Bayesian data-driven model discovery under uncertainty

M. Ceoloni, F. Fatone, F. Fedeli

Supervised by:

Prof. A. Guglielmi, Prof. A. Manzoni

February 18, 2021

Project purpose and goals

- **Model identification** and **parameters estimation** are of extreme importance in multiple fields of application.
- In the **recent past**, some techniques have been proposed in order to provide a **quantitative estimation of the uncertainty**, crucial for applications, exploiting **Bayesian tools**.
- We considered them with the **final goal** of finding a good application to real-world problems and in particular the **Covid-19 outbreak in Italy**.

IL FAMOSO RO

Coronavirus, ecco regione per regione l'indice di trasmissibilità della malattia

L' Ro indica il numero di infezioni prodotte da una persona nell'arco del suo periodo infettivo. Un dato associato al tempo che intercorre nel passaggio della malattia fra un infetto primario e quelli secondari

Project purpose and goals

- **Model identification** and **parameters estimation** are of extreme importance in multiple fields of application.
- In the **recent past**, some techniques have been proposed in order to provide a **quantitative estimation of the uncertainty**, crucial for applications, exploiting **Bayesian tools**.
- We considered them with the **final goal** of finding a good application to real-world problems and in particular the **Covid-19 outbreak in Italy**.

Research Question

Are those techniques suitable for modelling complex real-world systems? What are their performances? Can they be improved?

Roadmap - HMC sampling (Perdikaris et al., 2020)

We started by considering the work of **Perdikaris et al.** (2020), which provided a framework that seemed to **fit our research question**.

Its main features are:

- Based on a Gaussian likelihood on the error:
  $p(\boldsymbol{x}(t+\Delta t)|\boldsymbol{x}(t),\boldsymbol{\theta},\gamma) = \prod_{i=1}^{N} \mathcal{N}(\boldsymbol{x}(t_i+\Delta t_i); h_{\boldsymbol{\theta}}, \gamma^{-1})$
- Used a Lasso/Ridge-like shrinkage Bayesian model:
  $p(\boldsymbol{\theta}|\lambda) = Laplace/\mathcal{N}(\boldsymbol{\theta}; 0, \lambda^{-1})$
- Split in two semi-independent parts:
  1. Numerical (deterministic) estimation of the parameters' starting point for the HMC algorithm and the required gradients
  2. Sampling from the posterior through HMC

Roadmap - HMC sampling (Perdikaris et al., 2020)

We started by considering the work of **Perdikaris et al.** (2020), which provided a framework that seemed to **fit our research question**.

Its main features are:

- Based on a Gaussian likelihood on the error:
  $p(\boldsymbol{x}(t+\Delta t)|\boldsymbol{x}(t),\boldsymbol{\theta},\gamma) = \prod_{i=1}^{N} \mathcal{N}(\boldsymbol{x}(t_i+\Delta t_i); h_{\boldsymbol{\theta}}, \gamma^{-1})$
- Used a Lasso/Ridge-like shrinkage Bayesian model:
  $p(\boldsymbol{\theta}|\lambda) = Laplace/\mathcal{N}(\boldsymbol{\theta}; 0, \lambda^{-1})$
- Split in two semi-independent parts:
  1. Numerical (deterministic) estimation of the parameters' starting point for the HMC algorithm and the required gradients
  2. Sampling from the posterior through HMC

We applied it to a test case (Lotka-Volterra model) and thanks to the addition of an adaptive **mass matrix** we obtained **way better results than in the reference paper**.

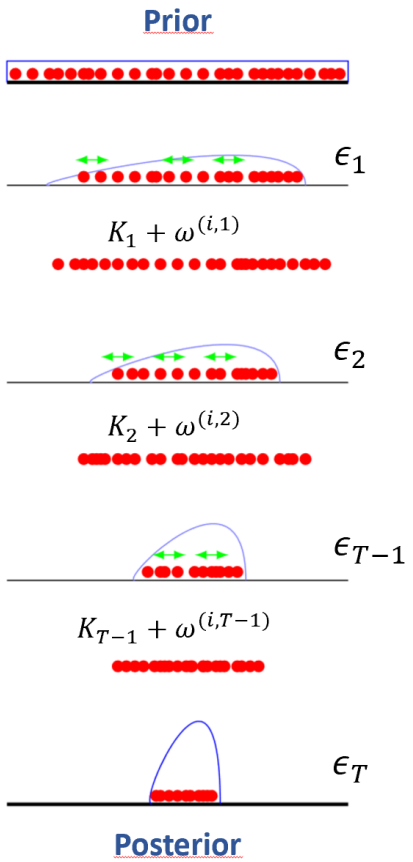
Roadmap - ABC-SMC algorithm (Toni et al., 2009)

Some issues arose when applied to real-world data, as soon as we extended this framework to the SIR model (**batch-feeding, poor performances**)

We relied then on the framework introduced by Toni et al. (2009), based on **ABC - Sequential Monte Carlo (ABC-SMC)** scheme

Algorithm 1: ABC - Sequential Monte Carlo

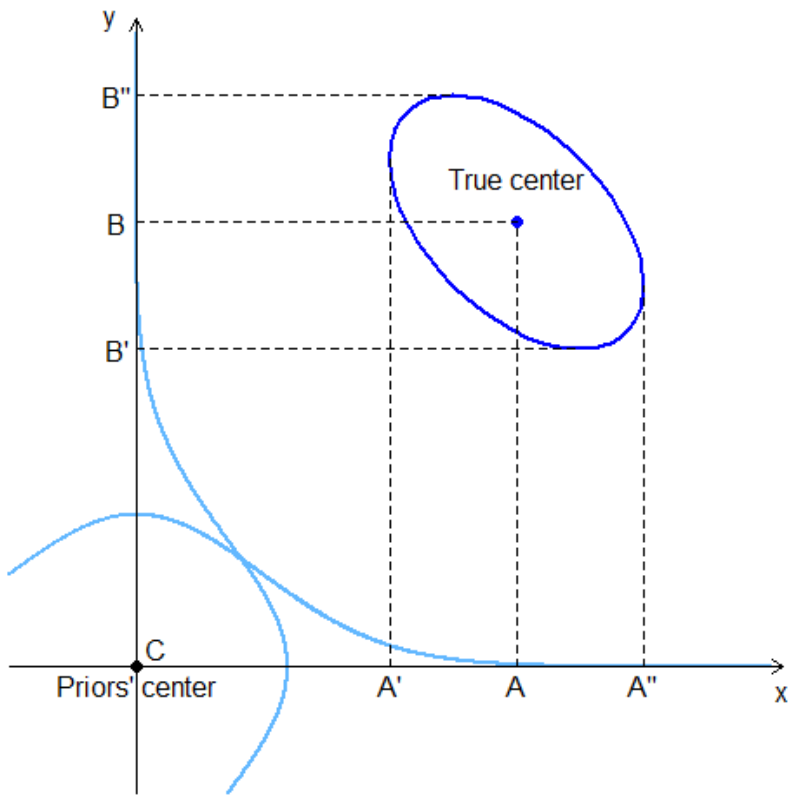
Result: A sample from $p_\epsilon(\boldsymbol{\theta}|x)$
Initialization: A precision schedule $\{\epsilon_t\}_{t\in 1:T}$;
while $t \leq T$ **do**
  while $n \leq N$ **do**
    if $t = 1$ **then**
      sample $\tilde{\theta}$ from $\pi(\theta)$;
    else
      sample θ from the previous population $\{\theta^{(i,t-1)}\}_i$ with weights $\{\omega^{(i,t-1)}\}_i$;
      sample $\tilde{\theta}$ from $K_t(\cdot|\theta)$ s.t. $\pi(\theta) > 0$;
    end
    compute $y = f(\cdot|\tilde{\theta})$;
    if $\Delta(y, x) \leq \epsilon_t$ **then**
      save $\tilde{\theta}$ and y;
    else
  end
  compute $\{\omega^{(i,t)}\}_i$ and normalize them;
end



Roadmap - Preconditioning

Standard ABC-SMC showed problems in case of priors being not diffuse enough.

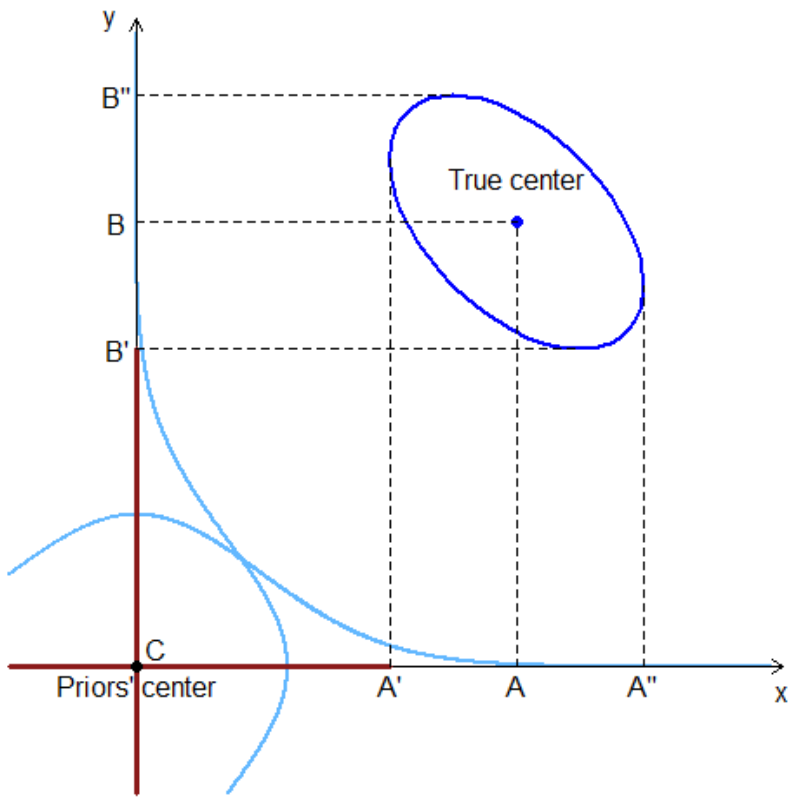
We introduced a **preconditioning phase**, exploiting the estimate of the true centre of the posterior from the numerical estimation phase.



Roadmap - Preconditioning

Standard ABC-SMC showed problems in case of priors being not diffuse enough.

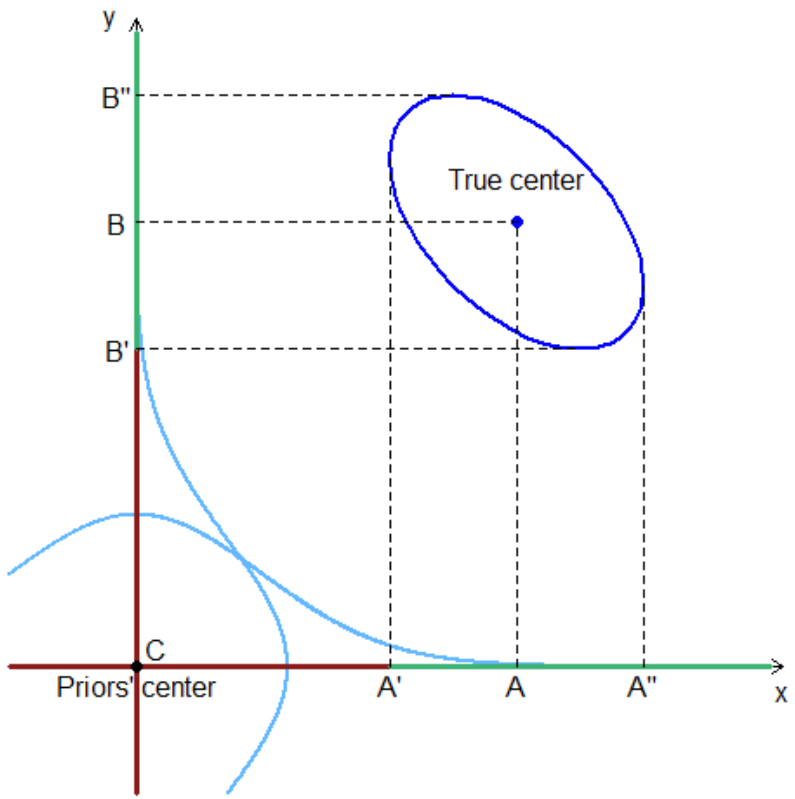
We introduced a **preconditioning phase**, exploiting the estimate of the true centre of the posterior from the numerical estimation phase.



Roadmap - Preconditioning

Standard ABC-SMC showed problems in case of priors being not diffuse enough.

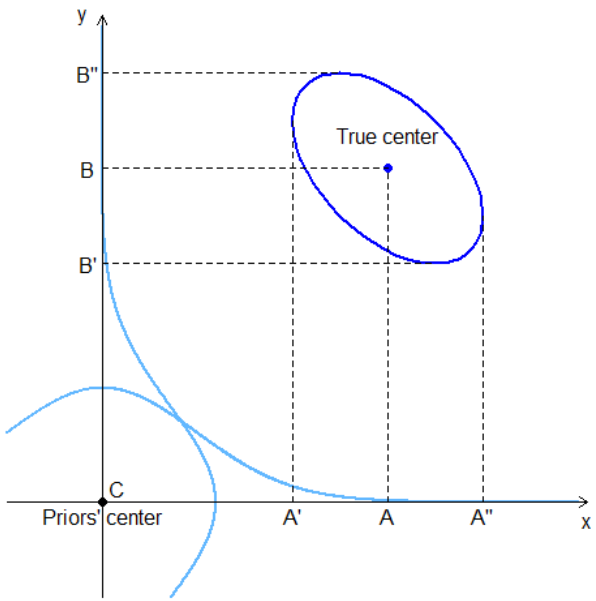
We introduced a **preconditioning phase**, exploiting the estimate of the true centre of the posterior from the numerical estimation phase.



Roadmap - Preconditioning

Standard ABC-SMC showed problems in case of priors being not diffuse enough.

We introduced a **preconditioning phase**, exploiting the estimate of the true centre of the posterior from the numerical estimation phase.



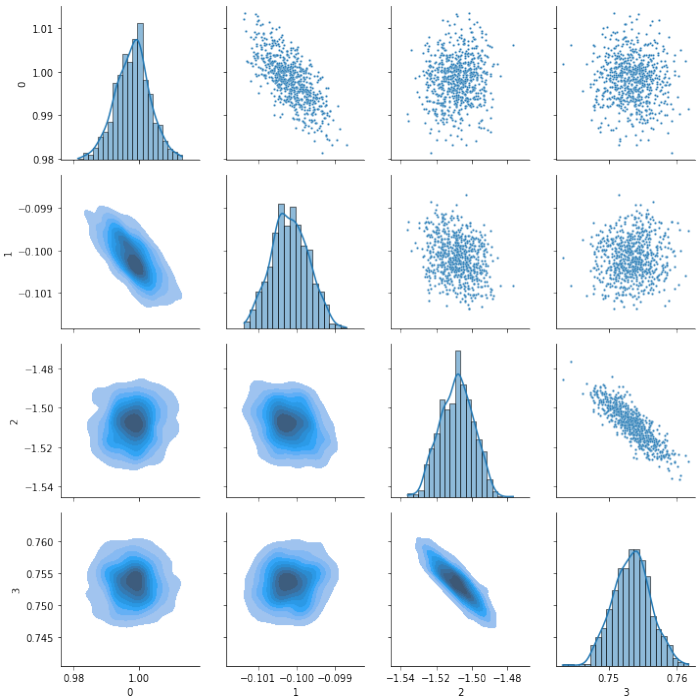
Algorithm 2: ABC-SMC 'empirical' preconditioning (1D)

Result: A′: estimate of the border of $\mathfrak{R}_{\epsilon_{start}}$
Initialization: A: estimate of the posterior centre, *tol* $\in (0,1)$;
while $n \leq N$ **do**
  if $A > q_\pi(1 - tol)$ *or* $A < q_\pi(tol)$ **then**
    sample P $\sim \mathcal{U}((C, A))$;
    sample $\tilde{\theta}$ from $\pi(\theta|\theta > P)$;
  else
    sample $\tilde{\theta}$ from $\pi(\theta)$;
  end
  compute $y = f(\cdot|\tilde{\theta})$;
  if $\Delta(y, x) \leq \epsilon$ **then**
    save $\tilde{\theta}$;
  else
end
return $min(\tilde{\theta})$

Roadmap - A comparison

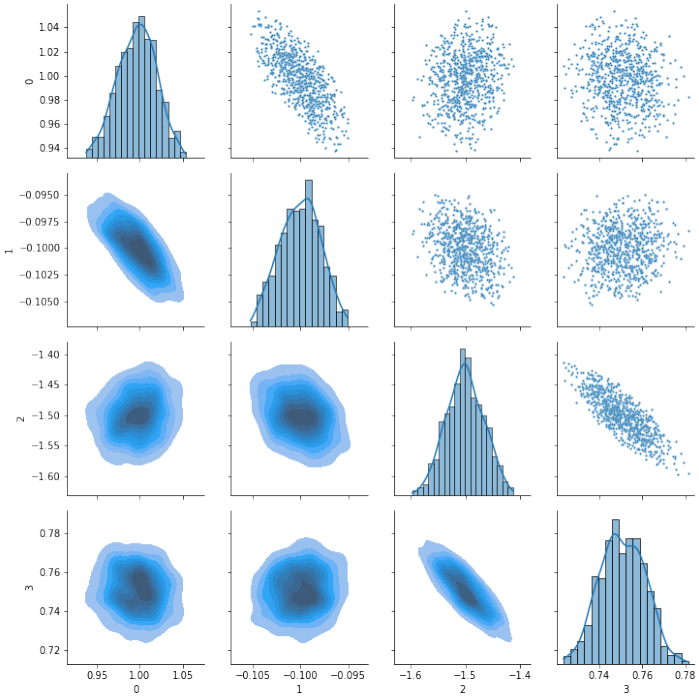
Finally we applied our idea to the **Lotka-Volterra model** with Ridge-like shrinkage, in order to assess the performances of the 2 frameworks on the same problem.

LV using HMC (Perdikaris)



Computational time: 1h 05min

LV using eABC-SMC

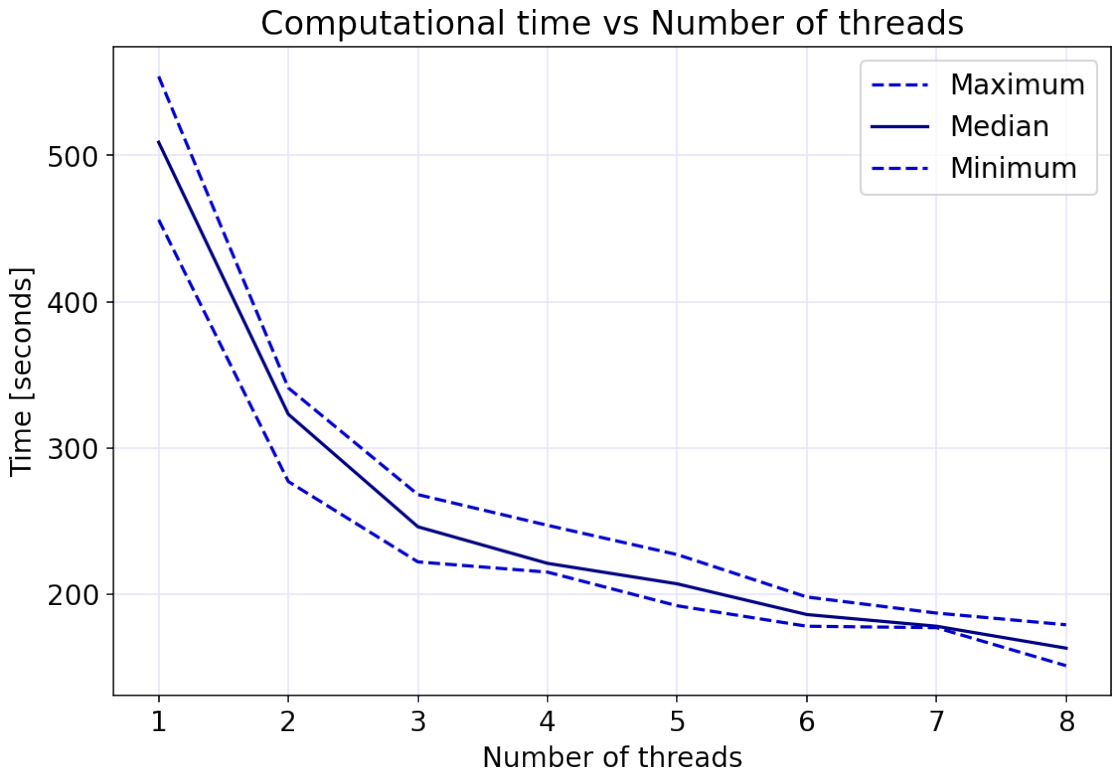


Computational time: 6 min (parallelized)

We observed a huge difference in computational time, while preserving the goodness of results.

Roadmap - Parallelization results

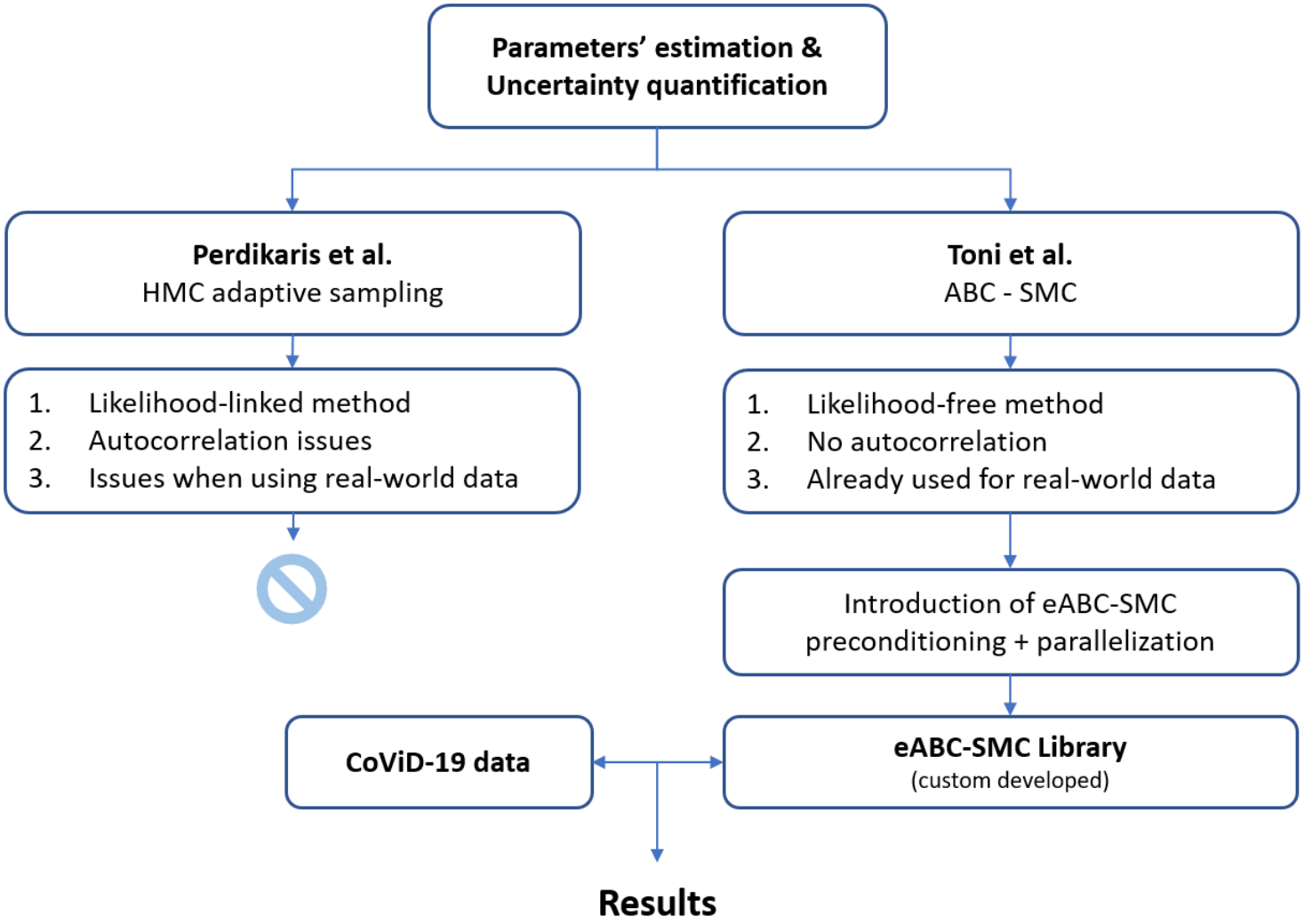
The eABC-SMC features a highly parallelizable architecture, that we fully exploited in our custom-made library. We carried out a performance analysis getting the following results:



The trend of the curve (on 4 Cores, 8 Threads Intel Core i7) shows a mean 25% reduction of the computation time for each added thread.

Roadmap

Our roadmap can be therefore summarized as follows:



Results - Covid-19 Italy - First outbreak

The dynamical system we considered in order to model the pandemic was a SIRD model with time varying parameters (Ianni, Rossi - 2020):

$$\begin{cases} \dot{S} = -\beta(t)\dfrac{SI}{N} \\ \dot{I} = \beta(t)\dfrac{SI}{N} - \gamma(t)I - \mu(t)I \\ \dot{R} = \gamma(t)I \\ \dot{D} = \mu(t)I \end{cases}$$

Results - Covid-19 Italy - First outbreak

The dynamical system we considered in order to model the pandemic was a SIRD model with time varying parameters (Ianni, Rossi - 2020):

$$\begin{cases} \dot{S} = -\beta(t)\dfrac{SI}{N} \\ \dot{I} = \beta(t)\dfrac{SI}{N} - \gamma(t)I - \mu(t)I \\ \dot{R} = \gamma(t)I \\ \dot{D} = \mu(t)I \end{cases} \qquad \gamma(t) = \gamma_0$$

Results - Covid-19 Italy - First outbreak

The dynamical system we considered in order to model the pandemic was a SIRD model with time varying parameters (Ianni, Rossi - 2020):

$$\begin{cases} \dot{S} = -\beta(t)\dfrac{SI}{N} \\ \dot{I} = \beta(t)\dfrac{SI}{N} - \gamma(t)I - \mu(t)I \\ \dot{R} = \gamma(t)I \\ \dot{D} = \mu(t)I \end{cases} \qquad \begin{aligned} \gamma(t) &= \gamma_0 \\ \beta(t) &= \beta_0 e^{-\omega t} \end{aligned}$$

Results - Covid-19 Italy - First outbreak

The dynamical system we considered in order to model the pandemic was a SIRD model with time varying parameters (Ianni, Rossi - 2020):

$$\begin{cases} \dot{S} = -\beta(t)\dfrac{SI}{N} \\ \dot{I} = \beta(t)\dfrac{SI}{N} - \gamma(t)I - \mu(t)I \\ \dot{R} = \gamma(t)I \\ \dot{D} = \mu(t)I \end{cases}$$

$$\gamma(t) = \gamma_0$$
$$\beta(t) = \beta_0 e^{-\omega t}$$
$$\mu(t) = \frac{\mu_0}{t+1}$$

Results - Covid-19 Italy - First outbreak

The dynamical system we considered in order to model the pandemic was a SIRD model with time varying parameters (Ianni, Rossi - 2020):

$$\begin{cases} \dot{S} = -\beta(t)\dfrac{SI}{N} \\ \dot{I} = \beta(t)\dfrac{SI}{N} - \gamma(t)I - \mu(t)I \\ \dot{R} = \gamma(t)I \\ \dot{D} = \mu(t)I \end{cases} \qquad \begin{aligned} \gamma(t) &= \gamma_0 \\ \beta(t) &= \beta_0 e^{-\omega t} \\ \mu(t) &= \frac{\mu_0}{t+1} \end{aligned}$$

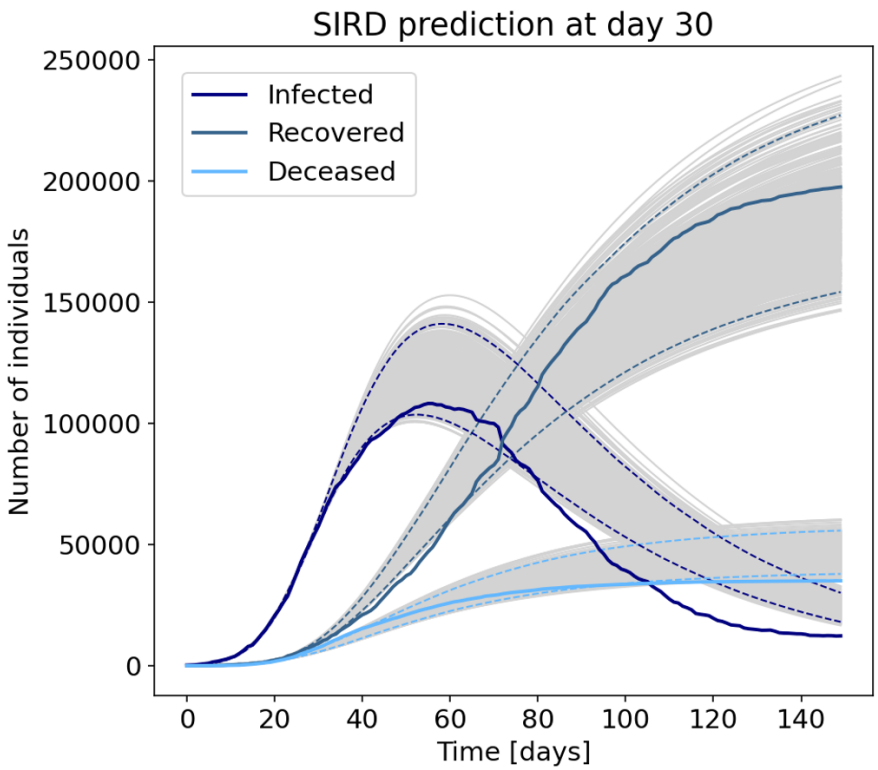
The following Bayesian model was taken for the application of eABC-SMC algorithm:

$$\begin{aligned} \beta_0 &\sim Gamma(\lambda_\beta \tilde{\beta}_0, \lambda_\beta) \\ \gamma_0 &\sim Gamma(\lambda_\gamma \tilde{\gamma}_0, \lambda_\gamma) \\ \mu_0 &\sim Gamma(\lambda_\mu \tilde{\mu}_0, \lambda_\mu) \\ N &\sim Gamma(\lambda_N \tilde{N}, \lambda_N) \\ \omega &\sim \mathcal{N}(\tilde{\omega}, \lambda_\omega) \end{aligned}$$

$$\text{with } \lambda_i \sim \mathcal{U}(\alpha_i, \beta_i)$$

Results - Covid-19 Italy - First outbreak

Using this framework we obtained pretty good results with the posterior:



Fit obtained using the first 30 days' data starting 2/24.

Priors obtained from Chinese data up to 3/25.

- With only few and extremely uncertain data we only slightly overestimated Infected and deceased.
- Our 95% HDP interval for the peak number of infected is:

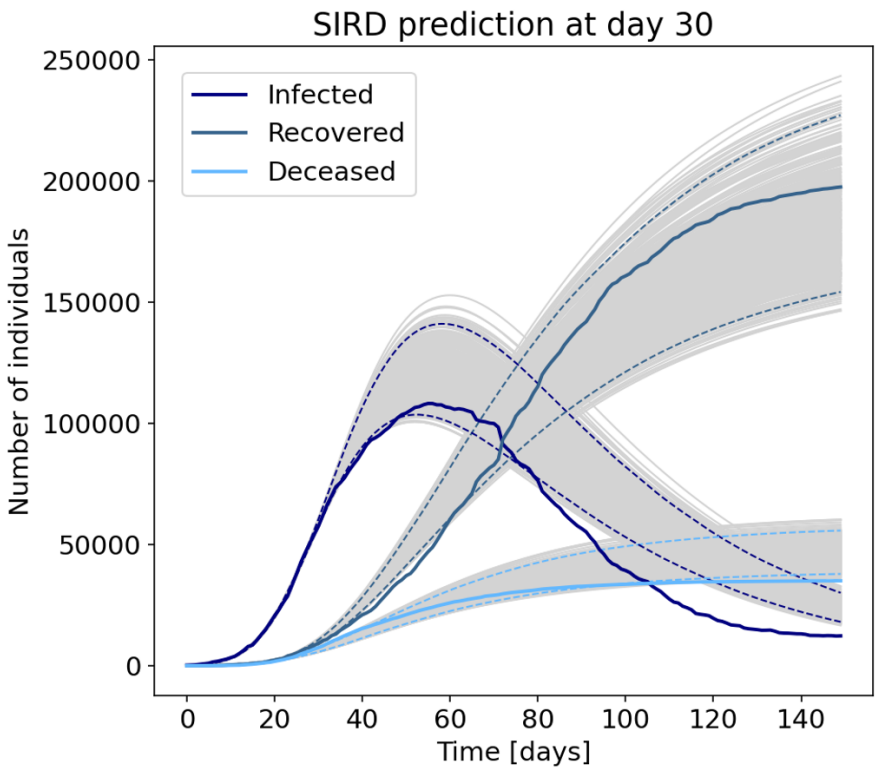
  $[103642, 141132]$ (true $= 108257$)

- Our 95% HDP interval for the peak time is:

  [April 16^{th}, April 22^{nd}] (true = April 18^{th})

Results - Covid-19 Italy - First outbreak

Using this framework we obtained pretty good results with the posterior:



Fit obtained using the first 30 days' data starting 2/24.

Priors obtained from Chinese data up to 3/25.

- With only few and extremely uncertain data we only slightly overestimated Infected and deceased.
- Our 95% HDP interval for the peak number of infected is:

  [103642, 141132] (true = 108257)

- Our 95% HDP interval for the peak time is:

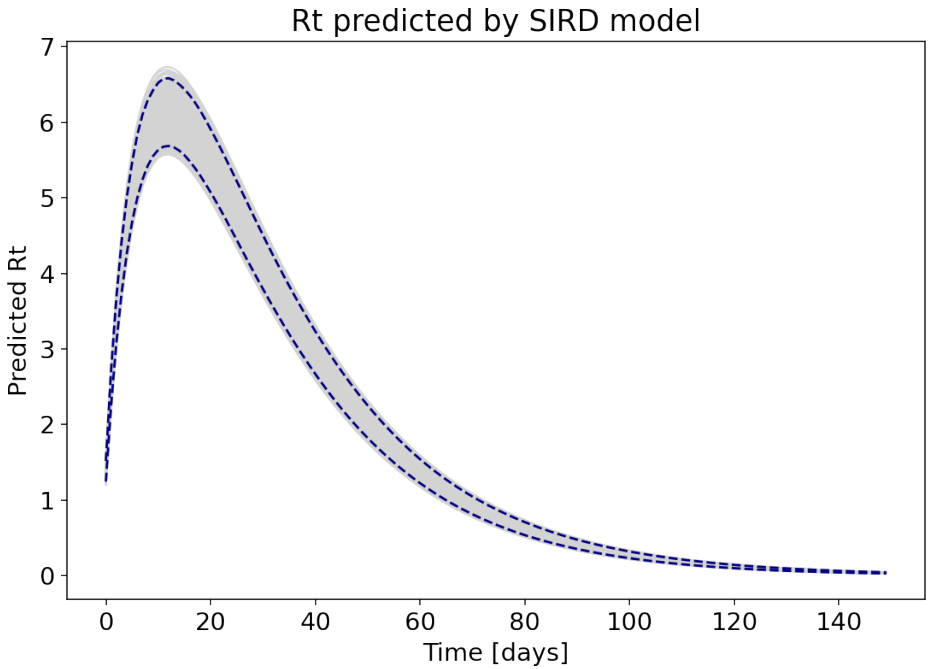
  [April 16^{th}, April 22^{nd}] (true = April 18^{th})


Coronavirus, per il picco dei contagi «serve ancora una settimana»

31 March 2020 - *CORRIERE DELLA SERA*

Results - Covid-19 Italy - First outbreak

We also carried out an analysis on the behaviour of the basic reproduction number during the first lockdown:



We define the **Basic reproduction number** or $\mathbf{R}_0$ as $R_0 = \frac{\beta(t)}{\gamma(t)+\mu(t)}$.

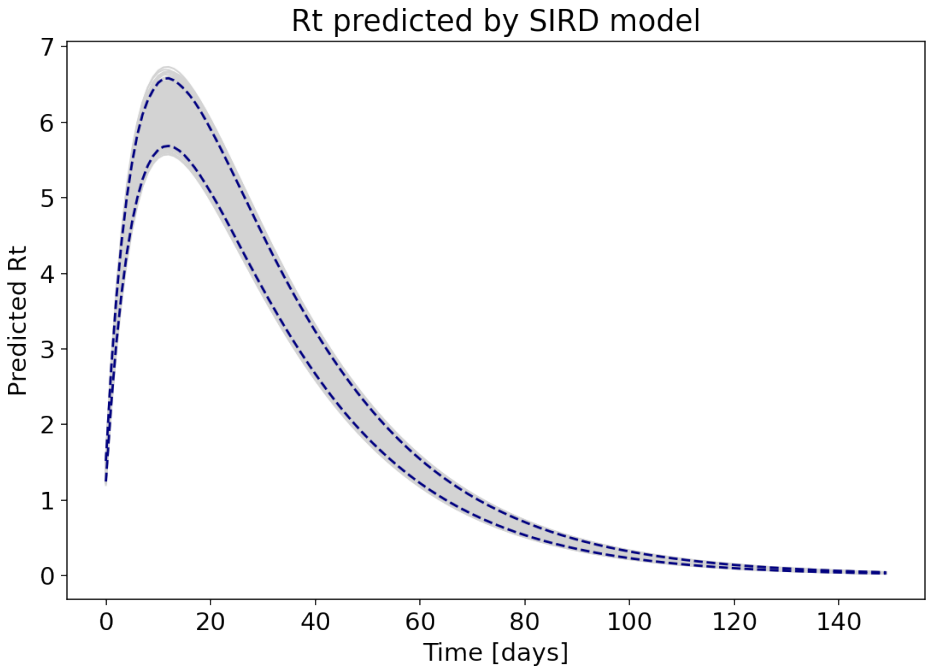
The 95% HPD interval for the peak time is

$$[\text{March } 5^{th}, \text{ March } 7^{th}]$$

Hard lockdown in Lombardia started on Sunday, March 8^{th}, the very same weekend.

Results - Covid-19 Italy - First outbreak

We also carried out an analysis on the behaviour of the basic reproduction number during the first lockdown:



We define the **Basic reproduction number** or $\mathbf{R}_0$ as $R_0 = \frac{\beta(t)}{\gamma(t)+\mu(t)}$.

The 95% HPD interval for the peak time is

$$[\text{March } 5^{th}, \text{ March } 7^{th}]$$

Hard lockdown in Lombardia started on Sunday, March 8^{th}, the very same weekend.

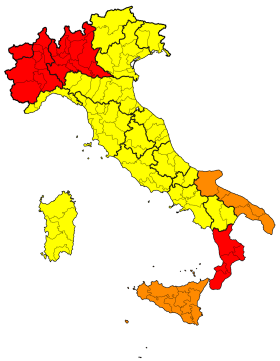
Another significant indicator of the contagion size was the **N parameter** (i.e. the **size of the initial population**). Regarding the Italian outbreak, a **95% HPD interval** for it is

$$[355054, 827385].$$

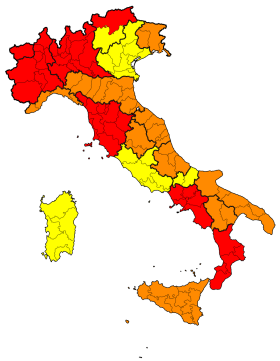
Results - Covid-19 Italy - Second outbreak

During the *second wave* in Fall 2020, Italian Government introduced a **color code** for the classification of risk for each Region and took **restrictive measures according to it**.

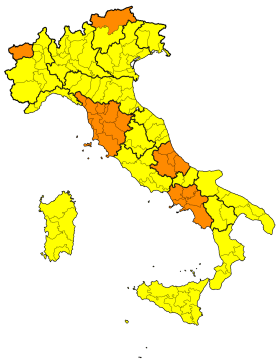
We report here the DPCMs trend for the period:



November 6^{th}



November 15^{th}



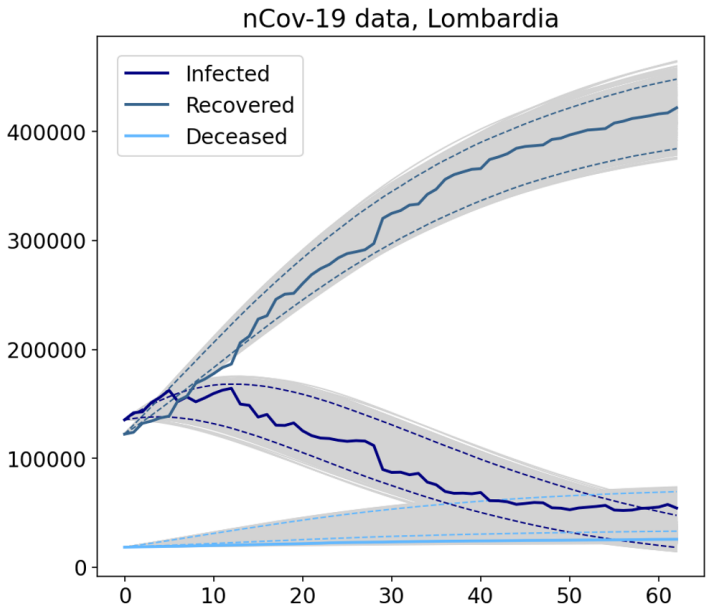
December 13^{th}

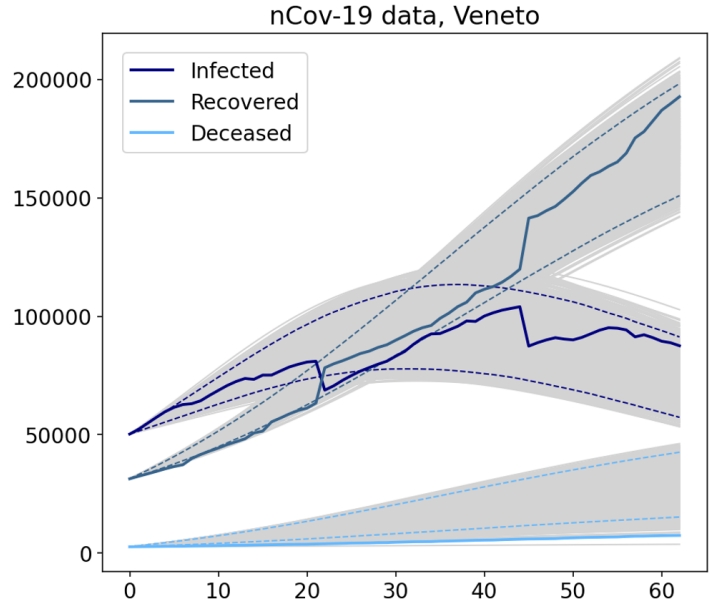
Our next goal was to evaluate possible **differences between Regions** and in particular trying to quantify them.

Were the differences among different colored Regions real? Do different measures have different impacts?

Results - Covid-19 Italy - Second outbreak

We considered in particular Lombardia (mostly been a *Red* Region) and Veneto (always been *Yellow* Region), obtaining the following results:





Curves were fitted for the period November 6^{th} 2020 - January 8^{th} 2021.

The goodness of the result gave us further confirmation of the good behaviour of the selected model in the description of the phenomenon.

Results - Covid-19 Italy - Second outbreak

More quantitatively, we considered some tests regarding the behaviour of $\beta(t)$ parameter:

$$\begin{cases} H_0 = \beta(t) \nearrow (\omega \leq 0) \\ H_1 = \beta(t) \searrow (\omega > 0) \end{cases}$$

We obtained the following $2logBF_{01}$ using the Bayes Factors for the test:

Lombardia: -2.54 **Veneto**: 2.59

Confirming a **general increase** in the contagion speed in **Veneto** and a **general decrease** in **Lombardia**, while always mantaining a lower mean β in Lombardia than in Veneto (confirmed by **Mann-Whitney** test)

$\Rightarrow$ different *colors* were **justified**, but meant **different evolutions** too.

Conclusions

- We built a library implementing eABC-SMC and applied it successfully to a real-world highly complex problem.
- Bayesian framework allowed us to get such promising results with a relatively simple model, thanks to its flexibility and the possibility to exploit previous knowledge.
- We provided a significant improvement to both the original ABC-SMC and HMC frameworks, obtaining really good results in terms of performances.
- Incidentally, we provided a good and robust framework for the Emergency Response in pandemics, though it may need some refinements to be adapted to different scenarios than Covid-19.

Bibliography

Thank You

Y. Yang, M.A. Bhouri, P. Perdikaris (2020). Bayesian differential programming for robust systems identification under uncertainty. ArXiv pre-print, submitted to *Proceedings of the Royal Society A*.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., Stumpf, M. P. (2009). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31), 187-202.

Filippi S., Barnes C. P., Cornebise J., Stumpf M. P. H. (2012). On optimality of kernels for approximate Bayesian computation using sequential Monte Carlo. arXiv:1106.6280 [stat.CO]

Ianni, A., Rossi, N. Describing the COVID-19 outbreak during the lockdown: fitting modified SIR models to data. *Eur. Phys. J. Plus* 135, 885 (2020).

https://github.com/CSSEGISandData/COVID-19_Unified-Dataset

Appendix 1 - Weights for the ABC-SMC [Toni et al., 2009]

Weights used to implement ABC-SMC algorithm are the following:

For $t \neq 1$

$$\omega^{(i,t)} = \frac{\pi(\theta^{(i,t)})}{\sum_{j=1}^{n} \omega^{(j,t-1)} K_t(\theta^{(i,t)} | \theta^{(j,t-1)})}$$

Furthermore, we used a Gaussian kernel, described as follows:

$$K_t(\theta_k | \theta_k^{(j,t-1)}) = \frac{1}{\sqrt{2\pi\sigma^2_{(k,t)}}} e^{\frac{-(\theta_k - \theta_k^{(j,t-1)})^2}{2\sigma^2_{(k,t)}}}$$

Common choice in literature (Toni et al., 2009, Filippi et al. 2012)

Appendix 2 - Complete eABC-SMC Algorithm

Algorithm 3: eABC-SMC

Result: A sample from $p_\epsilon(\boldsymbol{\theta}|x)$
Initialization: A precision schedule $\{\epsilon_t\}_{t\in 1:T}$, the estimated true center C ;
Preconditioning Estimate the borders through **Algorithm 2**;
while $t \leq T$ **do**
  while $n \leq N$ **do**
    if $t = 1$ **then**
      sample $\tilde{\theta}$ from $\pi(\theta|\mathfrak{R}_{bor})$, with $\mathfrak{R}_{bor}$ being the region delimited by borders;
    else
      sample θ from the previous population $\{\theta^{(i,t-1)}\}_i$ with weights $\{\omega^{(i,t-1)}\}_i$;
      sample $\tilde{\theta}$ from $K_t(\cdot|\theta)$ s.t. $\pi(\theta) > 0$;
    end
    compute $y = f(\cdot|\tilde{\theta})$;
    if $\Delta(y, x) \leq \epsilon_t$ **then**
      save $\tilde{\theta}$ and y;
    else
  end
  compute $\{\omega^{(i,t)}\}_i$ and normalize them;
end