

Analyzing Radioactive Decay with Parameter Estimation

Phsx 815- Project 3

Rakshak Adhikari

March 2021

1 Introduction

This project involves analyzing counts over a fixed time from a radioactive substance, which will be distributed like a Poisson distribution. We will write a total likelihood function for the distribution and maximize it to get the best estimate for the lambda value. We will try this for several different values of lambda to properly explore the parameter space of the distribution.

We are especially interested in the results for small parameter values (λ) because that is where the distribution behaves differently than a normal distribution.

While the code is testing to identify Radioactive substance, the code can be used to test any Poisson Distribution for given lambda values.

2 Theory

Poisson distribution is characterized by the following equation.

$$P(x|\lambda) = \frac{\exp(-\lambda)\lambda^x}{x!} \quad (1)$$

Because we have n number of counts, we have n values for x and hence n functions of Lambda as our likelihood function. Our total likelihood then is simply the product of all those individual likelihood functions.

$$\mathbb{L}_{total} = \prod_{x=x_1}^{x_n} \frac{\exp(-\lambda)\lambda^x}{x!} \quad (2)$$

The factorials do not contribute when we are maximizing the function. So we define a reduced likelihood function that we use in the code denoted by: \mathbb{L}

$$\mathbb{L} = \prod_{x=x_1}^{x_n} \exp(-\lambda)\lambda^x = \exp(-\lambda)\lambda^{\sum_{i=1}^n x_i} \quad (3)$$

The second term in our expression may still be very large, so for optimization, it is much better to maximize the Log of the reduced likelihood function. We note that this is possible because our function is positive everywhere in the domain.

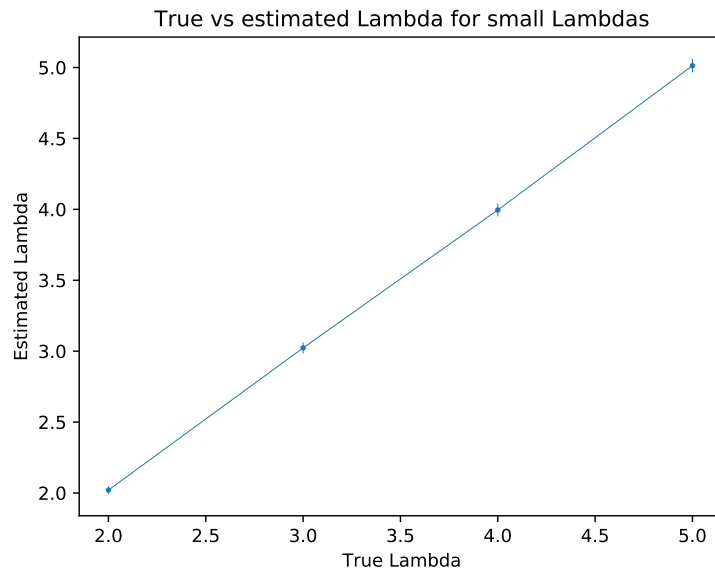
$$\text{Log}(\mathbb{L}) = -n\lambda + \ln(x) \sum_0^n x_i \quad (4)$$

3 Algorithm

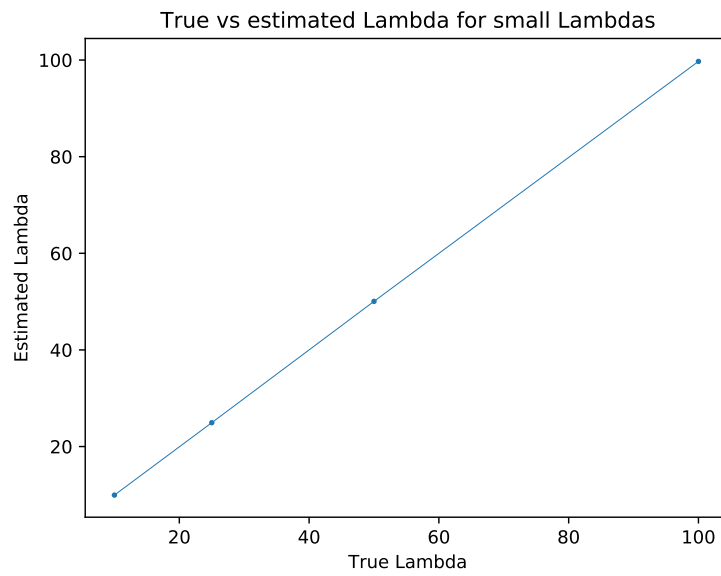
Simulating the Decay: *Simulator.py* simulates the radioactive decay using the python standard Poisson distribution generator. A series of lambda values is supplied as input and a distribution is outputted for each lambda values supplied. The number of measurements is also supplied. For the run, the following values were used:

$$\begin{aligned} \lambda &= [2, 3, 4, 5, 10, 25, 50, 100] \\ \text{no of experiments} &= 10^5 \\ \text{no of measurements per experiment} &= 1 \end{aligned}$$

Analyzing the Output: *Parameterestimator.py* uses the `scipy.optimize` function to minimize the negative log likelihood function for each dataset. It reads all the output text files produced by *simulator.py* and calculates the Likelihood function for each of the parameter values and maximizes the likelihood. It then outputs the maximized parameter value as well as the variance of the estimate. The `glob` function is used to read all the text files iteratively and `matplotlib` is used to plot.



(a) Small λ



(b) Large λ

4 Result

Since we wanted to probe for smaller values of Lambda where the Poisson distribution does not resemble a gaussian, we ran the script in two batches: one for smaller values of lambda and one for larger values of lambda. This allowed us to see the five sigma uncertainty which was small and was not visible if we had plotted all of the data together.

The errors were very small and are tabulated below:

True Lambda	Estimated Lambda	5-Sigma
2	2.0202	0.0275
3	3.0227	0.0389
4	3.9954	0.0446
5	5.0132	0.0481
10	9.9483	0.0704
25	24.9339	0.1107
50	50.0492	0.1585
100	99.7173	0.2409

5 Conclusion

The rate parameter can be estimated using minimization of the total likelihood function.

References

- [1] https://github.com/Raxxak/PHSX815_Project3/blob/main/simulator.py
- [2] https://github.com/Raxxak/PHSX815_Project2/blob/main/ParameterEstimator.py