ECMM450 Stochastic Processes
Simulation of Non-Homogeneous Poisson Processes

70054986
Department of Computer Science, University of Exeter
(Dated: April, 2023)

This project is about Non-Homogeneous Poisson processes and how to simulate them. We will be reviewing the thinning algorithm of Lewis and Shedler (1979) for simulating NHPP.

I. INTRODUCTION

Homogeneous Poisson processes are the natural representation of processes occuring in time. As they follow a fixed rate of 'event occurrence', they are unable to model processes with varying incidence rate, such as shocks before and after earthquakes [1], fluctuations in neural data [2], events with time-of-day patterns or those showing long-term growth or decay.
This report will be loosely structured according to the following points:

1. Explanation of a Non-Homogeneous Poisson Process (NHPP) accompanied by clear mathematical definition;
2. Review of the thinning algorithm of Lewis and Shedler (1979) [3] for simulating NHPP, with brief description of the algorithm, and its main benefits compared to other approaches;
3. Simulation of occurrence of 1000 successive events $t_1, t_2, ..., t_{1000}$ for a homogeneous Poisson process having a rate of 8 events per year, followed by visual representation of the same;
4. Determining whether the thinning algorithm can be used on the previous homogeneous Poisson process data to simulate an NHPP with a rate function that increases smoothly from 1 event per year at $t = 0$ by 1% per year, i.e. $\lambda(t) = (1.01)^t$;
5. Code to find occurence times for a NHPP having the rate function $\lambda(t) = (1.01)^t$ by using the thinning algorithm;
6. Comparison of the plots of HPP and NHPP to demonstrate differences between the two.

II. DESCRIPTION AND MATHEMATICAL DEFINITION

A non-homogeneous Poisson process can be thought of as a generalization of the homogeneous Poisson process. As opposed to its homogeneous counterpart where rate of occurrence of events is constant (denoted by λ), here the rate is a function of time, denoted by $\lambda(t)$. Thus, the number of occurrences in the interval $(0, T]$ follows Poisson distribution $Pois(\int_0^T \lambda(t)dt)$ [4]. More formally, allowing the rate parameter to vary with time results in the following definition.

Definition II.1 (Ross, 2006, p.339, Definition 5.4)**.** [5] The counting process $N(t), t \geq 0$ is said to be a nonhomogeneous Poisson process with intensity function $\lambda(t) \geq 0, t \geq 0$, if

1. $N(0) = 0$
2. The process has independent increments
3. $PN(t+h) - N(h) = 1 = \lambda(t)h + o(h)$
4. $PN(t+h) - N(h) \geq 2 = o(h)$

where $o(h)$ carries its usual meaning - a function $\phi(h)$, that satisfies the condition that $\frac{\phi(h)}{h} \to \infty$ as $h \to \infty$ [6].

Time sampling an ordinary Poisson process (with constant rate λ) results in a Non-Homogeneous Poisson process. Given $\{N(t), t \geq 0\}$, a Poisson process wih rate λ, if the event occurring at time t is counted with probability $p(t)$, then $\{N_c(t), t \geq 0\}$ is a Non-Homogeneous Poisson process [5].

III. REVIEW OF THE THINNING ALGORITHM

1. *Some Other Algorithms*

There exist other algorithms for simulating Poisson processes, a few of which are briefly described.

1. Time-scale transformation of a homogeneous Poisson process via inverse of the integrated rate function $\Lambda(x)$. Cinlar [7] states that the random variables $T1, T2, ...$ are event times corresponding to a nonhomogeneous Poisson process with expectation function $\Lambda(t)$ if and only if $\Lambda(T1), \Lambda(T2), ...$ are the event times corresponding to a homogeneous Poisson process with rate one. This of course requires the inverse of the integrated rate function $\Lambda(t)$ to be computed, which may prove costly and inefficient.
2. Another method is to generate intervals between the points individually [8]. It follows a similar inversion approach as the first method. Considering the ith inter-event time $X_i = T_i + 1 - T_i$, conditional on the first i event times $T_1 = t_1, T_2 =$

$t_2, ..., T_i = t_i$, its cdf can be derived as $Fti = 1 - e^{-\Lambda(ti+x)+\Lambda(ti)}$. Then $X_{i+1} - Xi$ can be generated as $X_{i+1} - Xi = F^{-1}(U_i)$, where $U_i \sim Unif(0,1)$. As stated in the original paper [3], this process involves not only the computation of the inverse cdf, but the distribution has different parameters for each interval.

3. Simulation of Non-Homogeneous Poisson variates can be done through by generating order statistics from a distribution with cdf $F(t) = \Lambda(t)/\Lambda(t_0)$ for $t \in [0, t_0]$ [9]. Formally, let $T_1, T_2, ...$ be time(s) to event for a Non-Homogeneous Poisson process with integrated rate function $\Lambda(t)$. Conditional on having observed $N(t_0) = n$ points in $(0, t_0]$, T_k is the kth order statistic from a distribution with cdf $F(t) = \Lambda(t)/\Lambda(t_0)$.

2. *The Thinning Algorithm*

To construct a Non-Homogeneous Poisson process $\{N(t), t \geq 0\}$, with rate parameter $\lambda(t)$, over the interval $(0, T]$, the algorithm starts with a Non-Homogeneous Poisson process $\{N^*(t), t \geq 0\}$, with rate parameter $\bar{\lambda}(t)$ that dominates the set $\lambda(t)$ for all $t \in (0, T]$, that is

$$\bar{\lambda}(t) \geq \lambda(t) \forall t \in (0, T]$$
$$\bar{\lambda}(t) = sup_{t\in(0,T]}\lambda(t)$$

Then, for all t, the point from the dominating NHPP is retained with probability $\lambda(t)/\bar{\lambda}$. The remaining points form a NHPP with rate parameter $\lambda(t)$. It is noted that since points are deleted independently, the number of points in $\{N(x) : x \geq 0\}$ in any set of non-overlapping intervals are mutually independent.

3. *Benefits Over Other Algorithms*

Each method has drawbacks with respect to computational efficiency. In its simplest implementation, the thinning method "*obviates the need for numerical integration of the rate function, for ordering of points, and for generation of Poisson variates*" [3].

IV. SIMULATION OF A HOMOGENEOUS POISSON PROCESS

Uniform random numbers are used to generate Poisson variates by using the following algorithm, that can be found in [4]. Full implementation in Python can be found in the Appendix. Fig 1 shows the result of simulating a homogeneous Poisson process.

Algorithm 1: (Lewis and Shedler, 1979, p.7, Algorithm 1) Simulation of an Inhomogeneous Poisson Process with Bounded Intensity Function $\lambda(t)$, on $[0, T]$

```
Input: $\lambda, T$
Initialize $n = m = 0,\ t_0 = s_0 = 0,$
  $\bar{\lambda}(t) = sup_{t\in(0,T]}\lambda(t)$;
while $s_m < T$ do
    Generate $u \sim$uniform(0,1);
    Let $w = -ln(u)/\bar{\lambda}$;
    Set $s_{m+1} = s_m + w$;
    Generate $D \sim$ uniform(0,1);
    if $D < \lambda(s_{m+1})/\bar{\lambda}$ then
        $t_{n+1} = s_{m+1}$;
        $n = n + 1$;
    end
    $m = m + 1$
end
if $t_n \leq T$ then
    return $\{t_k\}_{k=1,2,...,n}$
else
    return $\{t_k\}_{k=1,2,...,n-1}$
end
```

Algorithm 2: Simulation of a Homogeneous Poisson Process with Rate λ, on $[0, T]$ ¡Insert reference to Yuanda Chen¿

```
Input: $\lambda, N$
Initialize $n_0 = 0,\ t_0 = 0$;
while $True$ do
    Generate $u \sim$uniform(0,1);
    Let $w = -ln(u)/\bar{\lambda}$;
    Set $t_{n+1} = t_n + w$;
    if $n + 1 > N$ then
        return $\{t_k\}_{k=1,2,...,n}$
    else
        Set $n = n + 1$;
    end
end
```

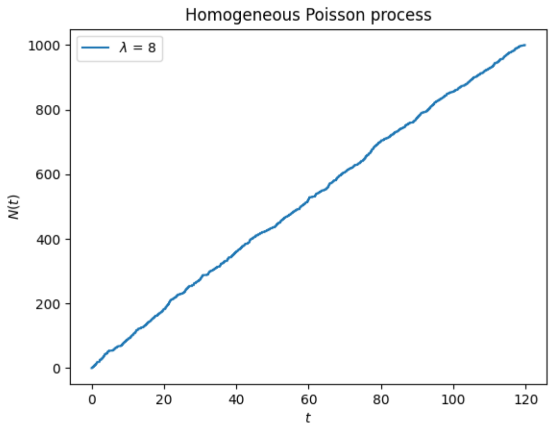

FIG. 1. Simulation of the time of occurrence of 1000 successive events for a homogeneous Poisson process having a rate of 8 events per year. $N(t)$ denotes the number of events occurring in the time period $[0, t]$.

The graph appears relatively linear. This would be expected as the rate of occurrence of events remains constant over any interval of time (a, b) and hence the slope corresponding to the number of events over time (i.e. rate) also remains the same. A comparison agains $E[N(t)] = \lambda t$ has been plotted in Fig **??** to demonstrate the linearity of the HPP.

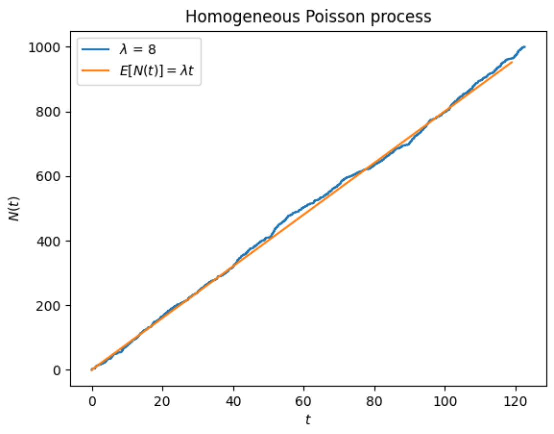

FIG. 2. Expected number of events follows a straight line with slope, $\lambda = 8$. $N(t)$ denotes the number of events occurring in the time period $[0, t]$.

V. NHPP WITH A SMOOTHLY INCREASING RATE FUNCTION

An NHPP is considered with a rate function that increases smoothly from 1 event per year at $t = 0$ by 1% per year, i.e. $\lambda(t) = (1.01)^t$. Looking at the previous homogeneous Poisson process from IV, it can be seen that the maximum time taken to accumulate 1000 events is

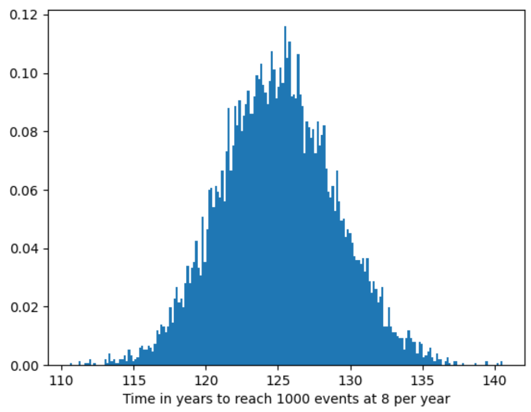

FIG. 3. Time in years to accumulate 1000 events at 8 per year. 10,000 iterations performed to generate histogram.

~ 140. Fig 3 shows this. Therefore, the maximum rate for the NHPP would be $(1.01)^{140} = 4.027$. The rate for the previous homogeneous Poisson process was 8. Hence, this is within the bounds of the original process. Thinning can be applied to simulate the NHPP from the previous HPP data.

VI. PERFORMING THE THINNING ALGORITHM WITH DATA FROM HOMOGENEOUS POISSON PROCESS

Using the thinning algorithm, the data points from the previous homogeneous process are time-sampled to construct a Non-Homogeneous process.

VII. GRAPHICAL COMPARISON OF HPP AND NHPP

Fig 4 showcases the cumulative events $N(t)$ against time t. It is observed that, as opposed to homogeneous Poisson process, the line is not quite linear - the slope increases as time progresses. This can be expected as the rate parameter $\lambda(t) = (1.01)^t$ also increases with time. Due to deletion of points from the original homogeneous process, it is also noted that the total number of points has decreased from 1000 to just over 300.
By integrating the rate function, we get

$$\int_0^t \lambda(t)dt = \int_0^t 1.01^t dt$$
$$= 1.01^t/ln(1.01) + C$$

Noting the boundary condition that at $t = 0$, $E[N(t)] = 0$, we have $C = \frac{1.01^0}{ln(1.01)} = 100.499$.

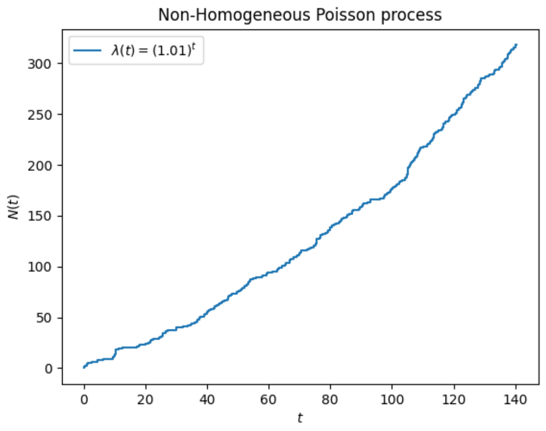

FIG. 4. Accumulated events following a NHPP with rate parameter $\lambda(t) = 1.01^t$.

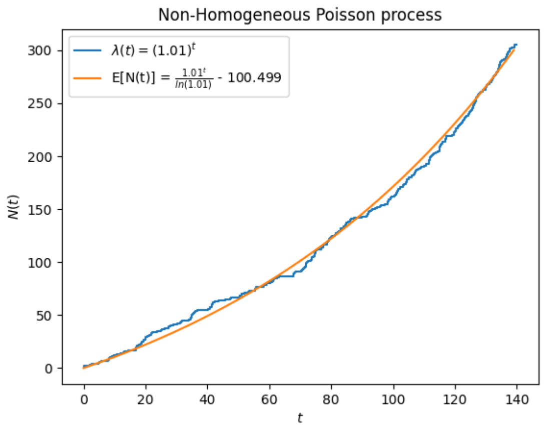

FIG. 5. $E[N(t)]$, denoted by $\frac{1.01^0}{ln(1.01)} - 100.499$ superimposed on the NHPP.

Thus, $E[N(t)] = \frac{1.01^0}{ln(1.01)} - 100.499$. Plotting this onto the earlier graph, Fig 5 is obtained. It can be seen that the orange line closely follows the blue stepped graph.

Appendix A: Code

Below is the code for section IV. It simulates a homogeneous Poisson process with rate parameter $\lambda = 8$ per year. Time to 1000 events is simulated and the graph generated from this code is Fig 1.

```
# Imports
import numpy as np
import matplotlib.pyplot as plt

# define hpp function
def gen_hpp(lmbda ,  N):
    '''
    param lmbda: rate parameter
    N: Number of events
    '''
    # inits
    t = [0]

    # begin loop
    while True:
        # generate uniform r.v. ~ Unif[0,1]
        u = np.random.uniform(0,1)
        # generate w ~ Exponential(lambda)
        w = - np.log(u)/lmbda
        t.append(t[-1] + w)
        # exit condition
        if len(t) > N:
           # get time to event & count the
           # number of events
           return t , np.arange(len(t))

if __name__ == '__main__': # main namespace
    l , N = 8 , 1000
    # generate the time(s) to event(s)
    # AND count of events
    hpp_event_times , events = \
    gen_hpp(lmbda = l , N = N)
    print(events , hpp_event_times) # debug

    # Make plots
    fig , ax = plt.subplots()
    # step graph
    ax.step(hpp_event_times , events ,
     label = f"$\lambda$ = {l}" , lw = 0.5)
    ax.set_xlabel(r'$t$')
    ax.set_ylabel(r'$N(t)$')
    ax.set_title('Homogeneous Poisson process')
    ax.legend(loc='best')
    plt.show()
```

Below is the code for section VI. It simulates a Non-Homogeneous Poisson process with rate parameter $\lambda(t) = 1.01^t$ that smoothly increases from 1 event per year. The time boundary is kept at 140 years as this is roughly the longest time it took the HPP to reach 1000 events. The graphs generated from this code are Fig 4 and Fig 5. In the latter, $E[N(t)]$ is shown along with the realizations from the NHPP to demonstrate that both the lines adhere to each other.

```
# Imports
import numpy as np
import matplotlib.pyplot as plt

# define hpp function
def gen_nhpp(lmbda_bar = 8 ,  T = 140):
    '''
    param lmbda_bar: rate parameter that
    dominates the rate param of the NHPP
    T : maximum time
    '''
    # inits
    s = [0]
    t = [0]

    # begin loop
    while s[-1] < T:
        # generate uniform r.v. ~ Unif[0,1]
        u = np.random.uniform(0,1)
        # generate w ~ Exponential(lambda)
        w = - np.log(u)/lmbda_bar
        s.append(s[-1] + w)
        # geenrate D ~ Unif[0,1]
        D = np.random.uniform(0,1)
        # acceptance criterion
        if D < (1.01)**s[-1] / lmbda_bar:
            t.append(s[-1])

        if t[-1] > T:
            num_events = np.arange(len(t[:-1]))
            print(f'Breakpoint 1: the number of events
            is {num_events[-1]}, and the time taken to \
            reach them is {t[:-1][-1]}')
            # get time to event & count the
            # number of events
            return t[:-1] , num_events
    else:
        num_events = np.arange(len(t))
        print(f'Breakpoint 2: the number of \
        events is {num_events[-1]}, and the \
        time taken to reach them is {t[-1]}')
        # get time to event & count the number
        # of events
        return t , num_events

# main namespace
if __name__ == '__main__':
    # generate the time(s) to event(s)
    # AND count of events
    nhpp_event_times , events = gen_nhpp()
    # Make plots
    fig , ax = plt.subplots()
    # step graph
    ax.step(nhpp_event_times , events ,
            label = f'$\lambda(t) = (1.01)^t$')
    # Integrating the rate function to get E[N(t)]
    x = np.arange(140)
    y = 1.01**x/np.log(1.01) - \
    (1.01)**0/np.log(1.01)
    ax.plot(x , y ,
    label = r'E[N(t)] = $\frac{1.01^t}\
            {ln(1.01)}$ - 100.499')
    # Auxiliaries
    ax.set_xlabel(r'$t$')
    ax.set_ylabel(r'$N(t)$')
    ax.set_title('Non-Homogeneous \
                 Poisson process')
    ax.legend(loc='best')
    plt.show()
```

[1] D. Vere-Jones, Stochastic models for earthquake occurrence, Journal of the Royal Statistical Society: Series B (Methodological) **32**, 1–45 (1970).

[2] F. Gabbiani and S. J. Cox, Stochastic processes, in *Mathematics for Neuroscientists* (Academic Press, 2010) Chap. 16, pp. 251–266.

[3] P. A. Lewis and G. S. Shedler, Simulation of nonhomogeneous poisson processes by thinning, Naval Research Logistics Quarterly **26**, 403–413 (1979).

[4] Y. Chen, Thinning algorithms for simulating point processes.

[5] S. M. Ross, *Introduction to Probability Models*, 10th ed. (Academic Press, 2006).

[6] P. Billingsley, *Probability and measure* (J. Wiley & Sons, 1995).

[7] E. Cinlar, *Introduction to stochastic processes* (Prentice-Hall, 1975).

[8] R. Pasupathy, Generating nonhomogeneous poisson processes.

[9] P. A. Lewis and G. S. Shedler, Simulation of nonhomogeneous poisson processes with log linear rate function, Biometrika **63**, 501–505 (1976).

[10] Poisson point process, `https://en.wikipedia.org/wiki/Poisson_point_process`, accessed: 2023-04-07.