Approximative Bayesian Computation (ABC) Methods

Christian P. Robert

Université Paris Dauphine and CREST-INSEE
http://www.ceremade.dauphine.fr/~xian

Joint works with M. Beaumont, J.-M. Cornuet, A. Grelaud, J.-M. Marin, F. Rodolphe, & J.-F. Tally

Athens, September 14, 2009

Outline

General purpose

Given a density π known up to a normalizing constant, and an integrable function h, compute

$$\Pi(h) = \int h(x)\pi(x)\mu(dx) = \frac{\int h(x)\tilde{\pi}(x)\mu(dx)}{\int \tilde{\pi}(x)\mu(dx)}$$

when $\int h(x)\tilde{\pi}(x)\mu(dx)$ is intractable.

Monte Carlo basics

Generate an iid sample $x_1, \ldots, x_N$ from π and estimate $\Pi(h)$ by

$$\hat{\Pi}_N^{MC}(h) = N^{-1} \sum_{i=1}^{N} h(x_i).$$

LLN: $\hat{\Pi}_N^{MC}(h) \xrightarrow{\text{as}} \Pi(h)$

If $\Pi(h^2) = \int h^2(x)\pi(x)\mu(dx) < \infty$,

CLT: $$\sqrt{N}\left(\hat{\Pi}_N^{MC}(h) - \Pi(h)\right) \overset{\mathscr{L}}{\rightsquigarrow} \mathscr{N}\left(0, \Pi\left\{[h - \Pi(h)]^2\right\}\right).$$

Monte Carlo basics

Generate an iid sample $x_1, \ldots, x_N$ from π and estimate $\Pi(h)$ by

$$\hat{\Pi}_N^{MC}(h) = N^{-1} \sum_{i=1}^{N} h(x_i).$$

LLN: $\hat{\Pi}_N^{MC}(h) \xrightarrow{\text{as}} \Pi(h)$

If $\Pi(h^2) = \int h^2(x)\pi(x)\mu(dx) < \infty$,

CLT: $$\sqrt{N}\left(\hat{\Pi}_N^{MC}(h) - \Pi(h)\right) \overset{\mathscr{L}}{\rightsquigarrow} \mathscr{N}\left(0, \Pi\left\{[h - \Pi(h)]^2\right\}\right).$$

Caveat

Often impossible or inefficient to simulate directly from Π

Importance Sampling

For Q proposal distribution such that $Q(dx) = q(x)\mu(dx)$, alternative representation

$$\Pi(h) = \int h(x)\{\pi/q\}(x)q(x)\mu(dx).$$

Importance Sampling

For Q proposal distribution such that $Q(dx) = q(x)\mu(dx)$, alternative representation

$$\Pi(h) = \int h(x)\{\pi/q\}(x)q(x)\mu(dx).$$

Principle

Generate an iid sample $x_1, \ldots, x_N \sim Q$ and estimate $\Pi(h)$ by

$$\hat{\Pi}^{IS}_{Q,N}(h) = N^{-1} \sum_{i=1}^{N} h(x_i)\{\pi/q\}(x_i).$$

Then

LLN: $\hat{\Pi}^{IS}_{Q,N}(h) \xrightarrow{\text{as}} \Pi(h)$ and if $Q((h\pi/q)^2) < \infty$,

CLT: $\sqrt{N}(\hat{\Pi}^{IS}_{Q,N}(h) - \Pi(h)) \overset{\mathscr{L}}{\rightsquigarrow} \mathscr{N}\left(0, Q\{(h\pi/q - \Pi(h))^2\}\right)$.

Then

LLN: $\hat{\Pi}^{IS}_{Q,N}(h) \xrightarrow{\text{as}} \Pi(h)$ and if $Q((h\pi/q)^2) < \infty$,

CLT: $\sqrt{N}(\hat{\Pi}^{IS}_{Q,N}(h) - \Pi(h)) \overset{\mathscr{L}}{\rightsquigarrow} \mathcal{N}\left(0, Q\{(h\pi/q - \Pi(h))^2\}\right)$.

Caveat

If normalizing constant unknown, impossible to use $\hat{\Pi}^{IS}_{Q,N}$

Generic problem in Bayesian Statistics: $\pi(\theta|x) \propto f(x|\theta)\pi(\theta)$.

Self-Normalised Importance Sampling

Self normalized version

$$\hat{\Pi}_{Q,N}^{SNIS}(h) = \left(\sum_{i=1}^{N} \{\pi/q\}(x_i) \right)^{-1} \sum_{i=1}^{N} h(x_i)\{\pi/q\}(x_i).$$

Self-Normalised Importance Sampling

Self normalized version

$$\hat{\Pi}_{Q,N}^{SNIS}(h) = \left(\sum_{i=1}^{N} \{\pi/q\}(x_i) \right)^{-1} \sum_{i=1}^{N} h(x_i)\{\pi/q\}(x_i).$$

$$LLN : \quad \hat{\Pi}_{Q,N}^{SNIS}(h) \xrightarrow{\text{as}} \Pi(h)$$

and if $\Pi((1+h^2)(\pi/q)) < \infty$,

$$CLT : \quad \sqrt{N}(\hat{\Pi}_{Q,N}^{SNIS}(h) - \Pi(h)) \overset{\mathscr{L}}{\rightsquigarrow} \mathcal{N}\left(0, \pi\left\{(\pi/q)(h - \Pi(h)\right\}^2)\right).$$

Self-Normalised Importance Sampling

Self normalized version

$$\hat{\Pi}_{Q,N}^{SNIS}(h) = \left(\sum_{i=1}^{N} \{\pi/q\}(x_i) \right)^{-1} \sum_{i=1}^{N} h(x_i)\{\pi/q\}(x_i).$$

$$LLN : \quad \hat{\Pi}_{Q,N}^{SNIS}(h) \xrightarrow{\text{as}} \Pi(h)$$

and if $\Pi((1+h^2)(\pi/q)) < \infty$,

$$CLT : \quad \sqrt{N}(\hat{\Pi}_{Q,N}^{SNIS}(h) - \Pi(h)) \overset{\mathscr{L}}{\rightsquigarrow} \mathcal{N}\left(0, \pi\left\{(\pi/q)(h - \Pi(h)\right\}^2)\right).$$

© The quality of the SNIS approximation depends on the choice of Q

Iterated importance sampling

Introduction of an algorithmic *temporal dimension* :

$$x_i^{(t)} \sim q_t(x|x_i^{(t-1)}) \qquad i = 1, \ldots, n, \quad t = 1, \ldots$$

and

$$\hat{\mathfrak{I}}_t = \frac{1}{n} \sum_{i=1}^{n} \varrho_i^{(t)} h(x_i^{(t)})$$

is still unbiased for

$$\varrho_i^{(t)} = \frac{\pi_t(x_i^{(t)})}{q_t(x_i^{(t)}|x_i^{(t-1)})}, \qquad i = 1, \ldots, n$$

PMC: Population Monte Carlo Algorithm

At time $t = 0$

Generate $(x_{i,0})_{1 \leq i \leq N} \overset{iid}{\sim} Q_0$

Set $\omega_{i,0} = \{\pi / q_0\}(x_{i,0})$

Generate $(J_{i,0})_{1 \leq i \leq N} \overset{\text{iid}}{\sim} \mathcal{M}(1, (\bar{\omega}_{i,0})_{1 \leq i \leq N})$

Set $\tilde{x}_{i,0} = x_{J_i,0}$

PMC: Population Monte Carlo Algorithm

At time $t = 0$

Generate $(x_{i,0})_{1\leq i\leq N} \overset{iid}{\sim} Q_0$
Set $\omega_{i,0} = \{\pi/q_0\}(x_{i,0})$

Generate $(J_{i,0})_{1\leq i\leq N} \overset{\text{iid}}{\sim} \mathcal{M}(1, (\bar{\omega}_{i,0})_{1\leq i\leq N})$
Set $\tilde{x}_{i,0} = x_{J_i,0}$

At time t $(t = 1, \ldots, T)$,

Generate $x_{i,t} \overset{\text{ind}}{\sim} Q_{i,t}(\tilde{x}_{i,t-1}, \cdot)$
Set $\omega_{i,t} = \{\pi(x_{i,t})/q_{i,t}(\tilde{x}_{i,t-1}, x_{i,t})\}$

Generate $(J_{i,t})_{1\leq i\leq N} \overset{\text{iid}}{\sim} \mathcal{M}(1, (\bar{\omega}_{i,t})_{1\leq i\leq N})$
Set $\tilde{x}_{i,t} = x_{J_{i,t},t}$.

[Cappé, Douc, Guillin, Marin, & CPR, 2009, Stat.& Comput.]

Notes on PMC

After T iterations of PMC, PMC estimator of $\Pi(h)$ given by

$$\bar{\Pi}_{N,T}^{PMC}(h) = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{N} \bar{\bar{\omega}}_{i,t} h(x_{i,t}).$$

Notes on PMC

After T iterations of PMC, PMC estimator of $\Pi(h)$ given by

$$\bar{\Pi}_{N,T}^{PMC}(h) = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\bar{\omega}_{i,t}h(x_{i,t}).$$

1. $\bar{\omega}_{i,t}$ means normalising over whole sequence of simulations
2. $Q_{i,t}$'s chosen arbitrarily under support constraint
3. $Q_{i,t}$'s may depend on whole sequence of simulations

Improving quality

The efficiency of the SNIS approximation depends on the choice of Q, ranging from optimal

$$q(x) \propto |h(x) - \Pi(h)|\pi(x)$$

to useless

$$\text{var}\,\hat{\Pi}_{Q,N}^{SNIS}(h) = +\infty$$

Improving quality

The efficiency of the SNIS approximation depends on the choice of Q, ranging from optimal

$$q(x) \propto |h(x) - \Pi(h)|\pi(x)$$

to useless

$$\text{var}\,\hat{\Pi}^{SNIS}_{Q,N}(h) = +\infty$$

Example (PMC=adaptive importance sampling)

Population Monte Carlo is producing a sequence of proposals Q_t aiming at improving efficiency

$$\text{Kull}(\pi, q_t) \leq \text{Kull}(\pi, q_{t-1}) \quad \text{or} \quad \text{var}\,\hat{\Pi}^{SNIS}_{Q_t,\infty}(h) \leq \text{var}\,\hat{\Pi}^{SNIS}_{Q_{t-1},\infty}(h)$$

[Cappé, Douc, Guillin, Marin, Robert, 04, 07a, 07b, 08]

Multiple Importance Sampling

Reycling: given several proposals $Q_1, \ldots, Q_T$, for $1 \leq t \leq T$ generate an iid sample

$$x_1^t, \ldots, x_N^t \sim Q_t$$

and estimate $\Pi(h)$ by

$$\hat{\Pi}_{Q,N}^{MIS}(h) = T^{-1} \sum_{t=1}^{T} N^{-1} \sum_{i=1}^{N} h(x_i^t)\omega_i^t$$

where

$$\omega_i^t \neq \frac{\pi(x_i^t)}{q_t(x_i^t)}$$

correct...

Multiple Importance Sampling

Reycling: given several proposals $Q_1, \ldots, Q_T$, for $1 \leq t \leq T$ generate an iid sample

$$x_1^t, \ldots, x_N^t \sim Q_t$$

and estimate $\Pi(h)$ by

$$\hat{\Pi}_{Q,N}^{MIS}(h) = T^{-1} \sum_{t=1}^{T} N^{-1} \sum_{i=1}^{N} h(x_i^t)\omega_i^t$$

where

$$\omega_i^t = \frac{\pi(x_i^t)}{T^{-1} \sum_{\ell=1}^{T} q_\ell(x_i^t)}$$

still correct!

Mixture representation

Deterministic mixture correction of the weights proposed by Owen and Zhou (JASA, 2000)

- The corresponding estimator is still unbiased [if not self-normalised]
- All particles are on the same weighting scale rather than their own
- Large variance proposals Q_t do not take over
- Variance reduction thanks to weight stabilization & recycling
- [K.o.] removes the randomness in the component choice [=Rao-Blackwellisation]

Global adaptation

Global Adaptation

At iteration $t = 1, \cdots, T$,

1. For $1 \leq i \leq N_1$, generate $x_i^t \sim \mathcal{T}_3(\hat{\mu}^{t-1}, \hat{\Sigma}^{t-1})$
2. Calculate the mixture importance weight of particle x_i^t

$$\omega_i^t = \pi\left(x_i^t\right) / \delta_i^t$$

where

$$\delta_i^t = \sum_{l=0}^{t-1} q_{\mathcal{T}(3)}\left(x_i^t; \hat{\mu}^l, \hat{\Sigma}^l\right)$$

Backward reweighting

3. If $t \geq 2$, actualize the weights of all past particles, x_i^l $1 \leq l \leq t-1$

$$\omega_i^l = \pi\left(x_i^t\right) / \delta_i^l$$

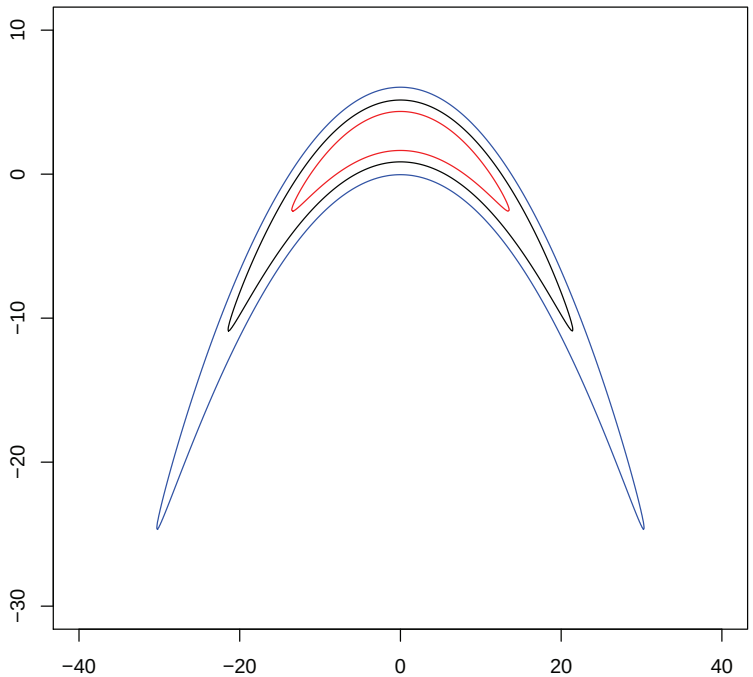
where

$$\delta_i^l = \delta_i^l + q_{\mathcal{T}(3)}\left(x_i^l; \hat{\mu}^{t-1}, \hat{\Sigma}^{t-1}\right)$$

4. Compute IS estimates of target mean and variance $\hat{\mu}^t$ and $\hat{\Sigma}^t$, where

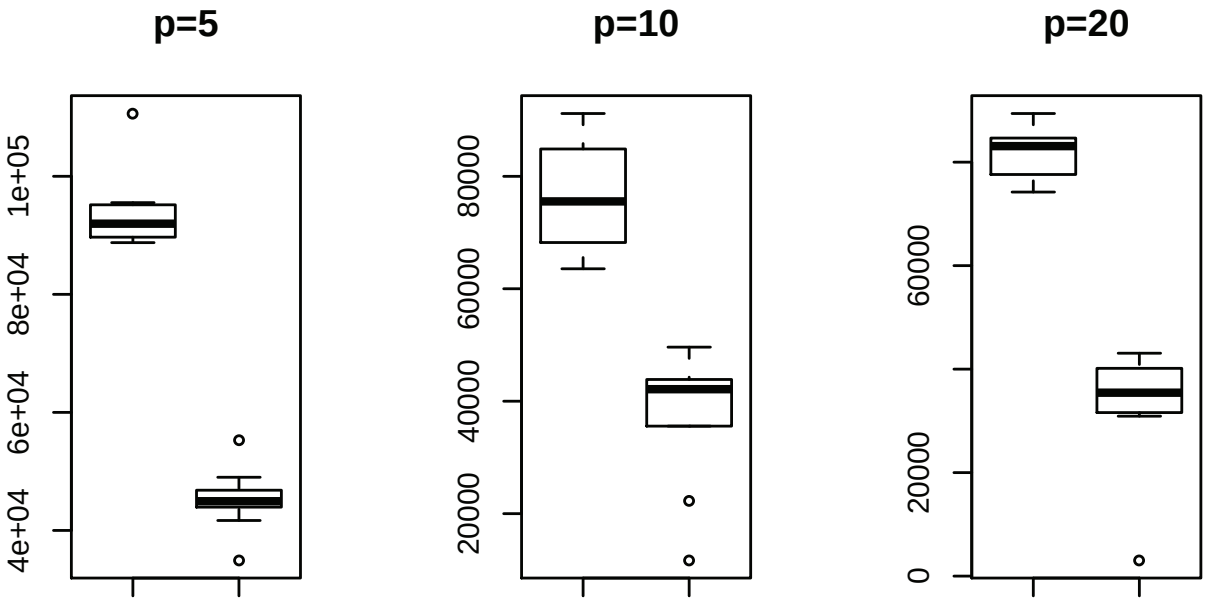
$$\hat{\mu}_j^t = \sum_{l=1}^{t}\sum_{i=1}^{N_1} \omega_i^l (x_j)_i^l \Big/ \sum_{l=1}^{t}\sum_{i=1}^{N_1} \omega_i^l \ \ldots$$

A toy example



Banana shape benchmark: marginal distribution of (x_1, x_2) for the parameters $\sigma_1^2 = 100$ and $b = 0.03$. Contours represent 60% (red), 90% (black) and 99.9% (blue) confidence regions in the marginal space.

A toy example



Banana shape example: boxplots of 10 replicate ESSs for the AMIS scheme (left) and the NOT-AMIS scheme (right) for $p = 5, 10, 20$.

Convergence of the AMIS estimator

Difficulty in establishing the convergence because of the backward structure: the weight of x_i^t at stage T depends on future as well as past x_j^{ℓ}'...
Regular Population Monte Carlo argument does not work for T asymptotics...

Convergence of the AMIS estimator

Difficulty in establishing the convergence because of the backward structure: the weight of x_i^t at stage T depends on future as well as past x_j^{ℓ}'...
Regular Population Monte Carlo argument does not work for T asymptotics...

[© Amiss estimator?!]

A modified version of the algorithm

Only consider AMIS with $p = 1$, $N = 1$ and $h(x) = x$.

Set the variances of the t distributions to be equal to 1 after rescaling, i.e. no learning process on the covariance matrix

Algorithmic setup

Our simplified algorithm then runs as follows:

$$x_0 \sim q_0(\cdot)\,,\; x_1 \sim T_3(u_1(x_0),1) \quad \text{where} \quad u_1(x_0) = \frac{\pi(x_0)x_0}{q_0(x_0)} = \hat{\mu}^0\,,$$

$$x_2 \sim T_3(u_2(x_{0:1}),1) \quad \text{where} \quad u_2(x_{0:1}) =$$

$$\frac{\pi(x_0)x_0}{q_0(x_0)+t_3(x_0;u_1(x_0),1)} + \frac{\pi(x_1)x_1}{q_0(x_1)+t_3(x_1;u_1(x_0),1)} = \hat{\mu}^1\,,$$

Algorithmic setup (2)

$$x_t \sim T_3(u_t(x_{0:t-1}), 1)$$

where $$u_t(x_{0:t-1}) = \sum_{k=0}^{t-1} \frac{\pi(x_k)x_k}{q_0(x_k) + \sum_{i=1}^{t-1} t_3(x_k; u_i(x_{0:i-1}), 1)}, \ldots$$

Stumbling block

Establishing that

$$\hat{\mu}^T = \sum_{k=0}^{T} \frac{\pi(x_k)x_k}{q_0(x_k) + \sum_{i=1}^{T} t_3(x_k; u_i(x_{0:i-1}), 1)} \xrightarrow[T\to\infty]{L_2} \mu = \int x\pi(x)dx\,.$$

proves to be surprisingly difficult (note that $\mathbb{E}(\hat{\mu}^T) \neq \mu$)

- Impossible to use PMC convergence theorems on triangular arrays of random variables.

Unbiased version of the estimator

Modified version of previous algorithm with two sequences:

$$x_0 \sim q_0(\cdot) \quad \text{and} \quad \tilde{x}_0 \sim q_0(\cdot)\,,$$

$$x_1 \sim T_3(u_1(\tilde{x}_0), 1) \quad \text{and} \quad \tilde{x}_1 \sim T_3(u_1(\tilde{x}_0), 1)$$
$$\text{where } u_1(\tilde{x}_0) = \frac{\pi(\tilde{x}_0)\tilde{x}_0}{q_0(\tilde{x}_0)} = \hat{\mu}^0\,,$$

$$x_2 \sim T_3(u_2(\tilde{x}_{0:1}), 1) \quad \text{and} \quad \tilde{x}_2 \sim T_3(u_2(\tilde{x}_{0:1}), 1) \text{ where } u_2(\tilde{x}_{0:1}) = \frac{\pi(\tilde{x}_0)\tilde{x}_0}{q_0(\tilde{x}_0) + t_3(\tilde{x}_0; u_1(\tilde{x}_0), 1)} + \frac{\pi(\tilde{x}_1)\tilde{x}_1}{q_0(\tilde{x}_1) + t_3(\tilde{x}_1; u_1(\tilde{x}_0), 1)} = \hat{\mu}^1\,,$$

Unbiased version of the estimator (2)

$$x_t \sim T_3(u_t(x_{0:t-1}), 1) \quad \text{and} \quad \tilde{x}_t \sim T_3(u_t(\tilde{x}_{0:t-1}), 1)$$

where $u_t(\tilde{x}_{0:t-1}) = \sum_{k=0}^{t-1} \frac{\pi(\tilde{x}_k)\tilde{x}_k}{q_0(\tilde{x}_k) + \sum_{i=1}^{t-1} t_3(\tilde{x}_k; u_i(\tilde{x}_{0:i-1}), 1)}, \ldots$

Let

$$\hat{\mu}_U^T = \sum_{k=0}^{T} \frac{\pi(x_k)x_k}{q_0(x_k) + \sum_{i=1}^{T} t_3(x_k; u_i(\tilde{x}_{0:i-1}), 1)}.$$

My questions

Clearly, we have

$$\mathbb{E}(\hat{\mu}_U^T) = \mu$$

and under mild conditions we should have

$$\hat{\mu}_U^T \xrightarrow[T\to\infty]{L_2} \mu$$

My questions

Clearly, we have

$$\mathbb{E}(\hat{\mu}_U^T) = \mu$$

and under mild conditions we should have

$$\hat{\mu}_U^T \underset{T\to\infty}{\xrightarrow{L_2}} \mu$$

Except for the compact case, i.e. when supp(π) is compact, this also proves impossible to establish...
The only indication we have is that var$(\hat{\mu}_U^T)$ is decreasing at each iteration

The ABC method

Bayesian setting: target is $\pi(\theta)f(x|\theta)$

The ABC method

Bayesian setting: target is $\pi(\theta)f(x|\theta)$
When likelihood $f(x|\theta)$ not in closed form, likelihood-free rejection technique:

The ABC method

Bayesian setting: target is $\pi(\theta)f(x|\theta)$
When likelihood $f(x|\theta)$ not in closed form, likelihood-free rejection technique:

ABC algorithm

For an observation $y \sim f(y|\theta)$, under the prior $\pi(\theta)$, keep *jointly* simulating

$$\theta' \sim \pi(\theta)\,, x \sim f(x|\theta')\,,$$

until the auxiliary variable x is equal to the observed value, $x = y$.

[Pritchard et al., 1999]

A as approximative

When y is a continuous random variable, equality $x = y$ is replaced with a tolerance condition,

$$\varrho(x, y) \leq \epsilon$$

where ϱ is a distance between summary statistics

A as approximative

When y is a continuous random variable, equality $x = y$ is replaced with a tolerance condition,

$$\varrho(x, y) \leq \epsilon$$

where ϱ is a distance between summary statistics
Output distributed from

$$\pi(\theta)\, P_\theta\{\varrho(x, y) < \epsilon\} \propto \pi(\theta|\varrho(x, y) < \epsilon)$$

ABC improvements

Simulating from the prior is often poor in efficiency

ABC improvements

Simulating from the prior is often poor in efficiency
Either modify the proposal distribution on θ to increase the density of x's within the vicinity of y...

[Marjoram et al, 2003; Bortot et al., 2007, Sisson et al., 2007]

ABC improvements

Simulating from the prior is often poor in efficiency
Either modify the proposal distribution on θ to increase the density of x's within the vicinity of y...

[Marjoram et al, 2003; Bortot et al., 2007, Sisson et al., 2007]

...or by viewing the problem as a conditional density estimation and by developing techniques to allow for larger ϵ

[Beaumont et al., 2002]

ABC improvements

Simulating from the prior is often poor in efficiency
Either modify the proposal distribution on θ to increase the density of x's within the vicinity of y...

[Marjoram et al, 2003; Bortot et al., 2007, Sisson et al., 2007]

...or by viewing the problem as a conditional density estimation and by developing techniques to allow for larger ϵ

[Beaumont et al., 2002]

...or even by including ϵ in the inferential framework [ABC_μ]

[Ratmann et al., 2009]

ABC-MCMC

Markov chain $(\theta^{(t)})$ created via the transition function

$$\theta^{(t+1)} = \begin{cases} \theta' \sim K(\theta'|\theta^{(t)}) & \text{if } x \sim f(x|\theta') \text{ is such that } x = y \\ & \text{and } u \sim \mathcal{U}(0,1) \leq \dfrac{\pi(\theta')K(\theta^{(t)}|\theta')}{\pi(\theta^{(t)})K(\theta'|\theta^{(t)})}\,, \\ \theta^{(t)} & \text{otherwise,} \end{cases}$$

ABC-MCMC

Markov chain $(\theta^{(t)})$ created via the transition function

$$\theta^{(t+1)} = \begin{cases} \theta' \sim K(\theta'|\theta^{(t)}) & \text{if } x \sim f(x|\theta') \text{ is such that } x = y \\ & \text{and } u \sim \mathcal{U}(0,1) \leq \frac{\pi(\theta')K(\theta^{(t)}|\theta')}{\pi(\theta^{(t)})K(\theta'|\theta^{(t)})}\,, \\ \theta^{(t)} & \text{otherwise,} \end{cases}$$

has the posterior $\pi(\theta|y)$ as stationary distribution

[Marjoram et al, 2003]

ABC$_\mu$

[Ratmann, Andrieu, Wiuf and Richardson, 2009, PNAS]

Use of a joint density

$$f(\theta, \epsilon|x_0) \propto \xi(\epsilon|x_0, \theta) \times \pi_\theta(\theta) \times \pi_\epsilon(\epsilon)$$

where x_0 is the data, and $\xi(\epsilon|x_0, \theta)$ is the prior predictive density of $\rho(S(x), S(x_0))$ given θ and x_0 when $x \sim f(x|\theta)$
Replacement of $\xi(\epsilon|x_0, \theta)$ with a non-parametric kernel approximation.

Questions about ABC$_\mu$

For each model under comparison, marginal posterior on ϵ used to assess the fit of the model (HPD includes 0 or not).

Questions about ABC$_\mu$

For each model under comparison, marginal posterior on ϵ used to assess the fit of the model (HPD includes 0 or not).

- Is the data informative about ϵ? [Identifiability]
- How is the prior $\pi(\epsilon)$ impacting the comparison?
- How is using both $\xi(\epsilon|x_0,\theta)$ and $\pi_\epsilon(\epsilon)$ compatible with a standard probability model?
- Where is there a penalisation for complexity in the model comparison?

ABC-PRC

Another sequential version producing a sequence of Markov transition kernels K_t and of samples $(\theta_1^{(t)}, \ldots, \theta_N^{(t)})$ $(1 \leq t \leq T)$

ABC-PRC

Another sequential version producing a sequence of Markov transition kernels K_t and of samples $(\theta_1^{(t)}, \ldots, \theta_N^{(t)})$ $(1 \leq t \leq T)$

ABC-PRC Algorithm

1. Pick a $\theta^\star$ is selected at random among the previous $\theta_i^{(t-1)}$'s with probabilities $\omega_i^{(t-1)}$ $(1 \leq i \leq N)$.
2. Generate
$$\theta_i^{(t)} \sim K_t(\theta|\theta^\star)\,, x \sim f(x|\theta_i^{(t)})\,,$$
3. Check that $\varrho(x, y) < \epsilon$, otherwise start again.

[Sisson et al., 2007]

ABC-PRC weight

Probability $\omega_i^{(t)}$ computed as

$$\omega_i^{(t)} \propto \pi(\theta_i^{(t)}) L_{t-1}(\theta^\star|\theta_i^{(t)}) \{\pi(\theta^\star) K_t(\theta_i^{(t)}|\theta^\star)\}^{-1},$$

where L_{t-1} is an arbitrary transition kernel.

ABC-PRC weight

Probability $\omega_i^{(t)}$ computed as

$$\omega_i^{(t)} \propto \pi(\theta_i^{(t)}) L_{t-1}(\theta^\star|\theta_i^{(t)}) \{\pi(\theta^\star) K_t(\theta_i^{(t)}|\theta^\star)\}^{-1} ,$$

where L_{t-1} is an arbitrary transition kernel.
In case

$$L_{t-1}(\theta'|\theta) = K_t(\theta|\theta') ,$$

all weights are equal under a uniform prior.

ABC-PRC weight

Probability $\omega_i^{(t)}$ computed as

$$\omega_i^{(t)} \propto \pi(\theta_i^{(t)}) L_{t-1}(\theta^\star|\theta_i^{(t)}) \{\pi(\theta^\star) K_t(\theta_i^{(t)}|\theta^\star)\}^{-1} ,$$

where L_{t-1} is an arbitrary transition kernel.
In case

$$L_{t-1}(\theta'|\theta) = K_t(\theta|\theta') ,$$

all weights are equal under a uniform prior.
Inspired from Del Moral et al. (2006), who use backward kernels L_{t-1} in SMC to achieve unbiasedness

ABC-PRC bias

Lack of unbiasedness of the method

ABC-PRC bias

Lack of unbiasedness of the method

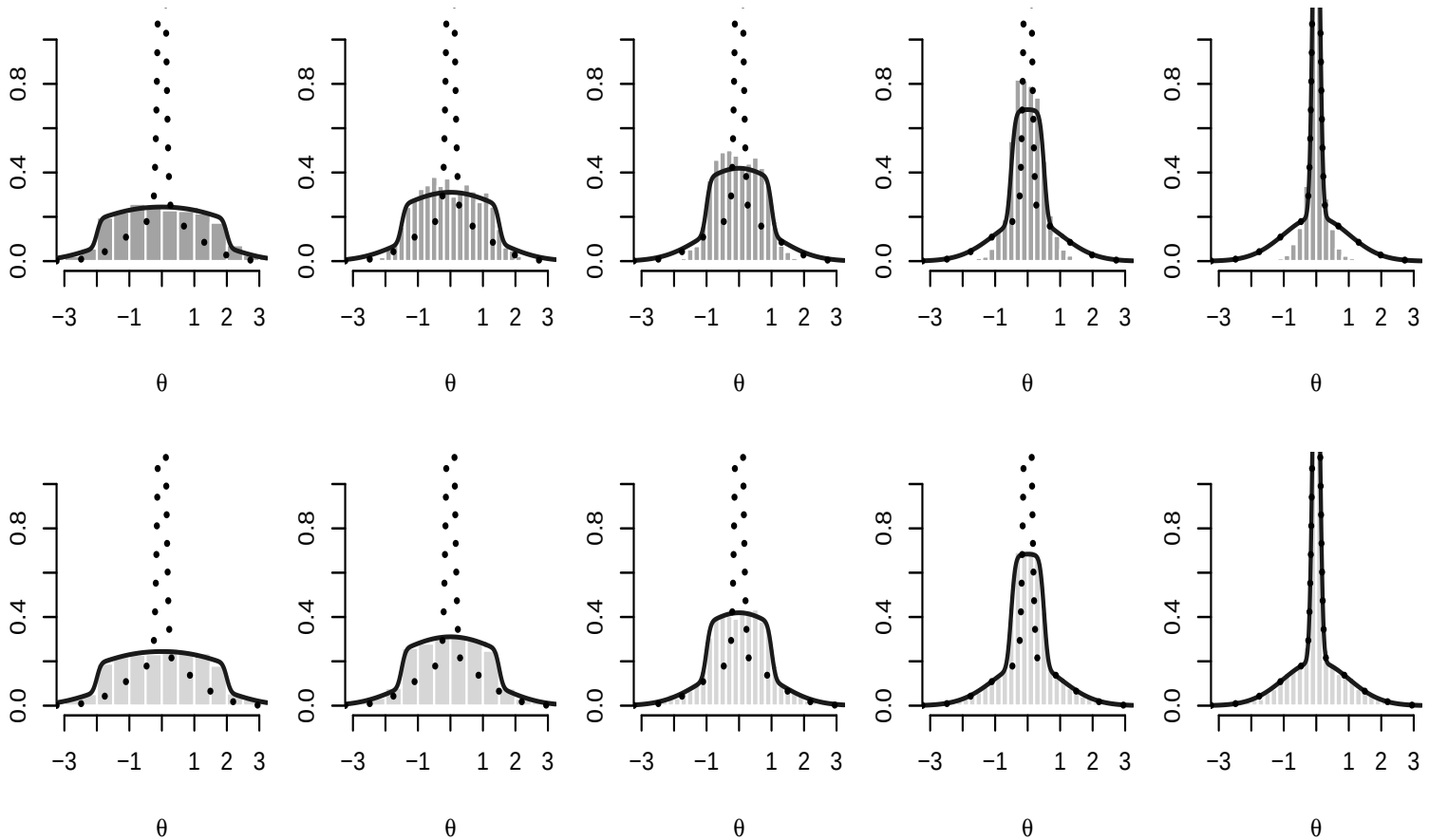
Joint density of the accepted pair $(\theta^{(t-1)}, \theta^{(t)})$ proportional to

$$\pi(\theta^{(t-1)}|y) K_t(\theta^{(t)}|\theta^{(t-1)}) f(y|\theta^{(t)}) \,,$$

For an arbitrary function $h(\theta)$, $E[\omega_t h(\theta^{(t)})]$ proportional to

$$\begin{aligned}
&\iint h(\theta^{(t)}) \, \frac{\pi(\theta^{(t)}) L_{t-1}(\theta^{(t-1)}|\theta^{(t)})}{\pi(\theta^{(t-1)}) K_t(\theta^{(t)}|\theta^{(t-1)})} \, \pi(\theta^{(t-1)}|y) K_t(\theta^{(t)}|\theta^{(t-1)}) f(y|\theta^{(t)}) \mathrm{d}\theta^{(t-1)} \mathrm{d}\theta^{(t)} \\
&\quad \propto \iint h(\theta^{(t)}) \, \frac{\pi(\theta^{(t)}) L_{t-1}(\theta^{(t-1)}|\theta^{(t)})}{\pi(\theta^{(t-1)}) K_t(\theta^{(t)}|\theta^{(t-1)})} \, \pi(\theta^{(t-1)}) f(y|\theta^{(t-1)}) \\
&\qquad \times K_t(\theta^{(t)}|\theta^{(t-1)}) f(y|\theta^{(t)}) \mathrm{d}\theta^{(t-1)} \mathrm{d}\theta^{(t)} \\
&\quad \propto \int h(\theta^{(t)}) \pi(\theta^{(t)}|y) \left\{ \int L_{t-1}(\theta^{(t-1)}|\theta^{(t)}) f(y|\theta^{(t-1)}) \mathrm{d}\theta^{(t-1)} \right\} \mathrm{d}\theta^{(t)} \,.
\end{aligned}$$

A mixture example



Comparison of $\tau = 0.15$ and $\tau = 1/0.15$ in K_t

A PMC version

Use of the same kernel idea as ABC-PRC but with IS correction
Generate a sample at iteration t by

$$\hat{\pi}_t(\theta^{(t)}) \propto \sum_{j=1}^{N} \omega_j^{(t-1)} K_t(\theta^{(t)}|\theta_j^{(t-1)})$$

modulo acceptance of the associated x_t, and use an importance weight associated with an accepted simulation $\theta_i^{(t)}$

$$\omega_i^{(t)} \propto \pi(\theta_i^{(t)}) / \hat{\pi}_t(\theta_i^{(t)}) \,.$$

© **Still likelihood free**

[Beaumont et al., 2008, arXiv:0805.2256]

The ABC-PMC algorithm

Given a decreasing sequence of approximation levels $\epsilon_1 \geq \ldots \geq \epsilon_T$,

1. At iteration $t = 1$,

   For $i = 1, ..., N$

   Simulate $\theta_i^{(1)} \sim \pi(\theta)$ and $x \sim f(x|\theta_i^{(1)})$ until $\varrho(x, y) < \epsilon_1$
   Set $\omega_i^{(1)} = 1/N$

   Take τ^2 as twice the empirical variance of the $\theta_i^{(1)}$'s

2. At iteration $2 \leq t \leq T$,

   For $i = 1, ..., N$, repeat

   Pick $\theta_i^\star$ from the $\theta_j^{(t-1)}$'s with probabilities $\omega_j^{(t-1)}$
   generate $\theta_i^{(t)}|\theta_i^\star \sim \mathcal{N}(\theta_i^\star, \sigma_t^2)$ and $x \sim f(x|\theta_i^{(t)})$

   until $\varrho(x, y) < \epsilon_t$
   Set $\omega_i^{(t)} \propto \pi(\theta_i^{(t)}) / \sum_{j=1}^N \omega_j^{(t-1)} \varphi\left(\sigma_t^{-1}\left\{\theta_i^{(t)} - \theta_j^{(t-1)})\right\}\right)$

   Take τ_{t+1}^2 as twice the weighted empirical variance of the $\theta_i^{(t)}$'s

ABC-SMC

[Del Moral, Doucet & Jasra, 2009]

True derivation of an SMC-ABC algorithm
Use of a kernel K_n associated with target π_{ϵ_n} and derivation of the backward kernel

$$L_{n-1}(z, z') = \frac{\pi_{\epsilon_n}(z')K_n(z', z)}{\pi_n(z)}$$

Update of the weights

$$w_{in} \propto_{i(n-1)} \frac{\sum_{m=1}^{M} \mathbb{A}_{\epsilon_\ltimes}(x_{in}^m}{\sum_{m=1}^{M} \mathbb{A}_{\epsilon_\ltimes -\nVdash}(x_{i(n-1)}^m}$$

when $x_{in}^m \sim K(x_{i(n-1)}, \cdot)$

A mixture example (0)

Toy model of Sisson et al. (2007): if

$$\theta \sim \mathcal{U}(-10, 10)\,, \quad x|\theta \sim 0.5\,\mathcal{N}(\theta, 1) + 0.5\,\mathcal{N}(\theta, 1/100)\,,$$

then the posterior distribution associated with $y = 0$ is the normal mixture

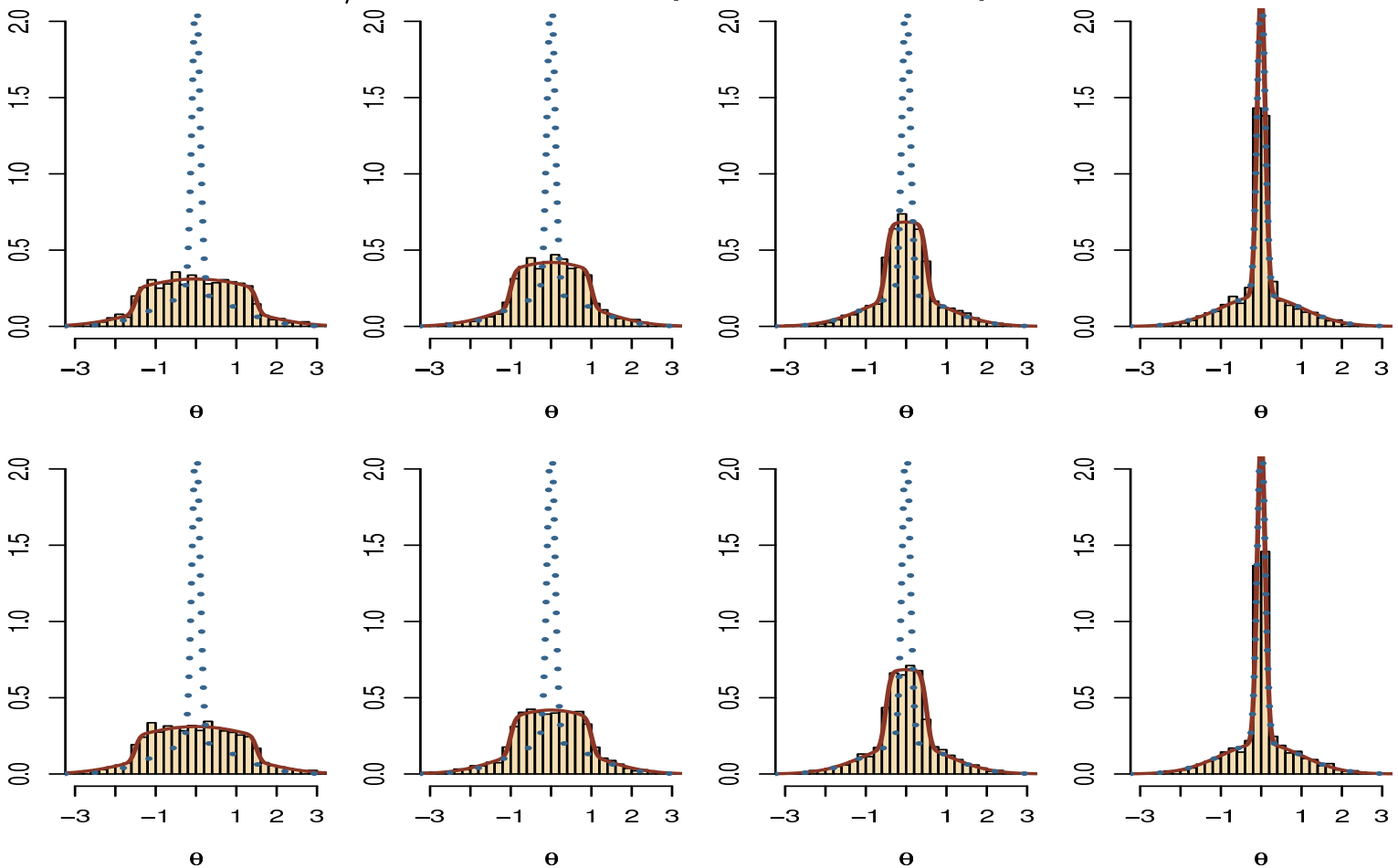
$$\theta|y = 0 \sim 0.5\,\mathcal{N}(0, 1) + 0.5\,\mathcal{N}(0, 1/100)$$

restricted to $[-10, 10]$.
Furthermore, true target available as

$$\pi(\theta||x| < \epsilon) \propto \Phi(\epsilon - \theta) - \Phi(-\epsilon - \theta) + \Phi(10(\epsilon - \theta)) - \Phi(-10(\epsilon + \theta))\,.$$

A mixture example (2)

Recovery of the target, whether using a fixed standard deviation of $\tau = 0.15$ or $\tau = 1/0.15$, or a sequence of adaptive τ_t's.



ABC for model choice

Gibbs random fields

Gibbs distribution

The rv $\mathbf{y} = (y_1, \ldots, y_n)$ is a **Gibbs random field** associated with the graph $\mathfrak{G}$ if

$$f(\mathbf{y}) = \frac{1}{\mathfrak{Z}} \exp\left\{ -\sum_{c \in \mathscr{C}} V_c(\mathbf{y}_c) \right\},$$

where $\mathfrak{Z}$ is the normalising constant, $\mathscr{C}$ is the set of cliques of $\mathfrak{G}$ and V_c is any function also called **potential**
$U(\mathbf{y}) = \sum_{c \in \mathscr{C}} V_c(\mathbf{y}_c)$ is the **energy function**

Gibbs random fields

Gibbs distribution

The rv $\mathbf{y} = (y_1, \ldots, y_n)$ is a **Gibbs random field** associated with the graph $\mathfrak{G}$ if

$$f(\mathbf{y}) = \frac{1}{\mathfrak{Z}} \exp\left\{ -\sum_{c \in \mathscr{C}} V_c(\mathbf{y}_c) \right\},$$

where $\mathfrak{Z}$ is the normalising constant, $\mathscr{C}$ is the set of cliques of $\mathfrak{G}$ and V_c is any function also called **potential**
$U(\mathbf{y}) = \sum_{c \in \mathscr{C}} V_c(\mathbf{y}_c)$ is the **energy function**

© **$\mathfrak{Z}$ is usually unavailable in closed form**

Potts model

Potts model

$V_c(\mathbf{y})$ is of the form

$$V_c(\mathbf{y}) = \theta S(\mathbf{y}) = \theta \sum_{l \sim i} \delta_{y_l = y_i}$$

where $l \sim i$ denotes a neighbourhood structure

Potts model

Potts model

$V_c(\mathbf{y})$ is of the form

$$V_c(\mathbf{y}) = \theta S(\mathbf{y}) = \theta \sum_{l \sim i} \delta_{y_l = y_i}$$

where $l \sim i$ denotes a neighbourhood structure

In most realistic settings, summation

$$Z_\theta = \sum_{\mathbf{x} \in \mathcal{X}} \exp\{\theta^{\mathsf{T}} S(\mathbf{x})\}$$

involves too many terms to be manageable and numerical approximations cannot always be trusted

[Cucala, Marin, CPR & Titterington, 2009]

Bayesian Model Choice

Comparing a model with potential S_0 taking values in $\mathbb{R}^{p_0}$ versus a model with potential S_1 taking values in $\mathbb{R}^{p_1}$ can be done through the Bayes factor corresponding to the priors π_0 and π_1 on each parameter space

$$\mathfrak{B}_{m_0/m_1}(\mathbf{x}) = \frac{\int \exp\{\theta_0^{\mathsf{T}} S_0(\mathbf{x})\}/Z_{\theta_0,0}\pi_0(\mathrm{d}\theta_0)}{\int \exp\{\theta_1^{\mathsf{T}} S_1(\mathbf{x})\}/Z_{\theta_1,1}\pi_1(\mathrm{d}\theta_1)}$$

Bayesian Model Choice

Comparing a model with potential S_0 taking values in $\mathbb{R}^{p_0}$ versus a model with potential S_1 taking values in $\mathbb{R}^{p_1}$ can be done through the Bayes factor corresponding to the priors π_0 and π_1 on each parameter space

$$\mathfrak{B}_{m_0/m_1}(\mathbf{x}) = \frac{\int \exp\{\theta_0^{\mathsf{T}} S_0(\mathbf{x})\}/Z_{\theta_0,0}\pi_0(\mathrm{d}\theta_0)}{\int \exp\{\theta_1^{\mathsf{T}} S_1(\mathbf{x})\}/Z_{\theta_1,1}\pi_1(\mathrm{d}\theta_1)}$$

Use of Jeffreys' scale to select most appropriate model

Neighbourhood relations

Choice to be made between M neighbourhood relations

$$i \overset{m}{\sim} i' \qquad (0 \leq m \leq M-1)$$

with

$$S_m(\mathbf{x}) = \sum_{i \overset{m}{\sim} i'} \mathbb{I}_{\{x_i = x_{i'}\}}$$

driven by the posterior probabilities of the models.

Model index

Formalisation via a model index $\mathcal{M}$ that appears as a new parameter with prior distribution $\pi(\mathcal{M} = m)$ and $\pi(\theta|\mathcal{M} = m) = \pi_m(\theta_m)$

Model index

Formalisation via a model index $\mathcal{M}$ that appears as a new parameter with prior distribution $\pi(\mathcal{M} = m)$ and $\pi(\theta|\mathcal{M} = m) = \pi_m(\theta_m)$
Computational target:

$$\mathbb{P}(\mathcal{M} = m|\mathbf{x}) \propto \int_{\Theta_m} f_m(\mathbf{x}|\theta_m)\pi_m(\theta_m)\,\mathrm{d}\theta_m\,\pi(\mathcal{M} = m)\,,$$

Sufficient statistics

By definition, if $S(\mathbf{x})$ sufficient statistic for the joint parameters $(\mathcal{M}, \theta_0, \ldots, \theta_{M-1})$,

$$\mathbb{P}(\mathcal{M} = m|\mathbf{x}) = \mathbb{P}(\mathcal{M} = m|S(\mathbf{x}))\,.$$

Sufficient statistics

By definition, if $S(\mathbf{x})$ sufficient statistic for the joint parameters $(\mathcal{M}, \theta_0, \ldots, \theta_{M-1})$,

$$\mathbb{P}(\mathcal{M} = m|\mathbf{x}) = \mathbb{P}(\mathcal{M} = m|S(\mathbf{x}))\,.$$

For each model m, own sufficient statistic $S_m(\cdot)$ and $S(\cdot) = (S_0(\cdot), \ldots, S_{M-1}(\cdot))$ also sufficient.

Sufficient statistics

By definition, if $S(\mathbf{x})$ sufficient statistic for the joint parameters $(\mathcal{M}, \theta_0, \ldots, \theta_{M-1})$,

$$\mathbb{P}(\mathcal{M} = m|\mathbf{x}) = \mathbb{P}(\mathcal{M} = m|S(\mathbf{x}))\,.$$

For each model m, own sufficient statistic $S_m(\cdot)$ and $S(\cdot) = (S_0(\cdot), \ldots, S_{M-1}(\cdot))$ also sufficient.
For Gibbs random fields,

$$\begin{aligned} x|\mathcal{M} = m \sim f_m(\mathbf{x}|\theta_m) &= f_m^1(\mathbf{x}|S(\mathbf{x}))f_m^2(S(\mathbf{x})|\theta_m) \\ &= \frac{1}{n(S(\mathbf{x}))} f_m^2(S(\mathbf{x})|\theta_m) \end{aligned}$$

where

$$n(S(\mathbf{x})) = \sharp\{\tilde{\mathbf{x}} \in \mathcal{X} : S(\tilde{\mathbf{x}}) = S(\mathbf{x})\}$$

© $S(\mathbf{x})$ **is therefore also sufficient for the joint parameters**

[Specific to Gibbs random fields!]

ABC model choice Algorithm

ABC-MC

- Generate m^* from the prior $\pi(\mathcal{M} = m)$.
- Generate $\theta^*_{m^*}$ from the prior $\pi_{m^*}(\cdot)$.
- Generate x^* from the model $f_{m^*}(\cdot|\theta^*_{m^*})$.
- Compute the distance $\rho(S(\mathbf{x}^0), S(\mathbf{x}^*))$.
- Accept $(\theta^*_{m^*}, m^*)$ if $\rho(S(\mathbf{x}^0), S(\mathbf{x}^*)) < \epsilon$.

Note When $\epsilon = 0$ the algorithm is exact

ABC approximation to the Bayes factor

Frequency ratio:

$$\begin{aligned}\overline{BF}_{m_0/m_1}(\mathbf{x}^0) &= \frac{\hat{\mathbb{P}}(\mathcal{M}=m_0|\mathbf{x}^0)}{\hat{\mathbb{P}}(\mathcal{M}=m_1|\mathbf{x}^0)} \times \frac{\pi(\mathcal{M}=m_1)}{\pi(\mathcal{M}=m_0)} \\ &= \frac{\sharp\{m^{i*}=m_0\}}{\sharp\{m^{i*}=m_1\}} \times \frac{\pi(\mathcal{M}=m_1)}{\pi(\mathcal{M}=m_0)}\,,\end{aligned}$$

ABC approximation to the Bayes factor

Frequency ratio:

$$\begin{aligned}\overline{BF}_{m_0/m_1}(\mathbf{x}^0) &= \frac{\hat{\mathbb{P}}(\mathcal{M}=m_0|\mathbf{x}^0)}{\hat{\mathbb{P}}(\mathcal{M}=m_1|\mathbf{x}^0)} \times \frac{\pi(\mathcal{M}=m_1)}{\pi(\mathcal{M}=m_0)} \\ &= \frac{\sharp\{m^{i*}=m_0\}}{\sharp\{m^{i*}=m_1\}} \times \frac{\pi(\mathcal{M}=m_1)}{\pi(\mathcal{M}=m_0)}\,,\end{aligned}$$

replaced with

$$\widehat{BF}_{m_0/m_1}(\mathbf{x}^0) = \frac{1+\sharp\{m^{i*}=m_0\}}{1+\sharp\{m^{i*}=m_1\}} \times \frac{\pi(\mathcal{M}=m_1)}{\pi(\mathcal{M}=m_0)}$$

to avoid indeterminacy (also Bayes estimate).

Toy example

iid Bernoulli model versus two-state first-order Markov chain, i.e.

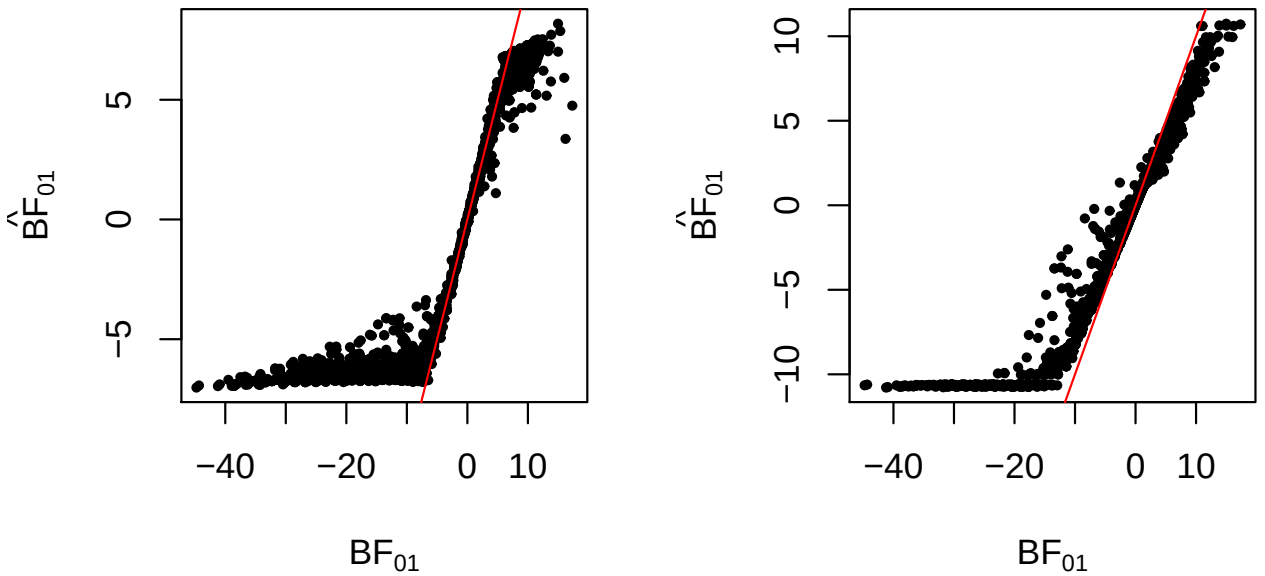
$$f_0(\mathbf{x}|\theta_0) = \exp\left(\theta_0 \sum_{i=1}^{n} \mathbb{I}_{\{x_i=1\}}\right) / \{1+\exp(\theta_0)\}^n \,,$$

versus

$$f_1(\mathbf{x}|\theta_1) = \frac{1}{2}\exp\left(\theta_1 \sum_{i=2}^{n} \mathbb{I}_{\{x_i=x_{i-1}\}}\right) / \{1+\exp(\theta_1)\}^{n-1} \,,$$

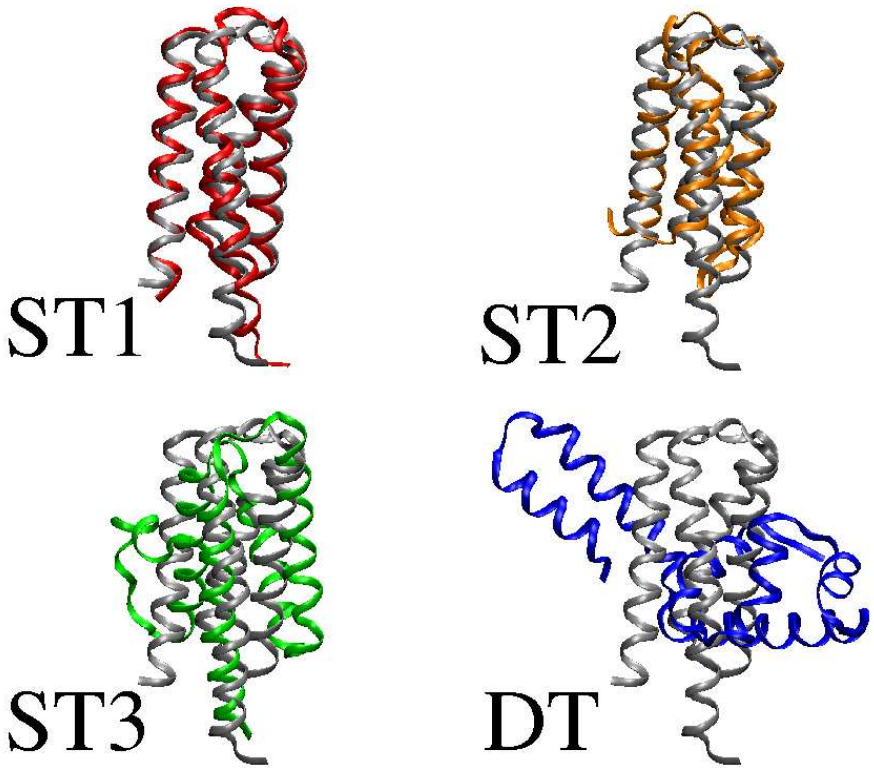
with priors $\theta_0 \sim \mathcal{U}(-5,5)$ and $\theta_1 \sim \mathcal{U}(0,6)$ (inspired by "phase transition" boundaries).

Toy example (2)



(left) Comparison of the true $BF_{m_0/m_1}(\mathbf{x}^0)$ with $\widehat{BF}_{m_0/m_1}(\mathbf{x}^0)$ (in logs) over $2,000$ simulations and 4.10^6 proposals from the prior. *(right)* Same when using tolerance ϵ corresponding to the 1% quantile on the distances.

Protein folding



Superposition of the native structure *(grey)* with the **ST1** structure *(red.)*, the **ST2** structure *(orange)*, the **ST3** structure *(green)*, and the **DT** structure *(blue)*.

Protein folding (2)

	% seq . Id.	TM-score	FROST score
1i5nA (**ST1**)	32	0.86	75.3
1ls1A1 (**ST2**)	5	0.42	8.9
1jr8A (**ST3**)	4	0.24	8.9
1s7oA (**DT**)	10	0.08	7.8

Characteristics of dataset. *% seq. Id.*: percentage of identity with the query sequence. *TM-score.*: similarity between predicted and native structure (uncertainty between 0.17 and 0.4) *FROST score*: quality of alignment of the query onto the candidate structure (uncertainty between 7 and 9).

Protein folding (3)

	NS/ST1	**NS/ST2**	**NS/ST3**	**NS/DT**
$\widehat{BF}$	1.34	1.22	2.42	2.76
$\widehat{\mathbb{P}}(\mathcal{M} = \textbf{NS}\|\mathbf{x}^0)$	0.573	0.551	0.708	0.734

Estimates of the Bayes factors between model **NS** and models **ST1**, **ST2**, **ST3**, and **DT**, and corresponding posterior probabilities of model **NS** based on an ABC-MC algorithm using $1.2\,10^6$ simulations and a tolerance ϵ equal to the 1% quantile of the distances.