Error Computation for Emulation

March 3, 2021

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

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

where $C_{\mathcal{L}}$ is the covariance matrix, and y is the clustering statistic we are emulating. Here y_{data} 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 $C_{\mathcal{L}}$ to be a $P \times P$ matrix.

This document outlines how we compute this covariance matrix, which we define as

$$C_{\mathcal{L}} = C_{\text{emu}} + C_{\text{data}} \tag{2}$$

where C_{emu} is the covariance of our emulator (defined below) and C_{data} is the covariance of the data set on which we are performing parameter recovery. These and all other covariance matrices in this documenet are of size $P \times P$.

To be consistent with our definition of the likelihood function, which uses the fractional error between the observation and the emulator prediction, all of the covariance matrices in this document are fractional errors. 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_{test} is the more well-measured quantity and we choose it for the denominator.

2 Emulator covariance

The overall emulator performance covariance C_{perf} is defined as

$$C_{\text{perf}} = C_{\text{emu}} + C_{\text{test}} \qquad (3)$$

where C_{test} is the covariance of the data you are testing the emulator on, or an estimate of this error (more on this later).

We obtain C_{perf} by computing the covariance of the fractional error between the emulator prediction and the observation. The performance covariance on a set with N observations indexed by n is then:

$$f_n = \frac{y_{n,\text{pred}} - y_{n,\text{test}}}{y_{n,\text{test}}} \qquad (4)$$

$$C_{\text{perf}} = \frac{1}{N-1} \sum_n^N f_n \cdot f_n^{\text{T}}. \qquad (5)$$

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

$$C_{\text{emu}} = C_{\text{perf}} - C_{\text{test}}. \qquad (6)$$

3 Aemulus simulation error

The Aemulus 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 ones 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 statistic is computed on all $7 \times 5 \times 100 = 3500$ mocks. The test boxes all have $L_{\text{aemulus}} = 1.05$ Gpc.

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.

$$\bar{y}_c = \frac{1}{B} \sum_b^B y_{b,c} \qquad (7)$$

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,

$$d_{b,c} = \frac{y_{b,c} - \bar{y}_c}{\bar{y}_c}. \tag{8}$$

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

$$C_{\text{aemulus1box}} = \frac{1}{BC - 1} \sum_b^B \sum_c^C d_{b,c} \cdot d_{b,c}^{\text{T}}. \tag{9}$$

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.

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   Measurement covariance with GLAM

We can estimate C_{test}, the covariance between the bins of the statistic, with other sets of simulations that may have advantages to Aemulus. Here we use GLAM, for which we have more simulations and a larger volume than Aemulus simulations, with side length $L_{\text{GLAM}} = 1$ Gpc. We compute the statistic on $M = 100$ GLAM mocks, which we index by m. The GLAM covariance matrix C_{GLAM} is given by

$$d_m = \frac{y_m - \bar{y}}{\bar{y}} \tag{10}$$

$$C_{\text{GLAM}} = \frac{1}{M - 1} \sum_m^M d_m \cdot d_m^{\text{T}} \tag{11}$$

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 GLAM simulation (GLAM errors will be smaller because it has more volume, so we need to inflate these).

5 Combining and scaling covariances

We combine these errors to obtain a final covariance matrix $C_{\mathcal{L}}$ for our likelihood. Covariances scale as the volume (covariance decreases with increasing volume; Klypin & Prada 2018) so we need to scale covariances to be consistent when combining them.

We have multiple options of how to estimate C_{perf} and C_{test}:

(A) C_{perf} is estimated using a test set that is composed of statistics computed on individual Aemulus boxes, and $C_{\text{test}} = C_{\text{aemulus1box}}$.

(B) C_{perf} is estimated with a test set is composed of the *mean* of statistics computed on B Aemulus box realizations, and $C_{\text{test}} = C_{\text{aemulusBbox}} = \frac{1}{B}C_{\text{aemulus1box}}$. (This choice of test set essentially increases the volume by a factor of B, so the error will be smaller.)

(C) C_{perf} is estimated with a test set is composed of statistics computed on individual Aemulus boxes, but we estimate C_{test} with the GLAM simulations, so

$$C_{\text{test}} = \left(\frac{L_{\text{GLAM}}}{L_{\text{aemulus}}} \right)^3 C_{\text{GLAM}}. \tag{12}$$

Note that $L_{\text{GLAM}} > L_{\text{aemulus}}$, so the GLAM error will be smaller than that of the test set used to compute C_{perf}, and needs to be inflated to match it.

(D) C_{perf} is estimated with a test set is composed of the mean statistics computed on B Aemulus boxes, and we estimate C_{test} with the GLAM simulations. Then,

$$C_{\text{test}} = \frac{1}{B} \left(\frac{L_{\text{GLAM}}}{L_{\text{aemulus}}} \right)^3 C_{\text{GLAM}}. \tag{13}$$

In our current tests, we choose option (D).

Once we have chosen the test set used to evaluate C_{perf} and estimated C_{test}, we combine them to estimate C_{emu} as in Equation 2. To compute the final covariance matrix $C_{\mathcal{L}}$ for performing parameter recovery on a data set, we need the covariance of that dataset C_{data}. This could be the real data (e.g. we plan to use BOSS and DESI). Then the final combined likelihood is given by Equation 1.

For our tests, we are performining parameter recovery on the Aemulus test simulations, so we have $C_{\text{data}} = C_{\text{test}}$. In this special case, we simply have $C_{\mathcal{L}} = C_{\text{perf}}$.