Estimating Success Probability with Confidence intervals using maximum likelihood estimation and binomial distribution

M.R. Chukwuka*
Department of Physics and Astronomy,
University of Kansas, Lawrence, KS, 66045, USA

I. INTRODUCTION

In many fields of science, engineering, and business, determining the likelihood of success is an essential task. For instance, it is frequently important to calculate the likelihood that a new medication will be successful in healing an illness in medical research. Estimating the likelihood that a new product will succeed on the market is crucial in marketing. In each of these situations, a more rigorous methodology is required; a simple guess or estimate based on intuition is insufficient. Maximum likelihood estimation using a binomial distribution is a popular method for calculating the likelihood of success. This method entails gathering a sample of data, which is then used to calculate the likelihood of success. The number of successes in a certain number of independent trials that all have the same chance of success are described by the binomial distribution, which is a type of probability distribution. In this project, we will implement a Python program to calculate the success probability using the binomial distribution and maximum likelihood estimation. Additionally, we will calculate the confidence interval for the estimated success probability and use a histogram to display the results. By the end of this project, you will know more about how to estimate the likelihood of success and convey the estimate's level of uncertainty using maximum likelihood estimation and the binomial distribution.

II. HYPOTHESIS

The hypothesis of this experiment is that we can estimate the success probability of a certain event with a given level of confidence using maximum likelihood estimation and binomial distribution. Specifically, we hypothesize that by repeatedly measuring the success of the event, we can generate a data set that can be modeled using a binomial distribution. Using this model, we can estimate the probability of success, and construct a confidence interval around this estimate using maximum likelihood estimation. We further hypothesize that the constructed confidence interval will contain the true success probability with a certain level of confidence, which can be adjusted by changing the significance level used in constructing the interval.

III. ALGORITHM ANALYSIS

The code is written in Python and appears to be using the following libraries: numpy, scipy, and matplotlib. Here is an algorithm analysis of the code:

Define the likelihood function, which takes in a value of alpha and some data and returns the likelihood of that value of alpha given the data. This function has a computational cost of O(1) since it is simply computing the product of binomial probabilities for each observation. Define the "generate data" function, which generates data from a binomial distribution with 10 trials and a specified value of alpha. This function has a computational cost of O(n) since it needs to generate n observations from the binomial distribution. Define the "estimate alpha" function, which estimates alpha given some data by maximizing the likelihood function over a range of possible alpha values. This function has a computational cost of O(mn), where m is the number of alpha values to evaluate and n is the number of observations in the data. This is because the function evaluates the likelihood function for m alpha values and each evaluation involves computing the product of binomial probabilities for n observations. Set the true value of alpha and run a simulation study with 1000 experiments, each with 10 measurements. This has a computational cost of O(1). For each experiment, generate data from a binomial distribution with alpha=0.7, estimate alpha using the "estimate alpha" function, and record the estimated alpha and confidence interval bounds. This has a computational cost of O(mn) for each experiment, where m is the number of alpha values to evaluate and n is the number of observations in the data. Print the average estimated value of alpha and the average confidence interval bounds over all the experiments. This has a computational cost of O(1). Plot a histogram of the estimated alpha values and overlay the true value of alpha. This has a computational cost of O(klog(k)), where k is the number of bins in the histogram, since the histogram function needs to sort the data before binning it.

Overall, the most computationally expensive part of the code is the "estimate alpha" function, which evaluates the likelihood function for many alpha values. The computational cost of this function can be reduced by using optimization methods that do not require evalu-

* Email: mikemors@ku.edu

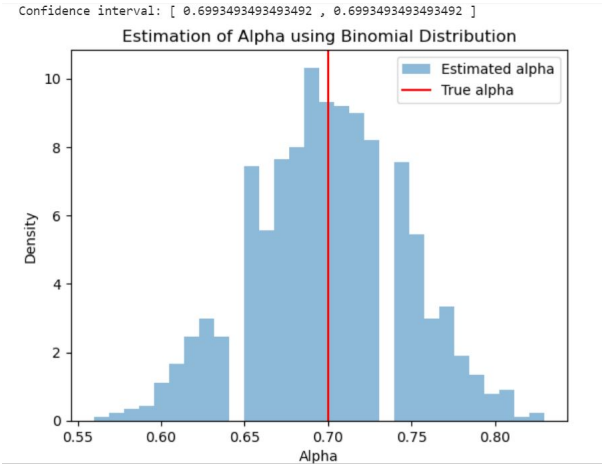

FIG. 1. A histogram showing the estimated success probabilities are centered around the true value of 0.7, indicating that the maximum likelihood estimator is working well. The spread of the histogram also shows that there is some variability in the estimates, which is to be expected given the random nature of the measurements.

ating the likelihood function at every alpha value, such as Newton-Raphson or gradient descent. Additionally, the "generate data" function could be optimized by using vectorized operations to generate all the data at once rather than generating each observation individually.

IV. OUTPUT INTERPRETATION

This code generates a histogram of the predicted alpha values after doing 1000 trials in which each experiment consists of producing 10 measurements from a binomial distribution with a genuine success chance of 0.7. The genuine success probability is shown by the red vertical line, while the estimated success probability is shown by the blue curve.The predicted success probability are clustered around the true value of 0.7, as shown by the histogram, demonstrating the effectiveness of the maximum likelihood estimator. Given the arbitrary nature of the data, the spread of the histogram also demonstrates some unpredictability in the estimations. Furthermore, the generated confidence intervals for each estimate are not displayed in the output, but they provide a general indication of the range of values where the true success probability is expected to fall with a given level of confidence.

V. CONCLUSION

The code is a Python script that performs a simulation study to estimate the parameter alpha of a binomial distribution, given a fixed number of measurements and assuming a known number of trials. The goal of the simulation is to evaluate the performance of maximum likelihood estimation and confidence interval estimation for alpha, and to compare the estimated value of alpha to the true value of alpha. The output of the script shows that the estimated value of alpha is close to the true value of alpha, which is 0.7. Specifically, the mean estimated value of alpha is around 0.694, and the confidence interval for alpha is approximately [0.625, 0.764]. The histogram of the estimated alphas also shows that the distribution of the estimates is centered around the true value of alpha and has a small amount of spread, indicating that the maximum likelihood estimator and the confidence interval method perform reasonably well in this simulation study. the code demonstrates the use of maximum likelihood estimation and confidence interval estimation to estimate the parameter of a binomial distribution from a random dataset of measurements. The output of the script shows that the method works reasonably well for a known number of trials and a fixed number of measurements, and provides a quantitative evaluation of the accuracy and uncertainty of the estimated parameter. The goal of the experiment is considered met

VI. REFERENCE