Comparing forecast performance on different synthetic pandemics

4/06/2024

Résumé

My abstract

1 Introduction

Here is the introduction. The simulations are available on this GitHub Repository.

2 Methods

The data

As the goal of this study is to compare some forecasts on many different pandemics, many synthetic pandemics need to be generated, with a particular attention on the diversity of these pandemics.

2.0.1 Covasim

To generate the pandemics, [5], a python librairy that can simulate the evolution of a pandemic was used. Covasim is an agent-based model that can model many different pandemics and has a high diversity of outputs. This model takes as an input many parameters such as the population type, the population size, the age repartition . . .and outputs a complete description of the pandemic, with real-time values of each relevant information, sucha as the number of severe, of asymptomatic... but also physical values such as the value of the reproduction number. Covasim enables to generate a huge diversity of pandemics, thanks to the plurality of parameters that can be given as the input of the model, but also with interventions that can be planned by the users. These interventions can simulate the impact of a vaccination campaign, with changes in the probability transmission, that can be different for all ages groups.

2.0.2 First pandemics

For the implementation and the first test of the models, two pandemics were generated. The first one focusing on the new deaths count and the second one focusing on the number of hospitalized count. Those pandemics will be referred to as pandemic 1 and pandemic 2.

The parameters used to generate these pandemics are described in the table 1 The parameters that are not specified are the default parameters of the Covasim librairy.

The different interventions were based on mobility reports from Västtrafik 1, the public transport company of the city of Gothenburg, which were reported during the Covid 19 pandemic and has been retrieved in [4]. These interventions correspond to 53 relative weekly variations of the mobility, with a reference value of 1 for the first week of the report, which correspond to the 9-th week of 2020.

2.0.3 Generating diverse pandemics

In order to evaluate the performances of our models on a wide range of pandemics, a training set of pandemics was generated. A huge diversity of pandemics is needed to determine which model is the more consistent. It is so relevant to identify the key parameters that generate this diversity. As

TABLE 1 – Table of the parameters used to simulate the firrst two pandemics.

Parameter	Pandemic 1	Pandemic 2
start day 1	2020-03-02	2020-03-02
end day	2020-07-01	2021-01-01
Population size	1000000	1000000
Interventions	interventions1	interventions2
population type	hybrid	hybrid
β initial	0.015	0.015
location	Sweden	Sweden
n infected initial	20	100

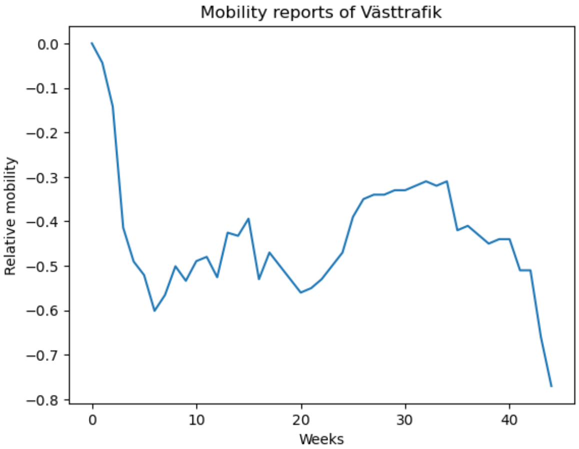


FIGURE 1 – Mobility reports from Västtrafik.

Covasim has a very huge set of inputs parameters, a first subset of key parameters was identified : the spread parameters and the severity parameters. The severity parameters are the 4 parameters that correspond to the probability for an agent to get from a compartment to another. The spread parameters are 9 parameters that represent the distribution of probability of the time spend by an agent in a compartment (such as infected, crictical...) once he entered it. This distribution is a log-normal distribution, but the spread-parameters correspond to the mean of this log-normal distribution All the parameters have a default value of 1, which correspond to keeping the reference value. We decided to select 4 parameters and to make them vary in $[0.5, 1, 2]$, leading to a set of 81 pandemics. To select the 4 parameters that generated the most diversity, different diversity metrics were computed.

Let Y_1 , $Y_2 \in \mathbb{R}^n$ be two time series of n days representing the number of hospitalized in two pandemics.

Let :

$$\mathcal{L}_1(Y_1, Y_2) = \|Y_1 - Y_2\|_{L_1}$$

$$\mathcal{L}_2(Y_1, Y_2) = \|(\frac{max(Y_1)}{max(Y_2)}; \frac{max(Y_1')}{max(Y_2')}; \frac{max(Y_1'')}{max(Y_2'')}, \|\tilde{Y_1} - \tilde{Y_2}\|_{L_1}, \|\tilde{Y_1'} - \tilde{Y_2'}\|_{L_1}, \|\tilde{Y_1''} - \tilde{Y_2''}\|_{L_1})\|_{L_2}$$

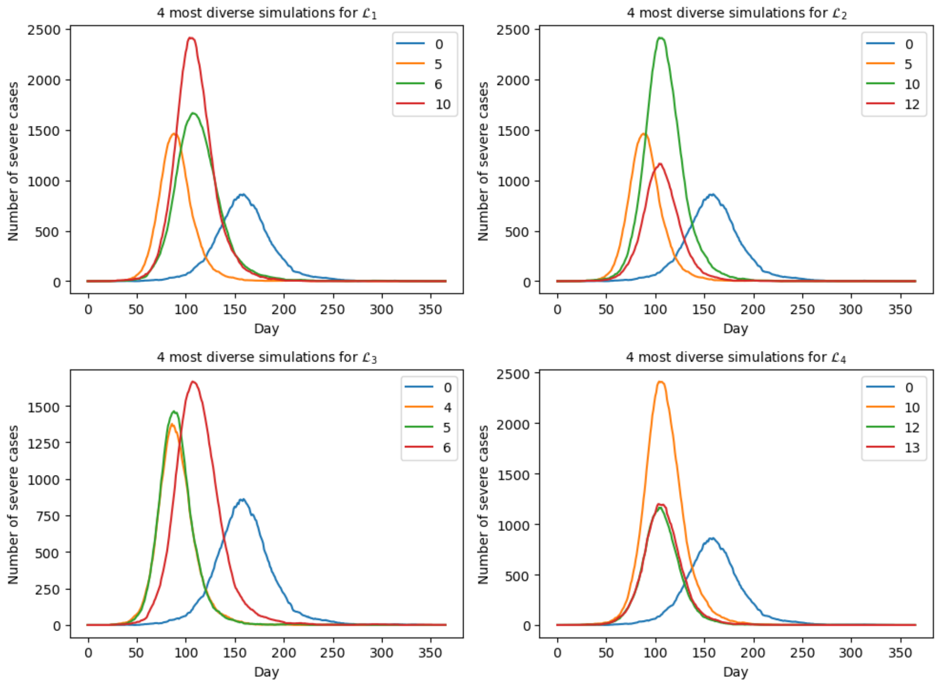
with Y' and $Y''$the first and second derivatives of$Y$

$\mathcal{L}_3 = \mathcal{W}(\tilde{Y_1} - \tilde{Y_2})$, with $\mathcal{W}$ the Wasserstein distance.

$$\mathcal{L}_4(Y_1, Y_2) = \|(\frac{max(Y_1)}{max(Y_2)}; \frac{max(Y_1')}{max(Y_2')}; \frac{max(Y_1'')}{max(Y_2'')}, \mathcal{W}(\tilde{Y_1} - \tilde{Y_2}), \mathcal{W}(\tilde{Y_1'} - \tilde{Y_2'}), \mathcal{W}(\tilde{Y_1''} - \tilde{Y_2''}))\|_{L_2}$$

with Y' and $Y''$the first and second derivatives of$Y$

It can be noted that $\mathcal{L}_2$ looks like the Sobolev norm $\|\tilde{Y_1} - \tilde{Y_2}\|_{W^{2,1}}$ with squared terms and with


4 most diverse simulations for $\mathcal{L}_1$

4 most diverse simulations for $\mathcal{L}_2$

4 most diverse simulations for $\mathcal{L}_3$

4 most diverse simulations for $\mathcal{L}_4$

FIGURE 2 – 4 most diverse pandemics according to each norm.

additionary terms taking into account the amplitude.

To determine which measure to use, we generated 14 pandemics. Each pandemic but the last one has default parameters except one of them which was doubled. The last pandemic has only default parameters.

For each norm $\mathcal{L}_k$, we determined S, the subset of 4 pandemics that maximized the following quantity :

$$\sum_{i,j \in S, i \neq j} \mathcal{L}_k(Y_i, Y_j)$$

The 4 most divers pandemics accorded to each norm are shown in the figure 2. We decided according to this figure, that $\mathcal{L}_2$ norm was the most relevant to determine the diversity of the pandemics. But, keeping the parameters `[0, 5, 10, 12]` would not be accurate, as the parameters were changed independantly, and the diversity did not take into account the correlation between some of them.

Finding the parameters that maximise the $\mathcal{L}_2$ diversity is equivalent to solve the following problem :

$S_{opt} = \underset{S' \subset S, |S'|=4}{argmax} \; \mathcal{L}(S')$, with $\mathcal{L}(S') = \sum_{s,t \in \mathcal{P}_g(S'), s \neq t} \mathcal{L}_2(s,t)$, and $\mathcal{P}_g(S')$ the set of the 81 pandemics generated with the 4 parameters of S'

However, generating a pandemic with `Covasim` is time consuming, and it is not possible to compute the diversity of each set of 4 parameters S' included in S.

A MCMC algorithm [3] was then implemented, to perform a clever grid search on the different subsets $S' \subset S$ of parameters. The MCMC algorithm is a method which is used to sample from a distribution that can't be directly sampled. The main idea is to construct a Markov Chain whose stationary distribution is the objective distribution.

Let $\mathcal{S} = \{S' \subset S; |S'| = 4\}$ be the support of the target distribution, which is, in our case, the set of all the 715 combinations of the 4 parameters among the 13 different possible, and let π be the target distribution on $\mathcal{S}$. $\forall s \in \mathcal{S}, \pi(s) = \frac{\mathcal{L}(s)}{\sum_{s \in \mathcal{S}} \mathcal{L}(s)}$. π is not directly computable as it is too time consuming to compute the denominator.

For each $s = [a, b, c, d] \in \mathcal{S}$, let $ne(s)$ be the set of the neighbourghs of s, i.e the set of all the elements of $\mathcal{S}$ who have only one parameter different from s. For instance, $[0, 3, 9, 12] \in ne([0, 3, 10, 12])$, but $[0, 3, 9, 12] \notin ne([0, 3, 8, 10])$.

Let U_n be a sequence of independent uniform random variables on $[0, 1]$ and $\forall s \in \mathcal{S}$, let $U_n^{ne(s)}$ be a sequence of independant uniform random variables on $ne(s)$. Let $s_0 \in \mathcal{S}$ and let S_n be the random sequence defined as follow :

$$\begin{cases} S_0 = s_0 \\ \forall n \in \mathbb{N}, \alpha_n = \dfrac{\mathcal{L}(U_n^{ne(S_n)})}{\mathcal{L}(\mathcal{S}_n)} \\ \forall n \in \mathbb{N}, S_{n+1} = U_n^{ne(S_n)} \mathbb{1}_{\{U_n < \alpha_n\}} + X_n \mathbb{1}_{\{U_n > \alpha_n\}} \end{cases}$$

As S_{n+1} is a function of S_n and of other independant random variables, the sequence S_n is a homogenous Markov Chain. This formula means that at each iteration, a neighbourgh of S_n is uniformly selected among all the neighbourghs of S_n (it is $U_n^{ne(S_n)}$) The Markov Chain moves to this neighbourgh if the value of $\mathcal{L}(U_n^{ne(S_n)})$ is higher than the value of the function $\mathcal{L}(S_n)$ at the current state. If the new value of $\mathcal{L}$ is smaller, the markov chain moves with a probability that is equal to the ratio of the two values. This way of moving on the different subsets prevents to be stucked in a local maxima but avoids exploring dummies areas, in which the diversity is very small.

The transition matrix of this Markov Chain is the following :

$$K(s, s') : \begin{cases} 0 \text{ if } s' \notin ne(s) \text{ and } s' \neq s \\ \dfrac{1}{Card(ne(s))} = \dfrac{1}{36} \text{ if } s' \in ne(s) \text{ and } \dfrac{\mathcal{L}(s')}{\mathcal{L}(s)} > 1 \text{ and } s' \neq s \\ \dfrac{1}{36} \times \dfrac{\mathcal{L}(s')}{\mathcal{L}(s)} \text{ if } s' \in ne(s) \text{ and } \dfrac{\mathcal{L}(s')}{\mathcal{L}(s)} \leq 1 \text{ and } s' \neq s \\ 1 - \displaystyle\sum_{s' \in \mathcal{S}, s' \neq s} K(s, s') \text{ if } s' = s \end{cases}$$

Let $(s, s') \in \mathcal{S}^2$. Let us suppose that $s' \neq s$, that $s' \in ne(s)$, and that $\mathcal{L}(s) < \mathcal{L}(s')$ (the other case is symmetric).

$$\begin{aligned} \pi(s)K(s, s') &= \frac{\mathcal{L}(s)}{\sum_{s \in \mathcal{S}} \mathcal{L}(s)} \times \frac{1}{36} \quad \text{as } \mathcal{L}(s) < \mathcal{L}(s') \\ &= \frac{\mathcal{L}(s)}{\sum_{s \in \mathcal{S}} \mathcal{L}(s)} \times \frac{1}{36} \times \frac{\mathcal{L}(s')}{\mathcal{L}(s')} \\ &= \frac{\mathcal{L}(s')}{\sum_{s \in \mathcal{S}} \mathcal{L}(s)} \times \frac{1}{36} \times \frac{\mathcal{L}(s)}{\mathcal{L}(s')} \\ &= \pi(s')K(s', s) \end{aligned}$$

Indeed, each subset s has 36 neighbourgh, as there are 13 parameters and one can replace each parameter of s by any of the 9 others.
Thus, π is **reversible** for K.
Let $(s, s') \in \mathcal{S}^2$. Let us note (a, b, c, d) and (a', b', c', d') the elements of s and s'. We note :
$s_1 = [a', b, c, d]$
$s_2 = [a', b', c, d]$

$s_3 = [a', b', c', d]$

$$\begin{aligned}
\mathbb{P}(S_{n+4} = s'|S_n = s) &\geqslant \mathbb{P}(S_{n+4} = s' \cap S_{n+3} = s_3 \cap S_{n+2} = s_2 \cap S_{n+1} = s_1|S_n = s) \\
&\geqslant \mathbb{P}(S_{n+4} = s'|S_{n+3} = s_3 \cap S_{n+2} = s_2 \cap S_{n+1} = s_1 \cap S_n = s) \\
&\qquad \times \mathbb{P}(S_{n+3} = s_3 \cap S_{n+2} = s_2 \cap S_{n+1} = s_1|S_n = s) \text{ (Baye' s Formula)} \\
&\geqslant \mathbb{P}(S_{n+4} = s'|S_{n+3} = s_3) \times \mathbb{P}(S_{n+3} = s_3|S_{n+2} = s_2 \cap S_{n+1} = s_1 \cap S_n = s) \\
&\qquad \times \mathbb{P}(S_{n+2} = s_2 \cap S_{n+1} = s_1|S_n = s) \text{ (by Markov's property)} \\
&\vdots \\
&\geqslant \mathbb{P}(S_{n+4} = s'|S_{n+3} = s_3) \times \mathbb{P}(S_{n+3} = s_3|S_{n+2} = s_2) \times \mathbb{P}(S_{n+2} = s_2|S_{n+1} = s_1) \\
&\qquad \times \mathbb{P}(S_{n+1} = s_1|S_n = s) \text{ (by Markov's property)} \\
&\geqslant (\frac{1}{36})^4 \times min(1, \frac{\mathcal{L}(s')}{\mathcal{L}(s)}) \times min(1, \frac{\mathcal{L}(s_3)}{\mathcal{L}(s_2)}) \times min(1, \frac{\mathcal{L}(s_2)}{\mathcal{L}(s_1)}) \times min(1, \frac{\mathcal{L}(s_1)}{\mathcal{L}(s)}) \\
&> 0
\end{aligned}$$

Thus, S_n is **irreducible**.

A Markov chain of transition matrix P on the support $\mathcal{S}$ is said to be aperiodic if :
$\forall s \in \mathcal{S}, \forall s' \in \mathcal{S}, \exists N \in \mathbb{N},$ s.t $\forall n > N, P(s, s')^n > 0$ [1, text]

First, note that $\forall s \in \mathcal{S}$, s is a local minimum (i.e if $\forall s' \in ne(s), \mathcal{L}(s') > \mathcal{L}(s)$) if and only if $K(s, s) = 0$

Thus, if s is not a local minimum, then $K(s, s) > 0$. Moreover, $\forall s \in \mathcal{S}, \forall s' \in ne(s)$, if $s \neq s'$, then $K(s, s') \neq 0$

Let $(s, s') \in \mathcal{S}^2$.

- If s' is not a local minimum,
  $\forall n > 4, \mathbb{P}(S_n = s'|S_0 = s) \geqslant \mathbb{P}(S_4 = s'|S_0 = s) \times K(s', s')^{n-4} > 0$
- If s' is a local minimum, $\forall s^* \in ne(s')$, s^* is not a local minimum and $K(s^*, s^*) \neq 0$.
  $\forall n > 5$, $\mathbb{P}(S_n = s'|S_0 = s) \geqslant \mathbb{P}(S_3 = s^*|S_0 = s) \times K(s^*, s^*)^{n-4} \times K(s^*, s') > 0$

Thus S_n is an **aperiodic** Markov Chain.

Finally, according to the **Theorem 5.5** from [1], as S_n is irreductible and aperiodic, as π is the stationary distribution, and as $\mathcal{S}$ is countable, S_n converges in distribution to π.

The most probable set that will be sampled by S_n is so the one that maximises the diversity. We implemented this MCMC algorithm to maximise $\mathcal{L}_2$ on $\mathcal{S}$. After 200 iterations, the set of parameters that maximised the diversity was `[0, 5, 10, 12]`. A new maxima was found at `[2, 4, 9, 10]` , which correspond to the parameters `[sym2sev, asym2rec, rel_symp_prob, rel_severe_prob]`. The $\mathcal{L}_2$- diversity increased from 62353 to 93553.

To create the most diverse set possible, we also created 4 different mobilities reports 3, corresponding to constant mobility, annual variations, lockdown scenario and the reports from Vasträffik from [4]. These time-varying mobilities enabled us to model more complex behaviours of the pandemics. We finally modelled 324 pandemics. Indeed each of the 4 parameters was scaled among 3 values : `[0.5, 1, 2]` and the 4 mobilities reports were used.

2.1 The models

In this study, we define a model h_θ as a function h defined on $\mathbb{N}$, with parameters θ and trained on the data $\mathcal{D}$. In the training phase, $\hat{\theta}$, an estimator of θ is computed form $\mathcal{D}$, and used for the prediction. We elaborated two types of models : the first type correspond to models which are only trained on the time series we want it to predict (the number of hospitalized in our case), and the second type

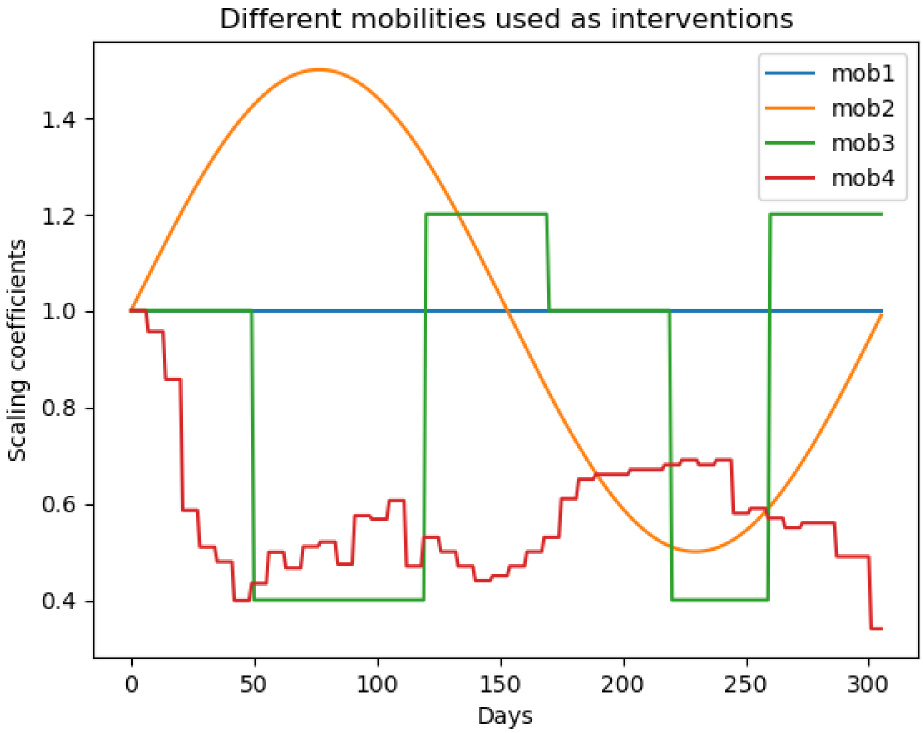


FIGURE 3 – Mobility reports.

of models are trained on the time series we want to predict, but also on other time series that can be relevant to predict the number of hospitalized (the mobility and the number of infected). All of these models were implemented in Python, and are available on the github repository provided with this article (1). During the training or predicting phase, the computation sometimes fail (for instance, when the matrix is non-inversible for the linear regression model). The model then outputs the value of the mving average model, which can be interpreted as a naive output when the computation fails.
Task of a model :

Each model h is given :

1. A training set $\mathcal{D}$
2. A reach of prediction r
3. A confidence threshold α

And outputs :

1. A prediction $\hat{Y}_r$
2. A $(1-\alpha)$ confidence interval on the prediction, $I_{\alpha,r}$

The model will train on the data $\mathcal{D}$ to compute $\hat{\theta}$ the parameter estimator and the output $\hat{Y}_r = h_{\hat{\theta}}(r)$.

2.1.1 The SIRH model

The SIRH model (4) is an extension of the classic compartemental SIR (Susceptible-Infectious-Recovered) model used to describe the spread of infectious diseases. In the SIRH model, a fourth compartment, "H" for "Hospitalized," is added. Each compartment correspond to the number of person in the state of health of the compartment. The evolution of the number of person in each compartment is described by a system of ordinary differential equations :

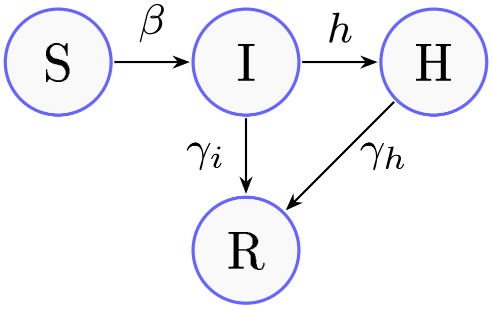


FIGURE 4 – Scheme of the SIRH model

$$\begin{cases} \frac{dS}{dt} = -\beta\frac{SI}{N} \\ \frac{dI}{dt} = \beta\frac{SI}{N} - \gamma_i I - hH \\ \frac{dR}{dt} = \gamma_i I + \gamma_h h \\ \frac{dH}{dt} = hI \end{cases} \tag{1}$$

At $t = 0$, the values of (S_0, I_0, R_0, H_0) is fixed to $(10^6 - 1, 1, 0, 0,)$. As the system of equation can't be directly solved, we use a Euler method to solve it :

$$\begin{cases} S_{t+dt} = S_t + dt\frac{dS}{dt} \\ I_{t+dt} = I_t + dt\frac{dI}{dt} \\ R_{t+dt} = R_t + dt\frac{dR}{dt} \\ H_{t+dt} = H_t + dt\frac{dH}{dt} \end{cases} \tag{2}$$

We chose to fix $dt = 0.001$.

To train this model, we minimize the least square between the curve of the number of hospitalized observed to the curve of the number of hospitalized of teh training data with respect to $\theta = (\beta, gamma_i, gamma_h, h)$. We implemented some variations of the model in which γ_i, γ_h or both were fixed to the value 0.2. In the prediction phase, a r day SIRH simulation is launched, with the parameter $\hat{\theta}$ computed during the training phase. The initial value for S and I correspond to the last value of the fit of the training phase. The initial value for H correspond to the last value of $\mathcal{D}$, the trainig data. The initial value of R is fixed by the previous values as the equation $S_t + I_y + R_t + H_t = N$ is always true. The confidence interval of the prediction is computd thanks to a linearization and the use of the delta-method (see 2.1.7)

A SIRH model of the second type was implemented. It has the same structure but uses the mobility data and the number of infected to be more precise. The idea is the same, but there are two differences :

— β varies with the time as a linear combination of the mobility : $\beta_t = a + b \times m_t$

— The data is fitted to both the number of hospitalized and the number of infected.

To write it more formally, let $H_\theta(t)$ and $I_\theta(t)$ be the number of hospitalized and infected at time t in the SIRH model with parameters θ. Let $Y_{H,t}$ and $Y_{I,t}$ be the number of hospitalized and infected at time t in the data. We have :

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^5}{\operatorname{argmin}} \sum_{t=1}^{n} \left(\frac{H_\theta(t) - Y_{H,t}}{\underset{t\in\{1,\dots,n\}}{max}(Y_{H,t})}\right)^2 + \left(\frac{I_\theta(t) - Y_{I,t}}{\underset{t\in\{1,\dots,n\}}{max}(Y_{I,t})}\right)^2$$

With $\theta = (a, c, \gamma_i, \gamma_h, h)$ and m_t the mobility at time t. The normalization factors enables to prevent the optimization to focus on the number of infected, which is bigger than the number of hospitalized. Once again, a variation of the SIRH model was implemented, in which the value of γ_h and γ_i is fixed to 0.2.

2.1.2 ARIMA and VAR models

The ARIMA and VAR models are used for time-series forecasting and have outperformed many models in pandemic prediction (see [6, text] and [10, text]) The $ARIMA(p, d, q)$ model is the sum of an $AR(p)$ and a $MA(q)$ model applied on the time series differenciated d times. It follows the equation :
$Y_t^d = \alpha + \sum_{i=1}^{p} \beta_{t-i}\, Y_{t-i}^d \, + \, \sum_{j=1}^{q} \phi_{t-j}\, \epsilon_{t-j}$
where Y_t^d is the time series at time t, d is the order of the differencing, α is a constant, p is the order of the autoregressive part, q is the order of the moving average part and ϵ_{t-j} is the difference between the prediction of the model and the real value at time $t - j$.
The coefficient are estimated through maximum likelihood estimation. This method is implemented in the `statsmodels` library, which directly provide prediction and confidence intervals. We realized a grid search on a single pandemic to identify the combinaition of parameters that would optimize the prediction accuracy. We found an optimal value for $p = 3, d = 0, q = 3$.

The VAR model is a multi-dimensional AR model, in which different variable are predicted. It so corresponds to a second type model. This model exploits the correlation between variables. The value of the parameters of a VAR have physical sense and can be interpreted to find correlations between variables. Let $Y_{1,t}, ..., Y_{k,t}$ be the times series (in our case, $k = 3$ and they correspond to the number of hospitalized, the number of infected and the mobility data).

$$VAR(p) : \begin{pmatrix} Y_{1,t} \\ Y_{2,t} \\ \vdots \\ Y_{k,t} \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{pmatrix} + \begin{pmatrix} \phi_{11,1} & \phi_{12,1} & \cdots & \phi_{1k,1} \\ \phi_{21,1} & \phi_{22,1} & \cdots & \phi_{2k,1} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{k1,1} & \phi_{k2,1} & \cdots & \phi_{kk,1} \end{pmatrix} \begin{pmatrix} Y_{1,t-1} \\ Y_{2,t-1} \\ \vdots \\ Y_{k,t-1} \end{pmatrix} + \cdots + \begin{pmatrix} \phi_{11,p} & \phi_{12,p} & \cdots & \phi_{1k,p} \\ \phi_{21,p} & \phi_{22,p} & \cdots & \phi_{2k,p} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{k1,p} & \phi_{k2,p} & \cdots & \phi_{kk,p} \end{pmatrix} \begin{pmatrix} Y_{1,t-p} \\ Y_{2,t-p} \\ \vdots \\ Y_{k,t-p} \end{pmatrix} + \begin{pmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \\ \vdots \\ \epsilon_{k,t} \end{pmatrix}$$

Again, the $\phi_{i,j,k}$ and c_i are estimated through maximum likelihood estimation with the `statsmodel` librairy. The confidence intervals are also directy provided by the librairy.

2.1.3 The moving average model

A mere moving average model was also implemented. It returns a constant prediction that correspond to the mean of the 7 past days. The confidence intervals are computed by assuming that the predictions follow a normal distribution of variance equal to the variance of the 7 past data-points ; This model is used as a baseline. A model that does not manage to outperform the moving average model would not be relevant.

2.1.4 Exponential regression

An exponential regression model was implemented. It correspond to fitting the data of the number of hospitalized (Y_t) to the function $E_{a,b,c}(t) = a \times e^{bt} + c$. The value of $\theta = (a, b, c)$ is computed through a least square method. The confidence interval on the prediction is estimated with the same method as SIRH (see 2.1.7).

An exponential regression of the second type was also implemented :
The data of the number of hospitalized is fitted to the function $E_{a,b,c,d,e}(t) = a \times e^{bm_{t-i} + ct + dinf_{t-j}} + e$. The value of $\theta = (a, b, c, d, e)$ is computed through a least square method. The optimal value of the time lag i and j is optimized during the training phase through a grid search among all the values between 0 and 14. The confidence interval on the prediction is estimated with the same method as SIRH (see 2.1.7).

2.1.5 Machine learning models

In order to implement machine learning regressor, we converted the time-series $Y_{t, t \in 1, \vdots n}$ in a training set (X_i, Y_i) such that :

$\forall i \in \{1, ..., n\}, X_i = (Y_{i-1}, Y_{i-2}, ..., Y_{i-20})$.

We then trained and optimized both regressors : the linear and the bayesain regression, which were the only one that did not output absurd results on the `scikit-learn` models among : linear regression, bayesian regression, Ridge, Gradient boosting regressor, Random Forest regressor, Bayesain Ridge and SVR. The confidence interval on linear regression was computed with the same method as SIRH (see 2.1.7). The confidence interval on bayesian regression was directly computed with the variance of the parameters given by the scikit-learn librairy and the delta method.

2.1.6 Ensemble model

It has been showned ([2] and TODO add other refs) that ensemble models, which combine the outputs of manu modles, can outperform by far individual models. We implemented an ensemble model which is a linear combinaition of the outputs of the 13 models described above. The weights of this model were found my minimizing the least-squared error between the prediction of the ensemble model and the real value of the number of hospitalized on a train set of approximately 80As the ensemble model only outputs a single value without confidence intervals, it is only evaluated with the RMSE metric.

2.1.7 Computing confidence intervals on the prediction

Assumption :

We suppose that the data of the pandemic observed follows the model h, of parameter $\theta^* \in \mathbb{R}^d$. Let Y_i, $i = 1, \ldots, n$ be the number of hospitalized at each day. We suppose that : $Y_i = h_{\theta^*}(i) + \epsilon_i$, with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, iid, and independent from all the other variables. The objective is to estimate θ^*. We use $\hat{\theta}$, the least square estimator of θ^* as an estimator of θ^* :

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{i=1}^{n} (Y_i - h_\theta(i))^2$$

Let :

$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

$$h_\theta = \begin{pmatrix} h_\theta(1) \\ \vdots \\ h_\theta(n) \end{pmatrix}$$

We have :

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^d}{\operatorname{argmin}} \|Y - h_\theta\|^2$$

Now, if θ is close enough to θ^*, we can write (from [9]) :

$$\forall i \in \{1, ..., n\} : h_\theta(i) = h_{\theta^*}(i) + (\theta - \theta^*)^T \nabla_\theta h_{\theta^*}(i)$$

which leads to :

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^d}{\operatorname{argmin}} \left\|Y - h_{\theta^*} - (\theta - \theta^*)^T \nabla_\theta h_{\theta^*}\right\|^2$$

Let us define :

$$\tilde{Y} = Y - h_{\theta^*}$$
$$\beta = \theta - \theta^*$$
$$\hat{\beta} = \theta - \hat{\theta}$$

and let us define the matrix $A \in \mathbb{R}^{n \times d}$ such that $\forall i \in \{1, ..., n\}, \forall j \in \{1, ..., d\}, A_{i,j} = \frac{dh_{\theta^*}}{d\theta_j}(i)$.
The previous problem can be re-written as :

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^d}{\operatorname{argmin}} \left\| \tilde{Y} - A\beta \right\|^2$$

This is a regression linear problem.
Let us solve this problem in the general case.
Let $(A_i, \tilde{Y}_i)$ be the observations Let $\mathbb{P}$ be the law from which the A_i are drawn, and let us assume that $Y_i = A_i\beta^* + \epsilon_i$, with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.
The solution of this problem is explicitly (from [8]) :

$$\hat{\beta} = (A^T A)^{-1} A^T \tilde{Y}$$

This least-square estimator is unbiased :

$$\mathbb{E}[\hat{\beta}] = \beta^*$$

$$\hat{\beta} = \left(\sum_{i=1}^{n} A_i^T A_i \right)^{-1} \times \left(\sum_{i=1}^{n} A_i^T \tilde{Y}_i \right)$$

$$\hat{\beta} = \frac{n}{n} \left(\sum_{i=1}^{n} A_i^T A_i \right)^{-1} \times \left(\sum_{i=1}^{n} A_i^T \tilde{Y}_i \right)$$

$$\hat{\beta} = \left(\frac{1}{n} \sum_{i=1}^{n} A_i^T A_i \right)^{-1} \times \left(\frac{1}{n} \sum_{i=1}^{n} A_i^T \tilde{Y}_i \right)$$

Let us denote :

$$\hat{D} = \frac{1}{n} \sum_{i=1}^{n} A_i^T A_i, \quad \text{and} \quad \hat{\delta} = \left(\frac{1}{n} \sum_{i=1}^{n} A_i^T \tilde{Y}_i \right)$$

We have :

$$\hat{\beta} = \hat{D}^{-1} \hat{\delta}$$

$$\hat{D} \underset{a.s}{\rightarrow} D = \mathbb{E}[A_i^T A_i]$$

$$\hat{\delta} \underset{a.s}{\rightarrow} \delta = \mathbb{E}[A_i^T \tilde{Y}_i]$$

$\hat{\beta} = \hat{D}^{-1} \hat{\delta} \underset{a.s}{\rightarrow} D^{-1} \delta$, as the following function ϕ is continuous :

$$\phi : \left\{ \begin{array}{rcl} \mathcal{GL}_n(\mathbb{R}) & \rightarrow & \mathcal{GL}_n(\mathbb{R}) \\ A & \mapsto & A^{-1} \end{array} \right.$$

Now, let us show that $\hat{\beta}$ is asymptotically normal :

$$\begin{aligned}
\sqrt{n}(\hat{\beta} - \beta^*) &= \sqrt{n}(\hat{D}^{-1}\hat{\delta} - \beta^*) \\
&= \sqrt{n}(\hat{D}^{-1}\hat{\delta} - \hat{D}^{-1}\hat{D}\beta^*) \\
&= \sqrt{n}\hat{D}^{-1}(\hat{\delta} - \hat{D}\beta^*) \\
&= \sqrt{n}\hat{D}^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} A_i^T\tilde{Y}_i - \frac{1}{n}\sum_{i=1}^{n} A_i^T A_i\beta^*\right) \\
&= \frac{\sqrt{n}}{n}\hat{D}^{-1}\left(\sum_{i=1}^{n} A_i^T(\tilde{Y}_i - A_i\beta^*)\right) \\
&= \frac{1}{\sqrt{n}}\hat{D}^{-1}\left(\sum_{i=1}^{n} A_i^T\epsilon_i\right)
\end{aligned}$$

This line is made of two terms. Let's show that each one of them converges in law.

$$\begin{aligned}
\frac{1}{\sqrt{n}}\left(\sum_{i=1}^{n} A_i^T\epsilon'_i\right) &= \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} A_i^T\epsilon'_i\right) \\
&= \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} A_i^T\epsilon'_i - 0\right) \\
&\xrightarrow{\mathcal{L}} \mathcal{N}(0, \mathrm{Var}(A_i^T\epsilon_i))
\end{aligned}$$

Yet, as ϵ_i and A_i are independant, and $\mathbb{E}[A_i^T\epsilon'_i] = 0$, $\mathrm{Var}(A_i^T\epsilon_i) = \mathbb{E}[A_iA_i^T\epsilon_i^2] = \mathbb{E}[A_iA_i^T]\sigma'^2$.
Finally, $\frac{1}{\sqrt{n}}\left(\sum_{i=1}^{n} A_i^T\epsilon'_i\right) \xrightarrow{\mathcal{L}} \mathcal{N}(0, D\sigma'^2)$.
On the other hand, $\hat{D}^{-1} \xrightarrow{\mathcal{L}} D^{-1}$, which is constant.
Finally, with Slutsky, we obtain that :

$$\begin{aligned}
\sqrt{n}(\hat{\beta} - \beta^*) &\xrightarrow{\mathcal{L}} D^{-1}\mathcal{N}(0, D\sigma'^2) \\
&\xrightarrow{\mathcal{L}} \mathcal{N}(0, D^{-1}(D\sigma'^2)(D^{-1})^T) \\
&\xrightarrow{\mathcal{L}} \mathcal{N}(0, D^{-1}D\sigma'^2(D^{-1})^T) \\
&\xrightarrow{\mathcal{L}} \mathcal{N}(0, \sigma'^2 D^{-1}) \\
&\xrightarrow{\mathcal{L}} \mathcal{N}(0, \sigma'^2(A^TA)^{-1})
\end{aligned}$$

Let's get back to the first problem :
As $\beta^* = 0$ and $\hat{\beta} = \hat{\theta} - \theta^*$, we have :

$$\sqrt{n}(\hat{\theta} - \theta^*) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \sigma^2(A^TA)^{-1})$$

and,

$$\hat{\theta} \sim \mathcal{N}(\theta^*, \frac{\sigma^2}{n}(A^TA)^{-1})$$

As a first conclusion, we have that $\hat{\theta}$ is asymptotically normal.
Let Σ be the covariance matrix estimated from the computation of $\hat{\theta}$. In our case, $\Sigma = \frac{\sigma^2}{n}(A^TA)^{-1}$.
As $\hat{\theta}$ is asymptotically normal, we can apply the delta-method :

$$\sqrt{n}(\hat{\theta} - \theta^*) \xrightarrow{\mathcal{L}} \mathcal{N}(0,\ \Sigma)$$

$$\sqrt{n}(h_{\hat{\theta}} - h_{\theta^*}) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \nabla_\theta h_\theta^T \Sigma \nabla_\theta h_\theta)$$

And finally :

$$h_{\hat{\theta}} \rightarrow \mathcal{N}(h_{\theta^*}, \frac{1}{n} \nabla_\theta h_\theta^T \Sigma \nabla_\theta h_\theta)$$

By estimating $\frac{1}{n}\Sigma$ from `curve_fit`, we can compute the confidence interval of the prediction with the quantiles of the normal distribution. The gradient of h_θ is approximated through numerical approximation :
$\nabla_\theta h_\theta[i] \simeq \frac{h_{\theta+d\theta_i} - h_\theta}{dt}$, with $dt = 0.0001$.

$$d\theta_i = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{i-1} \\ \theta_i + dt \\ \theta_{i+1} \\ \vdots \theta_n \end{pmatrix}$$

Assessing the performance of the models

Metrics

Two metrucs were used to assess the performance of the models. The first metric is the Weighted Confidence Interval (WIS), which is a metric commonly used in forecast evaluation (see [2, text] or [7]).

Let α be in $]0,1[$. Let $\hat{y}$ be the prediction of the model and y the real value. Let $[l,u]$ be the $(1-\alpha)$ confidence interval of the prediction. We define the Interval Score (IS) that way :
$IS_\alpha([l,u],\hat{y},y) = \frac{2}{\alpha} \times (\mathbb{1}_{\{y<l\}}(l-y) + \mathbb{1}_{\{y>u\}}(y-u) + (u-l))$. This metric is made of three terms : a term of overprediction that punishes a model predicting a confidence interval which os above the real value, a term of underprediction that punishes a model whose confidence interval is under the real prediction, and a term of range, that punishes too wide confidence intervals. Let $(\alpha_k)_{k\in\{1,\dots,K\}} \in]0,1[^K$
The WIS is defined as follow.

$WIS_\alpha([l,u],\hat{y},y) = \sum_{k=0}^{K} w_k IS_{\alpha_k}([l,u],\hat{y},y)$, with $(w_k)_{k\in\{1,\vdots,K\}} \in \mathbb{R}_+^K$ weights chosen by the user.

According to previous literature ([2]), we decided to set $(\alpha_k)_{k\in\{1,\vdots,K\}} = [0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$.
One can notice that the WIS does not take into account the point prediction, but focus on confidence interval accuracy.

The second metric chosen is the Root Mean Square Error $(RMSE)$. With the same notations as above, we define the $RMSE$ as follow :
$RMSE([l,u],\hat{y},y) = \sqrt{(y-\hat{y})^2}$
This metric focuses on the point prediction, and does not take into account the confidence intervals.