Numerical Recipes: Hand-In 3

Rens Kievit (s1948415)

April 19, 2023

Abstract

In this report we present the problems, solutions and scripts for the exercises from the third handout for the course Numerical Recipes.

Plotting styles in this report are set using the following code

```

def set_styles():
    """For consistent plotting scheme"""
    plt.style.use('default')
    mpl.rcParams['axes.grid'] = True
    plt.style.use('seaborn-darkgrid')
    mpl.rcParams['font.family'] = 'serif'
    mpl.rcParams['lines.linewidth'] = 1.5
```

Listing 1: Matplotlib Plotting Styles

1 Satellite Galaxies Around a Massive Central

In this section we will investigate the spherical distribution of satellite galaxies around a massive central galaxies, and attempt to fit a function to simulated data. Their density distribution n can be described as

$$n(x) = A \langle N_{sat} \rangle \left(\frac{x}{b}\right)^{a-3} \exp\left[-\left(\frac{x}{b}\right)^{c}\right] \tag{1}$$

Here x is the radius relative to the virial radius, i.e. $x \equiv r/r_{vir}$ with $x < x_{max} = 5$. a, b and c are free parameters, $\langle N_{sat} \rangle$ is the mean number of satellites per halo and $A = A(a,b,c)$ normalizes this profile such that $\int\int\int_V n(x)dV = \langle N_{sat} \rangle$. In this work we will mainly look at tt the number of satellites in the infinitesimal range $[x, x + dx\rangle$. This is given by

$$N(x)dx = n(x)4\pi x^2 dx \tag{2}$$

1.1 Maximization

We start by searching for the maximum of the distribution given by equation 2, for this we will assume $a = 2.4$, $b = 0.25$, $c = 1.6$. $x_{max} = 5$, $\langle N_{sat} \rangle = 100$ and $A = 256/(5\pi^{3/2})$. Instead of searching for the maximum, we instead search for the minimum of $-N(x)dx$ which gives the equivalent resulting x. Visually inspecting the distribution (Figure 1), we see a clear peak at $x \sim 0.5$. To be safe we set the edges of our initial bracket at $x_{min} = 0$ and $x_{max} = 5$ and then apply a bracketing algorithm to find a three-point bracket around the minimum. Then we use this bracket as input to the golden section search algorithm to find the x-value at the peak. We find the following brackets and minimization results:

```
Minimization Results
Bracket: [5.00000000e+00 1.00000000e-04 8.09010814e+00]
x at Max: 0.22998248152335327
N(x) Max: 267.84553313361295
```

Listing 2: Results of the maximization algorithm.

We also show the distribution and the exact location of this peak in Figure 1. We can see that the algorithms have perfectly discovered the maximium of this distribution.

1.2 Data Fitting

We import the five datafiles provided for this hand-in which contain satellite galaxies for halos in increasing mass bins ranging from $10^{11}M_{\odot}$ up to $10^{15}M_{\odot}$. Each file contains a number of halos, N_{halo}, and the distances of each satellite galaxy from its center massive galaxy. For each dataset we can immediately compute $\langle N_{sat} \rangle$ by dividing the total number of satellite galaxies by N_{halo}. We bin the data in 20 bins from $x = 10^{-4}$ to $x = 5$ in log-space. We set the lower limit above $x = 0$ for computational reasons because equation 2 is not defined there. We opt to show the distributions in log-space to better visualize the low x region. The decision for 20 bins is relatively arbitrary but appears to result in plots that are easily interpretable. We show the binned data together with N_{halo} for each dataset in Figure 2.

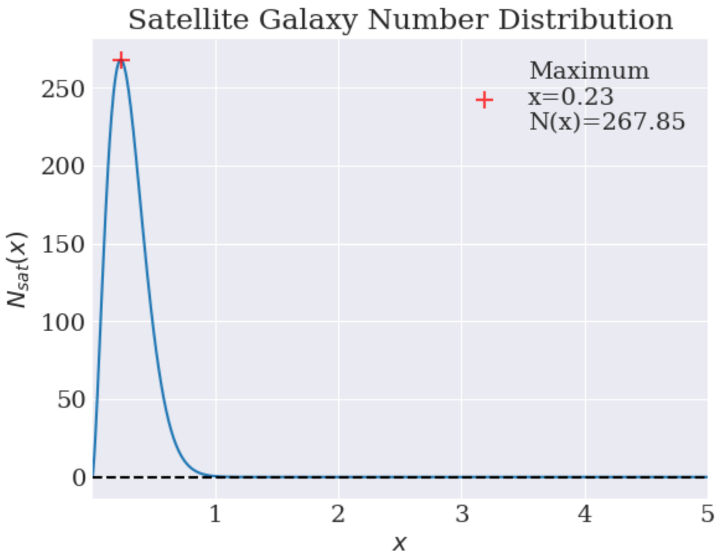

Figure 1: Distribution of the number of galaxies in the infinitesimal range $[x, x + dx\rangle$ described by equation 2. The red cross indicates the position of the maximum of this distribution discovered using the golden search algorihtm as described in Section 1.1

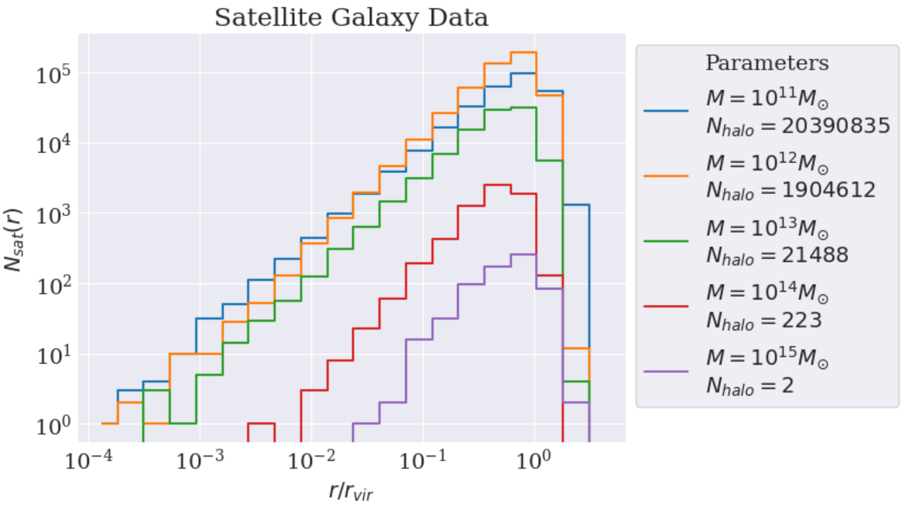

Figure 2: Binned satellite galaxy distance data divided into five mass bins of their massive galaxy halo as indicated on the right. The total number of halos over which the satellite galaxies are distributed is also indicated on the right.

In the next two subsections we will first describe our two approaches to fit equation 2 to these distributions. Because that equation is normalized such that the area under the curve equals $\langle N_{sat} \rangle$, we divide the histograms shown in Figure 2 by N_{halo} to recreate this. In Section 1.3 we present the results of our fitting algorithms applied to the data, combined with a brief statistical analysis.

1.2.1 Chi-Squared

These data are discrete counts, so therefore they should be fit by a Poisson distribution. However we start with an 'easy' χ^2 fit with Poisson variance (i.e. $\sigma^2 = \mu$) to compare to the proper unbiased fit. This means that we want to minimize the function

$$\chi^2 = \frac{(y_i - \mu(x_i|\boldsymbol{p}))}{\mu(x_i|\boldsymbol{p})}. \tag{3}$$

Where x_i and y_i are the bin center and bin counts respectively. μ is a function of the parameter describing the expected value in any particular bin, which is given by

$$\mu\left(x_i|\boldsymbol{p}\right)) = \tilde{N}_i = \int_{x_i}^{x_{i+1}} N(x)dx. \tag{4}$$

We minimize Equation 3 for each dataset separately using the Levenberg-Marquardt algorithm. As a good starting guess we use the same parameters as in Section 1.1, i.e. $a = 2.4$, $b = 0.25$, $c = 1.6$. Another important thing to note is that the normalization constant A is a function of these three parameters which keep changing. We therefore compute a new value for A each time the parameters are shift by first integrating over $N(x)$ from $x_{min} = 10^{-4}$ to $x_{max} = 5$ using Romberg integration and $A = 1$. We then compute A by dividing $\langle N_{sat} \rangle$ by the result of this integral.

1.2.2 Poisson

For the Poisson fit we have two options: we can either use the same binned data as is used for the χ^2 fit, or we can opt to not use bins at all. In this latter case we pretend as though we did bin the data, however with bin size sufficiently small such that each bin contains either 0 or 1 objects. Doing this allows us to make better use of the data at the cost of more computational load. In the worst case, namely the first dataset, this difference is a factor on the order of 10^4. Unfortunately while implemented, we have been unable to use this binless fitting as computing time exceeds the maximum alloted time of ten minutes. Therefore we opt to fit the data here using the same bins as in the previous section. The log-likelihood of a Poisson distribution is given as

$$-\ln \mathcal{L}(\boldsymbol{p}) = -\sum_{i=0}^{N-1} \left(y_i \ln\left[\mu(x_i|\boldsymbol{p}))\right] - \mu(x_i|\boldsymbol{p})) - \ln(y_i!)\right). \tag{5}$$

When minimizing this equation we can ignore the last term, $\ln(y_i!)$ because it is independent of the parameters we are fitting and therefore constant. We are then left with only the first two terms in the sum. For the minimization we will make use of the downhill simplex method where we initialize one of the vertices of the first simplex using the same parameters as before ($a = 2.4$, $b = 0.25$, $c = 1.6$). We initialize the other three vertices by adding one to parameter i for vertex i.

We also attempted to implement the Quasi-Newton method to minimize the log likelihood function, however this also has proven unsuccesful due to extremely steep gradients which often led the line minimization algorithm to regions in parameter space where the likelihood function is not defined. This only led to NaN values as a result for the parameters. While not used, the code is still included at the end of this report.

1.3 Fit Results