Accelerating Approximate Bayesian Computation with Kernels and Decision Making under Uncertainty

Michael U. Gutmann

School of Informatics
The University of Edinburgh
michael.gutmann@ed.ac.uk

January 12 2022

Program

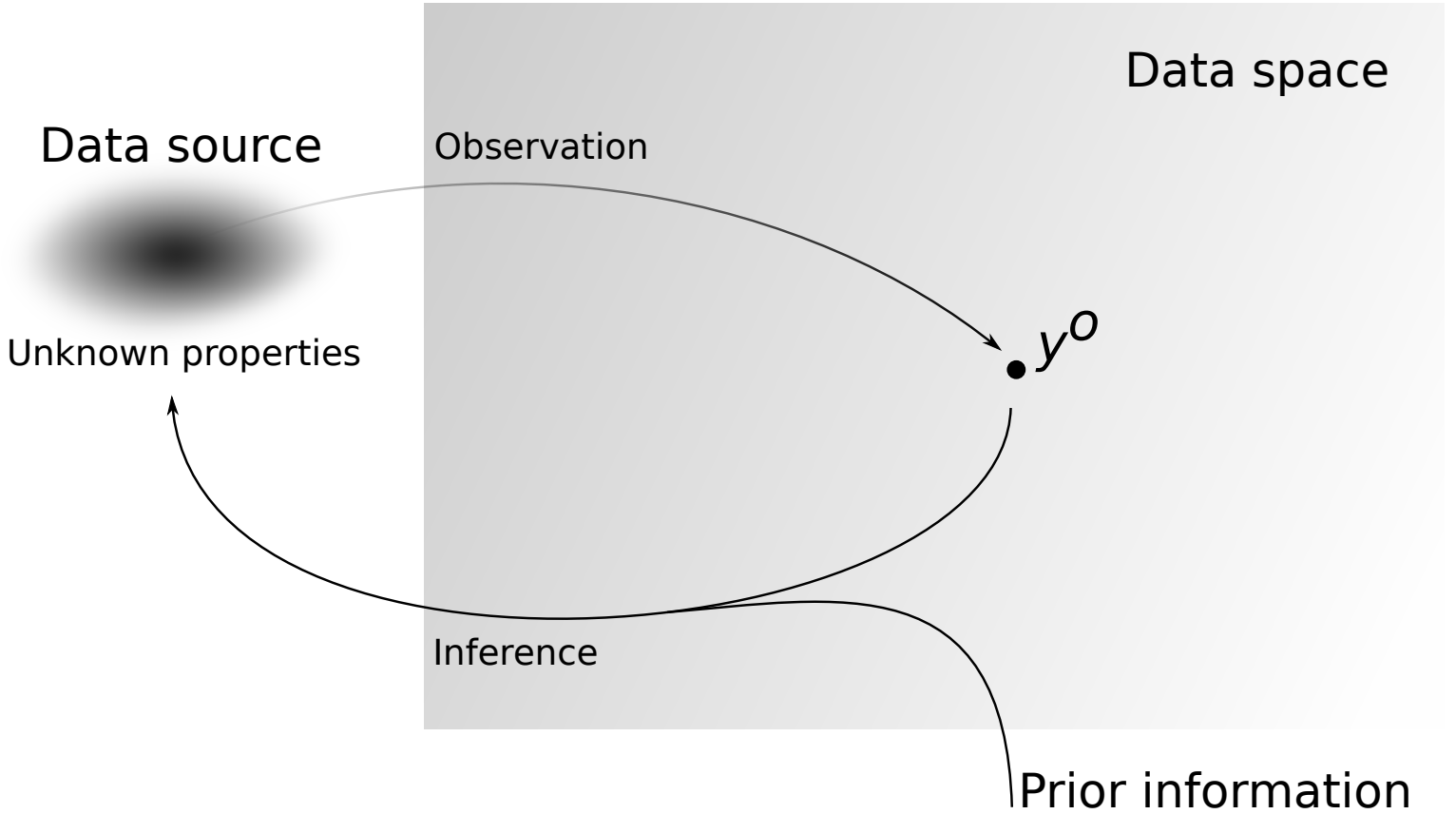
1. Simulator-based models
2. Classical algorithms for approximate Bayesian computation
3. Accelerating ABC

Program

1. Simulator-based models
   - Statistical inference
   - Simulator-based models
   - Implicit definition of the model pdf
2. Classical algorithms for approximate Bayesian computation
3. Accelerating ABC

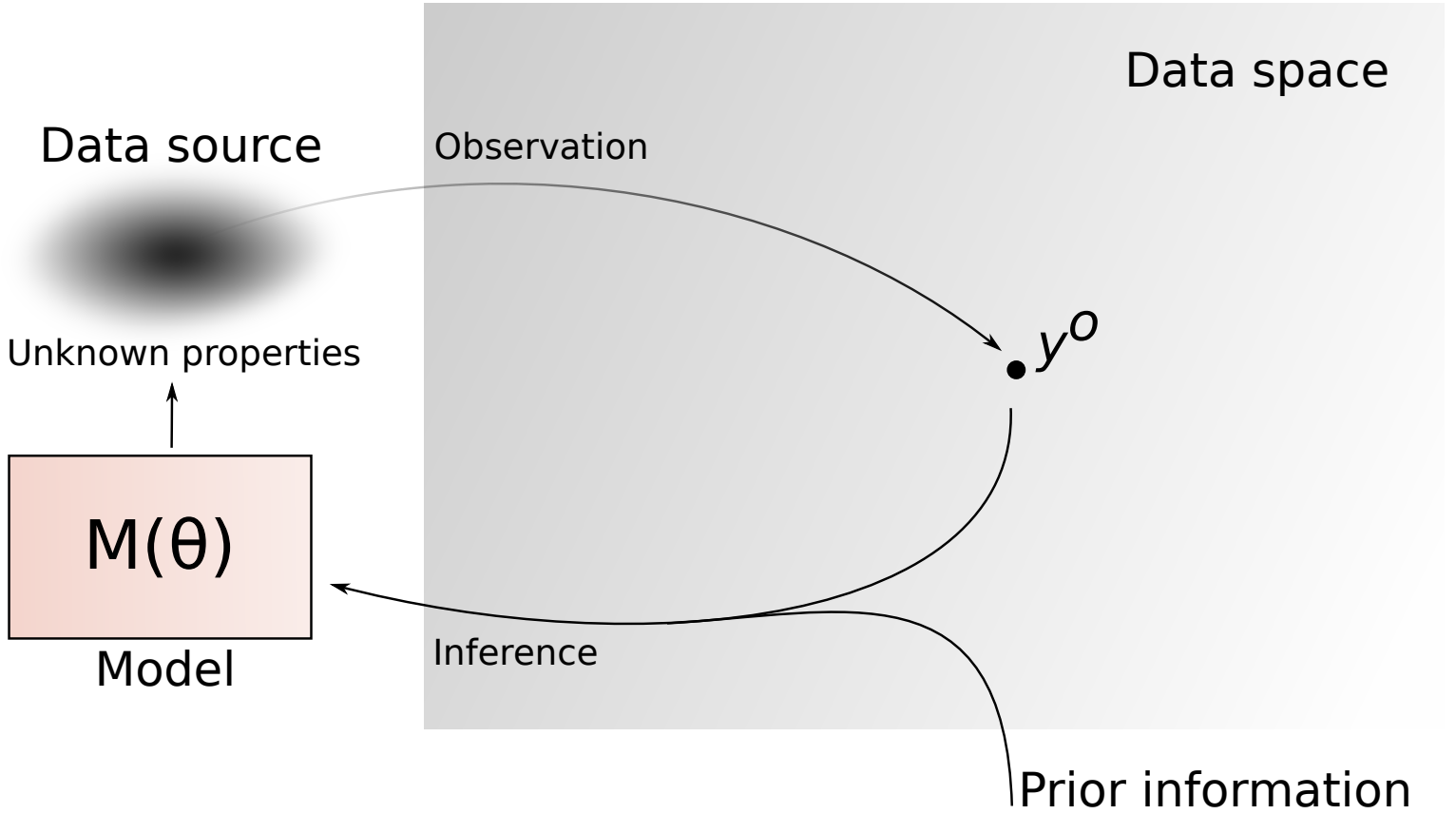
Big picture of statistical inference

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



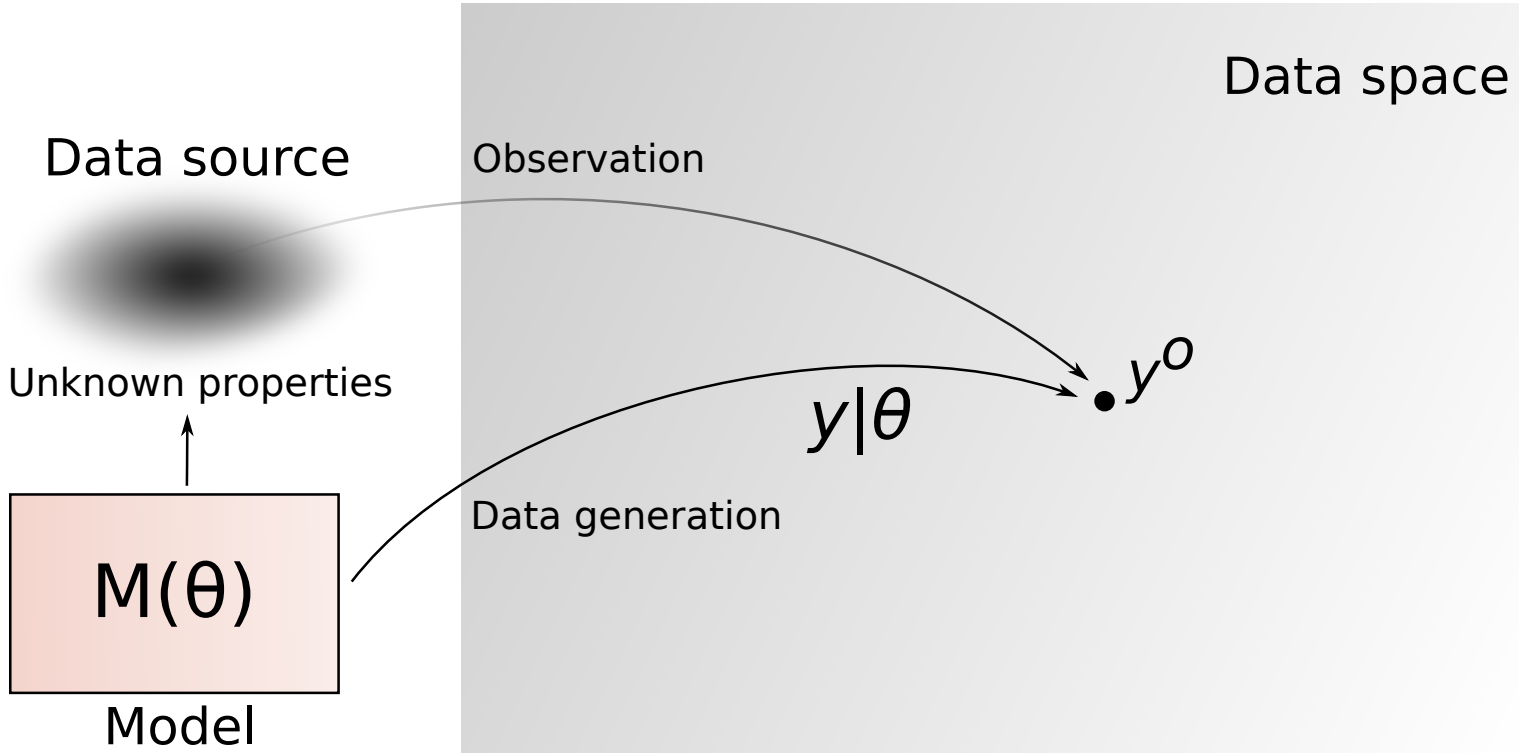
General approach

- Set up a model with potential properties θ (parameters)
- See which θ are reasonable given the observed data



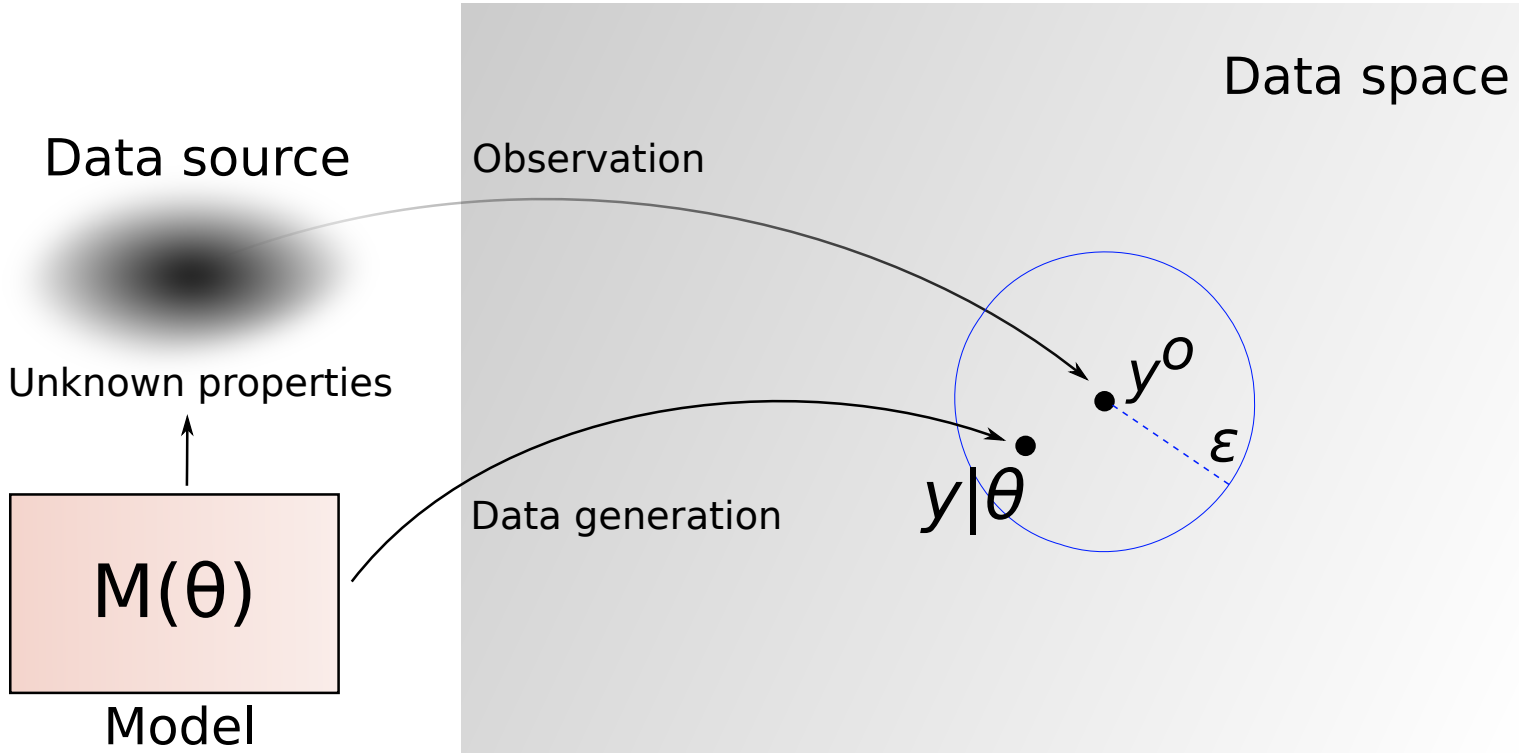
Likelihood function

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



Likelihood function

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



Likelihood function

- For discrete random variables:

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

- For continuous random variables:

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

Performing statistical inference

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

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L(\boldsymbol{\theta})$$

- Bayesian inference

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

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

- Solving the optimisation/sampling problem can be computationally very difficult.

Simulator-based models

- In this talk, we consider another difficulty:

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

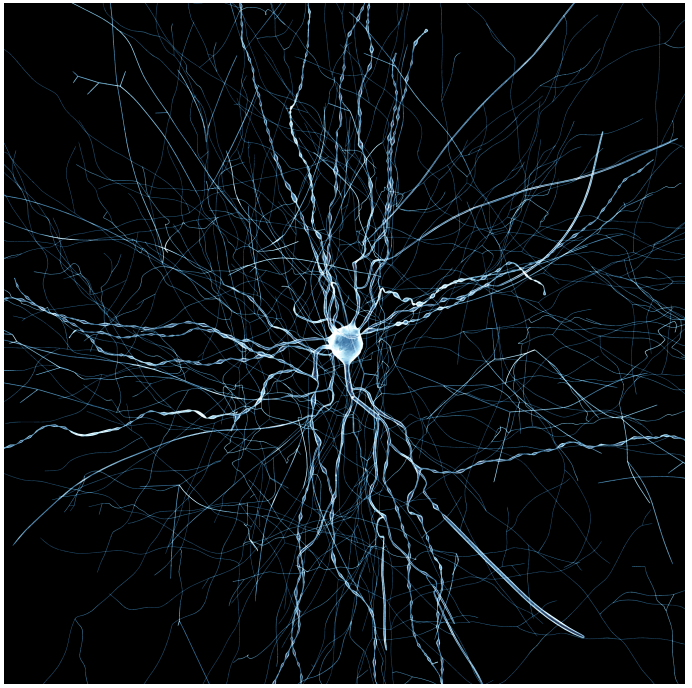
  models which are specified via a mechanism (rule) for generating data

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

Simulator-based models are widely used

- 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`)

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:

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

or

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

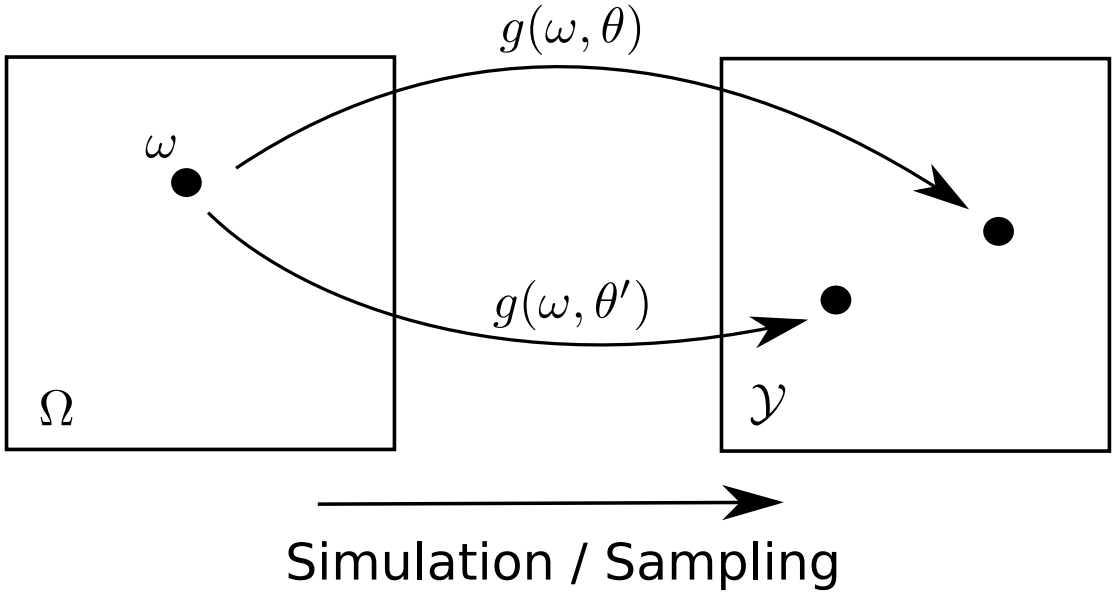
where ω and ν are independent random variables uniformly distributed on $(0, 1)$

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 but implicitly defined by a computer programme.



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.
- Modelling 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 easily be made more complicated than necessary ($\rightarrow$ possible identifiability issues).
- Statistical inference is difficult.

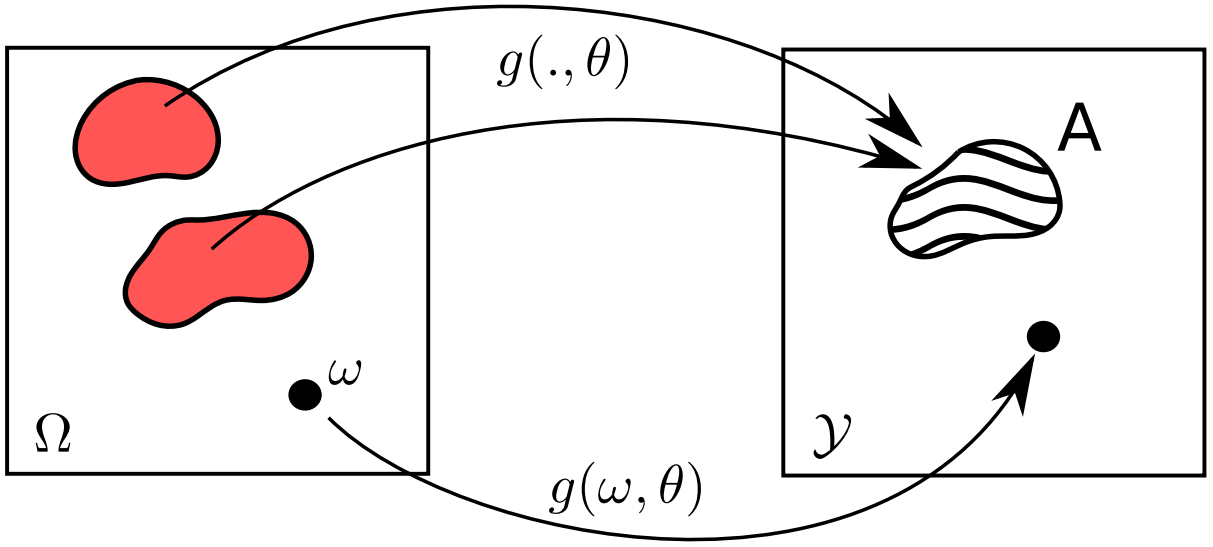
Family of pdfs induced by the simulator

- 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.

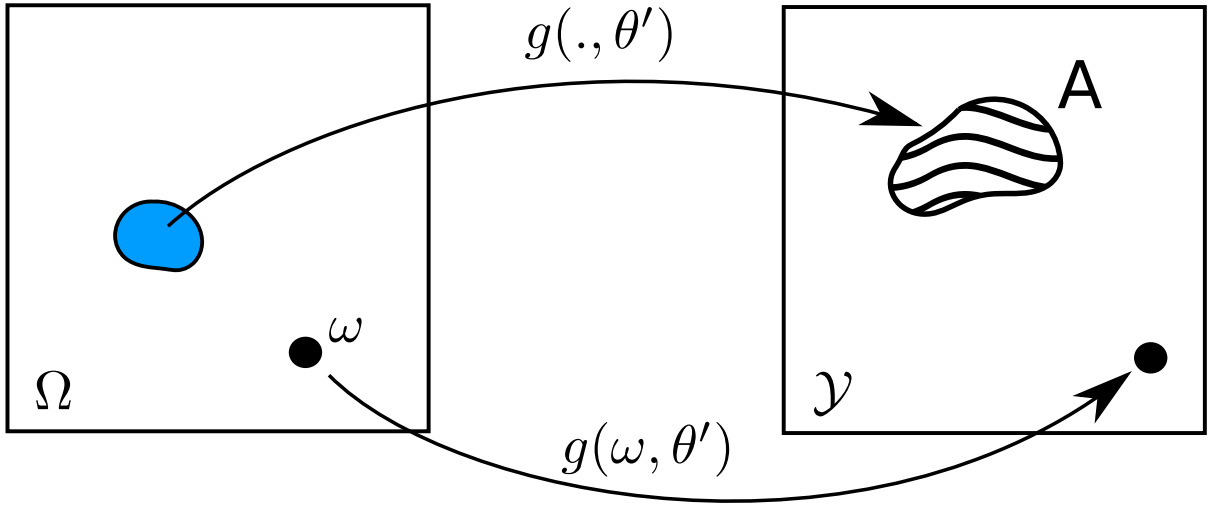
Implicit definition of the model distribution

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

Parameter value θ



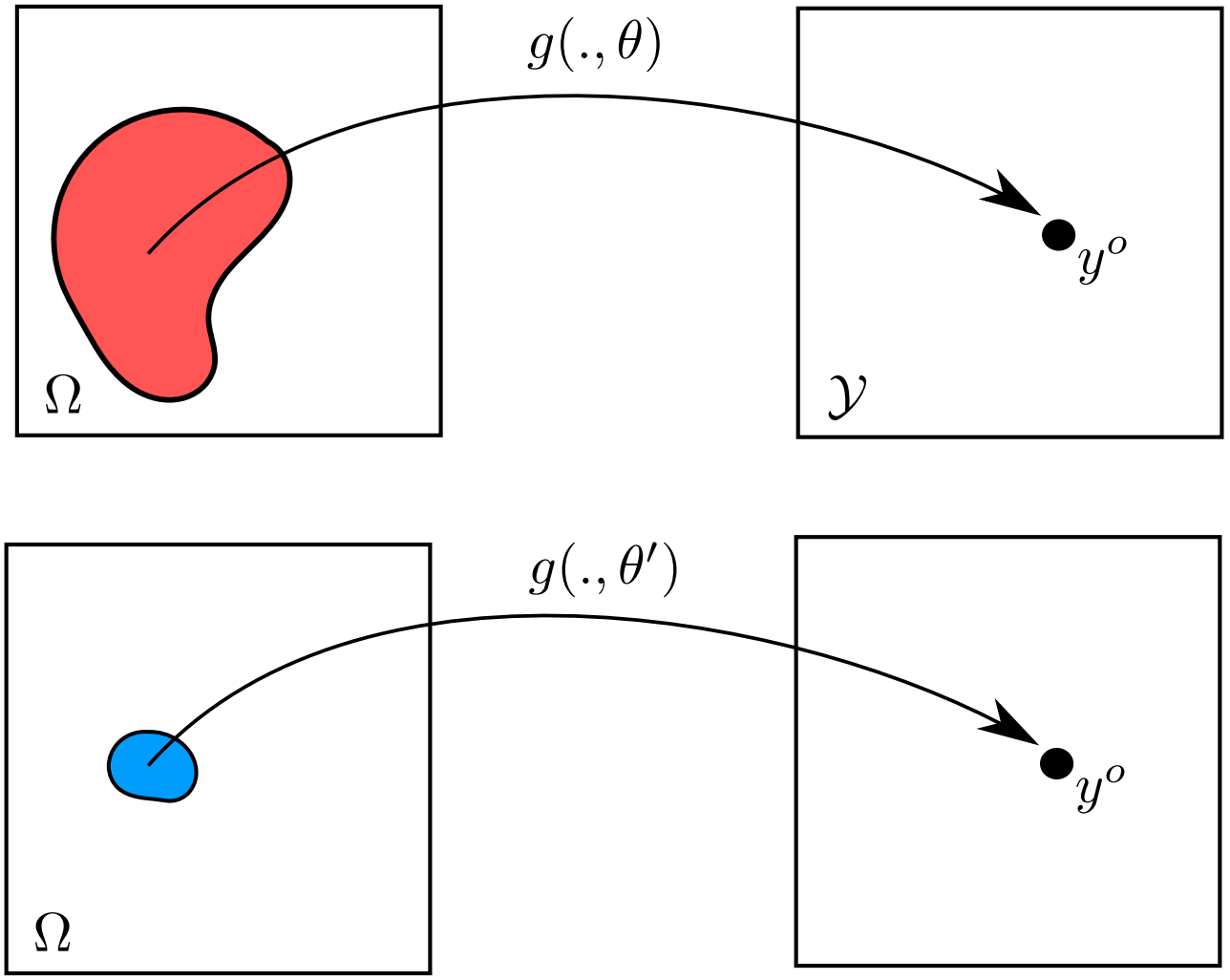
Parameter value θ'



Implicit definition of the likelihood function

For discrete random variables:

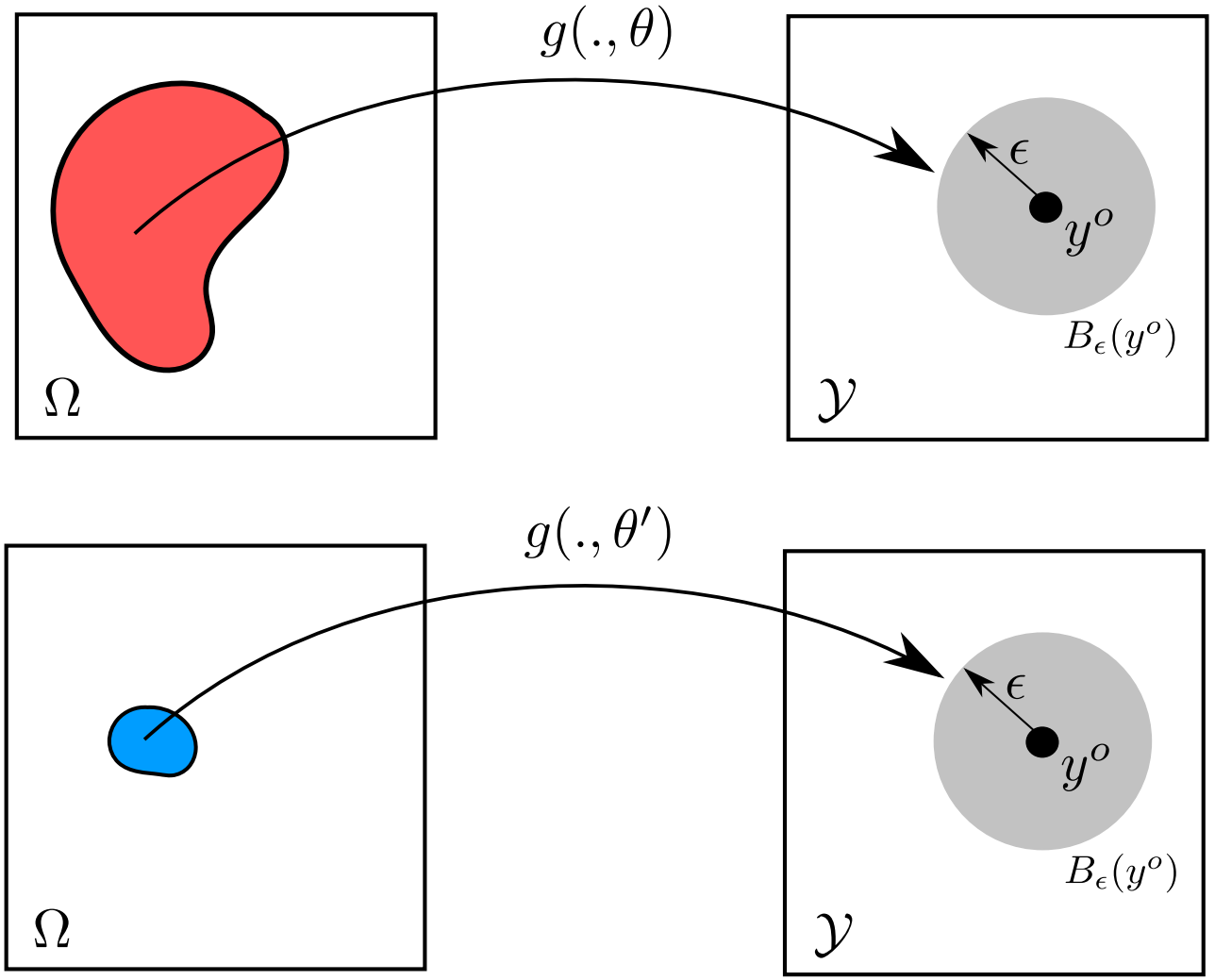
$$L(\theta) = \Pr\left(y = y^o \mid \theta\right) = \mathcal{P}\left(\{\omega : g(\omega, \theta) = y^o\}\right)$$



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 $\mathbf{y}^o$,

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

- 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.

Different inference approaches

- There are several flavors of parameter inference for simulator-based models. In Bayesian setting e.g.
  - Approximate Bayesian computation (ABC)
  - Synthetic likelihood (Wood, 2010; Price et al 2017)
  - Likelihood-free inference by ratio estimation (Thomas et al 2016; Hermans et al 2020)
  - ...
- Here: Focus on ABC.

Program

1. Simulator-based models
   - Statistical inference
   - Simulator-based models
   - Implicit definition of the model pdf
2. Classical algorithms for approximate Bayesian computation
3. Accelerating ABC

Program

1. Simulator-based models

2. Classical algorithms for approximate Bayesian computation
   - Exact inference
   - Need for approximations
   - Algorithms

3. Accelerating ABC

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 $p(\boldsymbol{\theta}|\mathbf{y}^o)$.

Limitations

- Only applicable to discrete random variables.
- And even for discrete random variables:
  Computationally infeasible in higher dimensions
- Reason: *The probability of the event* $\mathbf{y}_\theta = \mathbf{y}^o$ *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

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

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

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

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

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

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

- Inference is approximate due to
  - the summary statistics T and distance d
  - $\epsilon > 0$
  - the finite number of samples (Monte Carlo error)
- Robust but slow algorithm
  - ϵ needs to be small to reduce bias, but this causes a low acceptance rate
  - low acceptance rate when the likelihood is much more concentrated than the prior

Two widely used algorithms

- Two widely used algorithms which improve upon rejection ABC:
  1. Regression ABC (Beaumont et al, 2002, Blum and Francois, 2010)
  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 $\mathbf{t}_\theta = T(\mathbf{y}_\theta)$ 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 $\phi(\boldsymbol{\theta})\tilde{L}_\epsilon(\boldsymbol{\theta})$

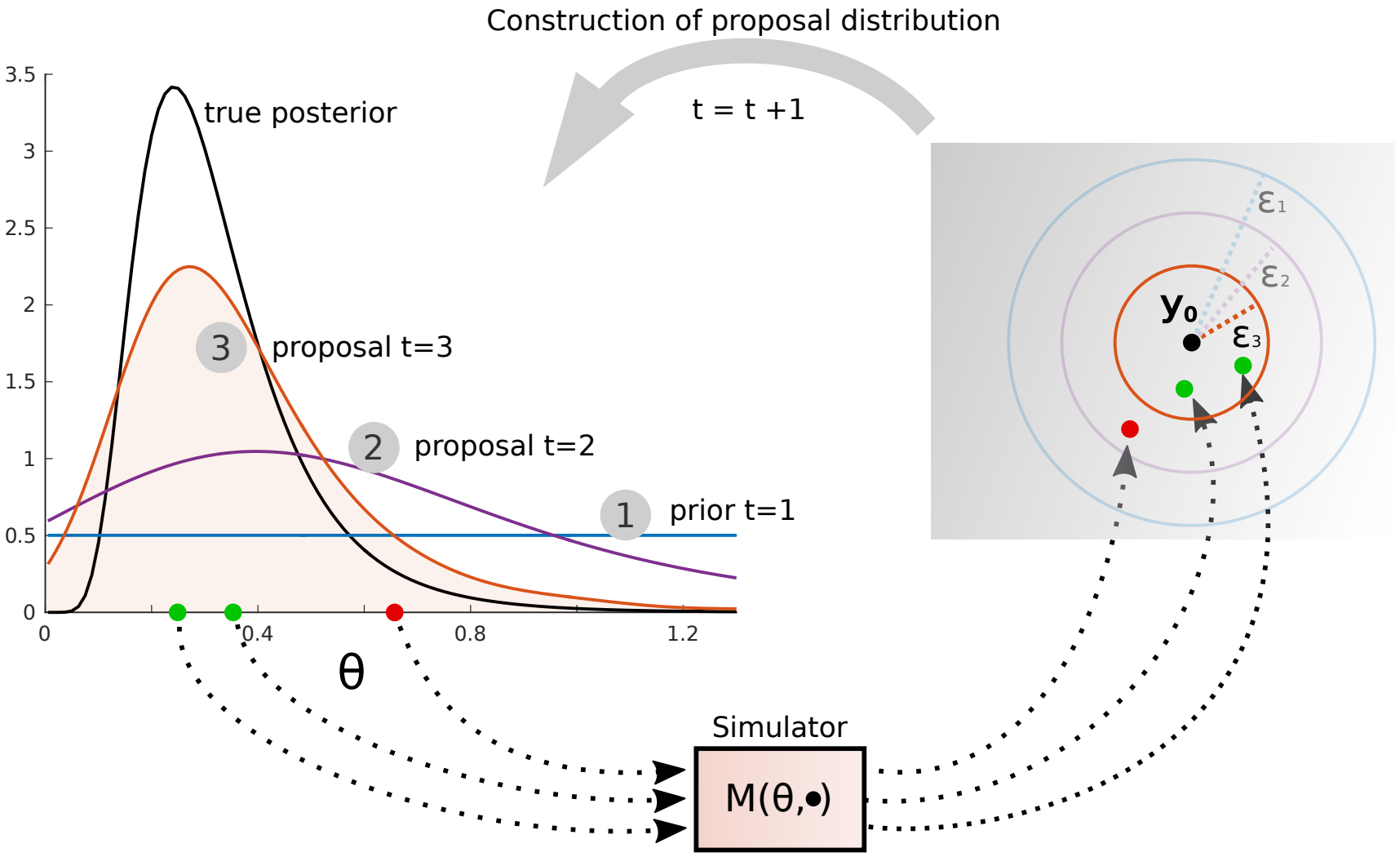
Basic idea of sequential Monte Carlo ABC

- 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).

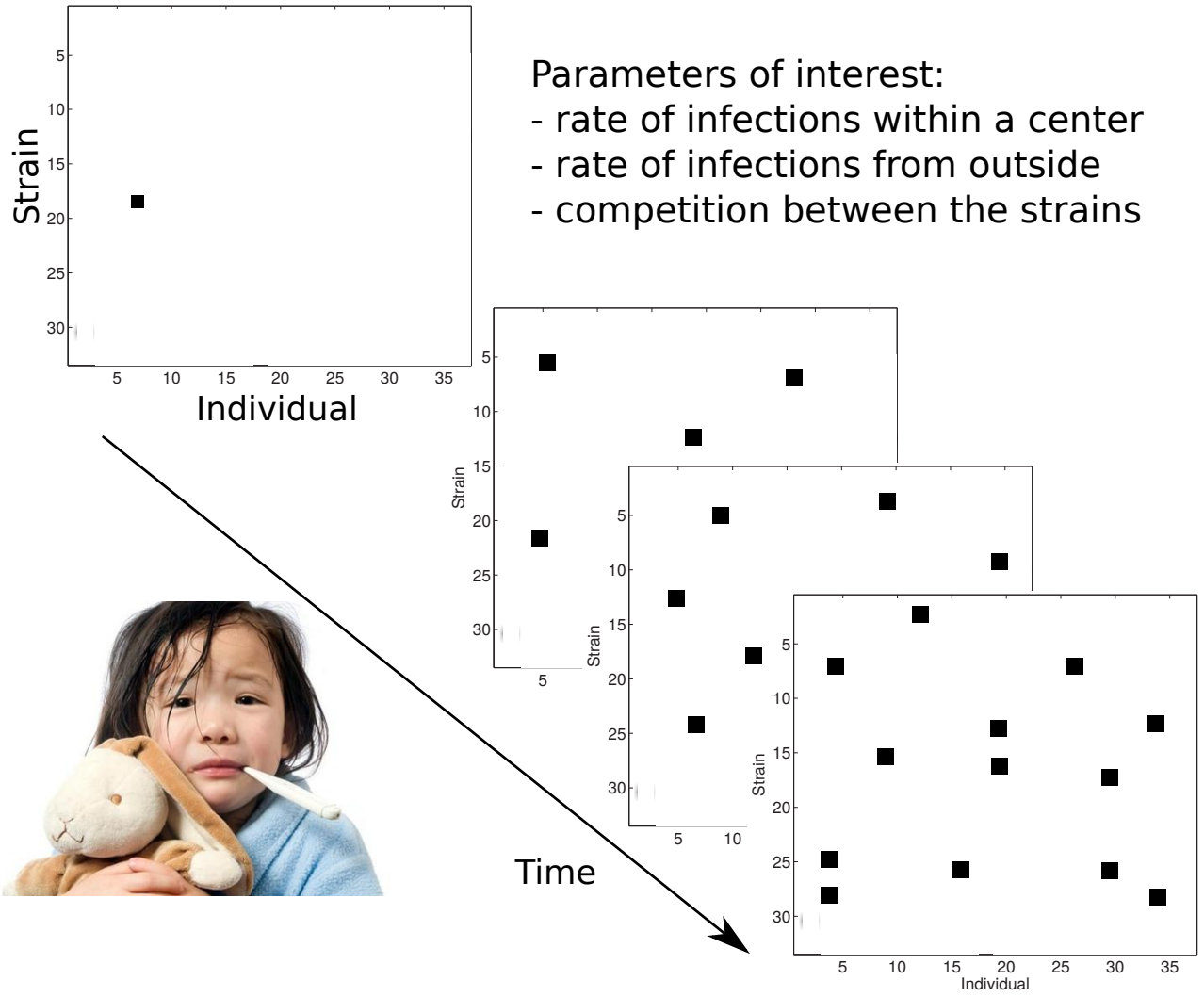
Basic idea of sequential Monte Carlo ABC



Example: Bacterial infections in child care centers

- Simulating bacterial transmissions in child day care centers

  (Numminen et al, 2013)

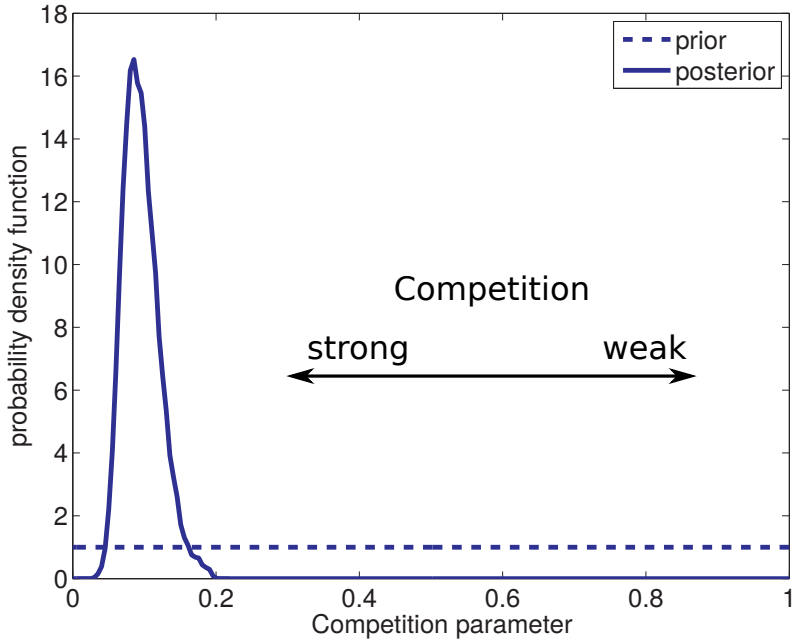


Example: Bacterial infections in child care centers

- Data: *Streptococcus pneumoniae* colonization for 29 centers
- Inference with Population Monte Carlo ABC
- Reveals strong competition between different bacterial strains

Expensive:

- 4.5 days on a cluster with 200 cores
- More than one million simulated data sets



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.
- Approximate likelihood function: Probability to generate data for which some discrepancy measure is less than some threshold.
- Rejection ABC: Trial and error scheme to find parameter values which produce simulated data resembling the observed data.
- Regression and sequential Monte Carlo ABC improve upon rejection ABC. But are still expensive.

Program

1. Simulator-based models

2. Classical algorithms for approximate Bayesian computation
   - Exact inference
   - Need for approximations
   - Algorithms

3. Accelerating ABC

Program

1. Simulator-based models

2. Classical algorithms for approximate Bayesian computation

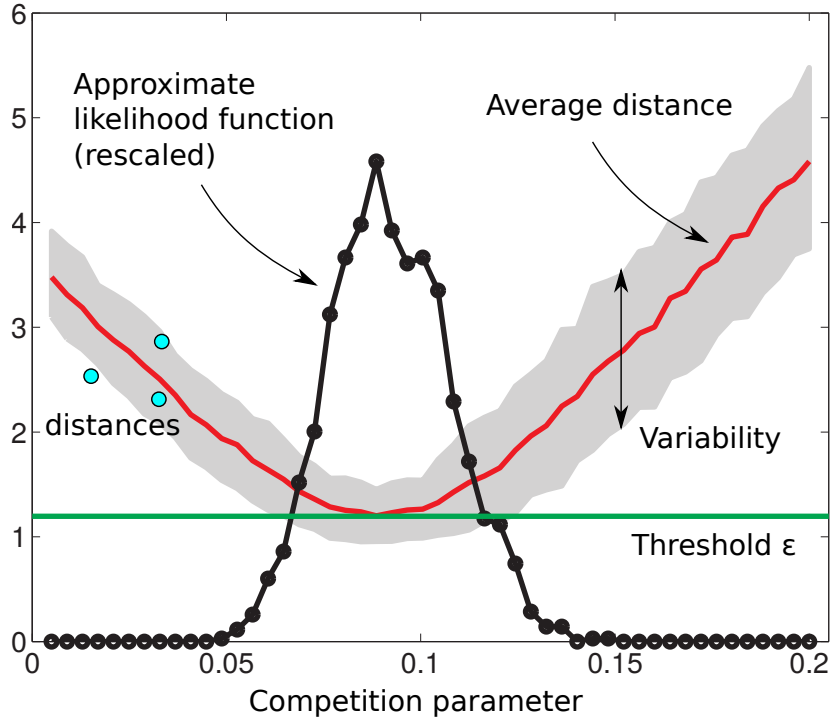
3. Accelerating ABC
   - Why are the classical algorithms so expensive?
   - Framework to accelerate the inference
   - Choice of the acquisition function

Why is the ABC algorithm so expensive?

1. It rejects most samples when ϵ is small
2. It does not make assumptions about the shape of $L(\theta)$
3. It does not use all information available
4. It does not take the finite computational budget into account

$$\tilde{L}_\epsilon(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\left(d(\mathbf{y}_\theta^{(i)}, \mathbf{y}^o) \leq \epsilon\right)$$

Approximate lik function for competition parameter. $N = 300$.



Proposed solution

(Gutmann and Corander, 2016)

1. It rejects most samples when ϵ is small
   $\Rightarrow$ Don't reject samples – learn from them
2. It does not make assumptions about the shape of $L(\boldsymbol{\theta})$
   $\Rightarrow$ Model the distances, assume average distance is smooth
3. It does not use all information available
   $\Rightarrow$ Incorporate new information using Bayes' theorem
4. It does not take finite computational budget into account
   $\Rightarrow$ Decide where to allocate the computational resources

equivalent strategy applies to inference with synthetic likelihood

Conceptual connection to classical algorithms

- The hallmarks of the proposed approach are
  (a) modelling (points 1 and 2)
  (b) using acquired information (data) to decide where to allocate the computational resources (points 3 and 4)
- Regression and SMC ABC have elements of the proposed approach:
  - Regression ABC: Fits an auxiliary (linear) model to perform the adjustment. $\rightarrow$ (a)
  - SMC: Proposal distribution is constructed based on previously simulated data, thus using previously simulated data to "decide" for which parameters to run the simulator next. $\rightarrow$ (b)
- Combining (a) & (b) is key to increasing the performance (e.g. Chen and Gutmann, 2019).
- Most modern algorithm for ABC do it (implicitly) in some way. In this talk, we will focus on Gaussian processes and Bayesian decision making.

Modelling

- Data $\mathcal{D}_t$ are tuples $(\boldsymbol{\theta}_i, \Delta_i)$, $i = 1, \ldots, t$, where $\Delta_i = d(\mathbf{y}_{\boldsymbol{\theta}}^{(i)}, \mathbf{x}^o)$
- Model the conditional distribution of Δ given $\boldsymbol{\theta}$
- Estimated model yields approximation $\hat{L}_\epsilon(\boldsymbol{\theta})$ for any choice of ϵ

$$\hat{L}_\epsilon(\boldsymbol{\theta}) \propto \widehat{\mathbb{P}}(\Delta \leq \epsilon \mid \boldsymbol{\theta})$$

  $\widehat{\mathbb{P}}$ is probability under the estimated model.
- Here: Use (log) Gaussian process as model (with squared exponential covariance function)

  (see Järvenpää et al, 2018, on GP model selection)

Decision making to allocate computational resources

- For which $\boldsymbol{\theta}$ should we run the simulator?
- Intuition: Give priority to regions in the parameter space where the distance tends to be small.
- Piggy-back on Bayesian optimisation to find such regions using the lower confidence bound acquisition function (e.g. Srinivas et al, 2012)

$$\mathcal{A}_t(\boldsymbol{\theta}) = \underbrace{\mu_t(\boldsymbol{\theta})}_{\text{post mean}} - \sqrt{\underbrace{\eta_t^2}_{\text{weight}} \underbrace{v_t(\theta)}_{\text{post var}}} \tag{16}$$

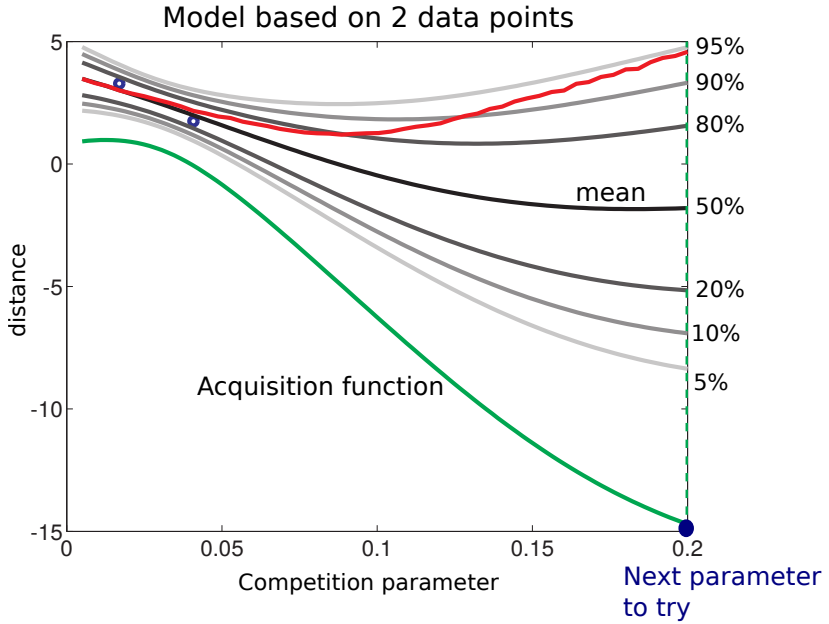
  t: number of samples acquired so far

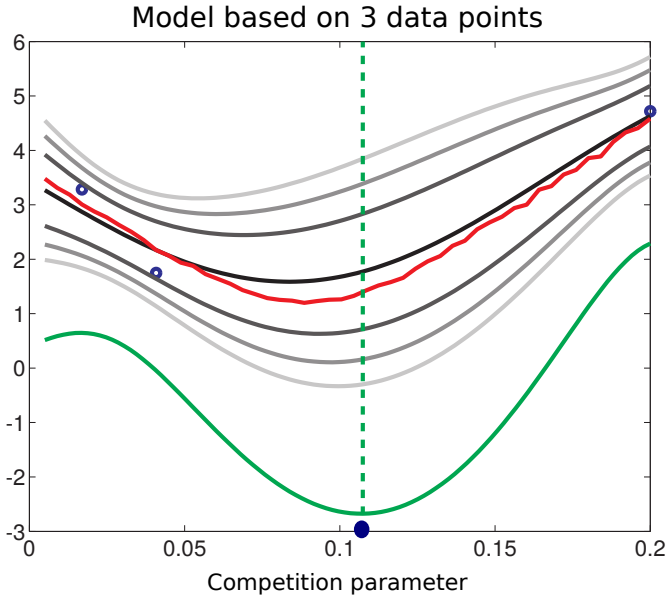
- Run simulator next for

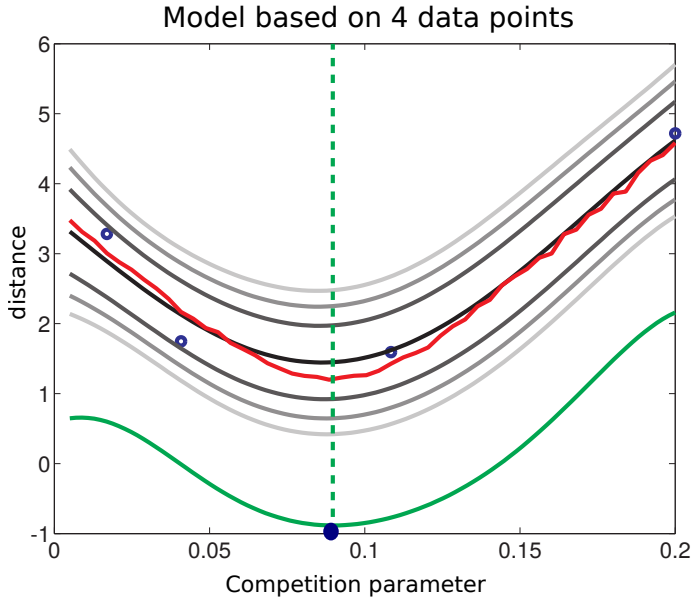
$$\boldsymbol{\theta}_{t+1}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{A}_t(\boldsymbol{\theta}) \tag{17}$$

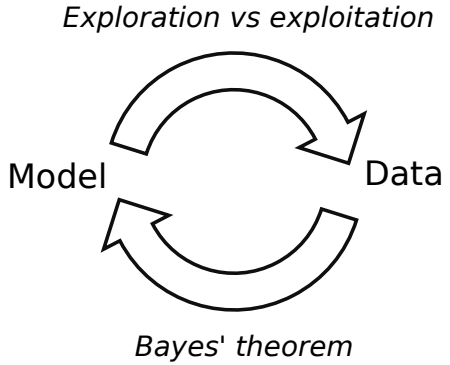
- Approach not restricted to this acquisition function.

Bayesian optimisation for likelihood-free inference









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.

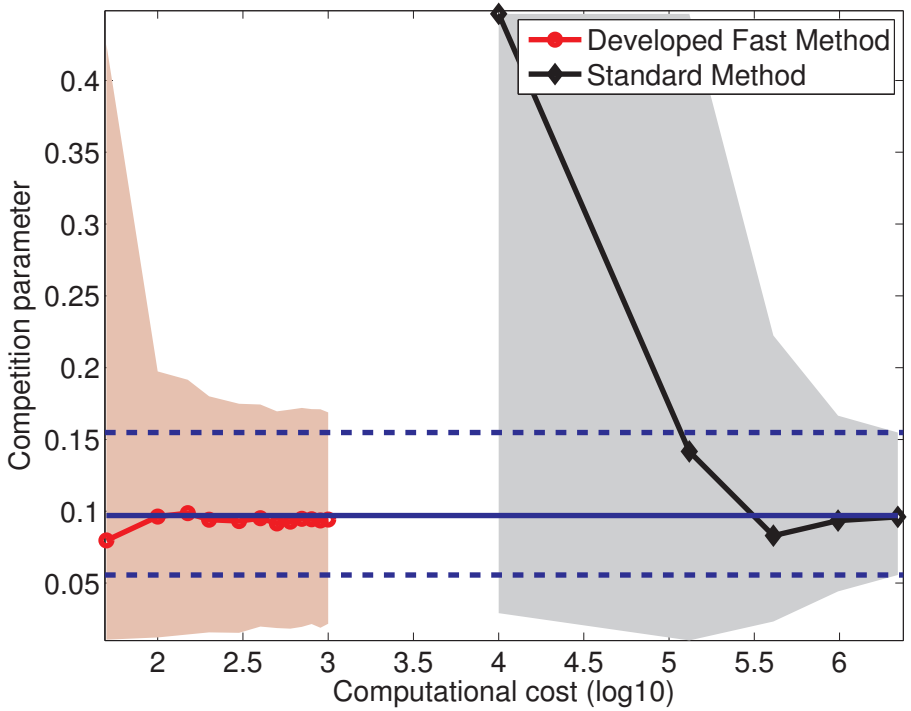
4.5 days with 200 cores

↓

90 minutes with seven cores

Posterior means: solid lines,

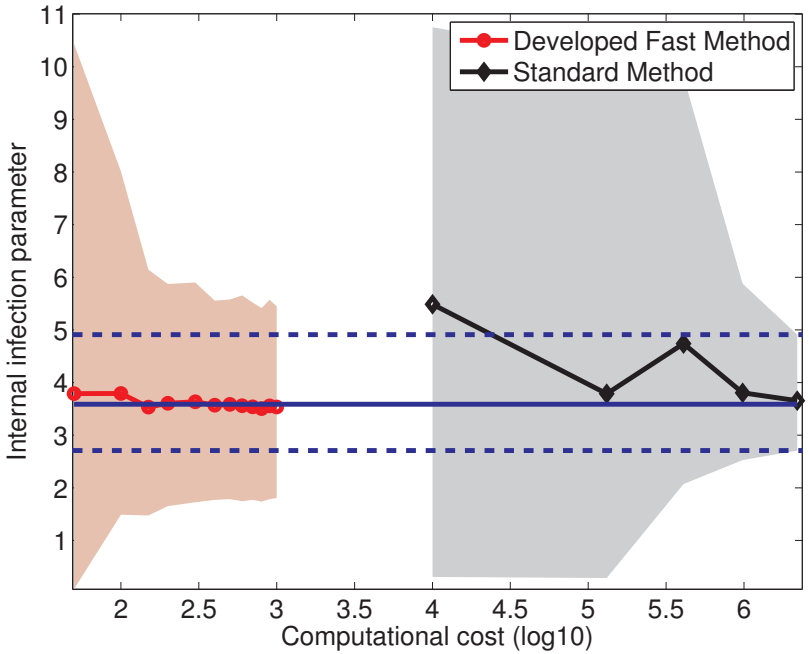
credibility intervals: shaded areas or dashed lines.

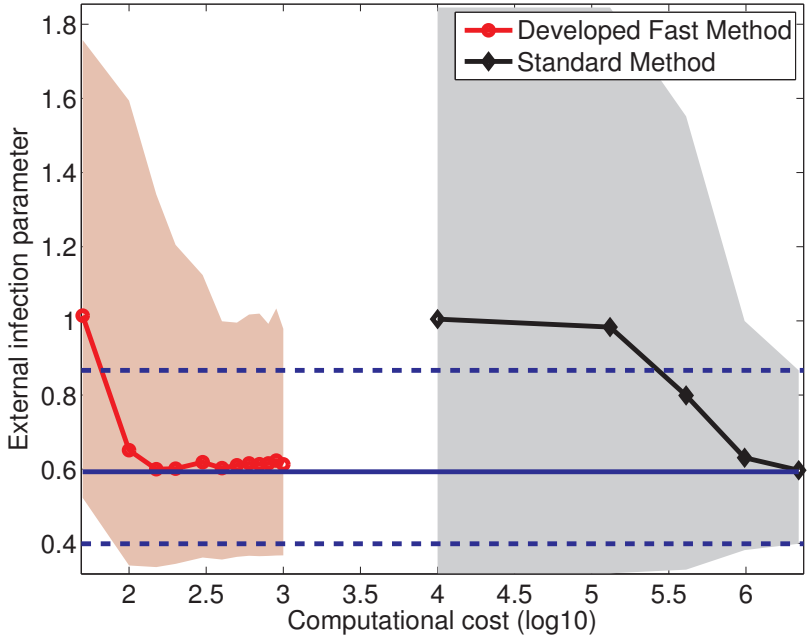


(Gutmann and Corander, 2016)

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.

Closer look at the decision making

- We piggy-backed on Bayesian optimisation to determine the parameter for which to run the simulator next.
- Advantages
  - Relatively easy, re-uses large body of work on Bayesian optimisation
  - Acquisition function is cheap to compute and does not depend on ϵ, which is often difficult to choose.
  - Some optimality results for the task of finding the minimum of $\mathbb{E}[\Delta|\boldsymbol{\theta}]$.
  - Minimising expected distance maximises a lower bound on the approximate log-likelihood. (Gutmann and Corander, 2016)
- Disadvantages
  - Acquisition function is not derived based on what we actually care most about: the posterior.
  - Does not incorporate the prior, which can lead to issues for confident mis-specified priors (Gutmann and Corander, 2016)

Acknowledgement

The following slides are based on slides kindly shared by Marko Järvenpää.

Going back to first principles

(Järvenpää et al, 2019)

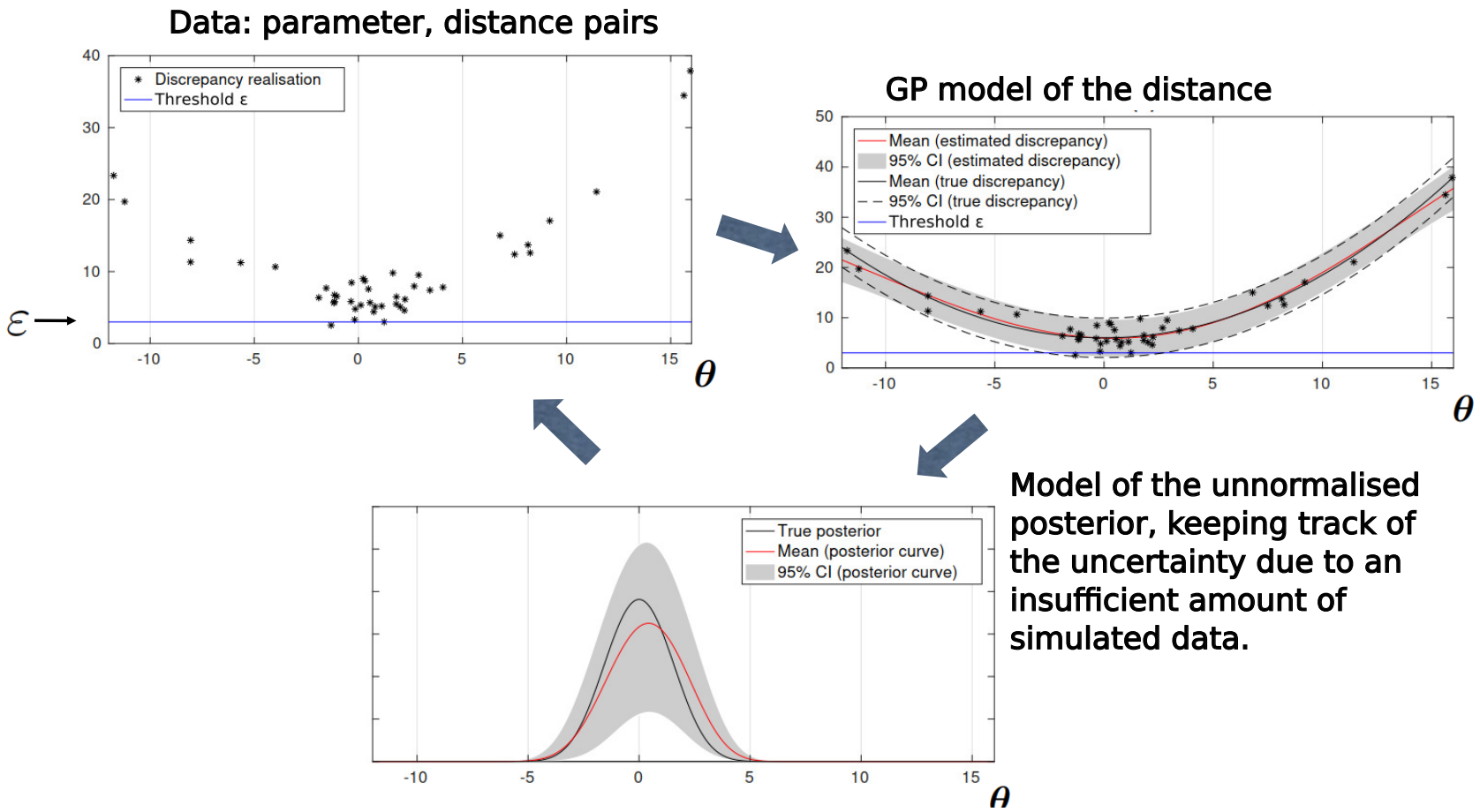
- Model $\Delta_{\boldsymbol{\theta}} = f(\boldsymbol{\theta}) + \nu$ where f is a GP and $\nu \sim \mathcal{N}(0, \sigma_n^2)$.
- If f and σ_n^2 were known, the ABC posterior π_{ABC}^f would be proportional to

$$\tilde{\pi}_{\text{ABC}}^f(\boldsymbol{\theta}) \propto p_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \mathbb{P}(f(\boldsymbol{\theta}) + \nu \leq \epsilon) \tag{18}$$

$$\propto p_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \Phi\left((\epsilon - f(\boldsymbol{\theta}))/\sigma_n\right) \tag{19}$$

  where $\Phi(\cdot)$ is the cdf of the standard Gaussian density.
- We don't know f but given acquired data $\mathcal{D}_t$, we have a distribution over it: $f \mid \mathcal{D}_t \sim \mathcal{GP}(m_t(\boldsymbol{\theta}), c_t(\boldsymbol{\theta}, \boldsymbol{\theta}'))$
- Uncertainty about f induces uncertainty about $\tilde{\pi}_{\text{ABC}}^f(\boldsymbol{\theta})$
- Choose next acquisition point $\boldsymbol{\theta}_t^*$ to reduce this uncertainty.

Illustration



Model of the unnormalised posterior, keeping track of the uncertainty due to an insufficient amount of simulated data.

Optimal selection of simulation locations

- We use Bayesian experimental design (Chaloner and Verdinelli, 1995)
- Define loss function $l(\pi^f_{\text{ABC}}, d)$ that quantifies the penalty of the decision to report d as our estimate of the ABC posterior while the true one is π^f_{ABC}.
- Compute the expected loss of the best decision

$$J_t(\boldsymbol{\theta}^*) = \mathbb{E}_{\Delta^*|\boldsymbol{\theta}^*, \mathcal{D}_t}\left(\min_d \mathbb{E}_{f|\mathcal{D}_t \cup \{(\Delta^*, \boldsymbol{\theta}^*)\}} l(\pi^f_{\text{ABC}}, d)\right). \tag{20}$$

- Depends on the "design" parameter $\boldsymbol{\theta}^*$: the parameter for which we run the simulator next
- Choose $\boldsymbol{\theta}^*$ such that above loss is minimised

$$\boldsymbol{\theta}^*_{t+1} = \operatorname*{argmin}_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}) \tag{21}$$

Expected integrated variance (EIV) criterion

(Järvenpää et al, 2019) $J_t(\boldsymbol{\theta}^*) = \mathbb{E}_{\Delta^*|\boldsymbol{\theta}^*,\mathcal{D}_t}\left(\min_d \mathbb{E}_{f|\mathcal{D}_t\cup\{(\Delta^*,\boldsymbol{\theta}^*)\}} l(\pi^f_{\text{ABC}}, d)\right)$

- Consider loss function

$$l(\pi^f_{\text{ABC}}, d) = \int_\Theta (\tilde{\pi}^f_{\text{ABC}}(\boldsymbol{\theta}) - \tilde{d}(\boldsymbol{\theta}))^2 \mathrm{d}\boldsymbol{\theta} \tag{22}$$

  between the unnormalised posteriors.

- The optimal decision (point estimate for unnormalised posterior) is

$$\tilde{d}_{opt}(\boldsymbol{\theta}) = \mathbb{E}_{f|D_t\cup\{(\Delta^*,\boldsymbol{\theta}^*)\}}(\tilde{\pi}^f_{\text{ABC}}(\boldsymbol{\theta})) \tag{23}$$

$$= p_{\boldsymbol{\theta}}(\boldsymbol{\theta})\Phi\left(\frac{\epsilon - m_t(\boldsymbol{\theta})}{\sqrt{\sigma_n^2 + s_t^2(\boldsymbol{\theta})}}\right) \tag{24}$$

  where $s_t^2(\boldsymbol{\theta})$ is the posterior variance for f.

Expected integrated variance (EIV) criterion

(Järvenpää et al, 2019) $J_t(\boldsymbol{\theta}^*) = \mathbb{E}_{\Delta^*|\boldsymbol{\theta}^*,\mathcal{D}_t}\left(\min_d \mathbb{E}_{f|\mathcal{D}_t\cup\{(\Delta^*,\boldsymbol{\theta}^*)\}} l(\pi^f_{\text{ABC}}, d)\right)$

- The minimal loss

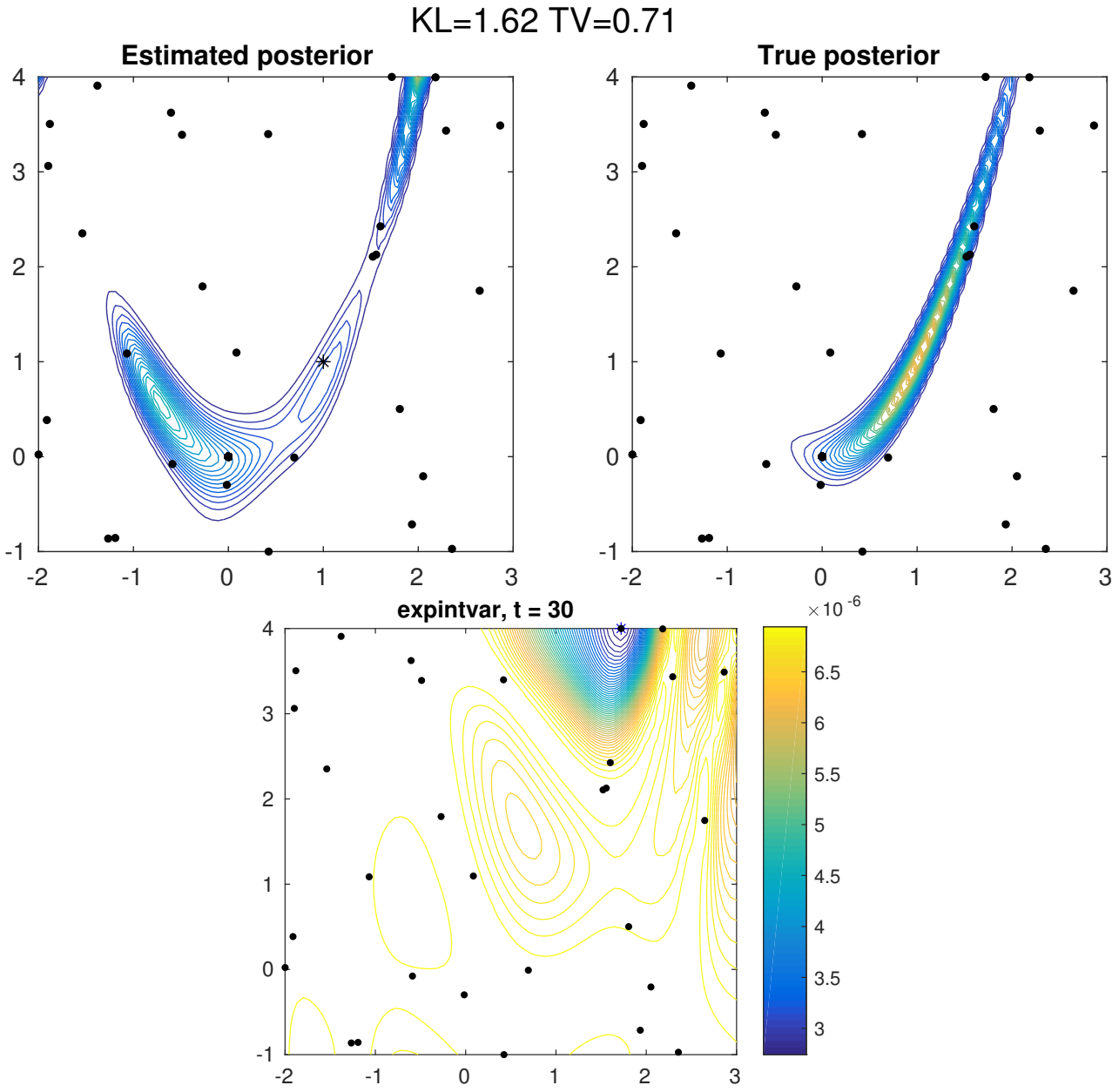
$$\min_d \mathbb{E}_{f|\mathcal{D}_t\cup\{(\Delta^*,\boldsymbol{\theta}^*)\}} l(\pi^f_{\text{ABC}}, d) \tag{25}$$

  equals the integrated variance of $\tilde{\pi}^f_{\text{ABC}}(\boldsymbol{\theta})$
- The expected loss $J_t(\boldsymbol{\theta}^*)$ thus equals the expected integrated variance.
- It can be derived in closed form

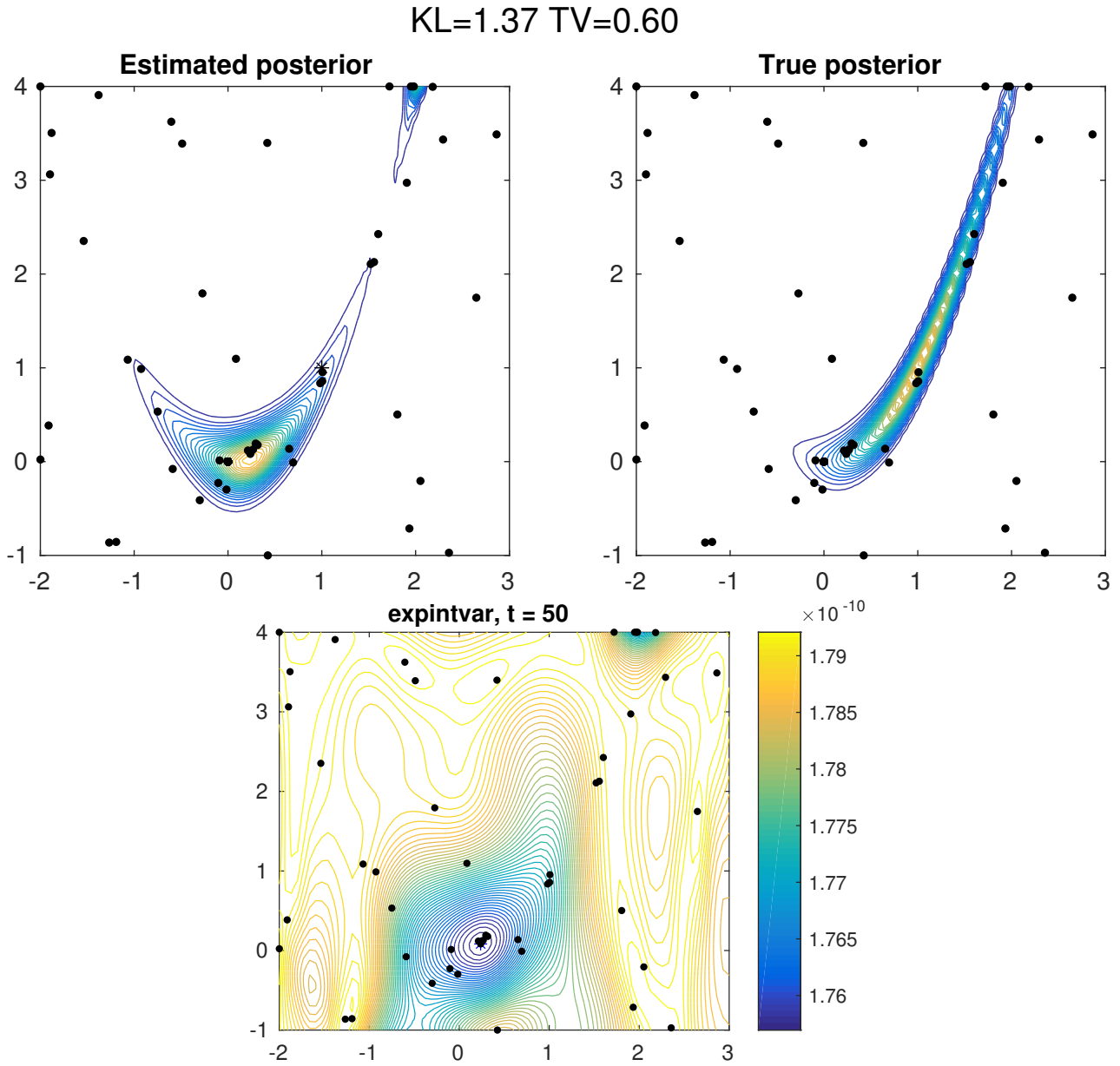
$$J_t(\boldsymbol{\theta}^*_t) = 2\int p^2_{\boldsymbol{\theta}}(\boldsymbol{\theta})\left[T\left(\frac{\epsilon - m_t(\boldsymbol{\theta})}{\sqrt{\sigma_n^2 + s_t^2(\boldsymbol{\theta})}}, \sqrt{\frac{\sigma_n^2 + s_t^2(\boldsymbol{\theta}) - \tau_t^2(\boldsymbol{\theta},\boldsymbol{\theta}^*)}{\sigma_n^2 + s_t^2(\boldsymbol{\theta}) + \tau_t^2(\boldsymbol{\theta},\boldsymbol{\theta}^*)}}\right)\right.$$
$$\left. - T\left(\frac{\epsilon - m_t(\boldsymbol{\theta})}{\sqrt{\sigma_n^2 + s_t^2(\boldsymbol{\theta})}}, \frac{\sigma_n}{\sqrt{\sigma_n^2 + 2s_t^2(\boldsymbol{\theta}))}}\right)\right] \mathrm{d}\boldsymbol{\theta}, \tag{26}$$

  where $\tau_t^2(\boldsymbol{\theta};\boldsymbol{\theta}^*) = c_t(\boldsymbol{\theta},\boldsymbol{\theta}^*)[c_t(\boldsymbol{\theta}^*,\boldsymbol{\theta}^*) + \sigma_n^2\mathbf{I}]^{-1}c_t(\boldsymbol{\theta}^*,\boldsymbol{\theta})$ and T is Owen's t-function.
- Integral approximated using importance sampling

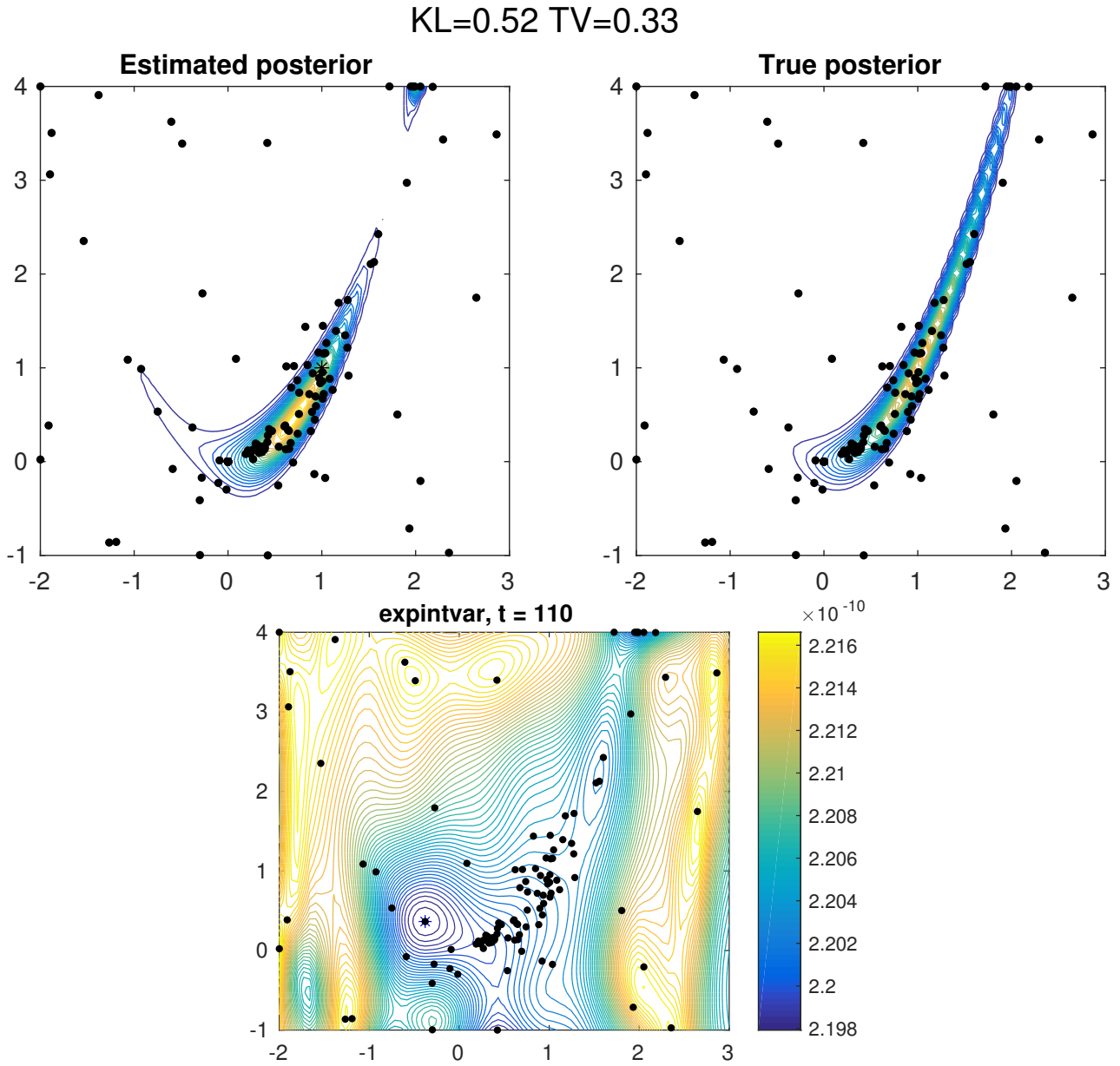
Example: Banana posterior



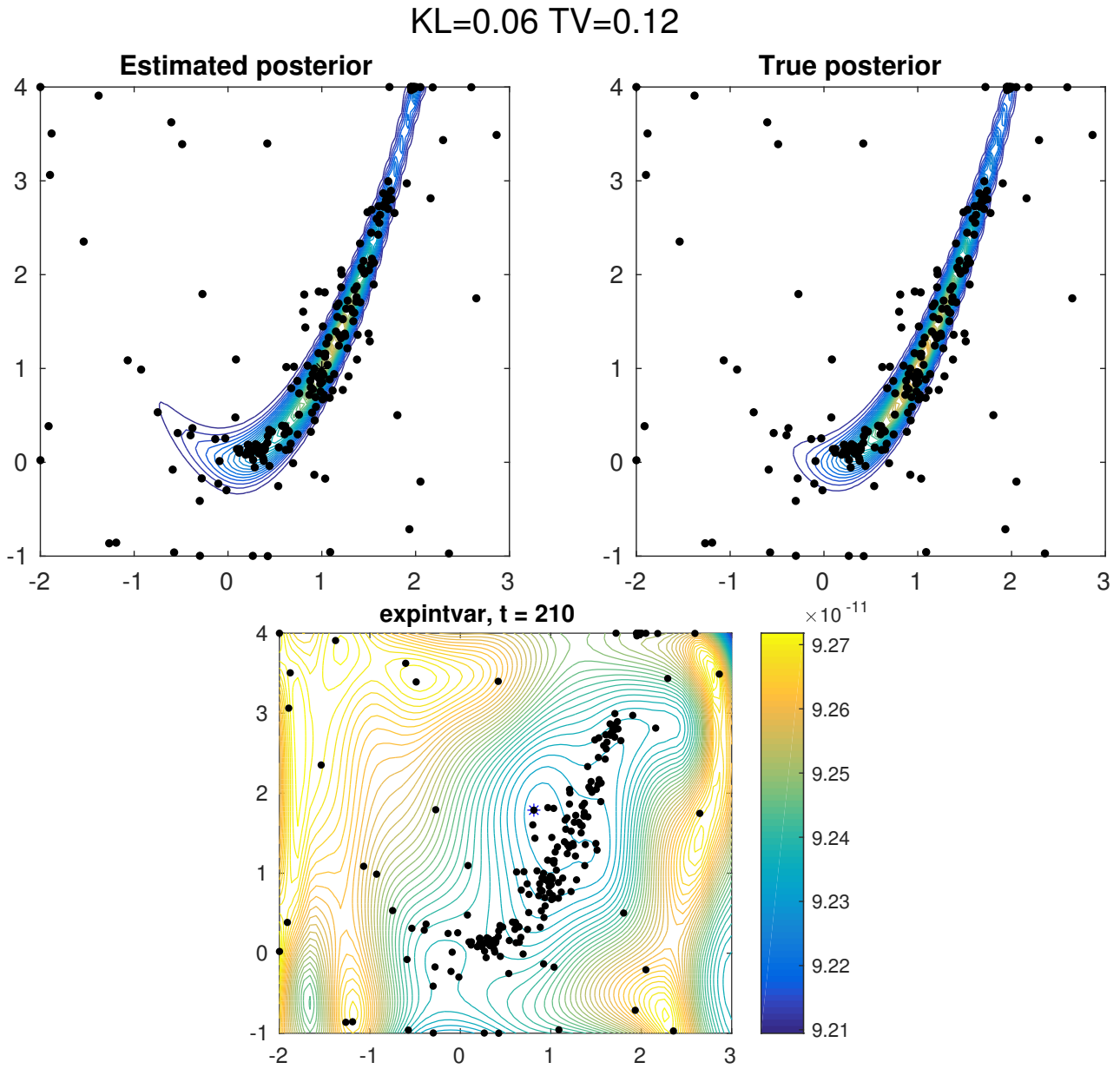
Example: Banana posterior



Example: Banana posterior



Example: Banana posterior



Comparison

(Many more results in the paper by Järvenpää et al, 2019)

- Metric: compute L_1 distance to reference posterior after each acquisition and report area under the curve
- Table below shows median over 100 experiments, normalised to performance of the expected integrated variance (expintvar).
- Other methods
  - maxvar: determine where the variance $\mathbb{V}(\tilde{\pi}_{\text{ABC}}(\boldsymbol{\theta}) \mid \mathcal{D}_{1:t})$ is largest
  - randmaxvar: stochastic version
  - LCB, EI: acquisition functions from Bayesian optimisation
  - unif: uniform sampling

	expintvar	maxvar	randmaxvar	LCB	EI	unif
Banana	**1.00**	1.23	1.09	1.08	1.67	1.47
Lotka-Volterra	**1.00**	1.37	1.10	1.15	1.85	1.62

Comparison

- Expected integrated variance yields consistently good performance.
- However, it is expensive to compute. Only worth it for expensive simulators.
- For more results, other loss functions, relationship to LCB, batch and parallel processing: see Järvenpää et al, 2019, 2020.

Summary

1. Simulator-based models
   - What they are
   - Why the likelihood function is intractable
2. Classical algorithms for approximate Bayesian computation
   - Need for approximations
   - 3 classical algorithms: rejection, regression, and SMC ABC.
3. Accelerating ABC
   - Discussed reasons why the classical algorithms so expensive
   - Framework to accelerate the inference based on (a) modelling and (b) decision making under uncertainty
   - LCB and new inference-targeted expected integrated variance (EIV) criteria

References: Part 1 and 2

- Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 2010.
- Price, Drovandi, Lee and Nott. Bayesian Synthetic Likelihood. *JCGS*, 2017.
- Thomas, Dutta, Corander, Kaski and Gutmann. Likelihood-Free Inference by Ratio Estimation. *Bayesian Analysis*, 2016, 2020
- Hermans, Begy, Loupe. Likelihood-free MCMC with Amortized Approximate Ratio Estimators. *ICML*, 2020.
- Beaumont, Zhang, Balding. Approximate Bayesian Computation in Population Genetics. *Genetics*, 2002.
- Blum and Francois. Non-linear regression models for Approximate Bayesian Computation. *Statistics and Computing*, 2010.
- Sisson, Fan, Tanaka. Sequential Monte Carlo without likelihoods. *PNAS*, 2007.
- Numminen, Cheng, Gyllenberg, Corander. Estimating the Transmission Dynamics of Streptococcus pneumoniae from Strain Prevalence Data. *Biometrics*, 2013.

References: Part 3

- Gutmann and Corander. Bayesian optimization for likelihood-free inference of simulator-based statistical models. JMLR 2016.
- Chen and Gutmann. Adaptive Gaussian Copula ABC, AISTATS 2019.
- Järvenpää, Gutmann,Vehtari and Marttinen. Gaussian process modeling in approximate Bayesian computation to estimate horizontal gene transfer in bacteria. *The Annals of Applied Statistics*, 2018.
- Srinivas, Krause, Kakade, Seeger. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting Information Theory. *IEEE Transactions on Information Theory*, 2012.
- Järvenpää, Gutmann, Vehtari, Marttinen. Efficient acquisition rules for model-based approximate Bayesian computation. *Bayesian Analysis*, 2019.
- Chaloner and Verdinelli. Bayesian Experimental Design: A Review Statistical Science. *Statistical Science*, 1995.
- Järvenpää, Vehtari, Marttinen. Batch simulations and uncertainty quantification in Gaussian process surrogate approximate Bayesian computation. UAI 2020.