Tutorial on Approximate Bayesian Computation

Michael Gutmann

https://sites.google.com/site/michaelgutmann

University of Helsinki   Aalto University

Helsinki Institute for Information Technology

16 May 2016

Content

Two parts:

1. The basics of approximate Bayesian computation (ABC)
2. Computational and statistical efficiency

What is ABC?

A set of methods for approximate Bayesian inference which can be used whenever sampling from the model is possible.

Part I

Basic ABC

Program

Program

Preliminaries

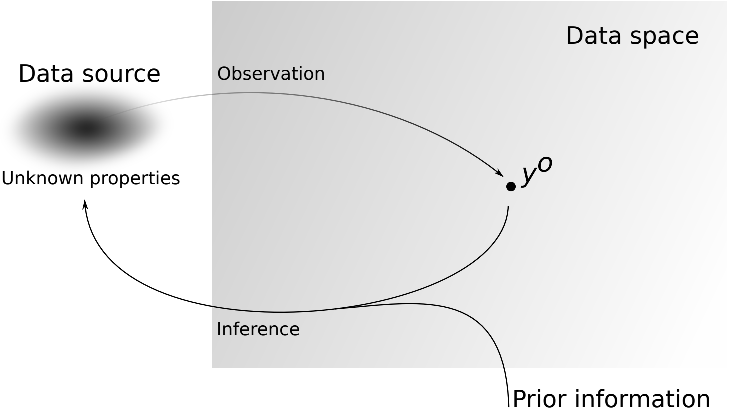
- Statistical inference
- Simulator-based models
- Likelihood function

Inference for simulator-based models

- Exact inference
- Approximate inference
- Rejection ABC algorithm

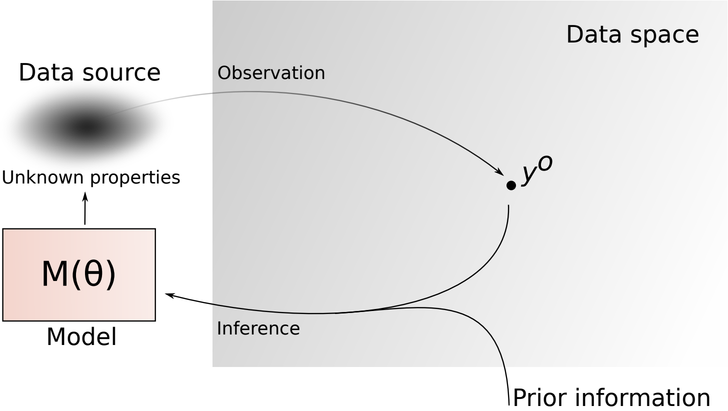
Big picture of statistical inference

- Given data $\mathbf{y}^o$, draw conclusions about properties of its source
- If available, possibly take prior information into account



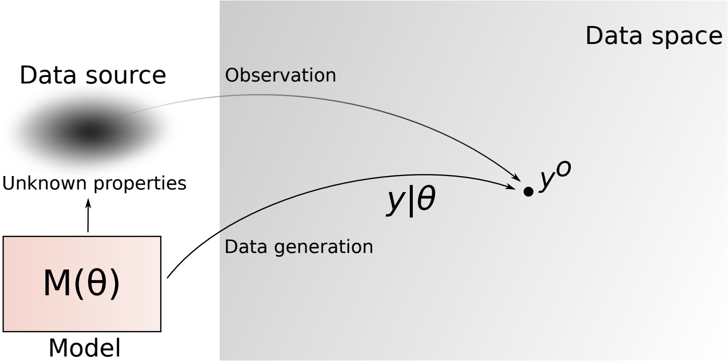
General approach

- Set up a model with potential properties $\boldsymbol{\theta}$ (parameters)
- See which $\boldsymbol{\theta}$ are reasonable given the observed data



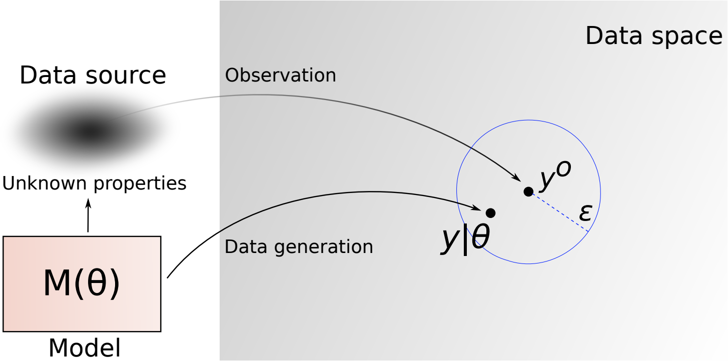
Likelihood function

- Measures agreement between $\boldsymbol{\theta}$ and the observed data $\mathbf{y}^o$
- Probability to see data $\mathbf{y}$ like $\mathbf{y}^o$ if property $\boldsymbol{\theta}$ holds



Likelihood function

- Measures agreement between $\boldsymbol{\theta}$ and the observed data $\mathbf{y}^o$
- Probability to see data $\mathbf{y}$ like $\mathbf{y}^o$ if property $\boldsymbol{\theta}$ holds



Likelihood function

- For discrete random variables:

$$L(\boldsymbol{\theta}) = \Pr(\mathbf{y} = \mathbf{y}^o | \boldsymbol{\theta}) \tag{1}$$

- For continuous random variables:

$$L(\boldsymbol{\theta}) = \lim_{\epsilon \to 0} \frac{\Pr(\mathbf{y} \in B_\epsilon(\mathbf{y}^o) | \boldsymbol{\theta})}{\mathrm{Vol}(B_\epsilon(\mathbf{y}^o))} \tag{2}$$

Performing statistical inference

- If $L(\boldsymbol{\theta})$ is known, inference boils down to solving an optimization/sampling problem
- Maximum likelihood estimation

$$\hat{\boldsymbol{\theta}} = \mathrm{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

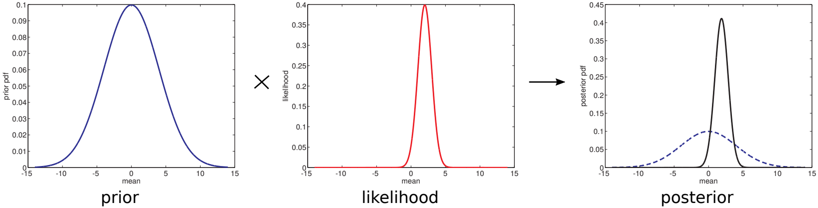
- Bayesian inference

$$p(\boldsymbol{\theta}|\mathbf{y}^o) \propto p(\boldsymbol{\theta}) \times L(\boldsymbol{\theta})$$

$$\text{posterior} \propto \text{prior} \times \text{likelihood}$$

Textbook case

- model $\equiv$ family of probability density/mass functions $p(\mathbf{y}|\boldsymbol{\theta})$
- Likelihood function $L(\boldsymbol{\theta}) = p(\mathbf{y}^o|\boldsymbol{\theta})$
- Closed form solutions are possible.



Simulator-based models

- Not all models are specified as family of pdfs $p(\mathbf{y}|\boldsymbol{\theta})$.
- Here: simulator-based models:

  models which are specified via a mechanism (rule) for generating data

Toy example

- Let $y|\theta \sim \mathcal{N}(\theta, 1)$
- Family of pdfs as model:

$$p(y|\theta) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y-\theta)^2}{2}\right) \tag{3}$$

- Simulator-based model:

$$y = z + \theta \qquad\qquad z \sim \mathcal{N}(0, 1) \tag{4}$$

or

$$y = z + \theta \qquad\qquad z = \sqrt{-2\log(\omega)}\cos(2\pi\nu) \tag{5}$$

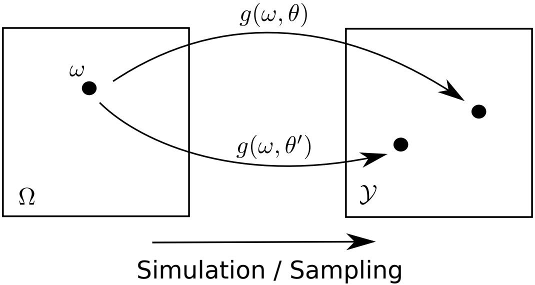
where ω and ν are independent random variables uniformly distributed on $(0, 1)$

Formal definition of a simulator-based model

- Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space.
- A simulator-based model is a collection of (measurable) functions $g(., \boldsymbol{\theta})$ parametrized by $\boldsymbol{\theta}$,

$$\boldsymbol{\omega} \in \Omega \mapsto \mathbf{y} = g(\boldsymbol{\omega}, \boldsymbol{\theta}) \in \mathcal{Y} \tag{6}$$

- The functions $g(., \boldsymbol{\theta})$ are typically not available in closed form.

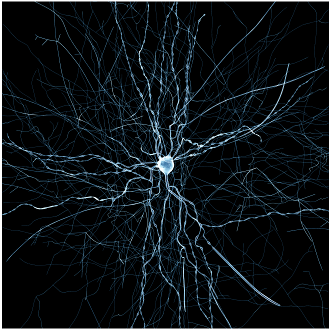


Other names for simulator-based models

- Models specified via a data generating mechanism occur in multiple and diverse scientific fields.
- Different communities use different names for simulator-based models:
  - Generative models
  - Implicit models
  - Stochastic simulation models
  - Probabilistic programs

Examples

- Astrophysics:
  Simulating the formation of galaxies, stars, or planets
- Evolutionary biology:
  Simulating evolution
- Neuroscience:
  Simulating neural circuits
- Ecology:
  Simulating species migration
- Health science:
  Simulating the spread of an infectious disease
- ...

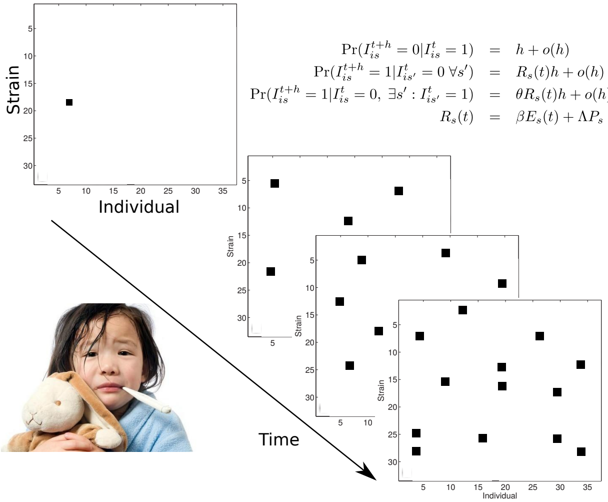


Simulated neural activity in rat somatosensory cortex
(Figure from `https://bbp.epfl.ch/nmc-portal`)

Example (health science)

- Simulating bacterial transmissions in child day care centers (Numminen et al, 2013)

$$\begin{aligned}
\Pr(I_{is}^{t+h}=0|I_{is}^{t}=1) &= h+o(h)\\
\Pr(I_{is}^{t+h}=1|I_{is'}^{t}=0\ \forall s') &= R_s(t)h+o(h)\\
\Pr(I_{is}^{t+h}=1|I_{is}^{t}=0,\ \exists s': I_{is'}^{t}=1) &= \theta R_s(t)h+o(h)\\
R_s(t) &= \beta E_s(t)+\Lambda P_s
\end{aligned}$$



Advantages of simulator-based models

- Direct implementation of hypotheses of how the observed data were generated.
- Neat interface with physical or biological models of data.
- Modeling by replicating the mechanisms of nature which produced the observed/measured data. ("Analysis by synthesis")
- Possibility to perform experiments in silico.

Disadvantages of simulator-based models

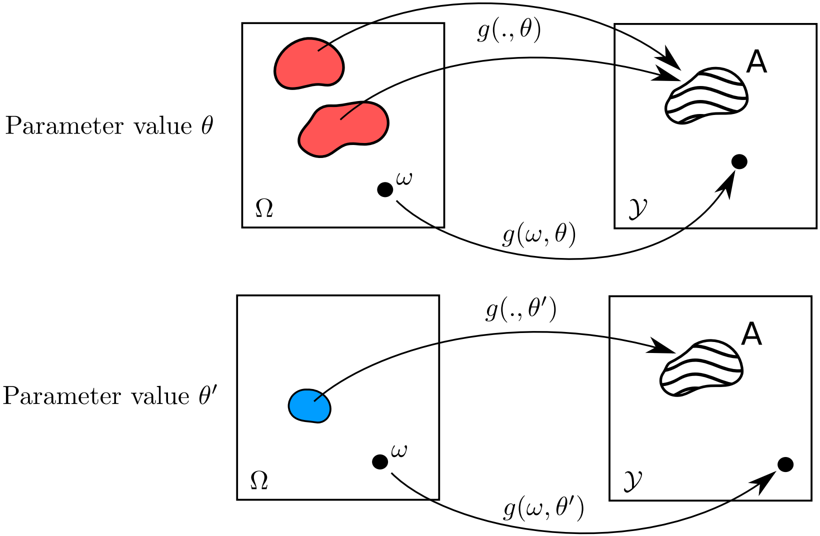
- Generally elude analytical treatment.
- Can be easily made more complicated than necessary.
- Statistical inference is difficult . . . but possible!

Family of pdfs induced by the simulator

- For any fixed $\boldsymbol{\theta}$, the output of the simulator $\mathbf{y}_{\boldsymbol{\theta}} = g(., \boldsymbol{\theta})$ is a random variable.
- No closed-form formulae available for $p(\mathbf{y}|\boldsymbol{\theta})$.
- Simulator defines the model pdfs $p(\mathbf{y}|\boldsymbol{\theta})$ implicitly.

Implicit definition of the model pdfs

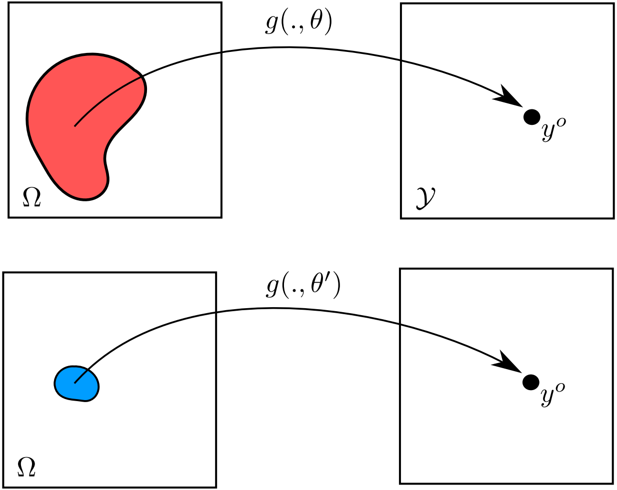
$$\Pr(y \in A \mid \theta) = \mathcal{P}(\{\omega : g(\omega, \theta) \in A\})$$



Implicit definition of the likelihood function

For discrete random variables:

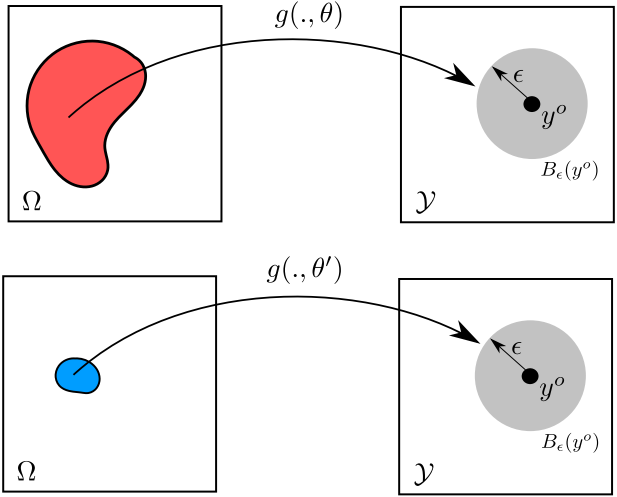
$$L(\theta) = \Pr(y = y^o \mid \theta) = \mathcal{P}(\{\omega : g(\omega, \theta) = y^o\})$$



Implicit definition of the likelihood function

For continuous random variables: $L(\boldsymbol{\theta}) = \lim_{\epsilon \to 0} L_\epsilon(\boldsymbol{\theta})$

$$L_\epsilon(\theta) = \frac{\Pr(y \in B_\epsilon(y^o) \mid \theta)}{V_\epsilon} = \frac{\mathcal{P}(\{\omega : g(\omega, \theta) \in B_\epsilon(y^o)\})}{V_\epsilon}$$



Implicit definition of the likelihood function

- To compute the likelihood function, we need to compute the probability that the simulator generates data close to $\mathbf{y}^o$,

$$\Pr(\mathbf{y} = \mathbf{y}^o | \boldsymbol{\theta}) \quad \text{or} \quad \Pr(\mathbf{y} \in B_\epsilon(\mathbf{y}^o) | \boldsymbol{\theta})$$

- No analytical expression available.
- But we can empirically test whether simulated data equals $\mathbf{y}^o$ or is in $B_\epsilon(\mathbf{y}^o)$.
- This property will be exploited to perform inference for simulator-based models.

Program

Exact inference for discrete random variables

- For discrete random variables, we can perform exact Bayesian inference without knowing the likelihood function.
- By definition, the posterior is obtained by conditioning $p(\boldsymbol{\theta}, \mathbf{y})$ on the event $\mathbf{y} = \mathbf{y}^o$:

$$p(\boldsymbol{\theta}|\mathbf{y}^o) = \frac{p(\boldsymbol{\theta}, \mathbf{y}^o)}{p(\mathbf{y}^o)} = \frac{p(\boldsymbol{\theta}, \mathbf{y} = \mathbf{y}^o)}{p(\mathbf{y} = \mathbf{y}^o)} \tag{7}$$

Exact inference for discrete random variables

- Generate tuples $(\boldsymbol{\theta}_i, \mathbf{y}_i)$:
  1. $\boldsymbol{\theta}_i \sim p_{\boldsymbol{\theta}}$ (iid from the prior)
  2. $\boldsymbol{\omega}_i \sim \mathcal{P}$ (by running the simulator)
  3. $\mathbf{y}_i = g(\boldsymbol{\omega}_i, \boldsymbol{\theta}_i)$ (by running the simulator)
- Condition on $\mathbf{y} = \mathbf{y}^o \Leftrightarrow$ Retain only the tuples with $\mathbf{y}_i = \mathbf{y}^o$
- The $\boldsymbol{\theta}_i$ from the retained tuples are samples from the posterior $p(\boldsymbol{\theta}|\mathbf{y}^o)$.

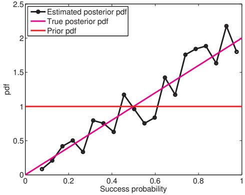
Example

- Posterior inference of the success probability θ in a Bernoulli trial.
- Data: $y^o = 1$
- Prior: $p_\theta = 1$ on $(0, 1)$
- Generate tuples (θ_i, y_i)
  1. $\theta_i \sim p_\theta$
  2. $\omega_i \sim U(0, 1)$
  3. $y_i = \begin{cases} 1 & \text{if } \omega_i < \theta_i \\ 0 & \text{otherwise} \end{cases}$
- Retain those θ_i for which $y_i = y^o$.

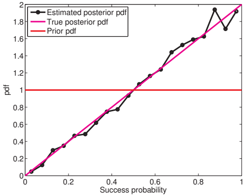
```
% Observed data
yobs = 1;

% Number of samples to generate from the posterior
N = 10000;

% Sample from prior, uniform on (0,1)
theta = rand(1,N);

% Run the "simulator"
omega = rand(1,N);
ysim = omega<theta;

% Check for simulated data which are equal to observed data
index = (ysim==yobs);

% Samples from the posterior
thetaPost = theta(index);
```

Example

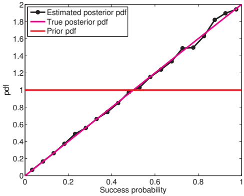
- The method produces samples from the posterior.
- Monte Carlo error when summarizing the samples as an empirical distribution or computing expectations via sample averages.
- Histogram for N simulated tuples (θ_i, y_i)



$N = 1000$



$N = 10,000$



$N = 100,000$

Limitations

- Only applicable to discrete random variables.
- And even for discrete random variables:
  Computationally not feasible in higher dimensions
- Reason: *The probability of the event* $\mathbf{y}_{\boldsymbol{\theta}} = \mathbf{y}^o$ *becomes smaller and smaller as the dimension of the data increases.*
- Out of N simulated tuples only a small fraction will be accepted.
  - The small number of accepted samples do not represent the posterior well.
  - Large Monte Carlo errors

Approximations to make inference feasible

- Settle for approximate yet computationally feasible inference.
- Introduce two types of approximations:
  1. Instead of working with the whole data, work with lower dimensional summary statistics $\mathbf{t}_{\boldsymbol{\theta}}$ and $\mathbf{t}^o$,

$$\mathbf{t}_{\boldsymbol{\theta}} = T(\mathbf{y}_{\boldsymbol{\theta}}) \qquad \mathbf{t}^o = T(\mathbf{y}^o). \tag{8}$$

  2. Instead of checking $\mathbf{t}_{\boldsymbol{\theta}} = \mathbf{t}^o$, check whether $\Delta_{\boldsymbol{\theta}} = d(\mathbf{t}^o, \mathbf{t}_{\boldsymbol{\theta}})$ is less than ϵ. (d may or may not be a metric)

Approximation of the likelihood function

$$L(\boldsymbol{\theta}) = \lim_{\epsilon \to 0} L_\epsilon(\boldsymbol{\theta}) \qquad L_\epsilon(\boldsymbol{\theta}) = \frac{\Pr(\mathbf{y} \in B_\epsilon(\mathbf{y}^o) | \boldsymbol{\theta})}{\mathrm{Vol}(B_\epsilon(\mathbf{y}^o))}$$

- Approximations are equivalent to:
  1. Replacing $\Pr\left(\mathbf{y} \in B_{\epsilon'}(\mathbf{y}^o) \mid \boldsymbol{\theta}\right)$ with $\Pr\left(\Delta_{\boldsymbol{\theta}} \leq \epsilon | \, \boldsymbol{\theta}\right)$
  2. Not taking the limit $\epsilon \to 0$
- Defines an approximate likelihood function $\tilde{L}_\epsilon(\boldsymbol{\theta})$,

$$\tilde{L}_\epsilon(\boldsymbol{\theta}) \propto \Pr\left(\Delta_{\boldsymbol{\theta}} \leq \epsilon \mid \boldsymbol{\theta}\right) \tag{9}$$

- Discrepancy $\Delta_{\boldsymbol{\theta}}$ is a (non-negative) random variable

$$\Delta_{\boldsymbol{\theta}} = d(\mathbf{t}^o, \mathbf{t}_{\boldsymbol{\theta}}) = d\left(T(\mathbf{y}^o), T(\mathbf{y}_{\boldsymbol{\theta}})\right)$$

Rejection ABC algorithm

- The two approximations made yield the rejection algorithm for approximate Bayesian computation (ABC):
  1. Sample $\boldsymbol{\theta}_i \sim p_{\boldsymbol{\theta}}$
  2. Simulate a data set $\mathbf{y}_i$ by running the simulator with $\boldsymbol{\theta}_i$ ($\mathbf{y}_i = g(\boldsymbol{\omega}_i, \boldsymbol{\theta}_i)$)
  3. Compute the discrepancy $\Delta_i = d(T(\mathbf{y}^o), T(\mathbf{y}_i))$
  4. Retain $\boldsymbol{\theta}_i$ if $\Delta_i \leq \epsilon$
- This is *the* basic ABC algorithm.

Properties

- Rejection ABC algorithm produces samples $\boldsymbol{\theta} \sim \tilde{p}_\epsilon(\boldsymbol{\theta}|\mathbf{y}^o)$,

$$\tilde{p}_\epsilon(\boldsymbol{\theta}|\mathbf{y}^o) \propto p_{\boldsymbol{\theta}}(\boldsymbol{\theta})\tilde{L}_\epsilon(\boldsymbol{\theta}) \tag{10}$$

$$\tilde{L}_\epsilon(\boldsymbol{\theta}) \propto \Pr\Big(\underbrace{d(T(\mathbf{y}^o), T(\mathbf{y}))}_{\Delta_{\boldsymbol{\theta}}} \leq \epsilon \mid \boldsymbol{\theta}\Big) \tag{11}$$

- Inference is approximate due to
  - the summary statistics T and distance d
  - $\epsilon > 0$
  - the finite number of samples (Monte Carlo error)

Part II

Computational and statistical efficiency

Brief recap

- Simulator-based models: Models which are specified by a data generating mechanism.
- By construction, we can sample from simulator-based models. Likelihood function can generally not be written down.
- Rejection ABC: Trial and error scheme to find parameter values which produce simulated data resembling the observed data.
- Simulated data resemble the observed data if some discrepancy measure is small.

Efficiency of ABC

1. Computational efficiency: How to efficiently find the parameter values which yield a small discrepancy?
2. Statistical efficiency: How to measure the discrepancy between the simulated and observed data?

Program

Computational efficiency
- Difficulties
- Solutions
- Recent work

Statistical efficiency
- Difficulties
- Solutions
- Recent work

Program

Computational efficiency
- Difficulties
- Solutions
- Recent work

Statistical efficiency
- Difficulties
- Solutions
- Recent work

Example

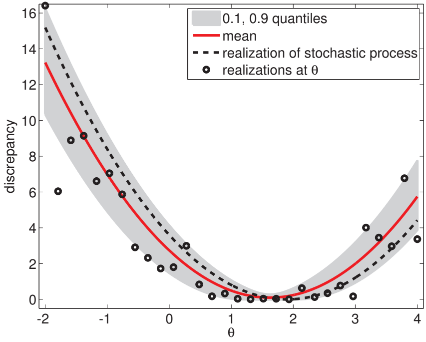
- Inference of the mean θ of a Gaussian of variance one.
- $\Pr(\mathbf{y} = \mathbf{y}^o | \boldsymbol{\theta}) = 0$.
- Discrepancy Δ_θ:

$$\Delta_\theta = (\hat{\mu}^o - \hat{\mu}_\theta)^2,$$

$$\hat{\mu}^o = \frac{1}{n}\sum_{i=1}^{n} y_i^o,$$

$$\hat{\mu}_\theta = \frac{1}{n}\sum_{i=1}^{n} y_i,$$

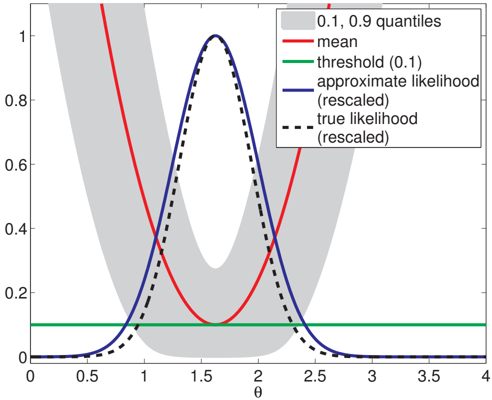
$$y_i \sim \mathcal{N}(\theta, 1)$$



Discrepancy Δ_θ is a random variable.

Example

Probability that Δ_θ is below some threshold ϵ approximates the likelihood function.



Example

- Here, $T(\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} y_i$ is a sufficient statistics for inference of the mean θ
- The only approximation is $\epsilon > 0$.
- In general, the summary statistics will not be sufficient.

Example

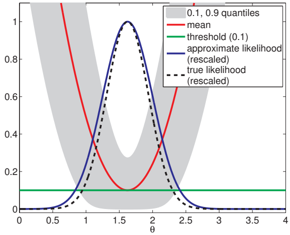
- In the Gaussian example, the probability for $\Delta_{\boldsymbol{\theta}} \leq \epsilon$ can be computed in closed form

$\Delta_\theta = (\hat{\mu}^o - \hat{\mu}_\theta)^2$

$$\Pr(\Delta_\theta \leq \epsilon) = \Phi\left(\sqrt{n}(\hat{\mu}^o - \theta) + \sqrt{n\epsilon}\right) - \Phi\left(\sqrt{n}(\hat{\mu}^o - \theta) - \sqrt{n\epsilon}\right)$$

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right) \mathrm{d}u$$

- For $n\epsilon$ small: $\tilde{L}_\epsilon(\theta) \propto \Pr(\Delta_\theta \leq \epsilon) \propto \sqrt{\epsilon} L(\theta)$
- For small ϵ good approximation of the likelihood function.
- But for small ϵ, $\Pr(\Delta_\theta \leq \epsilon) \approx 0$: Very few samples will be accepted



Two widely used algorithms

- Two widely used algorithms which improve computationally upon rejection ABC:
  1. Regression ABC (Beaumont et al, 2002)
  2. Sequential Monte Carlo ABC (Sisson et al, 2007)
- Both use rejection ABC as a building block.
- Sequential Monte Carlo (SMC) ABC is also known as Population Monte Carlo (PMC) ABC.

Two widely used algorithms

- Regression ABC consists in running rejection ABC with a relatively large ϵ and then adjusting the obtained samples so that they are closer to samples from the true posterior.
- Sequential Monte Carlo ABC consists in sampling $\boldsymbol{\theta}$ from an adaptively constructed proposal distribution $\phi(\boldsymbol{\theta})$ rather than from the prior in order to avoid simulating many data sets which are not accepted.

Basic idea of regression ABC

- The summary statistics $\mathbf{t}_{\boldsymbol{\theta}} = T(\mathbf{y}_{\boldsymbol{\theta}})$ and $\boldsymbol{\theta}$ have a joint distribution.
- Let $\mathbf{t}_i$ be the summary statistics for simulated data $\mathbf{y}_i = g(\boldsymbol{\omega}_i, \boldsymbol{\theta}_i)$.
- We can learn a regression model between the summary statistics (covariates) and the parameters (response variables)

$$\boldsymbol{\theta}_i = f(\mathbf{t}_i) + \boldsymbol{\xi}_i \tag{12}$$

  where $\boldsymbol{\xi}_i$ is the error term (zero mean random variable).
- The training data for the regression are typically tuples $(\boldsymbol{\theta}_i, \mathbf{t}_i)$ produced by rejection-ABC with some sufficiently large ϵ.

Basic idea of regression ABC

- Fitting the regression model to the training data $(\boldsymbol{\theta}_i, \mathbf{t}_i)$ yields an estimated regression function $\hat{f}$ and the residuals $\hat{\boldsymbol{\xi}}_i$,

$$\hat{\boldsymbol{\xi}}_i = \boldsymbol{\theta}_i - \hat{f}(\mathbf{t}_i) \tag{13}$$

- Regression ABC consists in replacing $\boldsymbol{\theta}_i$ with $\boldsymbol{\theta}_i^*$,

$$\boldsymbol{\theta}_i^* = \hat{f}(\mathbf{t}^o) + \hat{\boldsymbol{\xi}}_i = \hat{f}(\mathbf{t}^o) + \boldsymbol{\theta}_i - \hat{f}(\mathbf{t}_i) \tag{14}$$

- Corresponds to an adjustment of $\boldsymbol{\theta}_i$.
- If the relation between $\mathbf{t}$ and $\boldsymbol{\theta}$ is learned correctly, the $\boldsymbol{\theta}_i^*$ correspond to samples from an approximation with $\epsilon = 0$.

Basic idea of sequential Monte Carlo ABC

- We may modify the rejection ABC algorithm and use $\phi(\boldsymbol{\theta})$ instead of the prior $p_{\boldsymbol{\theta}}$.
  1. Sample $\boldsymbol{\theta}_i \sim \phi(\boldsymbol{\theta})$
  2. Simulate a data set $\mathbf{y}_i$ by running the simulator with $\boldsymbol{\theta}_i$ ($\mathbf{y}_i = g(\boldsymbol{\omega}_i, \boldsymbol{\theta}_i)$)
  3. Compute the discrepancy $\Delta_i = d(T(\mathbf{y}^o), T(\mathbf{y}_i))$
  4. Retain $\boldsymbol{\theta}_i$ if $\Delta_i \leq \epsilon$
- The retained samples follow a distribution proportional to $\phi(\boldsymbol{\theta})\tilde{L}_\epsilon(\boldsymbol{\theta})$

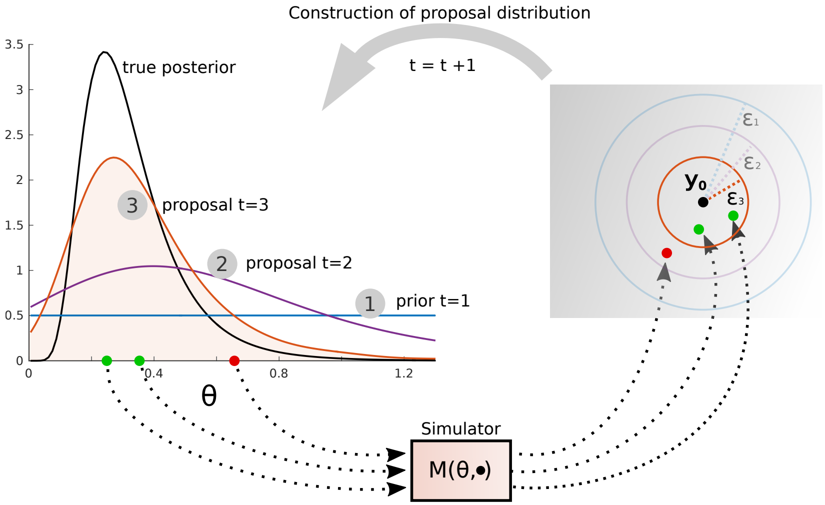
Basic idea of sequential Monte Carlo ABC

- Parameters $\boldsymbol{\theta}_i$ weighted with w_i,

$$w_i = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{\theta}_i)}{\phi(\boldsymbol{\theta}_i)}, \tag{15}$$

  follow a distribution proportional to $p_{\boldsymbol{\theta}}(\boldsymbol{\theta})\tilde{L}_\epsilon(\boldsymbol{\theta})$.
- Can be used to iteratively morph the prior into a posterior:
  - Use a sequence of shrinking thresholds ϵ_t
  - Run rejection ABC with ϵ_0.
  - Define ϕ_t at iteration t based on the weighted samples from the previous iteration (e.g Gaussian mixture with means equal to the $\boldsymbol{\theta}_i$ from the previous iteration).

Basic idea of sequential Monte Carlo ABC



Learning a model of the discrepancy

$\tilde{L}_\epsilon(\boldsymbol{\theta}) \propto \Pr(\Delta_{\boldsymbol{\theta}} \leq \epsilon \mid \boldsymbol{\theta})$

- The approximate likelihood function $\tilde{L}_\epsilon(\boldsymbol{\theta})$ is determined by the distribution of the discrepancy $\Delta_{\boldsymbol{\theta}}$
- If we knew the distribution of $\Delta_{\boldsymbol{\theta}}$ we could compute $\tilde{L}_\epsilon(\boldsymbol{\theta})$.
- We proposed to learn a model of $\Delta_{\boldsymbol{\theta}}$ and to approximate $\tilde{L}_\epsilon(\boldsymbol{\theta})$ by $\hat{L}_\epsilon(\boldsymbol{\theta})$,

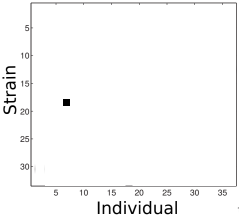
$$\tilde{L}_\epsilon(\boldsymbol{\theta}) \propto \widehat{\Pr}(\Delta_{\boldsymbol{\theta}} \leq \epsilon \mid \boldsymbol{\theta}) \tag{16}$$

- Model is learned more accurately in regions where $\Delta_{\boldsymbol{\theta}}$ tends to be small to make further computational savings.

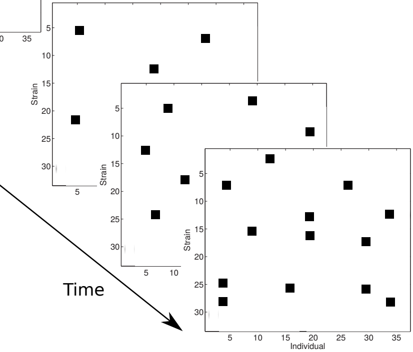
(Gutmann and Corander, *Journal of Machine Learning Research*, in press)

Example: Bacterial infections in child care centers

- Likelihood intractable for cross-sectional data
- But generating data from the model is possible



Parameters of interest:
- rate of infections within a center
- rate of infections from outside
- competition between the strains



(Numminen et al, 2013)

Example: Bacterial infections in child care centers

- Comparison of the proposed approach with a standard population Monte Carlo ABC approach.
- Roughly equal results using 1000 times fewer simulations.

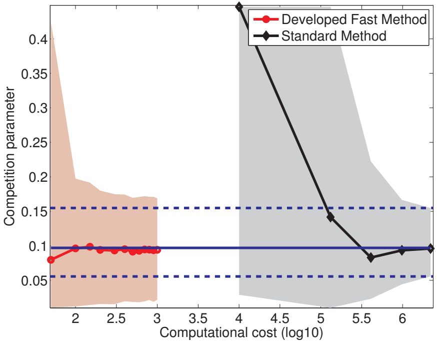
4.5 days with 200 cores

↓

90 minutes with seven cores

Posterior means: solid lines,

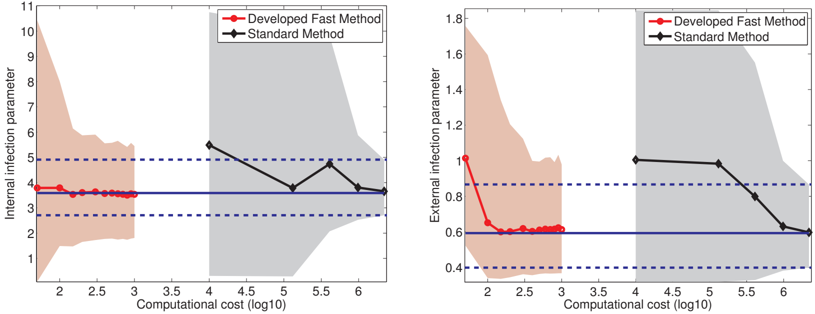
credibility intervals: shaded areas or dashed lines.



(Gutmann and Corander, 2015)

Example: Bacterial infections in child care centers

- Comparison of the proposed approach with a standard population Monte Carlo ABC approach.
- Roughly equal results using 1000 times fewer simulations.



Posterior means are shown as solid lines, credibility intervals as shaded areas or dashed lines.

Program

Computational efficiency
- Difficulties
- Solutions
- Recent work

Statistical efficiency
- Difficulties
- Solutions
- Recent work

- Discrepancy measure affects the accuracy of the estimates
- Bad discrepancy: estimated posterior = prior
- Bad discrepancy: vanishingly small acceptance probability
- Good discrepancy: good trade-off between loss of information and increase in acceptance probability

How to choose the discrepancy measure?

- Manually
  - Use expert knowledge about $\mathbf{y}^o$ to define summary statistics T.
  - Use Euclidean distance for d.

$$\Delta_{\boldsymbol{\theta}} = ||T(\mathbf{y}^o) - T(\mathbf{y}_{\boldsymbol{\theta}})||$$

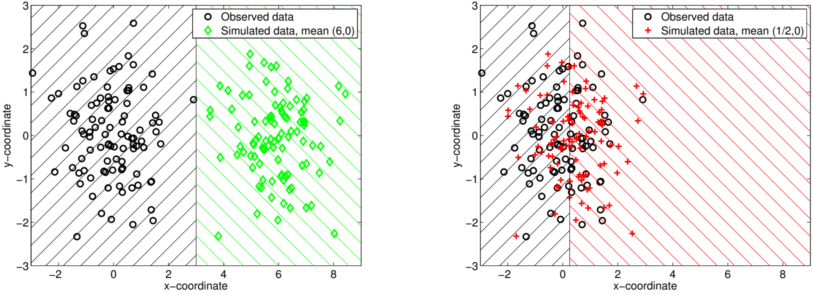
- Semi-automatic
  - Simulate pairs $(\boldsymbol{\theta}_i, \mathbf{y}_i)$
  - Define a large number of summary statistics $\tilde{T}$
  - Define T as a smaller number of (linear) combinations of them, automatically learned from the simulated pairs.
  - Use Euclidean distance for d.
- Combinations are typically determined via regression with the $\tilde{T}(\mathbf{y}_{\boldsymbol{\theta}})$ as covariates and parameters $\boldsymbol{\theta}$ as reponse variables.

  e.g. Nunes and Balding, 2010; Fearnhead and Prangle, 2012; Aeschbacher et al, 2012; Blum et al, 2013

Discrepancy measurement via classification

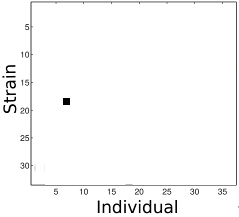
(Gutmann et al, 2014)

- Classification accuracy (discriminability) as discrepancy measure $\Delta_{\boldsymbol{\theta}}$.
- Discriminability of 100% indicates maximally different data sets; 50% indicates similar data sets.

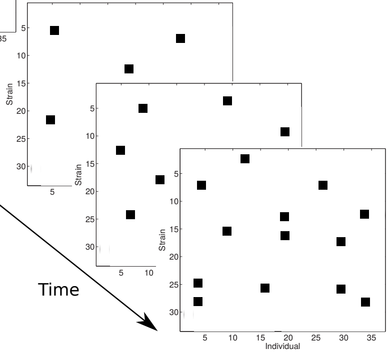


Example: Bacterial infections in child care centers

- Likelihood intractable for cross-sectional data
- But generating data from the model is possible



Parameters of interest:
- rate of infections within a center
- rate of infections from outside
- competition between the strains

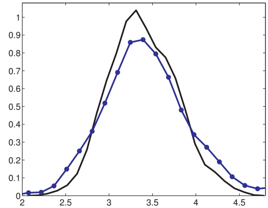


(Numminen et al, 2013)

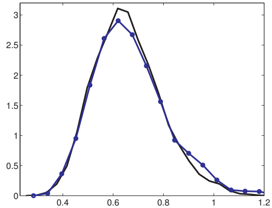
Example: Bacterial infections in child care centers

(Gutmann et al, 2014)

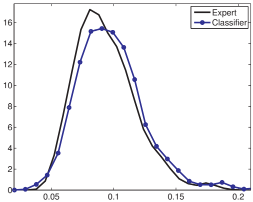
- Our classification-based distance measure does not use domain/expert knowledge.
- Performs as well as a distance measure based on domain knowledge (Numminen et, 2013).



(a) Posterior pdf for β



(b) Posterior pdf for Λ

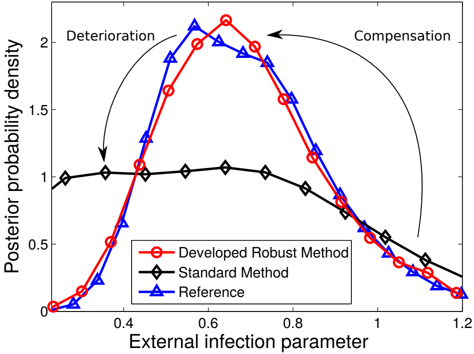


(c) Posterior pdf for θ

Example: Bacterial infections in child care centers

(Gutmann et al, 2014)

- Robustness is a concern when relying on expert knowledge
- Classification-based distance can automatically compensate errors in the expert input.



Summary

- The topic was Bayesian inference for models specified via a simulator (implicit / generative models).
- Introduced approximate Bayesian computation (ABC).
- Principle of ABC: Find parameter values which yield simulated data resembling the observed data.

Summary

- The topic was Bayesian inference for models specified via a simulator (implicit / generative models).
- Introduced approximate Bayesian computation (ABC).
- Principle of ABC: Find parameter values which yield simulated data resembling the observed data.
- Covered three classical algorithms:
  1. Rejection ABC
  2. Regression ABC
  3. Sequential Monte Carlo ABC

Summary

- The topic was Bayesian inference for models specified via a simulator (implicit / generative models).
- Introduced approximate Bayesian computation (ABC).
- Principle of ABC: Find parameter values which yield simulated data resembling the observed data.
- Covered three classical algorithms:
  1. Rejection ABC
  2. Regression ABC
  3. Sequential Monte Carlo ABC
- Choice of discrepancy measure between simulated and observed data

Summary

- The topic was Bayesian inference for models specified via a simulator (implicit / generative models).
- Introduced approximate Bayesian computation (ABC).
- Principle of ABC: Find parameter values which yield simulated data resembling the observed data.
- Covered three classical algorithms:
  1. Rejection ABC
  2. Regression ABC
  3. Sequential Monte Carlo ABC
- Choice of discrepancy measure between simulated and observed data
- Recent work of mine
  - Combining modeling of the discrepancy and optimization to increase computational efficiency
  - Using classification to measure the discrepancy

References

My papers: `https://sites.google.com/site/michaelgutmann/publications`

- Numminen et al. Estimating the Transmission Dynamics of Streptococcus pneumoniae from Strain Prevalence Data. *Biometrics*, 2013.
- Beaumont et al. Approximate Bayesian Computation in Population Genetics. *Genetics*, 2002.
- Sisson et al. Sequential Monte Carlo without likelihoods. *PNAS*, 2007.
- Nunes and Balding. On optimal selection of summary statistics for approximate Bayesian computation. *Statistical Applications in Genetics and Molecular Biology*, 2010.
- Fearnhead and Prangle. Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2012.
- Aeschbacher et al. A novel approach for choosing summary statistics in approximate Bayesian computation. *Genetics*, 2012.
- Blum et al. A Comparative Review of Dimension Reduction Methods in Approximate Bayesian Computation. *Statistical Science*, 2013.

Review papers

- Beaumont. Approximate Bayesian Computation in Evolution and Ecology. *Annual Review of Ecology, Evolution, and Systematics*, 2010.
- Hartig et al. Statistical inference for stochastic simulation models – theory and application.*Ecology Letters*, 2011.
- Marin et al. Approximate Bayesian computational methods. *Statistics and Computing*, 2012.
- Sunnaker et al. Approximate Bayesian Computation. *PLoS Computational Biology*, 2013.