

## Research Article

# COVID-19: Metaheuristic Optimization-Based Forecast Method on Time-Dependent Bootstrapped Data

Livio Fenga <sup>1</sup> and Carlo Del Castello <sup>2</sup>

<sup>1</sup>Italian National Institute of Statistics ISTAT, Rome 00184, Italy

<sup>2</sup>Kantar Group Ltd., Bologna, Italy

Correspondence should be addressed to Livio Fenga; [fenga@istat.it](mailto:fenga@istat.it)

Received 13 April 2020; Revised 7 December 2020; Accepted 24 January 2021; Published 11 February 2021

Academic Editor: Ramón M. Rodríguez-Dagnino

Copyright © 2021 Livio Fenga and Carlo Del Castello. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A compounded method—exploiting the searching capabilities of an operation research algorithm and the power of bootstrap techniques—is presented. The resulting algorithm has been successfully tested to predict the turning point reached by the epidemic curve followed by the COVID-19 virus in Italy. Future lines of research, which include the generalization of the method to a broad set of distribution, will be finally given.

## 1. Introduction

In general, predicting the time of a peak conditional to a set of time-dependent data is a nontrivial task. Often carried out in a multitasking fashion, requiring the availability of time and resources, the correct estimation of future turning points can be important in many instances but becomes crucial in the case of epidemic events. These are the typical circumstances when the forecasting exercise is conducted online and on a time series exhibiting a small sample size. However, under these conditions, the problem might become particularly complicated since statistical methods usually employed for these purposes—for example, of the type hidden Markov (e.g., Hamilton [1] and Koskinen and Oller [2]) or nonparametric (e.g., Delgado and Hidalgo [3]) models—not only are very demanding in terms of building and tuning procedures but typically requiring the availability of a “long” stretch of data. In addition to that, the time series related to epidemics usually show highly nonlinear dynamics, which, if not preprocessed, make them not suitable for standard linear models. On the other hand, attempting to fit nonlinear models, e.g., of the type self-exciting threshold autoregressive (for example, [4]) or the artificial neural network [1, 5], is not a viable option, due to the above-mentioned sample size issues. In any case, when an ill-tuned

model is fitted on a time series, reliable outcomes should not be reasonably expected. Therefore, an approach able to perform under the above-outlined conditions is proposed. In essence, the problem is solved by building a unified framework in which two powerful techniques—belonging to two different branches of computational statistics—are sequentially employed to lower the amount of uncertainty embedded in the observed data and to find a (possibly global) optimum through which the “best” statistical distribution for the dataset at hand is found.

## 2. The Unified Framework

As above stated, the approach studied in this study is rooted in a unified framework in which two powerful paradigms are exploited. The first one, which belongs to the so-called computer intensive statistical methods, is the bootstrap, which will be detailed in Section 4. By using this technique, a high number of “bona fide” replications of the original data are generated. In essence, each of the bootstrap series obtained “mimics” the observations recorded, so that the number of series observed—which in real life is typically equal to one—becomes (very) high. Repeating a mathematical operation (e.g., the computation of an estimator)  $B$  times makes possible (i) the assessment of the degree of uncertainty associated with the obtained

estimations and (ii) less-biased estimators. The latter goal is achievable by the design of the bootstrap method, as through its replications, the use of central tendency functions, such as mean or median, is possible. The second tool employed is an optimization method for the selection of the “best” parametrization of a class of statistical distribution commonly used in the literature. In practice, this step is performed in the so-called bootstrap world, meaning that it is sequentially repeated for each bootstrap sample. By doing so, the degree of uncertainty associated with the selected distribution is lower than the one obtainable by processing just one set of data (the real observations).

### 3. Data and Contagion Indicator

This study employs the data related to COVID-19, collected and regularly updated by the Italian National Institute of Health (an agency of the Italian Ministry of Health) and by the Italian Civil Protection Department. The whole dataset is freely and publicly available in a comprehensive database, accessible on the Internet at the web address <https://github.com/pcm-dpc/COVID-19/tree/master/dati-regioni>. The dataset includes 38 daily data points collected at national level during the period starting from January 19 to March 27. The used indicator—which will be referred to as the variable of interest—is obtained by subtracting, for each day, from the total number of people tested positive of corona virus both the number of the deaths and of the recovered.

### 4. The Resampling Method

The choice of the most appropriate resampling method is far from being an easy task, especially when the identical and independent distribution (i.i.d.) assumption (Efron’s initial bootstrap method) is violated. Under dependence structures embedded in the data, simple sampling with replacement has been proved (for example, Carlstein [6]) to yield suboptimal results. As a matter of fact, i.i.d.-based bootstrap schemes are not designed to capture and therefore replicate dependence structures. This is especially true under the actual conditions (small sample sizes and strong nonlinearity). In such cases, selecting the “right” resampling scheme becomes a particularly challenging task as many resampling schemes are not designed to capture the dynamics typically found in epidemiology. As an example, the well-known resampling method called sieve bootstrap—introduced by Buhlmann [7]—cannot be employed due to the quadratic shape almost always found in this type of time series.

In more details, while in the classic bootstrap, an ensemble  $\Omega$  represents the population of reference the observed time series is drawn from, and in MEB, a large number of ensembles (subsets), say  $\{\omega_1, \dots, \omega_N\}$ , become the elements belonging to  $\Omega$ , each of them containing a large number of replicates  $\{x_1, \dots, x_j\}$ . Perhaps, the most important characteristic of the MEB algorithm is that its design guarantees the inference process to satisfy the ergodic theorem. Formally, denoting by the symbol  $|\cdot|$  the cardinality function (counting function) of a given ensemble of time series  $\{x_i \in \omega_i; i = 1, \dots, N\}$ , the MEB procedure generates a set

of disjoint subsets  $\Omega_N \equiv \omega_1 \cap \omega_1 \dots \cap \omega_N$  s.t.  $E\Omega_N \bar{\mu}(x_t)$ , with  $\mu(\cdot)$  being the sample mean. Furthermore, basic shape and probabilistic structure (dependency) are guaranteed to be retained  $\forall x_{t,j}^* \in \omega_i \subset \Omega$ .

MEB resampling scheme has not negligible advantages over many of the available bootstrap methods; it does not require complicated tune-up procedures (unavoidable, for example, in the case of resampling methods of the type block bootstrap), and it is effective under nonstationarity. The MEB method relies on the entropy theory and the related concept of (un)informativeness of a system. In particular, the maximum entropy of a given density  $\delta f(x)$  is chosen so that the expectation of the Shannon information  $H = E(-\log \delta o(x))$  is maximized, i.e.,

$$\max_{\delta(x)} H = E(-\log \delta(x)). \quad (1)$$

Under mass- and mean-preserving constraints, this resampling scheme generates an ensemble of time series from a density function satisfying (4). Technically, the MEB algorithm can be broken down into the 8 steps detailed as follows:

- (1) A sorting matrix of dimension  $T \times 2$ , say  $S_1$ , accommodates in its first column the time series of interest  $x_t$  and an index set, i.e.,  $I_{\text{ind}} = \{2, 3, \dots, T\}$ , in the other one.
- (2)  $S_1$  is sorted according to the numbers placed in the first column. As a result, the order statistics  $x_{(t)}$  and the vector  $I_{\text{ord}}$  of sorted  $I_{\text{ind}}$  are generated and, respectively, placed in the first column and second column.
- (3) “Intermediate points,” averaging over successive order statistics, are computed, i.e.,  $c_t = ((x_{(t)} + x_{(t+1)})/2)$ ,  $t = 1, \dots, T-1$ , and intervals  $I_t$  constructed on  $c_t$  and  $r_t$  are derived, using ad hoc weights obtained by solving the following set of equations:
  - (i)  $f(x) = (1/r_1)\exp([x - c_1]/r_1)$ ,  
where  $x \in I_1$ ,  $r_1 = (3x_{(1)}/4) + (x_{(2)}/4)$ .
  - (ii)  $f(x) = (1/c_k - c_{k-1})$ ,  $x \in (c_k; c_{k+1})$ ,  
 $r_k = (x_{(k-1)}/4) + (x_{(k)}/2) + (x_{(k+1)}/4)$ ,  $k = 1, \dots, T-1$ .
  - (iii)  $f(x) = (1/r_T)\exp([c_{T-1} - x]/r_T)$ ,  
where  $x \in I_T$ ,  $r_T = (x_{(T-1)}/4) + (3x_T/4)$ .
- (4) From a uniform distribution in  $[0, 1]$ ,  $T$  pseudo-random numbers are generated and the interval  $R_t = (t/T; t+1/T)$  for  $t=0, 1, \dots, T-1$  is derived, in which each  $p_j$  falls.
- (5) A matching between  $R_t$  and  $I_t$  is created according to the following equations:

$$\begin{aligned} x_{j,t,\text{me}} &= c_{T-1} - |\theta| \ln(1 - p_j), & \text{if } p_j \in R_0, \\ x_{j,t,\text{me}} &= c_{T-1} - |\theta| \ln(1 - p_j), & \text{if } p_j \in R_{T-1}. \end{aligned} \quad (2)$$

so that a set of  $T$  values  $\{x_j, t\}$  as the  $j^{\text{th}}$  resample is obtained. Here,  $\theta$  is the mean of the standard exponential distribution.

- (6) A new  $T \times 2$  sorting matrix  $S_2$  is defined, and the  $T$  members of the set  $\{x_{j,t}\}$  for the  $j^{\text{th}}$  resample obtained in Step 5 is reordered in an increasing order of magnitude and placed in column 1. The sorted  $I_{\text{ord}}$  values (Step 2) are placed in column 2 of  $S_2$ .
- (7) Matrix  $S_2$  is sorted according to the second column so that the order  $\{1, 2, \dots, T\}$  is then restored. The jointly sorted elements of column 1 are denoted by  $\{x_{S,j,t}\}$ , where  $S$  recalls the sorting step.
- (8) Steps 1–7 are repeated a large number of times.

In order to give a clearer picture of the MEBOOT algorithm, in Figure 1, its flowchart is reported. As it can be noticed, four different functions characterize this resampling method. The sorting function plays a key role, as it operates in two different points of the algorithm, i.e., to order the values belonging to the original time series ( $S_1$ ) and to restore the given time sequence for each of the bootstrapped data ( $S_2$ ). Besides the pseudorandom function generator, whose employment is straightforward, the two remaining functions, i.e., the average and the matching, are, respectively, used to compute the mean of the maximum entropy density and to create a matched sequence of the intervals  $R_t$ 's.

## 5. Bootstrap-Driven Forecast Optimization

This section aims to define an alternative method to forecast our variable of interest by means of an optimization approach designed to fit a set of distribution functions applied to each bootstrap replication.

The variable of interest is assumed to approximately describe a logistic function, scaled by a normalizing parameter  $h$  (representing the asymptotic number of cases), as shown in Figure 2, so that its derivative is a Gaussian function rescaled accordingly, i.e.,

$$f(t|h, \mu, \sigma) = h \frac{1}{\sigma \sqrt{2\pi}} e^{-(1/2)(t-\mu/\sigma)^2}, \quad (3)$$

where  $t$  represents the number of days since pandemic has started in Italy,  $h$  represents the magnitude scale,  $\mu$  represents the peak of daily cases (scale), and  $\sigma$  represents the standard deviation (shape).

Now, given

- (i) The parameter vector  $\theta = (h, \mu, \sigma)$ , where  $\theta \in \Theta$
- (ii) The total active cases  $x_t$  since the infection spread
- (iii) A generic bootstrap distribution  $x_{t,i}^* \in \omega \subset \Omega$ , where  $i \in \{1..N\}$  is the  $i^{\text{th}}$  bootstrap within  $N$  replicates
- (iv)  $\hat{x}_{t,i}^* = f(t|h, \mu, \sigma)$ , where  $\hat{x}_{t,i}^*$  is the theoretical value; the objective function is defined as follows:

$$\begin{aligned} \forall i \in \{1..N\} \min_{\theta_i \in \Theta} \text{MSE}_i(x_{t,i}^*, \hat{x}_{t,i}^*) \\ = \min_{\theta_i \in \Theta} \text{Avg}_i(x_{t,i}^* - \hat{x}_{t,i}^*)^2. \end{aligned} \quad (4)$$

This is a nonlinear unconstrained optimization problem which cannot be addressed using standard global optimization

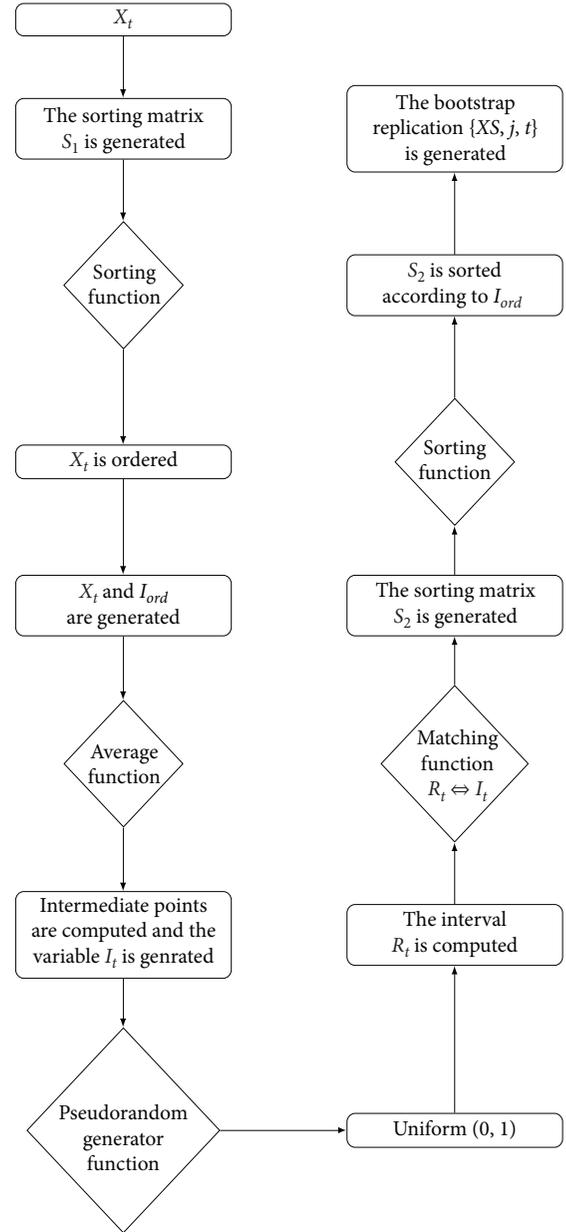


FIGURE 1: Flowchart of the MEBOOT algorithm.

methods (e.g., of the type simplex, branch and bound, or branch and cut), which are designed for linear programming (LP) [8] and mixed-integer linear programming (MILP) [9], within the field of discrete combinatorial problems [10].

On the other hand, local search simulated annealing metaheuristic designed to approximate global optimization can be used to solve unconstrained nonlinear problems in a large space.

## 6. Simulated Annealing Optimization

Simulated annealing (SA), following Van Laarhoven and Aarts [11], is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic used to approximate global optimization in a large search space for an optimization problem.

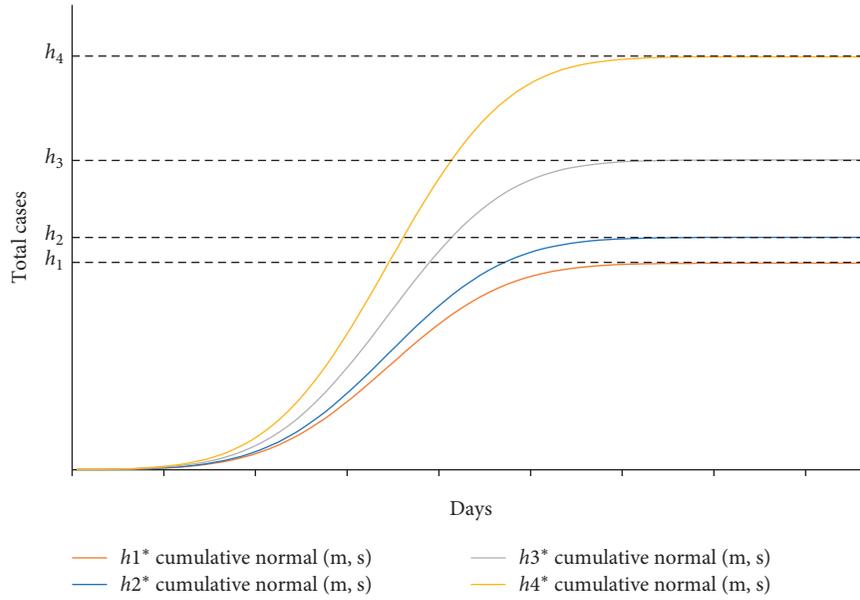


FIGURE 2: COVID-19 active cases.

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. Both are attributes of the material that depend on its thermodynamic free energy. Heating and cooling the material affects both the temperature and the thermodynamic free energy. Simulated annealing can be used to approximate the global minimum for a function with many variables. This approach was used by Kirkpatrick et al. [12] to solve the traveling salesman problem. They also proposed its current name, simulated annealing. The notion of slow cooling implemented in the simulated annealing algorithm is interpreted as a slow decrease in the probability of accepting worse solutions as the solution space is explored. Accepting worse solutions is a fundamental property of metaheuristics because it allows for a more extensive search for the global optimal solution.

In general, simulated annealing algorithms work as explained next. The temperature progressively decreases from an initial positive value to zero. At each time step, the algorithm randomly selects some neighbor state  $s^*$  of the current state  $s$ , measures its energy (in this case, the  $MSE_i(x^*_{t,i})$  on the bootstrap distribution), and decides between moving the system to the state  $s^*$  or staying in state  $s$  according to the temperature-dependent probability of selecting better or worse solutions. During the searching process, such a probability, respectively, can remain at 1 (or any positive number smaller than 1) or decrease towards zero.

**6.1. Simulated Annealing on Bootstrap Pseudocode.** The following pseudocode presents the simulated annealing heuristic applied to bootstrap replicates. For each bootstrap, it starts from a state  $s_0$  and continues searching solutions until temperature decay reaches a low temperature. In the process, the call  $\text{Neighbour}(s, \varphi)$  generates a randomly chosen neighbor of a given state  $s$ ; the call  $\text{Random } U(0, 1)$  picks and returns a value in the range  $[0, 1]$ , uniformly at random. The annealing schedule is defined by the temperature decay based on the fixed cooling rate  $\rho$ .

- (i) Let the current temperature set at  $T = t_0$
- (ii) Set the cooling rate  $\rho$
- (iii) For each bootstrap series  $i$  in  $\{1, \dots, N\}$ 
  - (i) Let current solution  $s = s_0$
  - (ii) Loop while temperature  $T > 1$ 
    - (i) Pick a random neighbor,  $s_{new} \leftarrow \text{Neighbor}(s, \varphi)$ , where  $\varphi$  is the radius around  $s$
    - (ii) If  $\text{Prob}(E(s), E(s_{new}) | T, k_B) \geq \text{Random } U(0, 1)$ :  $s \leftarrow s_{new}$
    - (iii)  $T \leftarrow T * (1 - \rho)$
  - (iii) Output the final state  $s$  on  $i^{\text{th}}$  bootstrap

Here, the function  $\text{Prob}(E(s), E(s_{new}) | T, k_B)$  is the acceptance probability at each iteration given temperature  $T$  and Boltzmann constant (Aarts and Korst [13]  $k_B$ ), i.e.,

$$\text{Prob}(E(s), E(s_{new}) | T, k_B) = \begin{cases} 1, & \text{if } E(s) > E(s_{new}), \\ \exp\left(\frac{E(s) - E(s_{new})}{k_B T}\right), & \text{otherwise.} \end{cases} \quad (5)$$

TABLE 1: COVID-19 bootstrap (AsymptoteClass, Peekday) frequency.

AsymptoteClass ±2500 Median point	Peekday												Total
	34	35	36	37	38	39	40	41	42	43	44	45	
100,000	13	19	3	8	3	—	—	1	—	—	—	—	47
102,500	22	27	6	5	3	—	—	—	—	—	—	—	63
105,000	18	13	12	6	3	1	2	—	—	—	—	—	55
107,500	16	5	6	4	—	—	—	—	—	—	—	—	31
110,000	31	10	6	1	1	2	1	—	—	—	—	—	52
112,500	10	3	7	2	1	—	1	—	—	—	—	—	24
115,000	23	7	4	1	1	1	—	—	—	—	—	—	37
117,500	11	1	2	3	1	—	—	—	—	—	—	—	18
120,000	6	2	4	1	3	—	—	—	—	—	—	—	16
122,500	3	—	—	3	—	2	—	—	—	—	—	—	8
125,000	3	—	—	2	3	3	2	—	—	—	1	—	14
127,500	—	—	—	—	1	—	1	—	—	—	—	—	2
130,000	1	—	—	1	—	1	1	—	—	—	—	—	4
132,500	—	—	—	1	1	3	—	1	1	—	—	—	7
135,000	—	—	—	2	2	1	1	1	—	—	—	—	7
137,500	—	—	—	—	1	3	1	1	—	—	—	—	6
140,000	1	—	—	1	—	1	2	1	—	1	—	—	7
142,500	—	—	—	—	—	1	—	—	—	—	—	—	1
145,000	—	—	—	—	—	—	2	—	—	—	—	—	2
147,500	—	—	—	—	1	—	—	—	—	—	—	—	1
150,000	—	—	—	—	—	—	1	6	—	—	—	1	8
155,000	—	—	—	—	—	1	—	—	1	—	—	—	2
157,500	—	—	—	—	—	1	1	1	1	—	—	—	4
160,000	—	—	—	—	—	—	2	1	—	—	—	1	4
162,500	—	—	—	—	—	1	2	3	2	—	1	—	9
165,000	—	—	—	—	—	—	2	4	3	—	1	—	10
167,500	—	—	—	—	—	1	3	3	4	—	—	—	11
170,000	—	—	—	—	—	—	4	6	3	3	2	—	18
172,500	—	—	—	—	—	—	—	6	9	2	2	—	19
175,000	—	—	—	—	—	1	1	9	—	1	1	—	13
Total	158	87	50	41	25	24	30	44	24	7	8	2	500

The employed criterion for the acceptance of bad solutions has been preferred to the Boltzmann criterion if  $\text{Prob}(E(s), E(s_{\text{new}})|T)$  (as suggested by the referee), since it exhibited a much higher flexibility.

### 7. Empirical Evidence

In order to improve local search speed, the parameter space  $\Theta$  can be bounded to  $\Theta^0 \subset \Theta$ , so that useless tails have been removed. Moreover, no information is lost under parameters space reduced to

$$\begin{aligned} h &\in [100000, 180000], \\ \mu &\in [35, 50], \\ \sigma &\in [7, 13]. \end{aligned} \tag{6}$$

The SA parameters have been iteratively tuned on a trial and error basis, to maximize the procedure overall efficiency. To this end, we set the initial temperature  $T_0=10000$ , the cooling rate  $\rho=0.0006$ , the Boltzmann constant  $k_B=100$ , and the radius  $\varphi=0.3 (\theta_{\text{max}}-\theta_{\text{min}})$ . The optimization procedure has been applied to 500 bootstrap replications, derived from the positive cases in Italy between February 19 and March 27. The related results have been used to create

the frequency distribution upon magnitude scales and peaks pairs, as shown in Table 1 and Figure 3.

The average values, computed for each parameter, are as follows:  $\text{Avg}(h) = 122178$ ,  $\text{Avg}(\mu) = 36.7$ , and  $\text{Avg}(\sigma) = 10.8$ .

Approximated confidence bounds are derived from the normal distribution (with 0.01 significance level) for each of the parameters, i.e.,

$$\begin{aligned} \text{Pr}(119401 \leq h \leq 124955) &= 0.99, \\ \text{Pr}(36.41 \leq \mu \leq 37.07) &= 0.99, \\ \text{Pr}(10.6 \leq \sigma \leq 10.9) &= 0.99. \end{aligned} \tag{7}$$

By evaluating  $\mu$  confidence intervals, a peak day of daily cases between March 25 and 26 is inferred, while the  $h$  magnitude parameter shows an asymptote in the curve of the total positive cases between 120000 and 125000. The new cases curve has an asymptotic behavior, so cutting the tail beyond the 0.1 cutoff for new infections, the pandemic time window is hypothetically over after May 16.

This behavior is clearly described in Figures 4 and 5, which have been built considering the Gaussian (Figure 4) and the cumulated Gaussian (Figure 5) curves around the 99% confidence lower and upper bounds for each parameter.

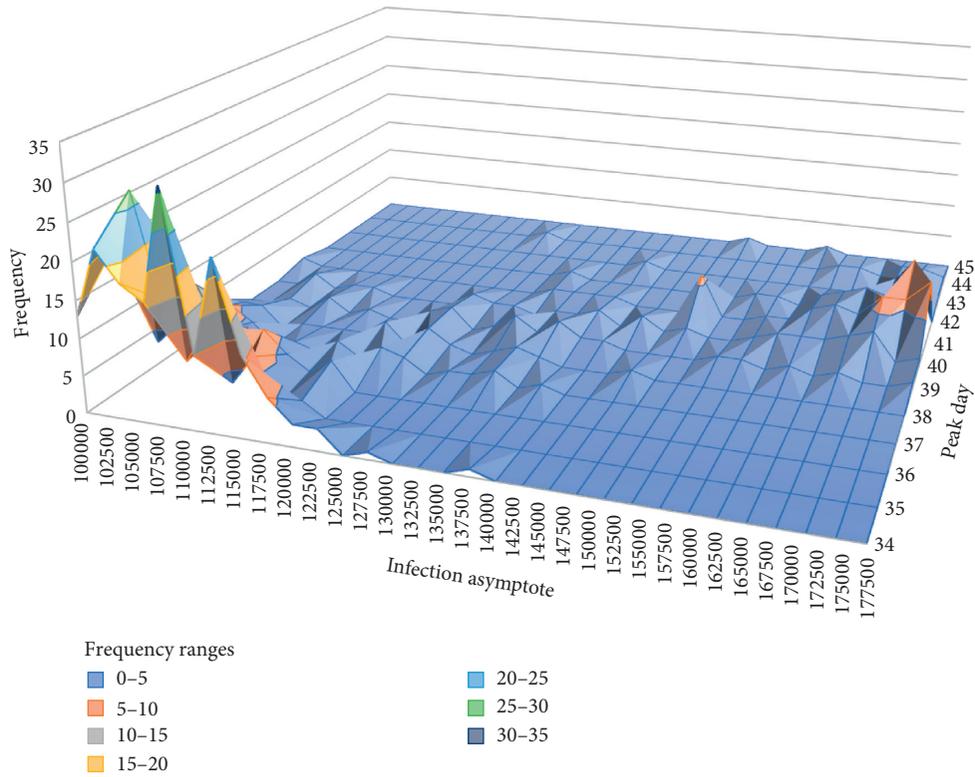


FIGURE 3: COVID-19 bootstrap (AsymptoteClass, Peak day) distribution.

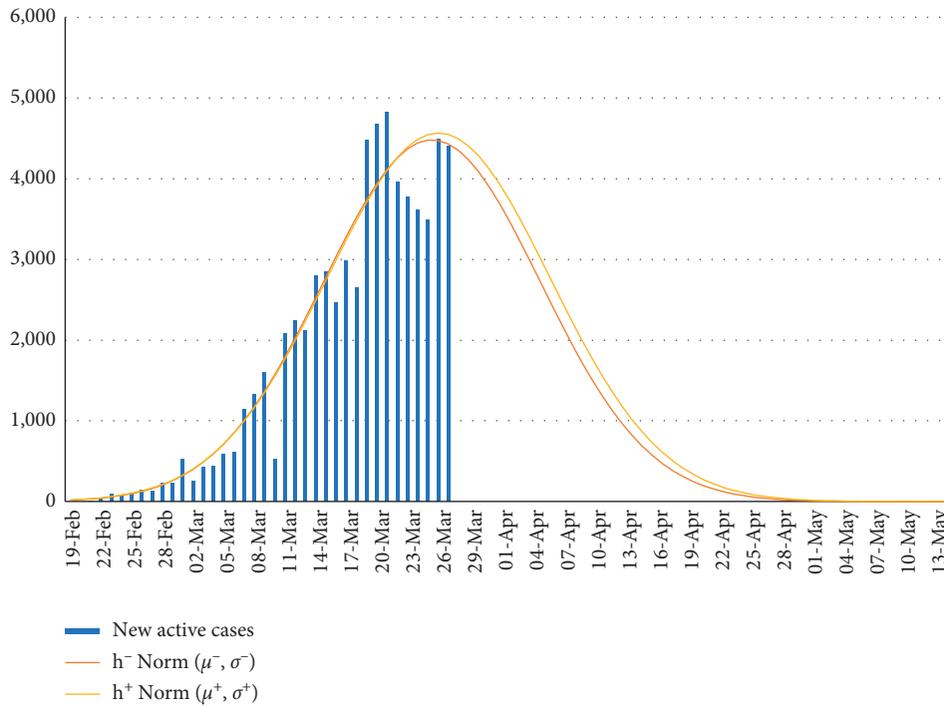


FIGURE 4: COVID-19 new active cases fitting with Gaussian.

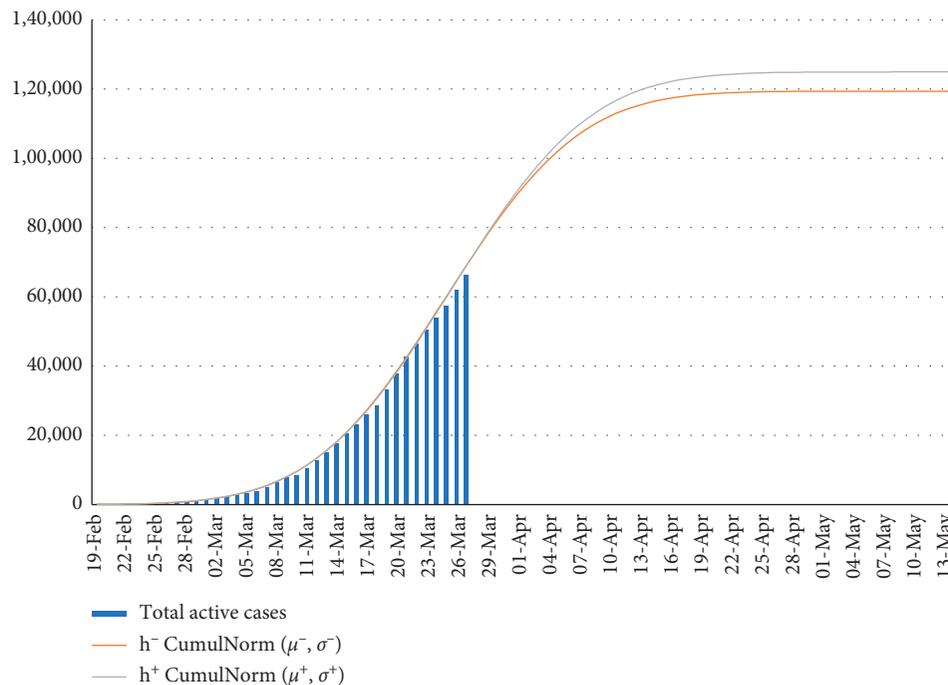


FIGURE 5: COVID-19 active cases fitting with cumulated Gaussian.

## 8. Further Developments

The SA optimization for fitting bootstraps derived from real data is applicable to any kind of distribution known in the literature and empirical distributions as well. This research highlights a great potential if the aforementioned procedure is enhanced by introducing an automatic routine for the “optimal” choice of either known ( $\xi_r$ ) or empirical ( $\xi_e$ ) distributions, where ( $\xi_r, \xi_e$ ) are in a predefined distribution space  $\Xi$ . In some more details, the algorithm could include a preprocessing light SA optimization (with a higher cooling rate  $\rho$  to cut down the number of SA iterations) designed to reduce the distribution space  $\Xi$  as well as the parameter space  $\Theta_\xi$  for each distribution  $\xi \in \Xi$  and thus boost the optimization search.

## Data Availability

The data used to support the findings of this study are publicly available, free of charge, at the web address <https://github.com/pcm-dpc/COVID-19/tree/master/dati-regioni>.

## Disclosure

The views and opinions expressed in this article are those of the authors and do not necessarily reflect the official policy or position of the Italian National Institute of Statistics or any other entities.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] J. D. Hamilton, “A new approach to the economic analysis of nonstationary time series and the business cycle,” *Econometrica: Journal of the Econometric Society*, vol. 57, pp. 357–384, 1989.
- [2] L. Koskinen and L.-E. Öller, “A classifying procedure for signalling turning points,” *Journal of Forecasting*, vol. 23, no. 3, pp. 197–214, 2004.
- [3] M. A. Delgado and J. Hidalgo, “Nonparametric inference on structural breaks,” *Journal of Econometrics*, vol. 96, no. 1, pp. 113–144, 2000.
- [4] M. P. Clements, P. H. Franses, J. Smith, and D. Van Dijk, “On SETAR non-linearity and forecasting,” *Journal of Forecasting*, vol. 22, no. 5, pp. 359–375, 2003.
- [5] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT press, Cambridge, MA, USA, 1995.
- [6] E. Carlstein, “The use of subseries values for estimating the variance of a general statistic from a stationary sequence,” *The Annals of Statistics*, vol. 14, no. 3, pp. 1171–1179, 1986.
- [7] P. Bühlmann, “Sieve bootstrap for time series,” *Bernoulli*, vol. 3, no. 2, pp. 123–148, 1997.
- [8] K. G. Buhlmann, *Linear Programming*, Springer, Berlin, Germany, 1983.
- [9] M. Bènichou, J. M. Gauthier, P. Girodet, G. Hentges, G. Ribière, and O. Vincent, “Experiments in mixed-integer linear programming,” *Mathematical Programming*, vol. 1, no. 1, pp. 76–94, 1971.
- [10] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Courier Corporation, Honolulu, HI, USA, 1998.
- [11] P. J. Van Laarhoven and E. H. Aarts, *Simulated Annealing: Theory and Applications*, pp. 7–15, Springer, Berlin, Germany, 1987.
- [12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [13] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*, Wiley, Hoboken, NJ, USA, 1988.