PS2: Introduction to Probablity and Statistics

Deepthi Gorthi
AY250: Stellar Populations

September 26, 2016

1 Problem 1

There is a minor mistake on the Initial Mass Function page of Wikipedia. What is it?
Solution:

I found at least two possible errors on the Wikipedia page for the IMF:

- Salpeter power law: The exponent quoted for the Salpeter power law is 2.35 whereas his paper from 1955 reports an exponent of only 1.35.
- Miller Scalo 79 IMF: In the initial mass function figure, the Miller Scalo '79 IMF is shown to flatten out from $\sim 1M_{\odot}$ while the IMF reported in their paper rises upto 10^2 at $0.1M_{\odot}$.

2 Problem 2

Consider a single power-law IMF of the form:

$$P(M|\theta) = c\, M^{-\alpha} \tag{1}$$

where

$$c = \frac{1}{\int_{M_{min}}^{M_{max}} M^{-\alpha} dM} \tag{2}$$

and $\theta = (M_{min}, M_{max}, \alpha)$.

For simplicity, assume perfect knowledge of the masses and that observational effects are negligible.

(a) Write code that generates a list of N stellar masses between a given M_{min} and M_{max} from a power-law distribution with an index of α.

(b) Write code that will perform inference on the set of fake data you generated in part (a) using `emcee`.

(c) Generate a fake dataset assuming $M_{min} = 3\,M_{\odot}$, $M_{max} = 15\,M_{\odot}$, $N = 1000$ and $\alpha = 1.35$. In your inference code, let α and M_{max} be free parameters (but fix $M_{min} = 3M_{\odot}$). Given this fake dataset, to what precision can you constrain α and M_{max}?

(d) Show how the precision to which α and M_{max} can be recovered depends on N, from $N \sim 10$ to $N \sim 10,000$. Summarize your results in plots. It is OK to discretely and uniformly select values of N in $\log_{10}$ space (e.g., $\log_{10}(N) = 1, 2, 3, 4$). *Hint: In the limit that N is a small number, you may want to generate multiple datasets to verify the fidelity of your confidence intervals, as stochastic effects can be important.*

Solution:

The code for this section is included in the repository as `powerlawimf.py`

(a) I learnt an interesting lesson through this exercise, which I should have realised through basic mathematics. The numpy power law distribution does not allow one to sample from a power law with a negative exponent. My hack to getting a negative exponent was to reflect the positive power law about the y axis and scale these to the M_{min} and M_{max} specified, i.e. `(Mmin-Mmax)*np.random.power(alpha,N)+ Mmax`. Though the histogram obtained at the naive first glance appeared like a power law, my failed attempts at solving the rest of the problem made me question this method.

I made a simple plot to compare the scaled positive exponent distribution and the actual negative exponent distribution, which is show in figure 1. I relearnt the valuable lesson from eighth grade that reflection about the y-axis just flips the sign of x, which will not change a positive power law distribution to a negative power law.

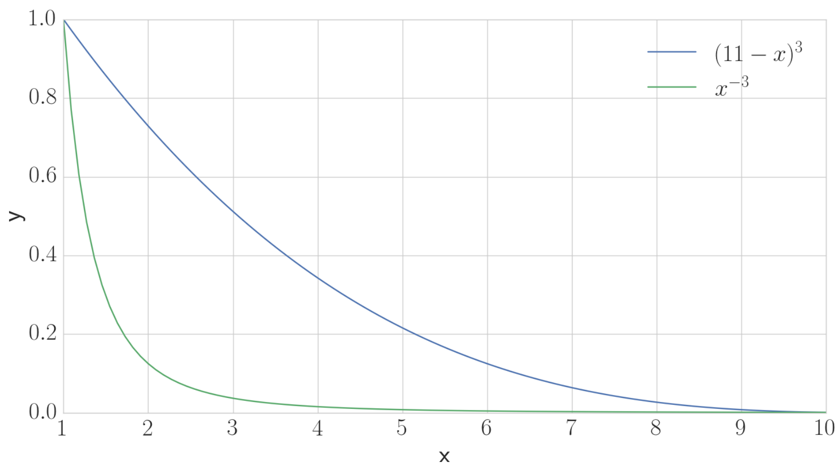

Figure 1: Comparision between the 'hacked' negative exponent distribution and an actual negative power law distribution. Reflecting a positive power law about the y-axis does not result in a negative power law distribution.

Then I implemented the actual (tougher) concept of *inverse transform sampling*, which takes random samples from a uniform distribution between $[0,1)$ and converts them into samples from a arbitrary PDF given that the CDF of this distribution is invertable.

The given PDF is:

$$P(M|\theta) = cM^{\alpha} \quad \forall x \in [M_{min}, M_{max}] \quad a < 0 \tag{3}$$

Some simple algebra gives the CDF of this distribution as:

$$F(x) = \begin{cases} 0 & x < M_{min} \\ \frac{x^{\alpha+1} - M_{min}^{\alpha+1}}{M_{max}^{\alpha+1} - M_{min}^{\alpha+1}} & M_{min} \leq x \leq M_{max} \\ 1 & x > M_{max} \end{cases} \tag{4}$$

The inverse of this distribution is given in equation 5 where u is a uniform random variable.

$$H^{-1}(u) = \left[M_{min}^{\alpha+1} + \left(M_{max}^{\alpha+1} - M_{min}^{\alpha+1}\right) u\right]^{\frac{1}{\alpha+1}} \tag{5}$$

Implementing this to drawn the N masses between $M_{min} = 3\,M_\odot$, $M_{max} = 15\,M_\odot$ with $\alpha = -1.35$ gave the right distribution (see figure 2).

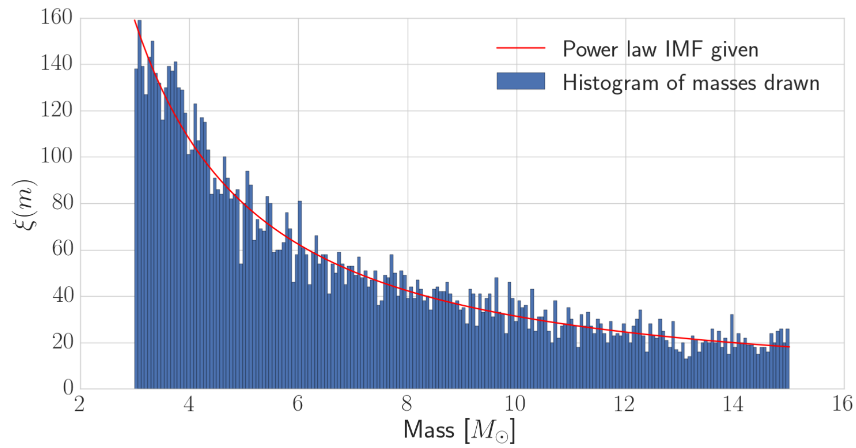

Figure 2: Histogram of the masses drawn from the negative power law cM^{-alpha} using inverse transform sampling, as compared to actual PDF.

(b) The trickiest part of this question was constructing the log-likelihood function. Initially I tried computing the probability of drawing the given masses, for the parameters $\theta = (M_{max}, \alpha)$ using the power law function (eq 3) but this did not work for some reason. The value of this estimator for the 'right' parameter set was somewhere inbetween the actual maximum and minimum log-probability that `emcee` reported. Hence I concluded that this might not be the right estimator and resort to fitting the histogram in $\log - \log$ space which would just be a straight line.

(c) This was the most troublesome, discouraging and teaching part of the question.

The log-likelihood estimator I described above, yielded the right value of the slope after correcting a few silly bugs but the reported M_{max} was very close to M_{min}. This took a few hours of serious mulling over! When I tried to infer M_{max} from the intercept of the fit, I realised that the intercept was itself wrong. Yet, I had normalised the CDF from which I drew my random masses, which should have resulted in a normalised histogram– did it? No!

Even if masses were drawn from a normalized power law, the resulting histogram of the masses was not normalized. Stranger still, the area under the curve varied with the number of bins I was trying to use. Only for very large number of bins was the area converging to 1, but hold your horses before you make bins=N. Histograms of power law distributions are not very indicative at the high mass end, intefering with a good fit for α. So the best resolution to this problem was re-normalizing the histogram before fitting for α and M_{max}.

It was a moment of euphoria to see the corner plot (figure 3) finally (finally!) converge on M_{max} and α of nearly the right values. `corner` seemed very confident with the estimate of the parameters but the autocorrelation time reported by `emcee` was ~ 25 prompting me to manually compute the confidence interval. The 25th and 75th percentiles of M_{max} and α are $(11.1, 11.2)$ and $(-1.29, -1.30)$ respectively.

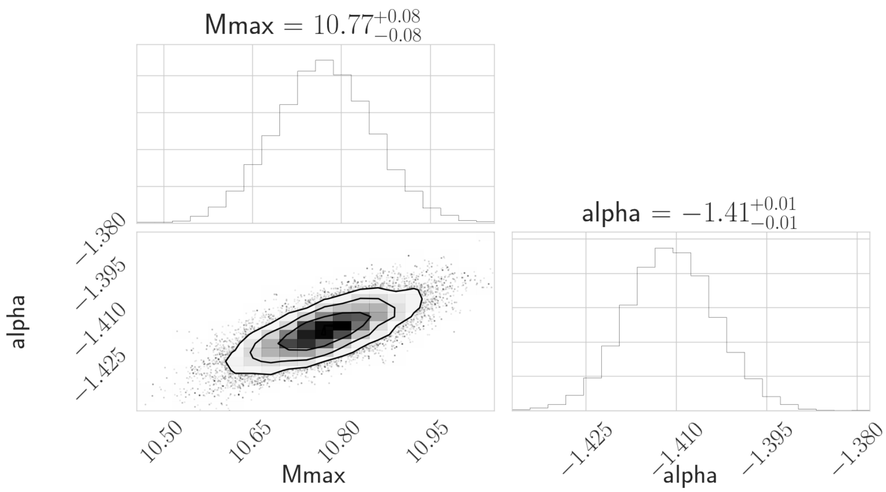

Figure 3: The corner plot obtained after normalizing the area under the histogram with 10 bins and running `emcee` for 500 steps using 300 walkers. The resulting values are close to the actual parameters of $\alpha = -1.35$ and $M_{max} = 15$.

After this experience it would have been worthwhile to check if the original estimator I tried would work if the area under the power law was normalized, but I did not have time to try this again.

(d) The plots obtained for different number of masses generated are shown in figures 4, 5, 6. As expected, the errors increase as the number of samples keeps getting smaller. The reported parameters for the N=10 case are so grossly erroneous that it makes me very skeptical of high mass IMF estimates, since that is the region which is most hurt by lack of data.

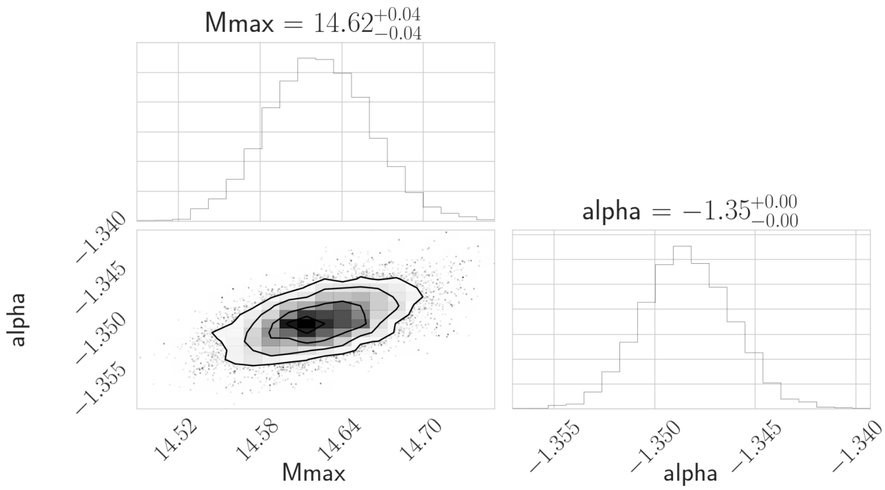

Figure 4: The corner plot obtained after normalizing the area under the histogram (bins=100) and running `emcee` for a 100 steps with 200 walkers. The reported parameters for $N = 10000$ samples are very close to the actual values though the errors estimated might be incorrect.

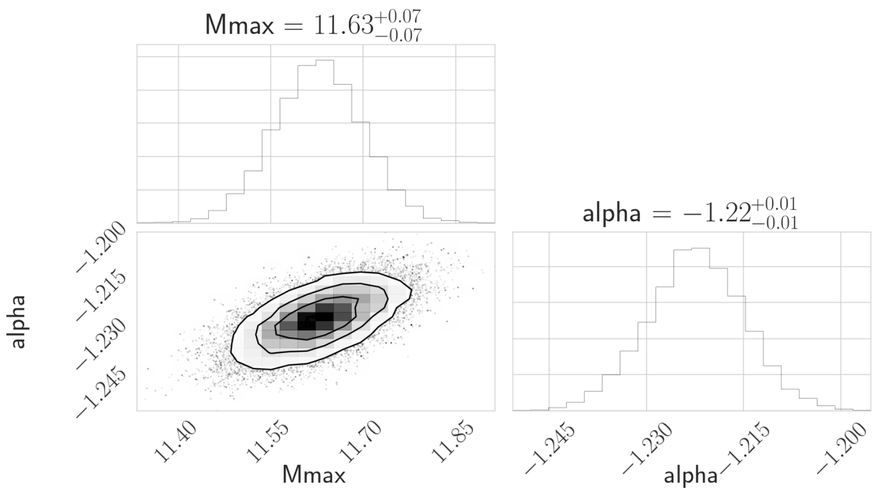

Figure 5: Corner plot obtained by using $N = 100$ samples with 10 bins, 300 walkers and 500 steps. The reported slope is off by 0.2 and the maximum mass by $5M_{\odot}$ making you wonder how scientists agree on the IMF even as much as they do!

3 Problem 3

In this problem, we will attempt to re-create Salpeter's original IMF measurement.

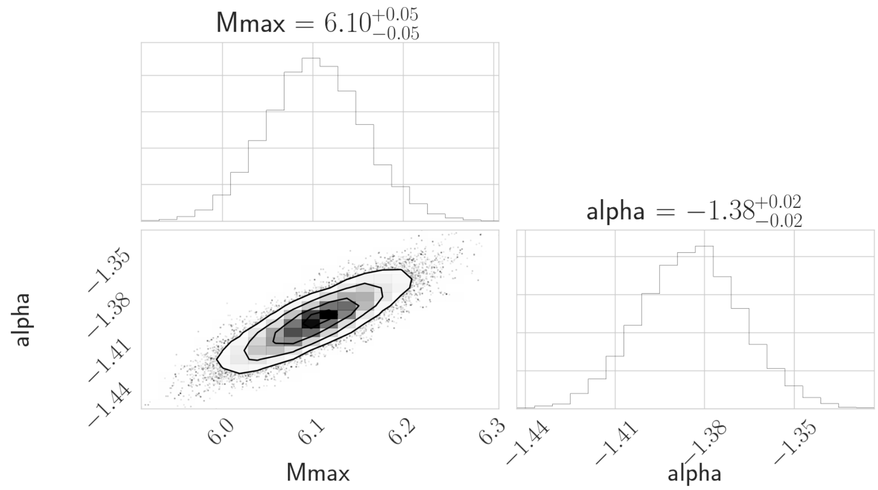

Figure 6: Corner plot obtained for $N = 10$ samples with 3 bins, 300 walkers and 500 steps. The results are completely off as you would expect, but unfortunately you sometimes have to try to infer the power law slope with such less data.

(a) Using the data for mass and number density in Table 2 and/or Figure 2 in Salpeter (1955), fit a power-law using an optimizer or least squares fitter (e.g., `scipy.optimize`).

(b) Same as part (a) only using your own inference code and `emcee`. Compare the two results: How close are they to one another? How close are they to the value reported in Salpeter (1955)?

Solution:

The code for this section is included in the repository under the very intuitive name of $\boxed{\texttt{q3.py}}$

(a) This part was relatively straight forward. I used `scipy.optimize.least_squares` to find the optimum value for the slope and intercept of the data in $\log - \log$ space. The slope reported by scipy was $\boxed{-1.40}$ as compared to Salpeter's slope of -1.35.

(b) This part was very easy as compared to the last question. `emcee` converged right away with a 100 walkers making 1000 steps to give a slope of -1.39. As expected, `scipy` and `emcee` are closer than Salpeter's estimate, though his value comes very close when you remember that his work was from

1955! Figure 7 compares the three results to each other and the original data used.

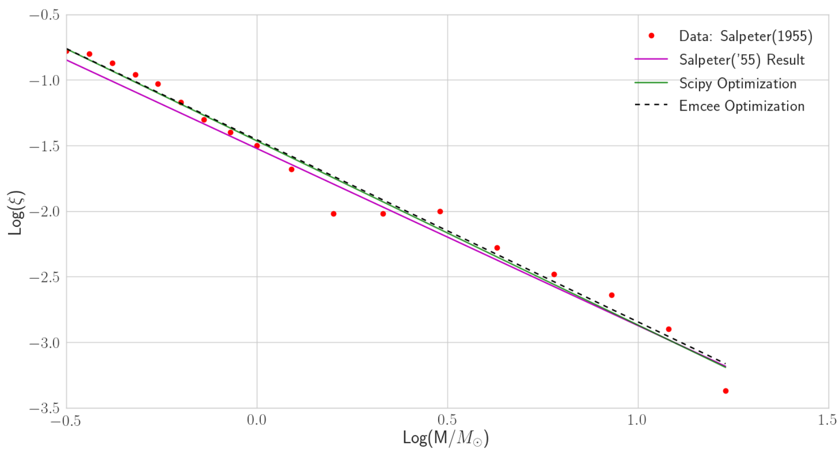

Figure 7: Comparision between the power law slope inferred by Salpeter in 1955, the modern day optimizing algorithm under the `scipy.optimize` package, and the MCMC algorithm implemented by `emcee`. The two 21st century algorithms agree better with each other than with Salpeter's calculations but the difference is almost negligible given the present day disagreements on the IMF.

4 Problem 4

There are claims in the literature that the low-mass IMF slope may systematically deviate from the Galactic value in low-mass dwarf galaxies (e.g., Wyse et al. 2002; Geha et al. 2013). However, these measurements are made over a fairly limited mass range (usually $\sim 0.5 - 0.8\, M_\odot$) and done so assuming that a power-law is a reasonable approximation for the low-mass IMF.

Suppose the true IMF for stars with $M < 1M_\odot$ in all galaxies is actually a Chabrier IMF, i.e., a log-normal at low-masses. Ignoring corrections for stellar multiplicity, this IMF has the functional form:

$$\xi(m)\Delta m = \frac{0.15}{m} \exp \frac{-(log(m) - log(0.08))^2}{(2 \times 0.69^2)} \tag{6}$$

(a) Using the Chabrier IMF from above, generate a list of N=10,000 (perfectly known) stellar masses between 0.5 and 0.8 $M_\odot$.

(b) Now, assuming a single-slope power-law IMF model (as done in the literature), infer the value of the spectral index α. How does this compare with the canonical Kroupa IMF found in the Milky Way?

Solution:

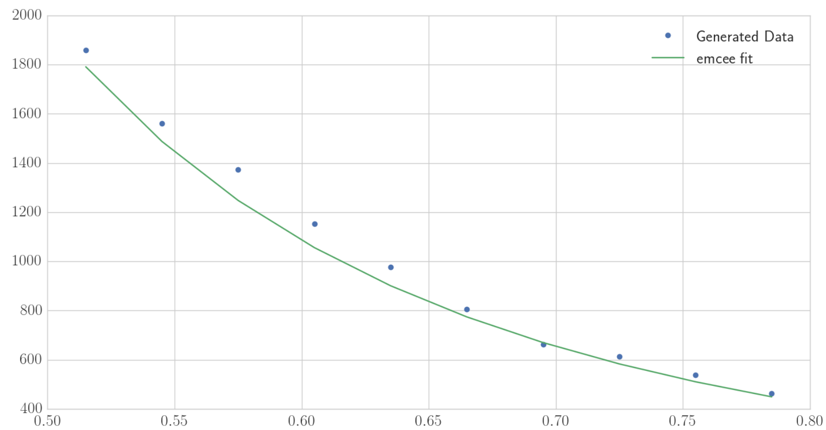

Figure 8

5 Problem 5

Using python-FSPS:

(1) Generate the spectrum for a 10 Myr simple stellar population (assume no dust, fixed metallicity, etc – the only variable of interest is age). Plot how the spectrum from $1500 - 10000$Å changes for three different high-mass IMF ($> 1\,M_\odot$) values: $\alpha = 0.8, 1.3, 1.8$, holding the lower portions of the IMF fixed.

(2) Generate the spectrum of a 10 Gyr simple stellar population. Plot how the spectrum from $5000 - 20000$Å changes for three different IMF forms: Salpeter IMF, a Kroupa IMF, and the van Dokkum IMF.

Solution

(1)

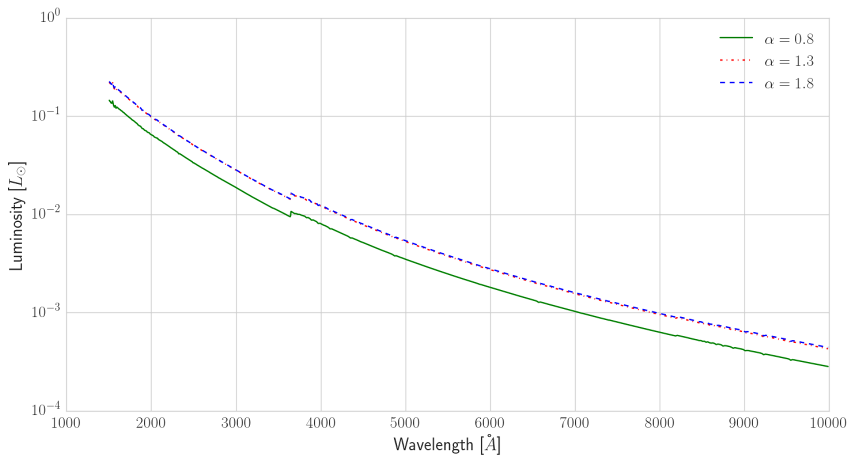

Figure 9

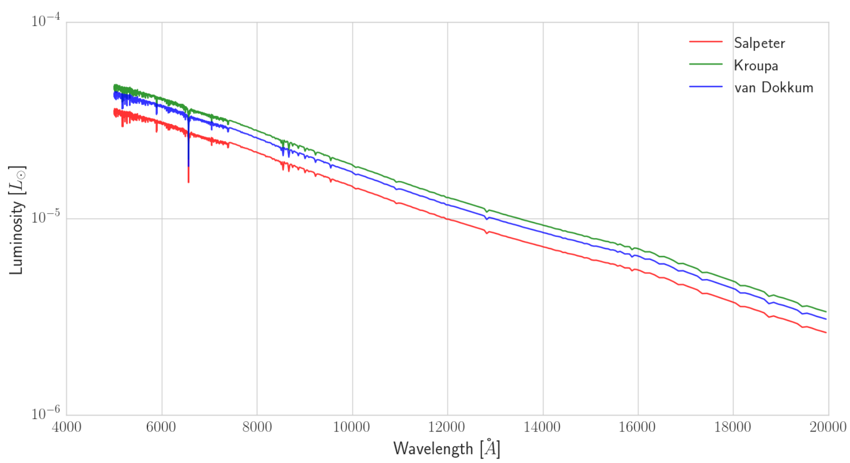

Figure 10