark: in $S_{\mathbf{y}}$ we have $\theta = \alpha + e$

Linear regression adjustment

1. Fit the regression model

The unknowns α and β are obtained by minimising the weighted least squares criterion

$$\sum_{i=1}^{N} w^{(i)} \left(\theta^{(i)} - \alpha - (S_{\mathbf{x}^{(i)}} - S_{\mathbf{y}})^{\top}\beta\right)^2$$

2. Compute the empirical residuals

$$\hat{e}^{(i)} = \theta^{(i)} - \hat{\alpha} - (S_{\mathbf{x}^{(i)}} - S_{\mathbf{y}})^{\top}\hat{\beta}$$

Linear regression adjustment

3. Deduce corrected parameters

$$\begin{aligned}\theta_c^{(i)} &= \hat{\alpha} + \hat{e}^{(i)} \\ &= \hat{\alpha} + (\theta^{(i)} - \hat{\alpha} - (S_{\mathbf{x}^{(i)}} - S_{\mathbf{y}})^\top \hat{\beta}) \\ &= \theta^{(i)} - (S_{\mathbf{x}^{(i)}} - S_{\mathbf{y}})^\top \hat{\beta}\end{aligned}$$

The $\theta_c^{(i)}$ values weighted by $w^{(i)} \propto K_\epsilon(\rho(S_{\mathbf{x}^{(i)}}, S_{\mathbf{y}})$ yield a sample from an approximated posterior distribution

Linear regression adjustment

For multidimensional $\boldsymbol{\theta}$, a linear adjustment can be performed on each dimension of $\boldsymbol{\theta}$, or some multivariate regression can be adopted

This method allows us to use a larger tolerance value and to substantially improve posterior accuracy

The assumed (linear) relationship needs to be exact

More flexible relationships can be used

Non-linear regression adjustment

[Blum and François, 2010] propose the more flexible non-linear conditional heteroscedastic model

$$\boldsymbol{\theta}^{(i)} = \mu(S_{\mathbf{x}^{(i)}}) + \sigma(S_{\mathbf{x}^{(i)}})e^{(i)}, \quad i = 1, \ldots, N,$$

where
$\mu(S_{\mathbf{x}^{(i)}}) = \mathbb{E}(\boldsymbol{\theta} \mid S_{\mathbf{x}^{(i)}})$,
$\sigma^2(S_{\mathbf{x}^{(i)}}) = \mathbb{V}(\boldsymbol{\theta} \mid S_{\mathbf{x}^{(i)}})$,
$e^{(i)}$ is the residual, still i.i.d. centred with common variance

These posterior quantities are estimated using feed-forward neural networks

Non-linear regression adjustment

Correction is performed as follows

$$\begin{aligned}\boldsymbol{\theta}_c^{(i)} &= \hat{\mu}(S_{\mathbf{y}}) + \hat{\sigma}(S_{\mathbf{y}})\hat{e}^{(i)} \\ &= \hat{\mu}(S_{\mathbf{y}}) + \hat{\sigma}(S_{\mathbf{y}}) \left\{ \frac{1}{\hat{\sigma}(S_{\mathbf{x}^{(i)}})} \left(\boldsymbol{\theta}^{(i)} - \hat{\mu}(S_{\mathbf{x}^{(i)}}) \right) \right\}.\end{aligned}$$

$\Rightarrow$ Reduces the influence of ϵ even more

$\Rightarrow$ The use of neural networks was motivated by their capacity to reduce the summary statistics space internally
This is a projection methods

Other regression adjustment

[Blum, 2010]
propose a quadratic relationship

[Leuenberger and Wegmann, 2010]
propose a generalised linear relationship

Penalisation in adjustment techniques

When assuming a local linear regression relationship, it is also possible to use the regularised weighted least squares

$$\sum_{i=1}^{N} w^{(i)} \left(\boldsymbol{\theta}^{(i)} - \alpha - (S_{\mathbf{x}^{(i)}} - S_{\mathbf{y}})^{\top} \beta \right)^2 + \lambda ||\beta||$$

See [Blum et al., 2013] for the ridge version,
and [Saulnier et al., 2017] for the LASSO version

$\Rightarrow$ Reduces the influence of uninformative summary statistics at the cost of an additional parameter λ

Regression adjustment techniques

Adjustment techniques might underestimate the posterior variances and give narrow credible intervals

R package `abc` [Csilléry et al., 2012]
for basic ABC and local linear, ridge, and neural network regression adjustment techniques

References I

Barnes, C. P., Filippi, S., and Stumpf, M. P. H. (2012).

Considerate approaches to constructing summary statistics for ABC model selection.

Statistics and Computing, 22:1181–1197.

Beaumont, M. A. (2010).

Approximate Bayesian computation in evolution and ecology.

Annual Review of Ecology, Evolution, and Systematics, 41(1):379–406.

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. (2009).

Adaptive approximate Bayesian computation.

Biometrika, 96(4):983–990.

Beaumont, M. A., Zhang, W., and Balding, D. (2002).

Approximate Bayesian computation in population genetics.

Genetics, 162(4):2025–2035.

References II

Biau, G., Cérou, F., and Guyader, A. (2015).

New insights into approximate Bayesian computation.

Annales de l'Institut Henri Poincaré B, Probability and Statistics, 51(1):376–403.

Blum, M. G. B. (2010).

Approximate Bayesian computation: a nonparametric perspective.

Journal of the American Statistical Association, 105(491):1178–1187.

Blum, M. G. B. and François, O. (2010).

Non-linear regression models for approximate Bayesian computation.

Statistics and Computing, 20:63–73.

Blum, M. G. B., Nunes, M., Prangle, D., and Sisson, S. A. (2013).

A comparative review of dimension reduction methods in approximate Bayesian computation.

Statistical Science, 28(2):189–208.

References III

Csilléry, K., François, O., and Blum, M. G. B. (2012).

abc: an R package for approximate Bayesian computation (ABC).

Methods in Ecology and Evolution, 3(3):475–479.

Del Moral, P., Doucet, A., and Jasra, A. (2012).

An adaptive sequential Monte Carlo method for approximate Bayesian computation.

Statistics and Computing, 22(5):1009–1020.

Drovandi, C. C. (2018).

ABC and indirect inference (in Handbook of Approximate Bayesian Computation), pages 179–210.

Chapman and Hall/CRC.

Drovandi, C. C. and Pettitt, A. N. (2011).

Estimation of parameters for macroparasite population evolution using approximate Bayesian computation.

Biometrics, 67(1):225–233.

References IV

Drovandi, C. C., Pettitt, A. N., and Lee, A. (2015).

Bayesian indirect inference using a parametric auxiliary model.

Statistical Science, 30(1):72–95.

Fearnhead, P. and Prangle, D. (2012).

Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation.

Journal of the Royal Statistical Society. B (Statistical Methodology), 74(3):419–474.

Filippi, S., Barnes, C. P., and Stumpf, M. P. H. (2012).

Contribution to the discussion of Fearnhead and Prangle (2012).

Journal of the Royal Statistical Society. B (Statistical Methodology), 74(3):459–460.

Gleim, E. and Pigorsch, C. (2013).

Approximate Bayesian computation with indirect summary statistics.

Technical report, University of Bonn, Bonn, Germany.

References V

Kingman, J. F. C. (1982).

Exchangeability and the evolution of large populations, pages 97–112.

North-Holland, Amsterdam.

Leuenberger, C. and Wegmann, D. (2010).

Bayesian computation and model selection without likelihoods.

Genetics, 184(1):243–252.

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. (2003).

Markov chain Monte Carlo without likelihoods.

Proceedings of the National Academy of Sciences, USA, 100(26):15324–15328.

Nunes, M. A. and Balding, D. J. (2010).

On optimal selection of summary statistics for approximate Bayesian computation.

Statistical Application in Genetics and Molecular Biology, 9(1):Article 34.

References VI

Prangle, D. (2017).

Adapting the ABC distance function.

Bayesian Analysis, 12(1):289–309.

Prangle, D. (2018).

Summary statistics (in Handbook of Approximate Bayesian Computation), pages 125–152.

Chapman and Hall/CRC.

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., and Feldman, M. W. (1999).

Population growth of human Y chromosomes: a study of Y chromosome microsatellites.

Molecular Biology and Evolution, 16:1791–1798.

Saulnier, E., Gascuel, O., and Alizon, S. (2017).

Inferring epidemiological parameters from phylogenies using regression-ABC: a comparative study.

PLOS Computational Biology, 13(3):e1005416.

References VII

Sedki, M. A. and Pudlo, P. (2012).

Contribution to the discussion of Fearnhead and Prangle (2012).

Journal of the Royal Statistical Society. B (Statistical Methodology), 74(3):466–467.

Sisson, S., Fan, Y., and Tanaka, M. (2009).

Sequential Monte Carlo without likelihoods: Errata.

Proceedings of the National Academy of Sciences, USA, 106(39):16889.

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007).

Sequential Monte Carlo without likelihoods.

Proceedings of the National Academy of Sciences, USA, 104(6):1760–1765.

Tavaré, S., Balding, D., Griffiths, R., and Donnelly, P. (1997).

Inferring coalescence times from DNA sequence data.

Genetics, 145(2):505–518.

References VIII

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. (2009).

Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems.

Journal of the Royal Society Interface, 6(31):187–202.

Wegmann, D., Leuenberger, C., and Excoffier, L. (2009).

Efficient approximate Bayesian computation coupled with Markov chain Monte Carlo without likelihood.

Genetics, 182(4):1207–1218.

Weiss, G. and von Haeseler, A. (1998).

Inference of population history using a likelihood approach.

Genetics, 149(3):1539–1546.