Approximating Poisson Distribution: An Monte Carlo Approach

Abstract

In this notebook, we illustrate how to construct an approximate distribution of Poisson Distribution, we mainly use the Monte Carlo method, we will demonstrate that after many many times of experiments, using the sample data collected from the experiments, we can derive a pretty good approximation of Poisson Distribution which has only a subtle differences in C.D.F. and P.D.F. when the amount of trials is large enough.

Poisson Distribution

A Poisson Distribution describes the distribution of the amount of happens of a specific event within a period, e.g., how many cars just passed a road in 1 hour, how many customers just enter a bank in 30 minutes, how many particles will burst out a radiative source in a period, let's say X cars will pass that road in 1 hour, then, the random variable X obeys a Poisson Distribution, denoted as

$X \approx \text{PoissonDistribution}[\lambda]$

where λ is the hourly average traffic of that road, then given an non-negative integer x, we can compute the probability of "4 cars pass that road in 1 hour", which is

$\Pr(X = 4) = \frac{\lambda^k e^{-\lambda}}{k!} = \left(10^4 e^{-10}\right)/4!$

let's say, in average, there has 10 cars pass that road, that is $\lambda = 10$, so that the probability of "4 cars pass that road in 1 hour" is $\left(10^4 e^{-10}\right)/4!$. Now hopefully you see the basics of Poisson Distribution.

How to Approximate a Poisson Distribution

Let's say that We don't know the exact expression of the cumulative distribution function of Poisson Distribution, and We only know that there's a random variable X which obeys a Poison Distribution of some parameter λ, i.e. $X \approx \text{PoissonDistribution}[\lambda]$, given only this, how can we compute the probability $\Pr(X \leq x)$, where x is a non negative integer ?

Shall notice that there is a basic assumption when We say some random variable X obeys a Poisson Distribution, it is usually assumed that in every interval, the mean amount of happens of that specific event does not change, simply speaking, let's take the car-pass-road case for example, we assumed that no matter which hour in a day is, the expectation of the amount of times the car pass

$\lambda = 4$

that road in that hour are unique, say that λ, the parameter of Poisson Distribution, is $\lambda = 4$, which equivalently means, that, the average traffic in 0 am is 4, and the average traffic in 1 am is 4, and the average traffic in 2,3,4,5, until 22, 23 hour of that day is 4, no matter which hour it is. This is what that basic assumption means. Hopefully you understand.

So, it that basic assumption useful? Yes it is useful. In Monte Carlo way, let's say that there are n intervals in total, that is, whole time range can be split into to n parts, and in each interval, the event are "expected" to happen λ times, which is the parameter of a Poisson Distribution, also the "mean traffic" in the car-pass-road case as we denoted, so after all n intervals gone, the event are "expected to" happened $n\lambda$ times altogether. Then, in Monte Carlo's point of view, each happening of the event (there are $n\lambda$ happenings altogether) can randomly "choose" which interval it "wanted to" happen at, let's take an example, say there is a array of event, which's length is $n\lambda$

events[] = [event[1], event[2], ..., event[n λ]]

Then, for each event in the array events[], say, firstly, it is event[1], then event[1] will randomly choose a interval, in

intervals[] = [interval[1], interval[2], ..., interval[n]]

say that event[1] just randomly picked interval[4], then event[2] randomly picked interval[7], then event[3] randomly picked interval[2], and so on until all event in events[] picked their own interval[i] to happen, then, for each interval in intervals[], we can counts that how many event[i]s just fallen into it, for example, event[2], event[4], event[7], event[11] all take interval[6], then, it will means that there are 4 event[i]s fallen into interval[6], we can count that, it is easy. Same operation perform on every interval, we can have such an table which maps interval[i] to the amount of event[i]s fallen to it, it may seems like below

counts = <| interval[1] -> 7, interval[2] -> 2, interval[3] -> 11, ... |>

then We only take the values from the table, we got

data = [7, 2, 11, ...]

then you will understand that this array, this array named data, shall distributed like Poisson Distribution.

Compute

First we cut the whole time range into $n = 8\,000\,000$ parts, the larger n is, the more accurate the result will be.

In[12]:= **n = 8 000 000**

Out[12]= 8 000 000

Just assign a concrete value to symbol λ, which is $\lambda = 4$, so that we can compute the numerical value, and will be easier to compare and plot. So, you can say that, We are gonna to approximate PoissonDistribution[λ=4].

In[13]:= **lambda = 4**

Out[13]= 4

Then We generate $n\lambda$ random integers that evenly distributed across 1 to n, which correspond to

interval[1] to interval[n] like we just said in previous section.

In[14]:= `sampleData = Ceiling[RandomVariate[UniformDistribution[{0, n}], n * lambda]];`
         向上取整   伪随机变数   均匀分布

Counts "how many events caught" for each interval[i].

In[15]:= `sampleDataCounts = Counts[sampleData];`
                            关联计数

Extract the values part.

In[16]:= `stats = Values[sampleDataCounts];`
                 获取值

Construct the empirical distribution using data we just collected, the array named stats.

In[18]:= `empiricalDistribution = EmpiricalDistribution[stats]`
                                   经验分布函数

Out[18]= DataDistribution [+ ▦ Type: Empirical
                                Data points: 7854033]

Try some value, to see the Pr($X \leq x$) computed from the empirical distribution.

In[26]:= `N[PDF[empiricalDistribution, {0, 1, 2, 3, 4, 5, 6, 7}]]`
             概率密度函数

Out[26]= {0., 0.0745656, 0.148999, 0.199489, 0.199073, 0.159294, 0.105902, 0.0606136}

Try some value, to see the Pr($X \leq x$) computed PoissonDistribution[λ], which is, of course, theoretical.

In[27]:= `N[PDF[PoissonDistribution[lambda], {0, 1, 2, 3, 4, 5, 6, 7}]]`
                         泊松分布

Out[27]= {0.0183156, 0.0732626, 0.146525,
          0.195367, 0.195367, 0.156293, 0.104196, 0.0595404}

Compare the plot of the empirical P.D.F. and the theoretical P.D.F..

In[24]:= `DiscretePlot[`
         离散图

         `{PDF[empiricalDistribution, x], PDF[PoissonDistribution[lambda], x]},`
             概率密度函数   泊松分布

         `{x, 0, 20}, PlotLegends → {"empirical", "theoretical"}]`
                      绘图的图例

Out[24]=
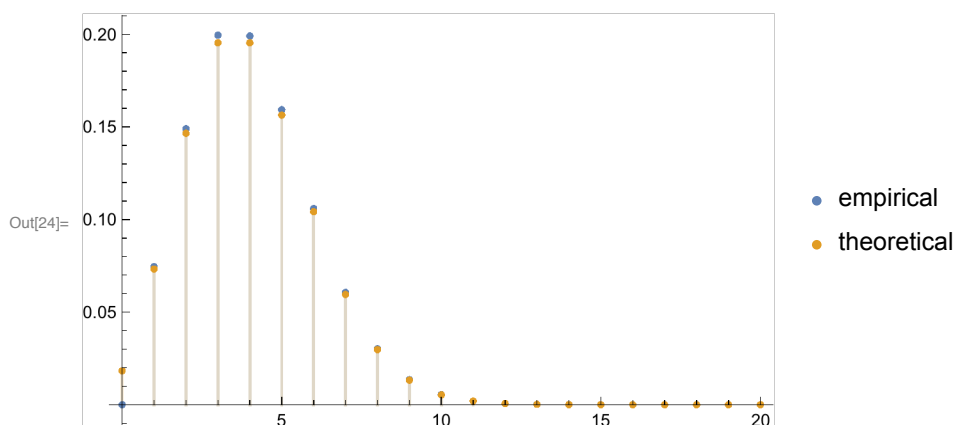
Compare the plot of the empirical C.D.F. and the theoretical C.D.F..

```
In[25]:= DiscretePlot[
         离散图
           {CDF[empiricalDistribution, x], CDF[PoissonDistribution[lambda], x]},
            累积分布函数                                泊松分布
           {x, 0, 20}, PlotLegends → {"empirical", "theoretical"}]
                                     绘图的图例
```
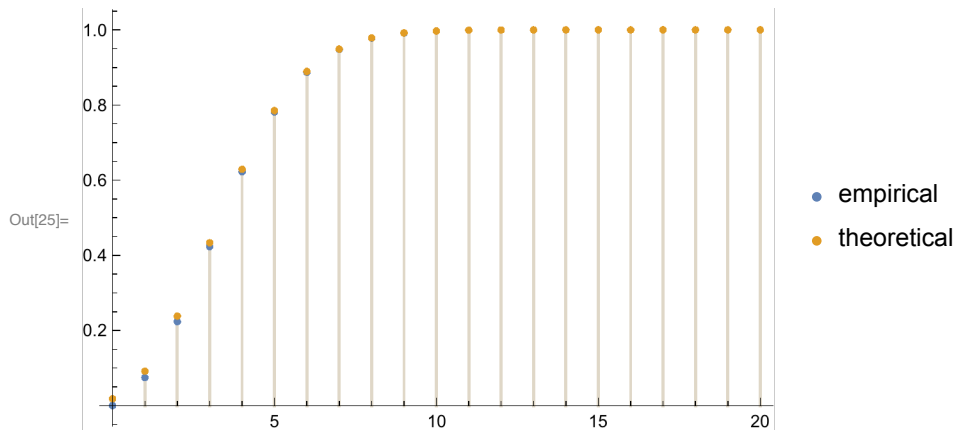
We can find that, there only subtle differences in C.D.F. or P.D.F., because *n* is so large, the empirical distribution and theoretical (which is accurate) PoissonDistribution are nearly fitted into one.