Phys 342 Lab#8 Due: Nov 23, 13:00

Two rods each of length b are placed on a table to form opposite sides of a rectangle. They are parallel to each other and a distance a apart. A point is chosen randomly on each rod, anywhere along the length of the rod according to the uniform distribution. The distance between the points is the random variable S and it can be shown that the probability density for S is given by:

$$f(s|a,b) = \frac{2s}{b^2}\left(\frac{b}{\sqrt{s^2-a^2}} - 1\right)$$

You do not need to derive this expression. For this exercise use $a = 1$. Retain the variable name a in your code, so that your program is easier to read and so it could be changed easily in the future.

1. Write a simple program that simulates this situation with $b = 4$ (randomly choosing two points and calculating the distance between them) to produce 100 independent values, $s_i, i = 1, \ldots, 100$. At the top of the program, call the seeding method "`np.random.seed(1234)`" to ensure that you will get the same results if you run the program at a later time.
2. Show the distribution of these values in a histogram with bin width $w = 0.1$, with the left edge of the first bin at $s = a$. Overlay on the histogram the probability distribution scaled appropriately (see p. 80). Start the curve at $s = a + w/2$, and use points separated by $w/5$ to define the curve.
3. We will use the maximum likelihood method to estimate the parameter b using a numerical optimizer. Write a function that calculates the negative of the log of the likelihood function. (We use the negative LL, because the optimizer finds the minimum of a function.) The function should have one parameter, the value of b for which the function is calculated. In other words:
```
def negLogLikelihood(b_val):
    statements to calculate the negative logarithm of the likelihood
    return answer
```
4. On a separate plot, show the negative log likelihood function. To avoid values of b for which the likelihood function is zero, first calculate the minimum value of b consistent with the dataset: $b_{\min}$. (This is given by $b_{\min}^2 = s_{\max}^2 - a^2$. You can use `np.amax(array)` to find the maximum value of an array.) Plot the function between $b_{\min} + w/10$ and $b_{\min} + 1$, using points separated by $w/10$ to define the curve.
5. Use methods provided by the scipy library to numerically find the maximum likelihood estimate, $\hat{b}$ and the standard deviation of the estimator, σ_b. Check out the file onePar.ipynb in the Jupyter Lab examples (under estimating parameters) to see how this is done.
6. As an alternative method, analytically differentiate the log likelihood function to find the equation to calculate the second derivative. What value do you get for the standard deviation of the estimator by this method? Is it close to the value found in Q5?
7. Repeat 1000 experiments (each with 100 values s_i) and plot the distribution of $\hat{b}$ (see Q5). Compare the standard deviation of that distribution with the values for $\sigma_{\hat{b}}$ found in Q5 and Q6. Is the maximum likelihood estimator biased?

Report your results in markdown cells and submit the .ipynb file to the course website for grading as usual.