Approximate Bayesian Computation

Michael Gutmann

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

University of Helsinki and Aalto University

1st December 2015

Content

Two parts:

1. The basics of approximate Bayesian computation (ABC)
2. ABC methods used in practice

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

Recap of Bayesian inference

- The ingredients for Bayesian parameter inference:
  - Observed data $\mathbf{y}^o \in \mathcal{Y} \subset \mathbb{R}^n$
  - A statistical model for the data generating process, $p_{\mathbf{y}|\boldsymbol{\theta}}$, parametrized by $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^d$.
  - A prior probability density function (pdf) for the parameters $\boldsymbol{\theta}$, $p_{\boldsymbol{\theta}}$
- The mechanics of Bayesian inference:

$$p_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y}^o) \propto p_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}^o|\boldsymbol{\theta}) \qquad \times p_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \tag{1}$$

$$\text{posterior} \propto \text{likelihood function} \times \text{prior} \tag{2}$$

- Often written without subscripts ("function overloading")

$$p(\boldsymbol{\theta}|\mathbf{y}^o) \propto p(\mathbf{y}^o|\boldsymbol{\theta}) \times p(\boldsymbol{\theta}) \tag{3}$$

Likelihood function

- Likelihood function: $L(\boldsymbol{\theta}) = p(\mathbf{y}^o|\boldsymbol{\theta})$
  - For discrete random variables:

$$L(\boldsymbol{\theta}) = p(\mathbf{y}^o|\boldsymbol{\theta}) = \Pr(\mathbf{y} = \mathbf{y}^o|\boldsymbol{\theta}) \tag{4}$$

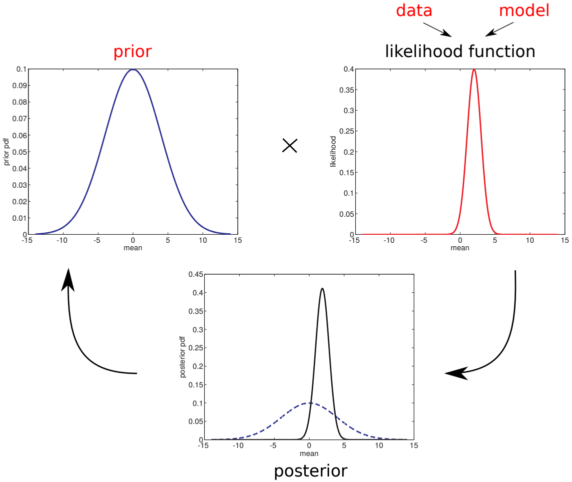
    Probability that data generated from the model, when using parameter value $\boldsymbol{\theta}$, are equal to $\mathbf{y}^o$.
  - For continuous random variables:

$$L(\boldsymbol{\theta}) = p(\mathbf{y}^o|\boldsymbol{\theta}) = \lim_{\epsilon \to 0} \frac{\Pr(\mathbf{y} \in B_\epsilon(\mathbf{y}^o)|\boldsymbol{\theta})}{\mathrm{Vol}(B_\epsilon(\mathbf{y}^o))} \tag{5}$$

    Proportional to the probability that the generated data are in a small ball $B_\epsilon(\mathbf{y}^o)$ around $\mathbf{y}^o$.
- $L(\boldsymbol{\theta})$ indicates to which extent different values of the model parameters are consistent with the observed data.

Example

$$p(\theta) = \frac{1}{\sqrt{2\pi \cdot 4^2}} \exp\left(-\frac{\theta^2}{2 \cdot 4^2}\right) \qquad y^o = 2 \qquad p(y|\theta) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y-\theta)^2}{2}\right)$$



data

model

likelihood function

prior

×

posterior

Different kinds of statistical models

- The statistical model was defined via the family of pdfs $p(\mathbf{y}|\boldsymbol{\theta})$.
- Statistical models can be specified in other ways as well.
- *In this lecture: models which are specified via a mechanism (rule) for generating data*
- Example: Instead of

$$p(y|\theta) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y-\theta)^2}{2}\right) \tag{6}$$

we could have specified the model via

$$y = z + \theta \qquad\qquad z = \sqrt{-2\log(\omega)}\cos(2\pi\nu) \tag{7}$$

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

Simulator-based models

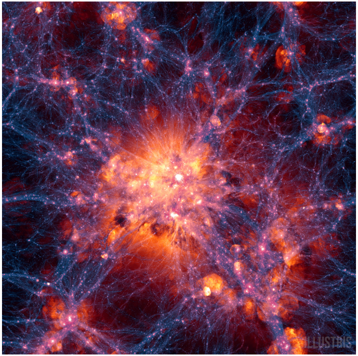
- Sampling from the model is straightforward. For example:
  1. Sampling ω_i and ν_i from the uniform random variables ω and ν,
  2. computing the nonlinear transformation $y_i = f(\omega_i, \nu_i, \theta) = \theta + \sqrt{-2\log(\omega_i)}\cos(2\pi\nu_i)$

  produces samples $y_i \sim p(y|\theta)$.
- Enables direct modeling of how data are generated.
- Names for models specified via a data generating mechanism:
  - Generative models
  - Implicit models
  - Stochastic simulation models
  - Simulator-based models

Examples

Simulator-based models are used in:

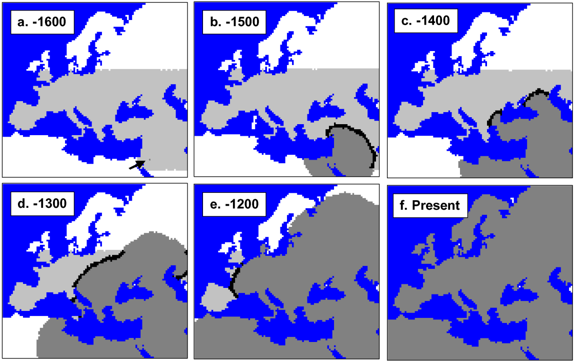
- Astrophysics:
  Simulating the formation of galaxies, stars, or planets
- Evolutionary biology:
  Simulating the evolution of life
- Health science:
  Simulating the spread of an infectious disease
- ...



Dark matter density simulated by the Illustris collaboration
(Figure from `http://www.illustris-project.org`)

Examples (evolutionary biology)

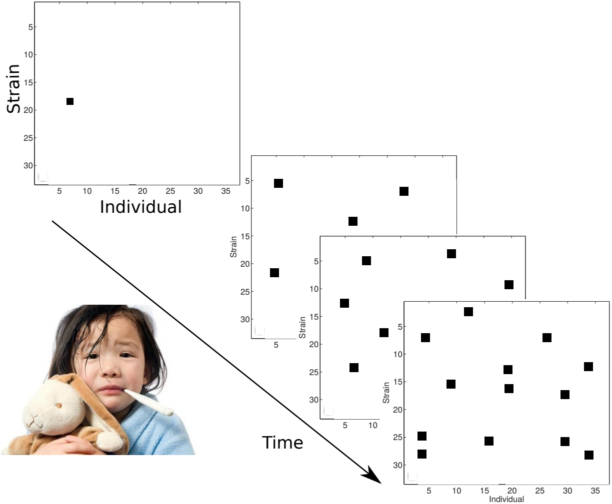
- Simulation of different hypothesized evolutionary scenarios
- Interaction between early modern humans (Homo sapiens) and their Neanderthal contemporaries in Europe



Immigration of Modern Humans into Europe from the Near East. Light gray: Neanderthal population. Dark: Homo sapiens. from (Currat and Excoffier, *Plos Biology*, 2004, 10.1371/journal.pbio.0020421). The numbers in the figures indicate generations. See also Pinhasi et al, The genetic history of Europeans, *Trends in Genetics*, 2012

Examples (health science)

- Simulation of bacterial transmission dynamics in child day care centers (Numminen et al, *Biometrics*, 2013)

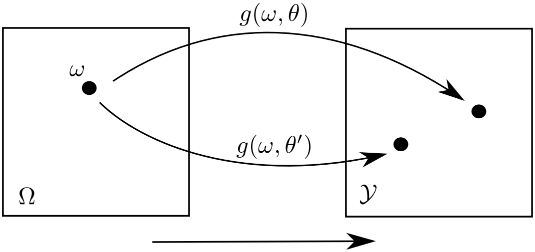


Formal definition of a simulator-based model

- 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 $\boldsymbol{\theta}$,

$$\boldsymbol{\omega} \in \Omega \mapsto \mathbf{y} = g(\boldsymbol{\omega}, \boldsymbol{\theta}) \in \mathcal{Y} \tag{8}$$

- For any fixed $\boldsymbol{\theta}$, $\mathbf{y}_{\boldsymbol{\theta}} = g(., \boldsymbol{\theta})$ is a random variable.



Simulation / Sampling

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!

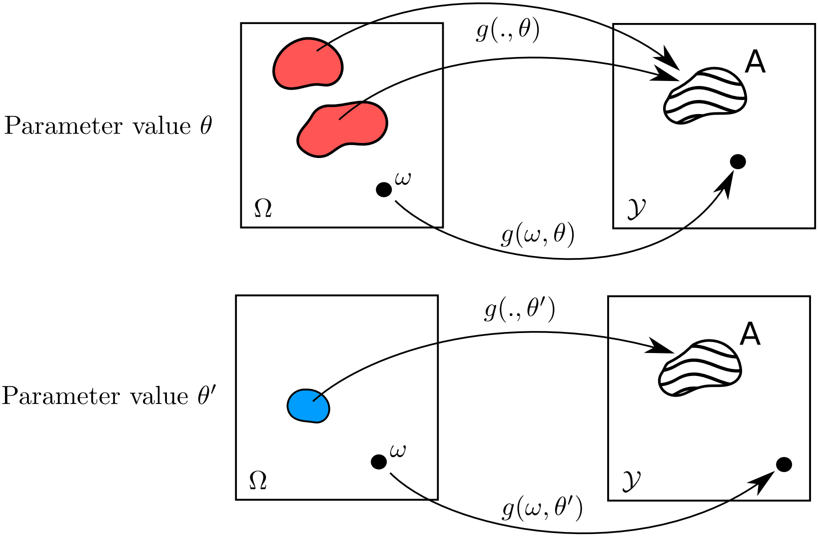
— *This lecture is about inference for simulator-based models* —

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.
- Generally, it is impossible to write down the pdf of $\mathbf{y}_{\boldsymbol{\theta}}$ analytically in closed form.
- No closed-form formulae available for $p(\mathbf{y}|\boldsymbol{\theta})$.
- Simulator defines the model pdfs $p(\mathbf{y}|\boldsymbol{\theta})$ implicitly.

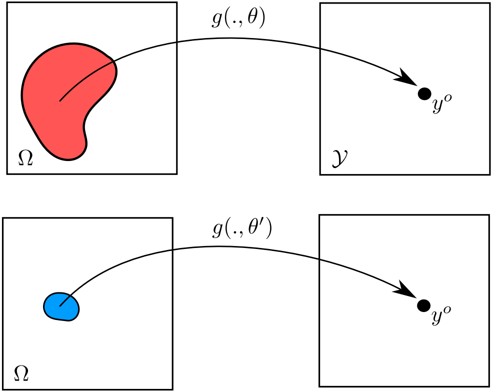
Implicit definition of the model pdfs

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



Implicit definition of the likelihood function

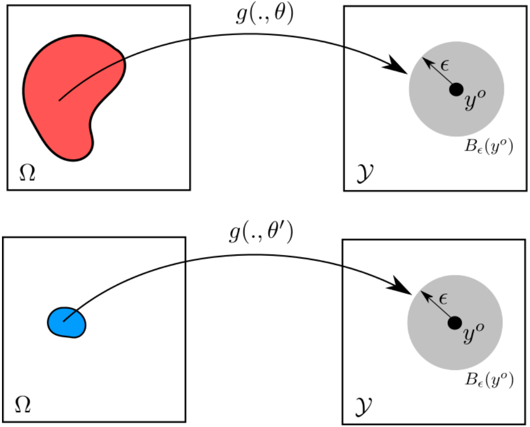
- The implicit definition of the model pdfs implies an implicit definition of the likelihood function. For discrete random variables:
$$L(\theta) = \Pr\left(y = y^o \mid \theta\right) = \mathcal{P}\left(\{\omega : g(\omega, \theta) = y^o\}\right)$$



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

Exact inference for discrete random variables

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

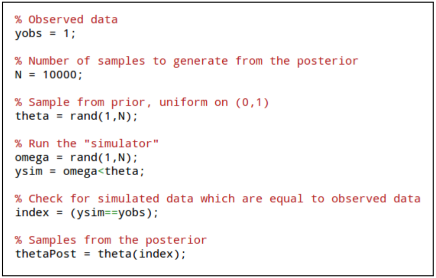
$$p(\boldsymbol{\theta}|\mathbf{y}^o) = \frac{p(\boldsymbol{\theta}, \mathbf{y}^o)}{p(\mathbf{y}^o)} = \frac{p(\boldsymbol{\theta}, \mathbf{y} = \mathbf{y}^o)}{p(\mathbf{y} = \mathbf{y}^o)} \tag{9}$$

- Given tuples $(\boldsymbol{\theta}_i, \mathbf{y}_i)$ where
  - $\boldsymbol{\theta}_i \sim p_{\boldsymbol{\theta}}$ (iid from the prior)
  - $\mathbf{y}_i = g(\boldsymbol{\omega}_i, \boldsymbol{\theta}_i)$ (obtained by running the simulator)

  retain only those where $\mathbf{y}_i = \mathbf{y}^o$.
- The $\boldsymbol{\theta}_i$ from the retained tuples are samples from the posterior $p(\boldsymbol{\theta}|\mathbf{y}^o)$.

Example

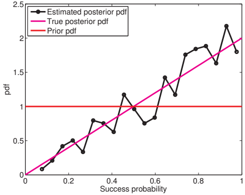
- Posterior inference of the success probability θ in a Bernoulli trial.
- Data: $y^o = 1$
- Prior: $p_\theta = 1$ on $(0, 1)$
- Data generating process:
  - Given $\theta_i \sim p_\theta$
  - $\omega_i \sim U(0, 1)$
  - $y_i = \begin{cases} 1 & \text{if } \omega_i < \theta_i \\ 0 & \text{otherwise}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



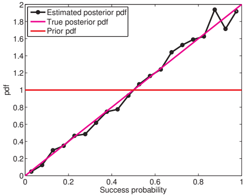
```
% Observed data
yobs = 1;

% 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

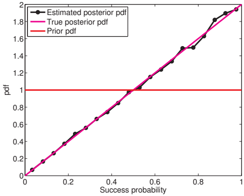
- The method produces samples from the posterior.
- Monte Carlo error when summarizing the samples as an empirical distribution or computing expectations via sample averages.
- Histogram for N simulated tuples (θ_i, y_i)



$N = 1000$



$N = 10,000$



$N = 100,000$

The good and the bad

- The method produces samples from $p(\boldsymbol{\theta}|\mathbf{y}^o)$.
- This is good.
- But 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
- This is bad.

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 $\mathbf{t}^o$,

$$\mathbf{t}_{\boldsymbol{\theta}} = T(\mathbf{y}_{\boldsymbol{\theta}}) \qquad \mathbf{t}^o = T(\mathbf{y}^o). \tag{10}$$

  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)
- In other words:
  1. Replace $\Pr(\mathbf{y} \in B_{\epsilon'}(\mathbf{y}^o) \mid \boldsymbol{\theta})$ with $\Pr(\Delta_{\boldsymbol{\theta}} \leq \epsilon | \, \boldsymbol{\theta})$
  2. Do not take the limit $\epsilon \to 0$
- Defines an approximate likelihood function $\tilde{L}_\epsilon(\boldsymbol{\theta})$,

$$\tilde{L}_\epsilon(\boldsymbol{\theta}) \propto \Pr(\Delta_{\boldsymbol{\theta}} \leq \epsilon \mid \boldsymbol{\theta}) \tag{11}$$

Example

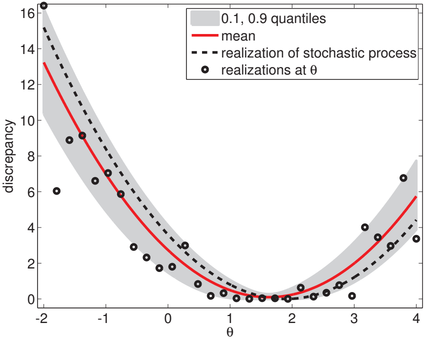
- Inference of the mean θ of a Gaussian of variance one.
- $\Pr(\mathbf{y} = \mathbf{y}^o | \boldsymbol{\theta}) = 0$.
- Discrepancy Δ_θ:

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

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

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

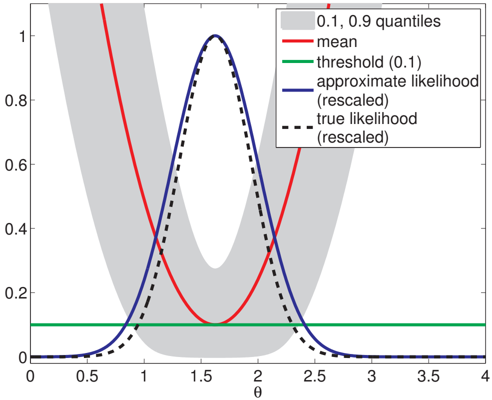
$$y_i \sim \mathcal{N}(\theta, 1)$$



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.

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

$$\tilde{p}_\epsilon(\boldsymbol{\theta}|\mathbf{y}^o) \propto p_{\boldsymbol{\theta}}(\boldsymbol{\theta})\tilde{L}_\epsilon(\boldsymbol{\theta}) \tag{12}$$

$$\tilde{L}_\epsilon(\boldsymbol{\theta}) \propto \Pr\Big(\underbrace{d(T(\mathbf{y}^o), T(\mathbf{y}))}_{\Delta_{\boldsymbol{\theta}}} \leq \epsilon \mid \boldsymbol{\theta}\Big) \tag{13}$$

Part II

ABC methods used in practice

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.

Critique of the rejection ABC algorithm

- The rejection ABC algorithm works.
- But it is computationally not efficient.
- The probability of the event $\Delta_{\boldsymbol{\theta}} \leq \epsilon$ is usually small when $\boldsymbol{\theta} \sim p_{\boldsymbol{\theta}}$. In particular for small ϵ.

Critique of the rejection ABC algorithm

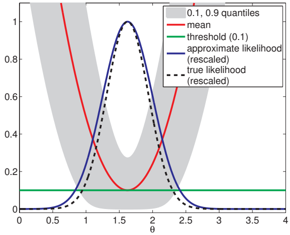
- In the Gaussian example, the probability for $\Delta_{\boldsymbol{\theta}} \leq \epsilon$ can be computed in closed form

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

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

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

- 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 upon rejection ABC:
  1. Regression ABC (Beaumont et al, *Genetics*, 2002)
  2. Sequential Monte Carlo ABC (Sisson et al, *PNAS*, 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)

$$\boldsymbol{\theta}_i = f(\mathbf{t}_i) + \boldsymbol{\xi}_i \tag{14}$$

  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 $\hat{\boldsymbol{\xi}}_i$,

$$\hat{\boldsymbol{\xi}}_i = \boldsymbol{\theta}_i - \hat{f}(\mathbf{t}_i) \tag{15}$$

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

$$\boldsymbol{\theta}_i^* = \hat{f}(\mathbf{t}^o) + \hat{\boldsymbol{\xi}}_i = \hat{f}(\mathbf{t}^o) + \boldsymbol{\theta}_i - \hat{f}(\mathbf{t}_i) \tag{16}$$

- 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

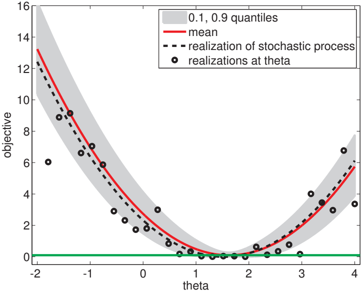
- Parameters $\boldsymbol{\theta}_i$ weighted with w_i,

$$w_i = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{\theta}_i)}{\phi(\boldsymbol{\theta}_i)}, \tag{17}$$

  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).
- More efficient than rejection ABC: $\phi_t(\boldsymbol{\theta})$ is close to the approximate posterior in the final iterations.

Another approach

- Evaluating $\Delta_{\boldsymbol{\theta}}$ is computationally costly. We are only interested in small $\Delta_{\boldsymbol{\theta}}$ (thresholding!)
- We could increase the computational efficiency by evaluating $\Delta_{\boldsymbol{\theta}}$ predominantly where it tends to be small.
- Use a combination of probabilistic modeling of $\Delta_{\boldsymbol{\theta}}$ and optimization to figure out where to evaluate $\Delta_{\boldsymbol{\theta}}$.



Learning a model of the discrepancy

- The approximate likelihood function $\tilde{L}_\epsilon(\boldsymbol{\theta})$ is determined by the distribution of the discrepancy $\Delta_{\boldsymbol{\theta}}$

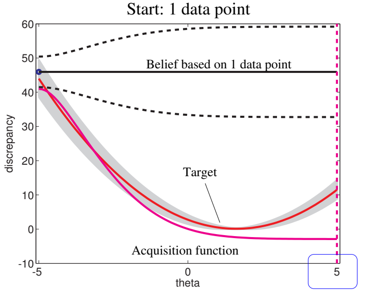
$$\tilde{L}_\epsilon(\boldsymbol{\theta}) \propto \Pr\left(\Delta_{\boldsymbol{\theta}} \leq \epsilon \mid \boldsymbol{\theta}\right)$$

- If we new the distribution of $\Delta_{\boldsymbol{\theta}}$ we could compute $\tilde{L}_\epsilon(\boldsymbol{\theta})$.
- In recent work, we proposed to learn a model of $\Delta_{\boldsymbol{\theta}}$ and to approximate $\tilde{L}_\epsilon(\boldsymbol{\theta})$ by $\hat{L}_\epsilon(\boldsymbol{\theta})$,

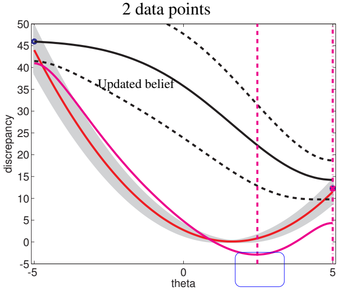
$$\tilde{L}_\epsilon(\boldsymbol{\theta}) \propto \widehat{\Pr}\left(\Delta_{\boldsymbol{\theta}} \leq \epsilon \mid \boldsymbol{\theta}\right), \tag{18}$$

  where $\widehat{\Pr}$ is the probability under the model of $\Delta_{\boldsymbol{\theta}}$.
  (Gutmann and Corander, *Journal of Machine Learning Research*, in press, 2015)
- Model is learned more accurately in regions where $\Delta_{\boldsymbol{\theta}}$ tends to be small, using techniques from Bayesian optimization.

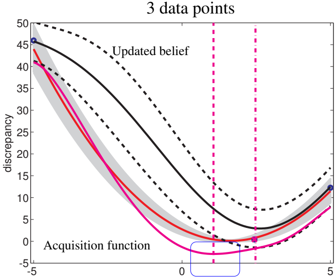
Bayesian optimization



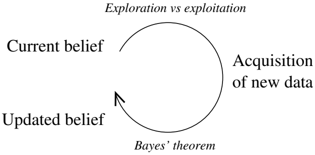
Where to evaluate next



Where to evaluate next



Where to evaluate next



Exploration vs exploitation

Current belief

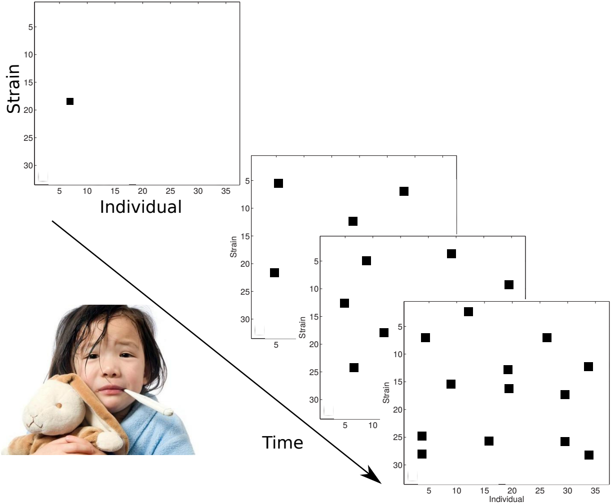
Acquisition of new data

Updated belief

Bayes' theorem

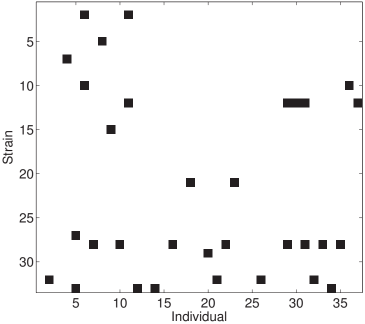
Application to epidemiology of infectious diseases

- Inference about bacterial transmission dynamics in child day care centers (Numminen et al, *Biometrics*, 2013)



Application to epidemiology of infectious diseases

Data: Colonization states of sampled attendees of 29 child day care centers (DCCs) in Oslo greater area.



Example data from a DCC. Each square indicates an attendee colonized with a strain of the bacterium *Streptococcus pneumoniae*.

Application to epidemiology of infectious diseases

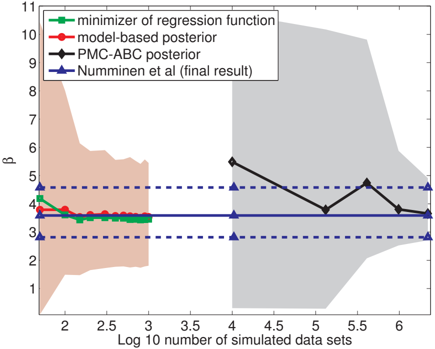
- Simulator-based model: latent continuous-time Markov chain for the transmission dynamics in a DCC and an observation model (Numminen et, Biometrics, 2013).
- The model has three parameters:
    - β: rate of infections within a DCC
    - Λ: rate of infections outside a DCC
    - θ: possibility to be infected with multiple strains
- Likelihood is intractable (data at a single time point are available only).

Application to epidemiology of infectious diseases

- Comparison of the model-based approach with a sequential/population Monte Carlo ABC approach.
- Roughly equal results using 1000 times fewer simulations.

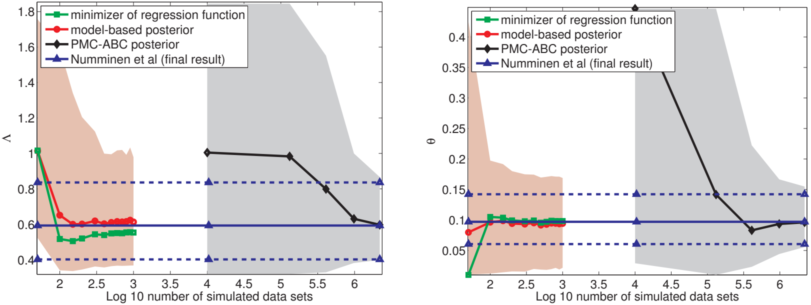
- The minimizer of the regression function under the model does not involve choosing a threshold ϵ.

Posterior means: solid lines with markers,
credibility intervals: shaded areas or dashed lines.



Application to epidemiology of infectious diseases

- Comparison of the model-based approach with a sequential/population Monte Carlo ABC approach.



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

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
- Introduced recent work which uses Bayesian optimization to increase the efficiency of the inference.
- Not covered: How to choose the summary statistics / the discrepancy measure between simulated and observed data.