Error Computation for Emulation

April 27, 2020

1 Introduction

Our goal is to compute the likelihood function used to estimate model parameters with our emulator. The log-likelihood $\mathcal{L}$ can be defined as

$$\mathcal{L} = -\frac{1}{2}\left(\frac{y_{\text{pred}} - y_{\text{obs}}}{y_{\text{obs}}}\right)^{\text{T}} C^{-1} \left(\frac{y_{\text{pred}} - y_{\text{obs}}}{y_{\text{obs}}}\right) \tag{1}$$

where C is the covariance matrix, and y is the clustering statistic we are emulating. Here y_{obs} is the statistic computed on a mock or data, and y_{pred} is the emulator prediction for that statistic. We take y to have P dimensions (radial bins in our case), and C to be a $P \times P$ matrix.

This document outlines how we compute this covariance matrix C, as well as other error measurements used in our emulation procedure.

2 Notation

The test set contains mock catalogs with $C = 7$ different cosmologies c. There are $B = 5$ boxes (or realizations) b for each cosmology. These are each populated with $H = 100$ different halo occupation distribution (HOD) models h, the same 100 for each cosmology. (There are also 10 random seeds used per HOD, but for now we only use one. We'll worry about this later.) The test boxes all have $L_{\text{test}} = 1.05$ Gpc. Covariances scale as the volume (covariance decreases with increasing volume; Klypin & Prada 2018) so we will scale all covariances to that of a single test

All of the covariance matrices in this document are fractional errors. This is consistent with our definition of the likelihood function, which uses the fractional error between the observation and the emulator prediction. We note that generally, it is not obvious whether to divide the physical error by the observation or the prediction to obtain the fractional error. In this case, the prediction is trained on many observations, so the observation y_{obs} is the more well-measured quantity and we choose it for the denominator.

The statistic is computed on all $7 \times 5 \times 100 = 3500$ mocks. The statistic is averaged over the 5 boxes, so our final test set contains $N = 700$ statistics. We will denote these with index n.

3 Test set error

We are interested in estimating the sample variance of a single simulation, as an estimate of the test set error. We compute this as follows. Choose a single HOD in the middle of the parameter space. Here we choose $h = 3$. For each cosmology, we compute the mean statistic $\bar{y}_c$ of the B boxes.

$$\bar{y}_c = \frac{1}{B} \sum_b^B y_{b,c} \tag{2}$$

We compute the deviation from this mean for each of box b, for a given cosmology c. This is defined as the fractional error between the box statistic and the mean statistic for that cosmology,

$$d_{b,c} = \frac{y_{b,c} - \bar{y}_c}{\bar{y}_c}. \tag{3}$$

We finally compute the covariance of these $B \times C$ deviations from the mean,

$$C^{\text{test}} = \frac{1}{BC - 1} \sum_b^B \sum_c^C d_{b,c} \cdot d_{b,c}^{\text{T}}. \tag{4}$$

Note that the expectation value of deviations from the mean should be zero, so we assume $\bar{d}_{b,c} = 0$ when computing the covariance. C^{test} is a matrix of size $P \times P$.

The square root of the diagonal of this matrix (a.k.a. the standard deviation) is used as an input to the Gassian Process to represent the error on the *training set*. We do this because have a better handle on this sample variance error using the test set as it has multiple realizations per cosmology, compared to using the training set which only has a single realization per cosmology. We note that the random seeds, which we have not yet included, estimate the shot noise contribution; we will update with this later.

4 Emulator performance and error

The overall emulator performance covariance C^{perf} is defined as

$$C^{\text{perf}} = C^{\text{emu}} + C^{\text{test}} \tag{5}$$

We obtain this by computing the covariance of the fractional error between the emulator prediction and the observation. Here, the observation $y_{n,\text{obs}}$ is the mean of the B boxes for each cosmology and HOD model, so we effectively have B times the volume. Thus when we compute the covariance, we need to scale it down by a factor of B to match the test set, a.k.a. our error will increase. The performance covariance is then:

$$f_n = \frac{y_{n,\text{pred}} - y_{n,\text{obs}}}{y_{n,\text{obs}}} \tag{6}$$

$$C^{\text{perf}} = B\left(\frac{1}{N-1}\right)\sum_{n}^{N} f_n \cdot f_n^{\text{T}}. \tag{7}$$

Once again we know the expectation value of these fractional errors should be zero, so we assume $\bar{f}_n = 0$ when computing the covariance.

We care about the emulator covariance C^{emu} separately from the test covariance, so we compute it using

$$C^{\text{emu}} = C^{\text{perf}} - C^{\text{test}}. \tag{8}$$

Both C^{emu} and C^{perf} are matrices of size $P \times P$.

5 Measurement covariance with Minerva

The covariance between the bins of the statistic is obtained with the Minerva mock catalogs. These are larger than the test simulations, with side length $L_{\text{minerva}} = 1.5$ Gpc. We compute the statistic on $M = 100$ Minerva mocks, which we index by m. The Minerva covariance matrix $C^{\rm minerva}$ is a $P \times P$ matrix, given by

$$\begin{aligned} d_m &= \frac{y_m - \bar{y}}{\bar{y}} && (9) \\ C^{\rm minerva} &= \left(\frac{L_{\rm minerva}}{L_{\rm test}}\right)^3 \left(\frac{1}{M-1}\right) \sum_m^M d_m \cdot d_m^{\rm T} && (10) \end{aligned}$$

where $\bar{y}$ is the mean of all M statistics. Note that we have rescaled the covariance by the ratio of volumes of the test simulation and the Minerva simulation (Minerva errors will be smaller because it has more volume, so we need to inflate these).

We combine these errors to obtain a final covariance matrix C for our likelihood. The final covariance matrix is

$$C = C^{\rm minerva} + C^{\rm emu}. \tag{11}$$