Lab 16- Epidemic processes

1.1 Introduction

In this report, we will present the Hawkes process and its application for epidemics. The Hawkes process is a point process, in which each previous event that occurred at the time $t_j < t$ generates new events at the rate $h(t−t_j)$ – also called the kernel of the Hawkes process. The event rate of a Hawkes process is a stochastic function dependent on previous event times, defined as:

$$\lambda(t|H_t) = \sigma(t) + m \Sigma_{j:\, tj<t} h(t - tj) \qquad (1)$$

This work will be devised in two parts:

First, we will simulate the epidemic process at an early stage, using the following parameters of the Hawkes process:

1. σ(t) is a constant equal to 20
2. h(t):
   2.1. h(t) is Uniform in t ∈ [0,20]
   2.2. h(t) follows this exponential distribution:
   h(t)=1/10*exp(-t/10)
3. m=2
4. t ∈ [0,100] i.e., this simulation will be on a time interval of 100 days.

Second, in the epidemic context, starting from the 20th day some restrictions will be implemented to reduce the stochastic intensity of this process of a factor $rh_o(t)$ which may be dynamically adjusted on a day-by-day basis.

We will assume for this part that the restrictions will lead to a cost equal to:

$$cost=rh_o^2(t)\, dt \qquad (2)$$

We will implement a model that minimizes this cost over one-year horizon, under the constraint that the average number of deaths does not exceed 20k.

1.2 Simulation

1.2.1 Input parameters:

The input parameters needed for this simulation are the followings:

- The population size.
- Sigma.
- Kernel of the Hawkes process.
- The total days the simulation will run for.
- The percentage of infected people that die.
- The maximum number of deaths over a year.
- Cost
- The seed value used to initialize a pseudorandom number generator.
- The number of runs: the number of times that we run our simulation. This is done to have more accurate results.

1.2.2 Output metrics:

The output metrics of the simulation for the early stage of the Hawkes Process and the generalization are:

- Number of infected/death individuals for the two choices of h(t).
- The length of the confidence interval.
- The relative error.

1.2.3 Main data structures

The main data structure used is NumPy arrays.

1.2.4 Main algorithm

1.2.4.1 Early-stage Hawkes Process

In order to simulate this epidemic process using a Hawkes process, we would follow the below steps:

1. First, understand the parameters of the Hawkes process: In this case, the function $\sigma(t)$ is equal to 20 for t ∈ [0, 10] days and 0 for all other times.

$h(t)$ (kernel function) is either uniform [0, 20] or an exponential decay with a decay rate of λ = 1/10 days.

m = 2, which represents the average number of secondary events caused by each primary event.

Primary events: These events are the "original" events that start the process. In this context of an epidemic, primary events represent the first infections in a population. These events are typically assumed to occur independently of one another and are usually modeled as a Poisson process with a certain intensity rate.

Secondary events: These events are caused by primary events. In an epidemic, secondary events represent the further spread of the disease in a population after the initial infections. The occurrence of secondary events is modeled as a point process that is dependent on the primary events. A Hawkes process model assumes that the occurrence of a secondary event increases the likelihood of additional secondary events in the near future, hence the name self-exciting.

2. Implement a simulation algorithm: One way to simulate a Hawkes process is to use the "thinning" algorithm, which is based on the probability distribution of the inter-arrival times between events. We will need to compute the conditional intensity function of the process at each time step and use it to generate the next event.

3. Generate the initial events: To begin the simulation, we need to generate the initial events (the "seeds") of the process. In this case, we have 2% of individuals that get infected and die, so we generate the number of initial events (seeds) based on this percentage of the total population.

4. Simulate the process over the specified time horizon: Once we have the initial events, we iterate over the time horizon (in this case, t ∈ [0, 100] days) and use the thinning algorithm to generate the next event at each time step.

5. Keep track of infected and death individuals: In this case, we need to keep track of how many individuals get infected, as well as how many of those individuals die after a while. We will do this by updating a counter for each event generated.

- **Uniform Kernel:**

We start by defining the kernel of the Hawkes process to a uniform distribution for t ∈ [0,20] and obtain to following results:

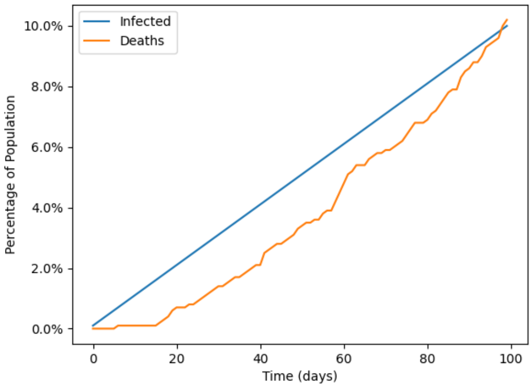

Fig 01. Percentage of infected/deaths individuals over time for uniform h(t)

- **Exponential Kernel:**

We then define the kernel of the Hawkes process to an exponential distribution with lambda=1/10 and obtain to following results:

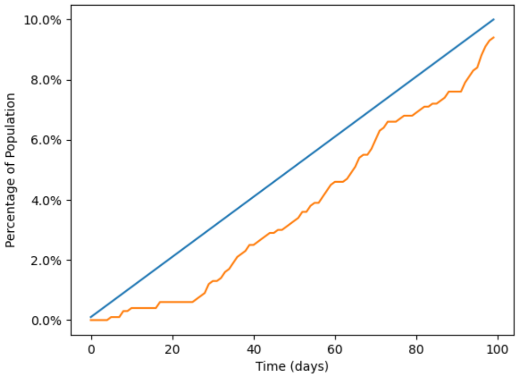

Fig 02. Percentage of infected/deaths individuals over time for exponential h(t)

- **Interpretation:**

From the generated plots, we can notice that number of infected individuals is not dependent on the kernel function used. The kernel function is used to calculate the intensity function. This intensity function is only used to calculate the next event, which is the number of new infections, and this new infection then follows the recovery or death process, which is independent of the choice of h(t). In this case, the number of infected individuals is determined primarily by the other parameters such as the branching parameter m and the number of seeds (initial infected individuals). As we change the parameter m or the number of seeds the results of the plot will be different.

Concerning the number of deaths, we can notice the difference between the uniform kernel and the exponential one. For the exponential case, the number of deaths is decreasing over time in comparison with the uniform case.

1.2.4.2 Generalization

In this section, we are generalizing to include non-pharmaceutical interventions (NPI) that reduce the stochastic intensity of the process by a factor $\rho(t)$.

Here are the steps followed:

1. Introduce the concept of $\rho(t)$: This is the factor by which the stochastic intensity of the process is reduced at each time step, and it can be adjusted on a day-by-day basis. We assume that $\rho(t)$ starts at 1 (max value) for $t < 20$ and can be set to a value between 0 and 1 for $t \geq 20$.

2. Implement the NPI reduction factor into the simulation: Modify the simulation algorithm to include the NPI reduction factor $\rho(t)$ in the computation of the conditional intensity function. Specifically, multiply the conditional intensity function by $\rho(t)$ at each time step.

3. Implement the cost function: The cost function is proportional to the integral of $\rho(t)$^2 over time. So, we will have to keep track of the $\rho(t)$^2 values and integrate them at the end of the simulation.

4. Design a strategy: We need to design a strategy that attempts to minimize the cost over the 1-year horizon while keeping the average number of deaths below 20,000. This will involve adjusting $\rho(t)$ over time to achieve the best balance between controlling the spread of the disease and minimizing the cost.

We will implement a strategy that follows a rule-based approach. This approach will allow us to set of predefined rules to adjust $\rho(t)$ based on the cost and the number of infections but also the number of deaths. The value of $\rho(t)$ can be for example increased if the number of infected individuals is high and decreased if the cost becomes too high.

5. Simulate the process over the specified time horizon: Once we have the initial events, we iterate

over the time horizon (in this case, t ∈ [0, 365] days) and use the modified simulation algorithm to generate the next event at each time step.

6. Keep track of infected, death and cost over time: We will have to keep track of how many individuals get infected and how many of them die after a while, as well as the cost of the restrictions. We do this by updating a counter for each event generated and cost at each time step.

Below are the results obtained after the implementation of the results:

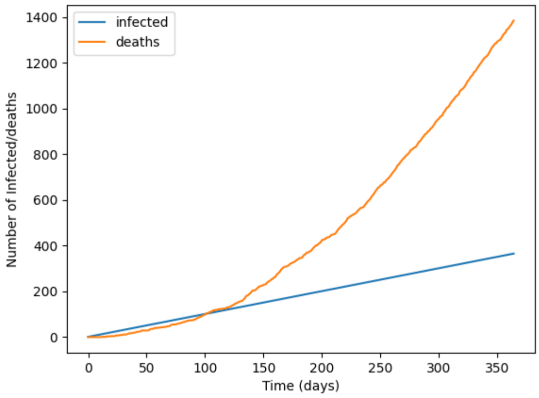

Fig 05. Evolution over a year interval of infected/death individuals

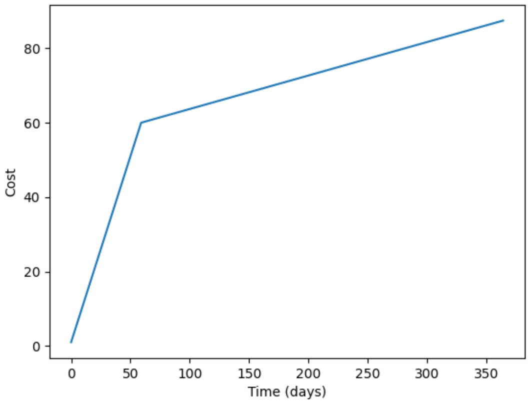

Fig 06. Evolution of the cost over a year interval

- **Interpretation:**

We can see from Fig 05 that the curve of infected people in comparison with the infected has been affected by the NPI changes implemented and hence the number of infections has decreased.

Fig 06 shows how the total cost of NPI changes over time. This overall trend of the cost shows the effectiveness of the NPI implemented.

The cost in this experiment is computed as the sum of the square of the values of $\rho(t)$ at each time step t, over the whole year. The cost is proportional to the square of the value of $\rho(t)$. A higher value of $\rho(t)$ means a higher intensity of the interventions, which in turn leads to a higher cost.

A value of $\rho(t)$ = 1 in the implemented code means that the intensity of the interventions is at its maximum level. This would mean that the highest level of restrictions and measures are being implemented (such as school closures, quarantines, social distancing, and lockdowns, etc.). This could lead to the highest level of costs for society and the economy, but also the highest chance to curb the spread of the disease. A value of $\rho(t)$ < 1 means that the interventions are less intense and less restrictive, leading to a lower cost but also a lower chance of curbing the spread of the disease.