CoViD–19: Meta-heuristic optimization based forecast method on time dependent bootstrapped data

Livio Fenga
Italian National Institute of Statistics
ISTAT, Rome, Italy 00184
livio.fenga@istat.it

Carlo Del Castello
Kantar, Italy
carlo.delcastello@kantar.com

Abstract:
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. Futures lines of research, which include the generalization of the method to a broad set of distribution, will be finally given.

KEYWORDS: CoViD–19; maximum entropy bootstrap; model uncertainty; simulated annealing; non-linear optimization; peak prediction.

1. Introduction

In general, predicting the time of a peak conditional to a set of time dependent data is a non trivial task. Often carried out in a multi-tasking 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 on-line 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 this purposes – for example of the type hidden Markov (see, e.g. Hamilton (1989) and Koskinen and Öller (2004)) or non parametric (see, e.g., Delgado and Hidalgo (2000)) models – not only are very demanding in terms of building and tuning procedures, but typically require the availability of a "long" stretch of data. In addition to that, the time series related to epidemics usually show highly non-linear dynamics, which, if not pre-processed, make them not suitable for standard linear models. On the other hand, attempting to fit non-linear models – e.g. of the type Self Exciting Threshold Autoregressive (see, for example Clements et al. (2003)) or artificial neural network (Hassoun et al. (1995)) – it is not a viable options, 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 data set at hand is found.

2. The unified framework

As above stated, the approach studied in this paper, 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 4. By using this technique, an 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 design of the bootstrap method, as through its replications the use of central tendency functions, such as mean or median, are 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 paper uses the official data diffused daily by the National Institute of Health – an agency of the Italian Ministry of health – and the Department of Civil Protection. The data set includes 38 daily data points collected at national level during the period starting from January 19^{th} to March 27^{th}. The used indicator – which will refer to as the variable of interest – is obtained by subtracting, for each day, from the total number of people tested positive of Corona virus the number of the deaths and of the recovery.

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 *iid* assumption (Efron’s initial bootstrap method) is violated. Under dependence structures embedded in the data, simple sampling with replacement has been proved – see, for example Carlstein et al. (1986) – to yield suboptimal results. As a matter of fact, *iid*–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 non-linearity). In such cases, selecting the “right” resampling scheme becomes a particularly challenging task as many resamplig 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 Bühlmann et al. (1997) – 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 $\boldsymbol{\Omega}$ represents the population of reference the observed time series is drawn from, in *MEB* a large number of ensembles (subsets), say $\{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N\}$ becomes the elements belonging to $\boldsymbol{\Omega}$, each of them containing a large number of replicates $\{x_1, \ldots, x_J\}$. 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 $\{x_t \in \boldsymbol{\omega}_i;\ i = 1, \ldots, N\}$, the *MEB* procedure generates a set of disjoint subsets $\boldsymbol{\Omega_N} \equiv \omega_1 \cap \omega_1 \cdots \cap \omega_N$

s.t. $\mathbb{E}\boldsymbol{\Omega_N} \approx \mu(x_t)$, being $\mu(\cdot)$ the sample mean. Furthermore, basic shape and probabilistic structure (dependency) is guaranteed to be retained $\forall x^*_{t,j} \subset \omega_i \subset \Omega$.

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 non-stationarity. *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 $\delta(x)$, is chosen so that the expectation of the Shannon Information $H = \mathbb{E}(-\log \delta(x))$, is maximized, i.e.

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

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

---

1. A sorting matrix of dimension $T \times 2$, say $\mathcal{S}_1$, accommodates in its first column the time series of interest x_t and an Index Set – i.e. $I_{ind} = \{2, 3, \ldots, T\}$ – in the other one;
2. $\mathcal{S}_1$ is sorted according to the numbers placed in the first column. As a result, the order statistics $\boldsymbol{x_{(t)}}$ and the vector I_{ord} of sorted I_{ind} are generated and respectively placed in the first and second column;
3. compute "intermediate points", averaging over successive order statistics, i.e. $c_t = \frac{x_{(t)}+x_{(t+1)}}{2}$, $t = 1, \ldots T-1$ and define intervals I_t constructed on c_t and r_t, using *ad hoc* weights obtained by solving the following set of equations:

   i)
   $$f(x) = \frac{1}{r_1} \exp(\frac{[x - c_1]}{r_1}); \quad x \in I_1; r_1 = \frac{3x_{(1)}}{4} + \frac{x_{(2)}}{4}$$

   ii)
   $$f(x) = \frac{1}{c_k - c_{k-1}}; \quad x \in (c_k; c_{k+1})],$$
   $$r_k = \frac{x_{(k-1)}}{4} + \frac{x_{(k)}}{2} + \frac{x_{(k+1)}}{4}; \ k = 1, \ldots, T-1;$$

   iii)
   $$f(x) = \frac{1}{r_T} \exp \frac{\Big([c_{T-1} - x]\Big)}{r_T}; x \in I_T; \quad r_T = \frac{x_{T-1}}{4} + \frac{3x_T}{4};$$
4. from a uniform distribution in $[0, 1]$, generate T pseudorandom numbers and define the interval $R_t = (t/T; t + 1/T]$ for $t = 0, 1, \ldots, T-1$, in which each p_j falls;
5. create a matching between R_t and I_t according to the following equations:

   $$\begin{aligned} x_{j,t,me} &= c_{T-1} - |\theta| \ln(1 - p_j) \quad && \text{if } \ p_j \in R_0, \\ x_{j,t,me} &= c_1 \quad\ \ - |\theta||ln(1 - p_j)| && \text{if } \ p_j \in R_{T-1}, \end{aligned}$$

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

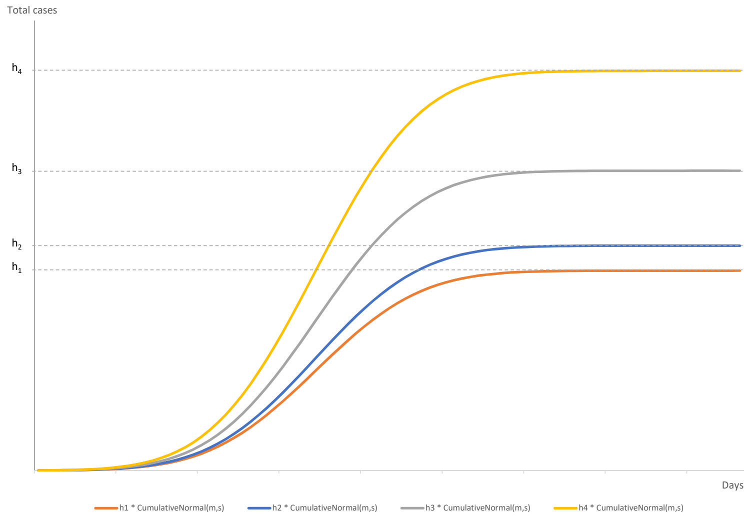
6. a new $T \times 2$ sorting matrix $\mathcal{S}_2$ is defined and the T members of the set $\{x_{j,t}\}$ for the j^{th} resample obtained in Step 5 is reordered in an increasing order of magnitude and placed in column 1. The sorted I_{ord} values (Step 2) are placed in column 2 of $\mathcal{S}_2$;
7. matrix S_2 is sorted according to the second column so that the order $\{1, 2, \ldots, T\}$ is there restored. The jointly sorted elements of column 1 is denoted by $\{x_{\mathcal{S},j,t}\}$, where $\mathcal{S}$ recalls the sorting step;
8. Repeat Steps 1 to 7 a large number of times.

5. Bootstrap driven forecast optimization

This section aims to define an alternative method to forecast our variable of interest using an optimization approach to fit a set of distribution functions on bootstrap replications.

The variable of interest is assumed to approximately describe a logistic function, scaled by a normalizing parameter h (representing the asymptotic number of cases)

Fig. 1. **COVID19 Active cases**



and its derivative is a Gaussian function re-scaled by the parameter h.

$$f(t \mid h, \mu, \sigma) = h \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}$$

where $t =$ days since pandemic has started in Italy
$h =$ magnitude scale

$\mu =$ peak of daily cases (scale)
$\sigma =$ standard deviation (shape)

So, given

- the parameter vector $\theta = (h, \mu, \sigma)$, where $\theta \in \Theta$
- the total active cases x_t since the infection spread
- a generic bootstrap distribution $x^*_{t,i} \in \omega \subset \Omega$ where $i \in \{1..N\}$ is the i-th bootstrap within N replicates
- $\hat{x}^*_{t,i} = f(t \mid h, \mu, \sigma)$ where $\hat{x}^*_{t,i}$ is the theoretical value

the objective is

$$\forall i \in \{1..N\} \quad \min_{\theta_i \in \Theta} MSE_i(x^*_{t,i}, \hat{x}^*_{t,i}) = \min_{\theta_i \in \Theta} AVG_i(x^*_{t,i} - \hat{x}^*_{t,i})^2$$

This is a non-linear unconstrained optimization problem which cannot be addressed using standard global optimization methods (Simplex, Branch and Bound or Branch and Cut algorithm), which are designed for Linear Programming LP (Murty (1983)) and Mixed-Integer Linear Programming MILP (Bénichou et al. (1971)), within the field of discrete combinatorial problems (Papadimitriou and Steiglitz (1998)).

Local search Simulated Annealing meta-heuristic to approximate global optimization can be used to solve unconstrained non-linear problems in a large space.

6. Simulated Annealing Optimization

Simulated annealing (SA), following Van Laarhoven and Aarts (1987), 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.

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. In 1983, this approach was used by Kirkpatrick et al. (1983) for a solution of the traveling salesman problem. They also proposed its current name, simulated annealing.

This 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 meta-heuristics because it allows for a more extensive search for the global optimal solution.

In general, simulated annealing algorithms work as below explained. 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 probabilities of selecting better or worse solutions, which during the search respectively remain at 1 (or positive) and 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 Neighbour(s, ϕ) should generate a randomly chosen neighbour of a given state s; the call Random $U(0, 1)$ should pick and return 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 ρ.

```
• Let current temperature $T = t_0$
• Set Cooling rate $\rho$
• For each bootstrap series $i$ in $\{1, \dots, N\}$
    – Let current solution $s = s_0$
    – Loop while temperature $T > 1$
        ⋆ Pick a random neighbor, $s_{new} \leftarrow$ Neighbor$(s, \phi)$, where $\phi$ is the radius around $s$
        ⋆ if $Prob(E(s), E(s_{new}) \,|\, T, k_B) \geq$ Random $U(0, 1)$:
              $s \leftarrow s_{new}$
        ⋆ $T \leftarrow T * (1 - \rho)$
    – Output the final state $s$ on $i$-th bootstrap
```

where $Prob(E(s), E(s_{new}) \,|\, T, k_B)$ is the acceptance probability at each iteration given temperature T and Boltzmann constant (see Aarts and Korst (1988)) k_B

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

7. Empirical evidences

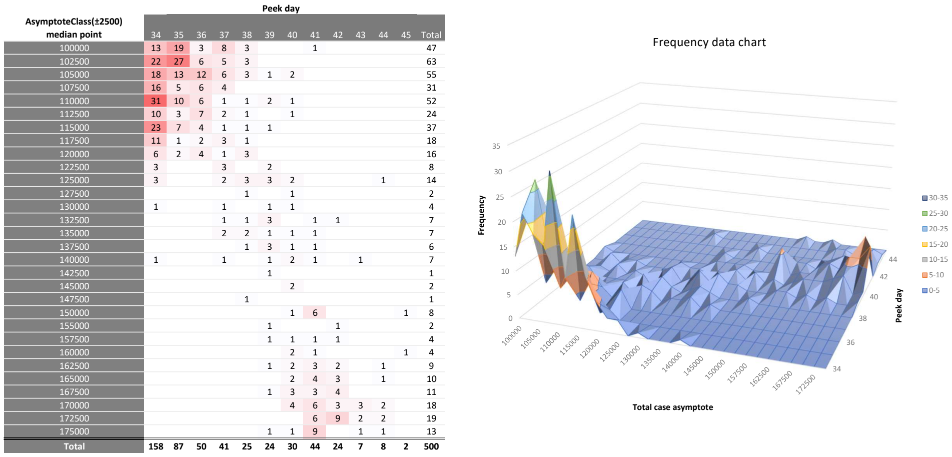
In order to improve local search speed the parameter space Θ can be bounded to $\Theta' \subset \Theta$ removing useless tails. No information is lost if parameters space is reduced to

$$h \in [100000, 180000], \quad \mu \in [35, 50], \quad \sigma \in [7, 13]$$

SA parameters has been recursively tuned and the procedure improved using an initial temperature $T_0 = 10000$, a cooling rate $\rho = 0.0006$, Boltzmann constant $k_B = 100$ and radius $\phi = 0.3(\theta_{max} - \theta_{min})$.

Optimization procedure applied to 500 Bootstraps, derived from active case in Italy from February 19^{th} to March 27^{th}, shows

Fig. 2. **COVID19 bootstrap [AsymptoteClass,Peekday] frequency**



AsymptoteClass(±2500) median point	Peek day												
	34	35	36	37	38	39	40	41	42	43	44	45	Total
100000	13	19	3	8	3			1					47
102500	22	27	6	5	3								63
105000	18	13	12	6	3	1	2						55
107500	16	5	6	4									31
110000	31	10	6	1	1	2	1						52
112500	10	3	7	2	1		1						24
115000	23	7	4	1	1	1							37
117500	11	1	2	3	1								18
120000	6	2	4	1	3								16
122500	3			3		2							8
125000	3			2	3	3	2				1		14
127500					1		1						2
130000	1			1		1	1						4
132500				1	1	3		1	1				7
135000				2	2	1	1	1					7
137500					1	3	1	1					6
140000	1			1		1	2	1		1			7
142500						1							1
145000							2						2
147500					1								1
150000							1	6				1	8
155000						1			1				2
157500						1	1	1	1				4
160000							2	1				1	4
162500						1	2	3	2		1		9
165000							2	4	3		1		10
167500						1	3	3	4				11
170000							4	6	3	3	2		18
172500								6	9	2	2		19
175000						1	1	9		1	1		13
Total	**158**	**87**	**50**	**41**	**25**	**24**	**30**	**44**	**24**	**7**	**8**	**2**	**500**

with $Avg(h) = 122178, \quad Avg(\mu) = 36.7, \quad Avg(\sigma) = 10.8.$

As an approximation, confidence bounds from normal distibution are derived for each of the parameters, such that with 0.01 significance level

$$Pr(119401 \leq h \leq 124955) = 0.99$$

$$Pr(36.41 \leq \mu \leq 37.07) = 0.99$$

$$Pr(10.6 \leq \sigma \leq 10.9) = 0.99$$

μ confidence interval points out a peak day of daily cases between March 25^{th} and 26^{th}, while h magnitude parameter shows an asymptote of totale active cases curve between 120000 and 125000. The new cases curve has an asymptotic behavior, so cutting tail beyond a 0.1 cut-off for new infections, the pandemic time window is hypothetically over after May 16^{th}.

This behavior is clearly described in the below charts built considering Gaussian and Cumulated Gaussian around 99% confidence lower and upper bound for each parameter.

Fig. 3. **COVID19 Active cases fitting with Cumulated Gaussian**

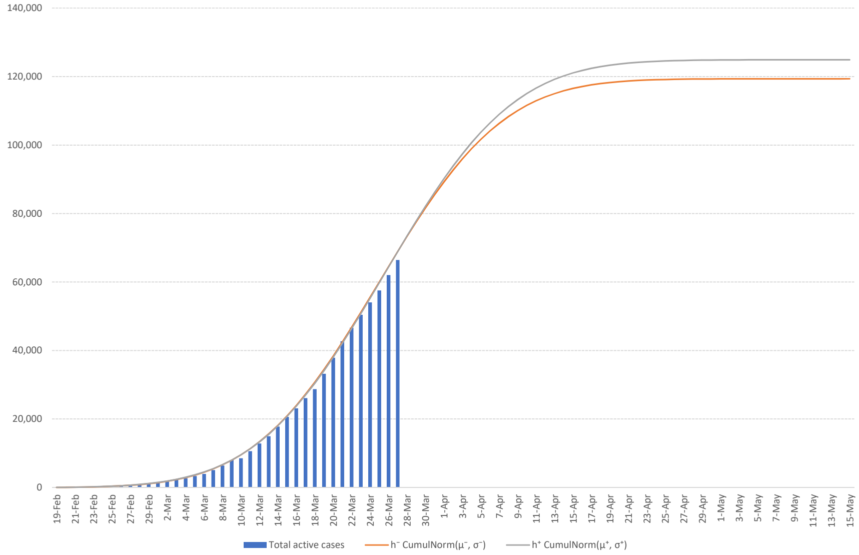
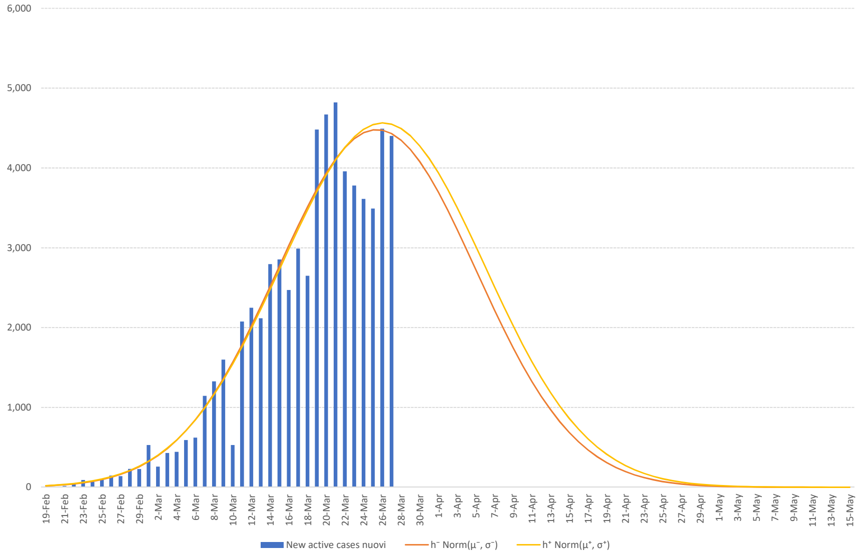


Fig. 4. **COVID19 New active cases fitting with Gaussian**



8. Further developments

The SA optimization for fitting bootstraps derived from real data is applicable to any kind of distribution known in literature and empirical distributions as well.

This kind of research highlights a great potential if the aforementioned procedure is enhanced with the automatic choice of known distributions ξ_r or empirical ones ξ_e where (ξ_r, ξ_e) are in a predefined distribution space Ξ. More in detail, the algorithm could include a pre-processing light SA optimization (with an higher cooling rate ρ to cut down the number of SA iterations) to reduce the distribution space Ξ and the parameter space Θ_ξ for each distribution $\xi \in \Xi$ and boost the optimization search performed by the main SA process.

9. Disclaimer

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 Entity.

References and links

Aarts, E., and Korst, J. (1988), "Simulated annealing and Boltzmann machines," , .

Bénichou, M., Gauthier, J.-M., Girodet, P., Hentges, G., Ribière, G., and Vincent, O. (1971), "Experiments in mixed-integer linear programming," *Mathematical Programming*, 1(1), 76–94.

Bühlmann, P. et al. (1997), "Sieve bootstrap for time series," *Bernoulli*, 3(2), 123–148.

Carlstein, E. et al. (1986), "The use of subseries values for estimating the variance of a general statistic from a stationary sequence," *The annals of statistics*, 14(3), 1171–1179.

Clements, M. P., Franses, P. H., Smith, J., and Van Dijk, D. (2003), "On SETAR non-linearity and forecasting," *Journal of Forecasting*, 22(5), 359–375.

Delgado, M. A., and Hidalgo, J. (2000), "Nonparametric inference on structural breaks," *Journal of Econometrics*, 96(1), 113–144.

Hamilton, J. D. (1989), "A new approach to the economic analysis of nonstationary time series and the business cycle," *Econometrica: Journal of the Econometric Society*, pp. 357–384.

Hassoun, M. H. et al. (1995), *Fundamentals of artificial neural networks* MIT press.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983), "Optimization by simulated annealing," *science*, 220(4598), 671–680.

Koskinen, L., and Öller, L.-E. (2004), "A classifying procedure for signalling turning points," *Journal of Forecasting*, 23(3), 197–214.

Murty, K. G. (1983), *Linear programming* Springer.

Papadimitriou, C. H., and Steiglitz, K. (1998), *Combinatorial optimization: algorithms and complexity* Courier Corporation.

Van Laarhoven, P. J., and Aarts, E. H. (1987), "Simulated annealing," in *Simulated annealing: Theory and applications* Springer, pp. 7–15.