Statistics for Astronomers
Solutions to Homework #8

Prof. Sundar Srinivasan

January 17, 2021

Note: the solutions below use the script `hw8.py`.

1. (a) The module `hw8q1` propagates the uncertainties and generates a distribution of flux values, from which the mean, median, and mode are computed.

   Note that direct application of `scipy.stats.mode` to the flux values always returns the lowest flux value regardless of sample size in this case. I'm not sure, but this could be due to the module detecting a multimodal distribution. As a workaround, I use the `KDEUnivariate` method from the `statsmodels.nonparametric.kde` package to compute a KDE for the distribution of flux values and then identify the location of the maximum as its mode. The code prints out
   `The mean, median, and mode flux in mJy are 0.871, 0.801, and 0.731 respectively.`

   Fig. 1 shows the resulting flux distribution in KDE form as well as histograms computed using Knuth's method as well as Bayesian blocks. The mean, median, and mode are also shown. Note that the KDE predicts values below zero, which are not physical.

   (b) The code prints out
   `The equal-tailed interval is [0.478, 1.236].`
   Fig. 1 shows the interval as a horizontal line.

2. (a) Given the large dynamic range over which the function is to be evaluated, in the code `hw8q2` we construct a logarithmic grid for x with $N = 10000$ values between $\log x = -1.5$ and $\log x = 2$. We then evaluate $p(x)$ for this grid.

   The mean of the proposal distribution $q(x)$ is chosen to be at the same location as the mode of the target distribution $p(x)$. This makes it easier to select the scale in the next part of the problem. The code prints out
   `Mean for q(x) = location of maximum of p(x) = 2.821.`

   The target distribution is heavily skewed to the right of its mode, so we set the standard deviation of $q(x)$ as follows. As we want $q(x) > p(x)$ for all $x > 0$, we first find the x for which $p(x)$ has fallen to $e^{-3^2/2}$ of its maximum value. For a Gaussian, this corresponds to the location of the 3σ point. We then set the standard deviation of $q(x)$ to one-third of the distance between this x value and the location of the maximum. The code prints out
   `Stdev for q(x) = 1/3*(dist.  from mode to point where target is exp(-0.5*3**2)) = 2.879.`

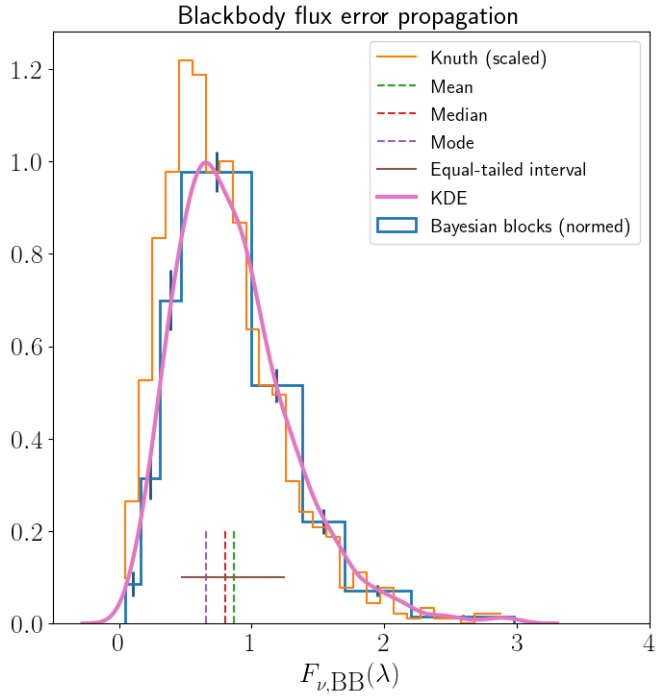

Figure 1: The distribution of blackbody fluxes resulting from propagating the errors in the radius, distance, and temperature, in the form of a kernel density estimate (*pink*) and histograms computed using Knuth's method (*orange*) and Bayesian blocks (*blue*). For comparison, the Knuth histogram has been scaled to its maximum value, and the Bayesian blocks version is normalised so it integrates to unity. The Bayesian blocks histogram also displays errors in bin values. Vertical dashed lines show the locations of the mean (*green*), median (*red*), mode (*purple*), and the horizontal line is the equal-tailed interval.

An important point to consider in the rejection sampling procedure is that draws from the proposal distribution (a Gaussian) may result in negative x values. **The rejection step must check for this in addition to comparing the function evaluation to a random uniform number**, otherwise the target distribution won't be faithfully reproduced.

(b) Now that the mean and standard deviation of $q(x)$ have been determined, we ensure that $q(x) > p(x)$ for all values of x significantly different from zero. We scale $q(x)$ such that its peak is slightly (5%) higher than $p(x)$:
`Scale factor = max(px)/max(qx)*1.05 = 1.659`
Fig. 2 shows a comparison of the target and proposal distributions.

(c) The required fraction is

$$\frac{\int_0^\infty dx\ q(x) - \int_0^\infty dx\ p(x)}{\int_0^\infty dx\ q(x)} = 1 - \frac{\int_0^\infty dx\ p(x)}{1.659 \int_0^\infty dx\ h(x)} = 1 - \frac{1}{1.659} \approx 0.397. \tag{1}$$

In the above, $h(x)$ is the probability distribution for the normal distribution with mean and standard deviation computed in the previous sections. The fractional area computed above is a

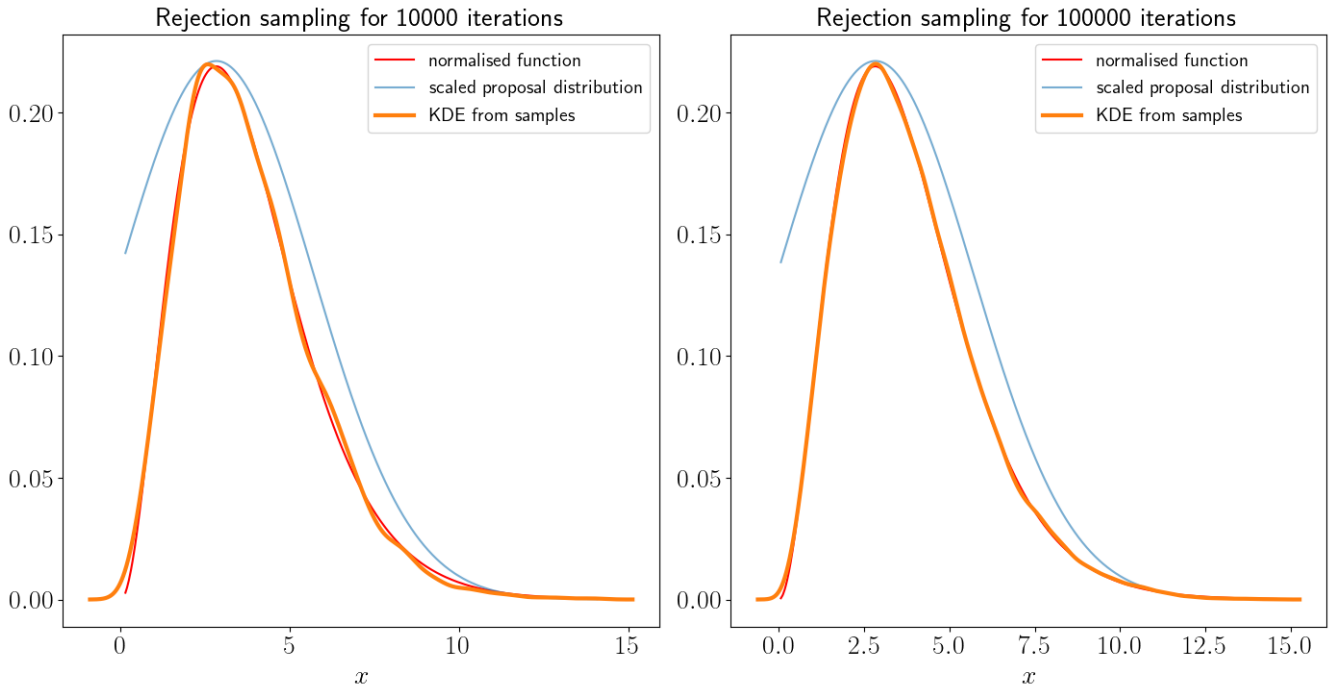

Figure 2: Results of rejection sampling using 10 000 (*left*) and 100 000 (*right*) iterations. In each plot, we compare the result of the rejection sampling (*shown as a kernel density estimate, orange*) to the target distribution (*red*) and the scaled proposal distribution (*blue*).

function only of the factor by which the normal distribution is scaled up to compute the proposal distribution.

(d) The code in `hw8q2` performs rejection sampling, and also prints out the percentage of points rejected:
`Approximately 37.0% of points rejected.`
Note that this percentage only depends on the relative shapes of the target and proposal distributions, once the scale factor has been chosen. This can be verified by changing the number of iterations in the code. The large fraction of rejected points is due to the fact that we are removing any negative x values from the draws.

(e) Fig. 2 compares the target distribution to the KDE of the distribution after rejection sampling. There is very good agreement overall. The KDE has some small-scale structure which is smoothed out when the number of iterations are increased.