NUMERICAL OPTIMIZATION FOR LARGE SCALE PROBLEMS AND STOCHASTIC OPTIMIZATION PROBLEM
Stochastic project 01

Eleonora Carletti e Gianluca Guzzetta

A.Y. 2021/2022

1 Introduction to the project

In this report a static stochastic approach has been used to predict and minimize the cost of an epidemic simulation considering both health system and social distancing costs given a certain probability. A stochastic gradient method has been used in order to identify a plausible good local minimum of a neural network.
In order to accomplish this task, a simulation of an epidemic model was needed, in this case the Reed-Frost model, which is a stochastic epidemic model arose around the early 1900, belonging to the family of S-I-R (**S**usceptible-**I**nfection-**R**emoval) models.
The Reed-Frost model describes the evolution of an infection in generations analyzing the infected number of contagiousness of illness for a *closed population* over a certain interval and the number of infected at the next step in time (i.e. $t+1$), in which the number of infected individuals depend on a binomial random variable with $S(t)$ and p, respectively number of susceptible individuals at time t and the probability of one-to-one infection.
Then, based on this distribution of random values, we performed the simulation of the model on a set of predefined probabilities, in order to understand

how the simulations could vary depending on these parameters over a certain period, these considerations give us a raw setup of what we could expect from this model.

2 Analysis of the Stochastic model

The Reed-Frost model provided us with the possibility to run an experimental simulation of human epidemic disease using a *stochastic approach*, in which the system's state can be represented as $\mathcal{S} = (S_n, I_n)$ which can be mathematically written as a binomial random variable of S_n and p:

$$I_{n+1} \sim Bin(S_n, 1 - (1-p)^{I_n}) \quad \textit{condition 1}$$

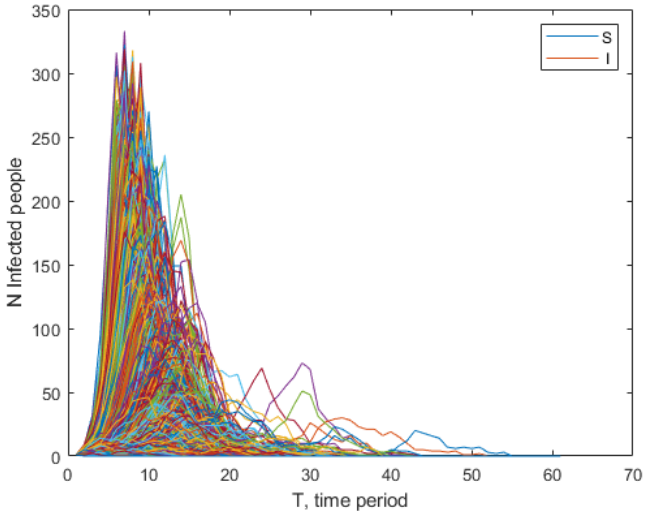
and

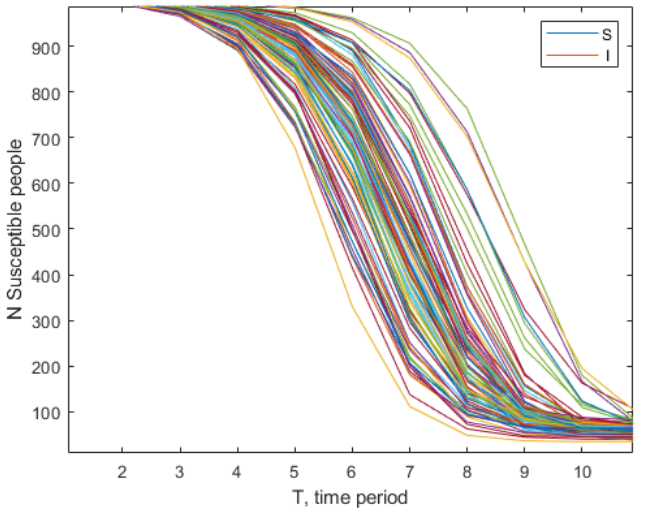
$$S_{n+1} = S_n - I_{n+1} \quad \textit{condition 2}$$

We consider as hypothesis running the simulation of the model for 1000 times each one with an interval period of "60 days". Furthermore, as initial values a closed population of $N = 1000$, with initial conditions $S_0 = 0$, and $I_0 = 1$. We consider as initial closed population of $N = 1000$, with initial conditions $S_0 = 0$, and $I_0 = 1$.
A set of predefined probabilities were given, and they can be considered as a vector, for further storage reasons as providing the average cost produced by each of the probability during the whole period considered given *nsim* (number of performed simulations of the model.)

$$p : [0.003, 0.00275, 0.0025, 0.002, 0.00175, 0.0015, 0.00125, 0.001, 0.00075, 0.0005]$$

Running the model produces the following first consideration (after multiple runs): observing figure (1) we reach the sharpest decline from [4,7] in the first 10 days there is a significant decrease number of susceptible people, proved in a growth of infected people.





As expected since we may encounter a huge number of infected people in the first period, after a certain time they will become immune/dead, till we reach a general trend at around decreasing over time and a peak max at N infected around 340. as expected in the SIR model. The number of infected tends to be more random that one above.

3 Evaluating the average cost

We should reckon the average cost depends on the cost of each iteration among each people, S and I. The social distancing cost expressed for each iteration is

a value which depends on a given grid of probabilities, and every time there is at least one infection, social distancing precautions has to be taken. The cost of the healthy system for each infective is of unit cost, therefore the total cost for each t is given by the sum of the social distancing cost $c(p)$ and the health system cost. The social distancing cost is represented as:

$$c(p) = \left(\frac{0.003}{p}\right)^9 - 1$$

This provided us the output layer for our *neural network NN*, the outputs to be analyzed to train our NN and to obtain the minimal average cost.

4 Generating a neural network to train dataset

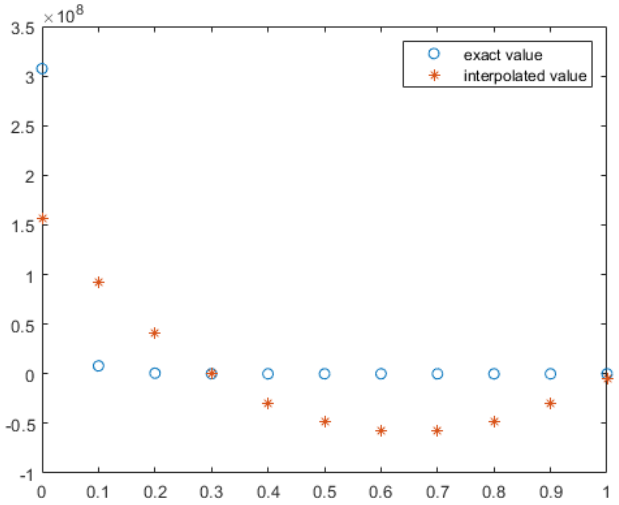
After a first sight to the model, the next step was to train a neural network to approximate the average cost in the whole continuous interval $p \in [0.003, 0.00125]$. We used a back propagation approach in order to train the neural network then the optimal solution is reached using an optimization gradient method, our choice was the Spall method, in order to perform only two function evaluations at each time.
The algorithm is based on the Gosavi book constants, the value of the set learning was set as constant to 0.05 and we used 2 levels of optimization in order to normalize the dataset and increasing accuracy, since the model is a stochastic process, using these "tricks" may lead to closer approximation, considering that picking a smaller value outside the interval probability set $p = [0.003, 0.00125]$ the value of the objective function would explode and NN would be hard to be fit to such level of non-linearity.

$$M = max(\bar{x}) - min(\bar{x})$$

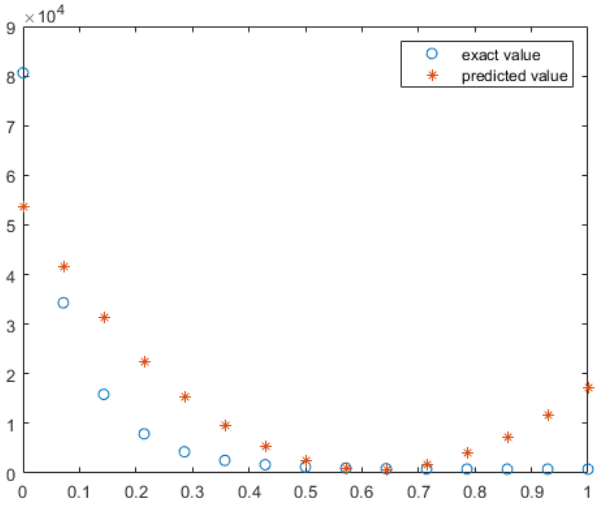
and the sigmoid as:

$$1 + e^{(-\frac{u}{M})^1}$$

in which M is the mean difference between the maximum value and the minimum value. We obtained the following result:



Now we tried to predict the average cost values based on our neural network in a set of probabilities between $[0.003, 0.00125]$.



As previously discussed, given the non-linearity (exponential) nature of the average cost function, it is not trivial to keep values as close as possible, but given these pictures we we can expect a probability close to 0.003 hence the average cost tends to go to 0.

5 Looking for the optimal p: the Spall method

The Spall method is a very efficient way to find an approximation of a derivative since it performs only 2 function evaluations each time. In order to find the optimal p we need to find a way to approximate the average cost function and to create a derivative based on functions evaluations.

1. Method able to work with a multinomial random variable (in this case b.r.v.)
2. Approximate with a cost of 2 computation per iteration to approximate the gradient using the Spall Method.
3. Using the Spall method to find a somehow reasonable optimal local minimum, further analyses has to be done to understand if it may be a good global optimum, moreover a Stochastic process brings already a noise, hence is reasonable to accept it with a proper acknowledgment.

Firstly, we computed h as a binomial random variable between (-1,1) in order to get a perturbation function at $f(x \pm ch)$ point starting from a random point $x_0 = Bin(0.00125, 0.003)$. Simultaneous perturbation stochastic approximation (Spall) method has been used with the following parameters taken from Gosavi textbook: $A = 10$, $B = 100$, $l = 2$, $C = 0.1$ $t = 1/6$, $tol = 0.001$. We chose as *perturbation coefficient* $c^m = C/m^t$ where $C = 0.1$ and $t = 1/6$. We proceeded to approximate the derivative of our average cost function into something which may be interpreted to find a minimum of this surface. Hence, covering the prediction with a proper way to find a derivative as just discussed, was the key to find a locally acceptable optimum, indeed evaluations of the function Applying this criteria and ran some times using different parameters (we passed from l=1, to l=2 in order to get "closer" results), finally we found an **optimal local minimum at** $x_0 = 2.787e - 02$, which appears to be a reasonable probability considered the previous assumptions.

6 Appendix: MATLAB implementation code