Likelihood-free Sequential Transport Monte Carlo

ISBA World Meeting

Cecilia Viscardi[a] – University of Florence
joint with Dennis Prangle – University of Bristol

June 27, 2022

[a]cecilia.viscardi@unifi.it

Outline of the talk

1. Setup and key idea
2. Two ingredients
   - Sequential ABC methods
   - Normalising flows
3. Transport PMC-ABC
4. Preliminary results
5. Discussion and future work

Part I

Setup and notation

- θ: parameters to infer
- y_0: observed data

GOAL:

$$\underbrace{p(\theta|y_0)}_{\text{posterior}} \propto \underbrace{\pi(\theta)}_{\text{prior}} \underbrace{p(y_0|\theta)}_{\text{Intractable likelihood}}$$

- When $p(y_0|\theta)$ is intractable
- BUT we are able to produce **pseudo-data** y from a *simulator*
- we can resort to SBI, such as ABC.

ABC methods

At each iteration

1. Draw θ from $\underbrace{\pi(\cdot)}_{\text{prior}}$
2. Simulate y from $\underbrace{p(\cdot|\theta)}_{\text{simulator}}$
3. Accept θ if $d(y, y_0) \leq \epsilon$

TARGET:

$$\underbrace{p_\epsilon(\theta|y_0)}_{\text{approximate posterior}} \propto \underbrace{\pi(\theta)\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}[d(y_i, y_0) \leq \epsilon]}_{\text{approximate likelihood}}$$

At a given iteration

$$p_\epsilon(\theta_i|y_0) \propto \pi(\theta_i)\mathbb{1}[d(y_i, y_0) \leq \epsilon]$$

ABC algorithms based on IS

ABC algorithms often involve IS steps:

1. Draw θ from $\underbrace{q(\cdot)}_{\text{proposal distribution}}$;
2. Draw pseudo-data y from $\underbrace{p(\cdot|\theta)}_{\text{simulator}}$;
3. Give a weight $\underbrace{\frac{\pi(\theta)}{q(\theta)}\mathbb{1}[d(y, y_0) \leq \epsilon]}_{\text{target/proposal}}$

A good proposal may be any distribution *close* to the target to:

- get small ϵ values;
- AND/OR get proper ESS.

The main idea

The main idea of this work is to combine two ingredients:

1. **Likelihood-free sequential methods**
2. **Normalising flows (NFs)**

Many recent methods are based on them

- **Distilled Importance Sampling** [Prangle and Viscardi, 2019]:
- **Annealed Flow Transport Monte Carlo** (not likelihood-free) [Arbel et al., 2021];
- Sequential Neural Posterior and Likelihood Approximation [Wiqvist et al., 2021]
- etc.

Part II

Sequential ABC methods

Sequential methods usually

- Get samples from a sequence of **tempered target** densities (using $\epsilon_1 \geq \epsilon_2 ... \geq \epsilon_K$);
- Are based on **IS steps** (SIS-ABC, SMC-ABC, PMC-ABC)

  [Beaumont et al., 2009, Del Moral et al., 2012]

Algorithm 1 PMC-ABC

```
Initialise ϵ_1 = ∞
for k = 1, 2, ... do
    Let i = 0 (number of acceptances).
    while i < N do
        Sample θ* from q_k(θ) = (Σ_{i=1}^N w_i^{k−1} K(θ|θ_i^{k−1})) / (Σ_{i=1}^N w_i^{k−1})
        Sample y* from p(·|θ*) and let d* = d(y*, y_0)
        if d* ≤ ϵ_k then
            θ_i^k = θ*, d_i^k = d*, w_i^k = π(θ*)/q_k(θ*) and increment i by 1
        end if
    end while
    Calculate ϵ_{k+1} as the α quantile of the distances.
end for
```

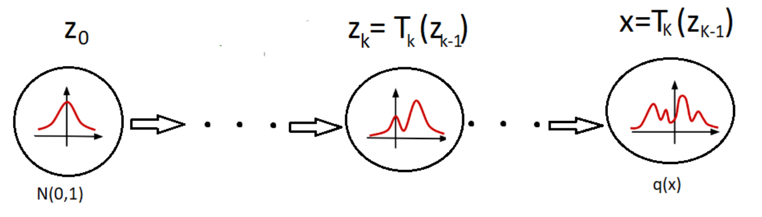
Normalising flows (NFs)

- NFs are useful for tasks of [Dinh et al., 2016, Papamakarios et al., 2021]
  - Density estimation – trained to match observed data;
  - Approximate inference – trained to match intractable distributions.
- NFs are probabilistic models deforming a simple distribution to match some complex distribution (or viceversa):

$$Z \sim N(0,1); \quad X = T(Z); \quad X \sim q(\cdot)$$

- They usually apply a composition of transformations:

$$T = T_K \circ T_{K-1} \circ \ldots \circ T_1$$



Training NFs

- Each transformation T_k is parametrised by ϕ_k – e.g. $T_k(z) = a(z, \phi_k)z + b(z, \phi_k)$;
- The distribution $q(x; \phi)$ is retrieved using the *change of variable formula*
$$q(x; \phi) = \Phi(z)/|\nabla T(z; \phi)|;$$
- Transformations must be bijective and the determinant of the Jacobian should be fast to compute;
- Flows are trained to estimate $\phi = (\phi_1, ..., \phi_k)$ leading to $q(x; \phi)$ as close as possible to a **target** function $p(\cdot)$;
- Optimal ϕ can be computed minimising

$$L(\phi) = KL\big(\underbrace{p(x)}_{\text{target}}||q(x; \phi)\big) = E_p[\log p(x) - \log q(x; \phi)].$$

Part III

Transport ABC-PMC

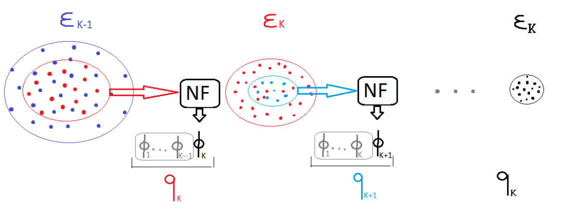
$$\text{PMC-ABC} + \text{NFs} = \text{Transport PMC-ABC}$$

We alternate

- IS steps;
- Steps training NFs.

At each step k

- Samples from the current tempered target p_{ϵ_k} are used to train the NF and get q_{k+1};
- q_{k+1} will transport particles toward $p_{\epsilon_{k+1}}$.



Algorithm 2 Transport PMC-ABC

Let $q_1(\theta) = \pi(\theta)$ (the prior) and $\epsilon_1 = \infty$.
for $k = 1, 2, 3, \ldots$ **do**
  Importance sampling: Sample $N_{\text{train}} + N_{\text{test}}$ weighted particles (θ, ω) as follows. Sample $\theta \sim q_k$ until $d(y, y_0) \leq \epsilon_k$. Let $\omega = \pi(\theta)/q_k(\theta)$.
  Select ϵ_{k+1} as the α quantile of the training distances.
  Train proposal: Sequentially optimise

$$L_k(\phi) = \sum_{(\theta,\omega)\in\text{train}} \underbrace{\omega \cdot \mathbb{1}[d(y, y_0) \leq \epsilon_{k+1}]}_{\text{reweighting}} \log q(\theta; \phi)$$

  and select ϕ^* from the results which maximises the test loss

$$L_{k,}(\phi) = \sum_{(\theta,\omega)\in\text{test}} \underbrace{\omega \cdot \mathbb{1}[d(y, y_0) \leq \epsilon_{k+1}]}_{\text{reweighting}} \log q(\theta; \phi)$$

end for

Transport PMC-ABC: details

- To estimate q_{k+1} *as close as possible* to $p_{\epsilon_{k+1}}$ we wish to minimise

$$E_{p_{\epsilon_{k+1}}}[\log p_{\epsilon_{k+1}}(\theta) - \log q_{k+1}(\theta;\phi)] \approx \frac{1}{N}\sum_{i=1}^{N}[\underbrace{\log p_{\epsilon_{k+1}}(\theta_i)}_{\text{constant}} - \log q_{k+1}(\theta_i;\phi)\,]$$

- The loss function is

$$L_k(\phi) = \sum_{i=1}^{N} \log q_{k+1}(\theta_i;\phi) \quad \theta_i \sim p_{\epsilon_{k+1}}$$

- $(\theta_1, \omega_1)\ldots(\theta_N, \omega_N)$ is a weighted sample from p_{ϵ_k} with

$$\omega(\theta_i) = \frac{\pi(\theta_i)}{q_k(\theta_i)}$$

- **Reweighting**:

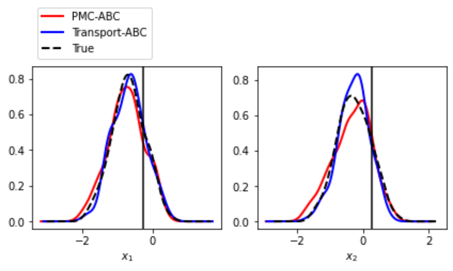
$$\omega^*(\theta_i) = \omega(\theta_i)\mathbb{1}[d(y_i, y_0) \leq \epsilon_{k+1}] = \frac{\pi(\theta_i)\mathbb{1}[d(y_i, y_0) \leq \epsilon_{k+1}]}{q_k(\theta_i)}$$

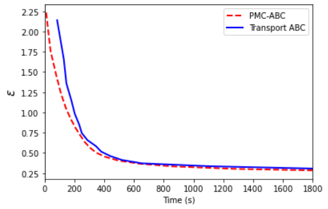
- **IS estimate:** $L_k(\phi) = \sum_{i=1}^{N} \omega^*(\theta_i) \log q_{k+1}(\theta_i;\phi)$.

Part IV

Toy example: MVN

- $Y \sim MVN(x, \Sigma)$ where $x = [\mu_1, \mu_2]$ $\quad \Sigma = \begin{bmatrix} 1.3862 & 1.4245 \\ 1.4245 & 1.5986 \end{bmatrix}$
- $X \sim MVN(\mu_0, \Sigma_0)$ where $\mu_0 = [0, 0]$ $\quad \Sigma_0 = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$
- $N = 1000$ and $\alpha = 0.5$

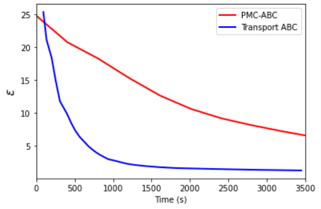


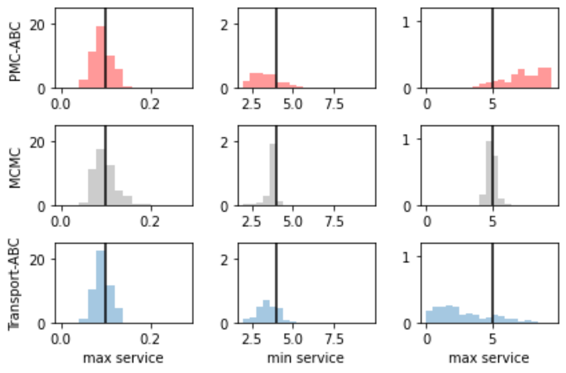


M/G/1 example

- We consider a M/G/1 queuing model of a single queue of customers;
- Times between arrivals at the back of the queue are $\text{Exp}(\theta_1)$;
- Service time is $U(\theta_2, \theta_3)$;
- y_0 is a synthetic dataset of $m = 20$ observations;
- Parameters to be inferred: $\theta = (\theta_1, \theta_2, \theta_3)$;
- Prior distributions: $\theta_1 \sim U(0, 1/3)$, $\theta_2 \sim U(0, 10)$, $\theta_3 - \theta_2 \sim U(0, 10)$;
- Summary statistics: quartiles of the inter-departure times.

PMC-ABC vs Transport PMC





DIS vs Transport PMC-ABC

Distilled Importance Sampling (DIS) reached better results ($\epsilon = 0.13$) and further work is needed to get competitive results.

- **DIS** can *orient* the simulator to get a faster decrease of ϵ;
- The object of the inference is $x = (\theta, h)$ where h are latent variables and pseudo-random numbers;
- $y : \mathcal{X} \to \mathcal{Y}$;
- $q_k(x)$ close to $p_{\epsilon_k}(x|y_0)$ will produce pseudo-data closer to the observed data.

Transport PMC-ABC

- can *orient* the simulator (as DIS does);
- exploits sequentiality to freeze the transformations that have been learned at previous iterations. At each iteration k, we must learn a flow from p_{ϵ_k} to $p_{\epsilon_{k+1}}$ (rather than from π to $p_{\epsilon_{k+1}}$);
- HOWEVER requires a very good fit of each normalising flow (hard with high-dimensional x).

Part V

Discussion

- **NFs** can help to define good proposal distributions in sequential likelihood-free methods;
- Proposal distributions close to the tempered target allow a faster decrease of ϵ values;
- **Transport PMC-ABC** guarantees that ϵ reduces in every iteration;
- **Transport PMC-ABC** (as DIS) allows *orienting* the simulator;
- Using forward KL has practical and theoretical advantages.

Future work

- The method can be extended to a SMC-ABC sampling scheme;
- Further work is needed to get the method competitive compared to DIS;
- Test the method at work on discrete variables;
- The choice of N and α should be further investigated;
- Compare Transport ABC with other SBI methods

$$\vdots$$

Arbel, M., Matthews, A. G. D. G., and Doucet, A. (2021).
Annealed flow transport monte carlo.

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. (2009).
Adaptive approximate bayesian computation.
Biometrika, 96(4):983–990.

Del Moral, P., Doucet, A., and Jasra, A. (2012).
An adaptive sequential monte carlo method for approximate bayesian computation.
Statistics and computing, 22(5):1009–1020.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016).
Density estimation using real nvp.
arXiv preprint arXiv:1605.08803.

Papamakarios, G., Nalisnick, E. T., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021).
Normalizing flows for probabilistic modeling and inference.
J. Mach. Learn. Res., 22(57):1–64.

Papamakarios, G., Pavlakou, T., and Murray, I. (2017).
Masked autoregressive flow for density estimation.
In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Prangle, D. and Viscardi, C. (2019).
Distilling importance sampling.

References ii

Shestopaloff, A. Y. and Neal, R. M. (2014).
On bayesian inference for the m/g/1 queue with efficient mcmc sampling.

Wiqvist, S., Frellsen, J., and Picchini, U. (2021).
Sequential neural posterior and likelihood approximation.

Details on the experiments

- Masked Piecewise Rational Quadratic Autoregressive Transform [Papamakarios et al., 2017]
- *https://github.com/bayesiains/nflows*
- *https://github.com/dennisprangle/DistillingImportanceSampling*
- MCMC scheme for M/G/1 example from [Shestopaloff and Neal, 2014]