Notes On Abacus BAO Analysis

Yutong Duan [1]*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

This serves as detailed notes on the procedures of BAO analysis with AbacusCosmos to accompany the code. Part 1 is on the calculation of correlation functions and covariance matrices. Part 2 is on the fitting methods of the BAO fitter.

Key words: keyword1 – keyword2 – keyword3

1 INTRODUCTION

Precision measurement of the BAO signal from galaxy correlation functions requires understanding the systematics. The standard approach is producing many mock catalogues of galaxies/quasars, test the analysis pipelines, and evaluate the sensitivity to systematics. The pipelines largely consist of two parts, statistics and fitting. The statistics of galaxy catalogues are correlation functions and covariance matrices. Then the statistics are fed to the fitting procedures, yielding best-fit α, the BAO scale parameter with respect to the fiducial model.

2 CORRELATION FUNCTIONS AND COVARIANCE MATRIX

2.1 Reading Halo Catalogue

Given a cosmology and a redshift, there are 16 boxes with varied phases. For each phase, load the halo catalogue produced by the Rockstar halo-finder. The virial mass and radius fields have inconsistent naming so add the following columns if they are not already present

$$\texttt{halo\_mvir} = \texttt{halo\_m} \tag{1}$$

$$\texttt{halo\_rvir} = \texttt{halo\_r} \tag{2}$$

to keep the Abacus catalogues compatible with halotools, where these fields are assumed to be present. The halo mass field `halo_mgrav` is preferred over `halo_mvir` in calculations because `halo_mvir` only works well for large halos and misbehaves for small halos as well as subhalos. For our analysis the distinction is not important though.

The NFW profile $\rho(r)$ is a mass distribution model for dark matter halos as a function of radius. The profile is completely characterised by `halo_mvir` and the the concentration $c_{\mathrm{NFW}} \equiv R_{\mathrm{vir}}/R_s$, where R_s is the scale radius of the halo (see halotools implementation for formulae). The Klypin definition of the halo scale radius R_{s} is considered more stable than the usual R_{s} for small halos. The NFW concentration field is added to halo table as well.

$$\texttt{halo\_nfw\_conc} = \frac{\texttt{halo\_rvir}}{\texttt{halo\_klypin\_rs}}. \tag{3}$$

2.2 Populating Halos with Galaxies

Halotools treats all halos as spherical and uses NFW profiles to paint galaxies. Several mainstream HOD models are built in to halotools: `['zheng07', 'leauthaud11', 'tinker13', 'hearin15', 'zu_mandelbaum15', 'zu_mandelbaum16', 'cacciato09']`. We tune the model parameters to generate LRG-like samples. For a $(1100\,\mathrm{Mpc/h})^3$ simulation box, typical numbers are

$$N_{\mathrm{halos}} = 8 \times 10^6 \tag{4}$$

$$N_{\mathrm{galaxies}} = 5 \times 10^5. \tag{5}$$

The Zheng et al. (2007) model used in SDSS is an essential model with 5 parameters, $(M_{\mathrm{cut}}, \sigma, M_0, M_1, \alpha)$. The mean halo occupation numbers per halo for central and satellite galaxies are given by

$$\begin{aligned}\left\langle N_{\mathrm{cen}}(M)\right\rangle &= \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\log M - \log M_{\mathrm{cut}}}{\sigma}\right)\right] \\ &= \frac{1}{2}\mathrm{erfc}\left(\frac{\log(M_{\mathrm{cut}}/M)}{\sigma}\right)\end{aligned} \tag{6}$$

$$\left\langle N_{\mathrm{sat}}(M)\right\rangle = \left\langle N_{\mathrm{cen}}(M)\right\rangle \left(\frac{M - M_0}{M_1}\right)^{\alpha}. \tag{7}$$

Since Abacus enables direct access to DM particles beyond standard halo metadata, a DM particle-based approach can be used to populate halos, without invoking NFW concentrations. Satellite galaxies are assigned to each DM particle within the halo with a certain probability, such that the resulting mean occupation is as specified by the HOD model. This avoids sphericalising halos and respects the morphology of halos when assigning synthetic galaxies. For the first BOSS analyses White et al. (2011) used Zheng's

[*] E-mail: dyt@physics.bu.edu

formalism but with slightly different parametrisations,

$$\left\langle N_{\text{cen}}(M) \right\rangle = \frac{1}{2}\text{erfc}\left(\frac{\ln(M_{\text{cut}}/M)}{\sqrt{2}\sigma'}\right) \quad (8)$$

$$\left\langle N_{\text{sat}}(M) \right\rangle = \left\langle N_{\text{cen}}(M) \right\rangle \left(\frac{M - \kappa M_{\text{cut}}}{M_1}\right)^{\alpha} . \quad (9)$$

where the argument inside erf has an extra factor of $\ln 10/\sqrt{2}$, and M_0 is replaced with κM_{min}. The reason for the trivial change of parametrisation was that "our definition of σ can be interpreted as a fractional 'scatter' in mass at threshold". Conversion of SDSS/BOSS empirical values to the new parameters $(M_{\text{cut}}, \sigma', \kappa, M_1, \alpha)$ can be done simply by

$$\sigma' = \frac{\ln 10}{\sqrt{2}}\sigma \quad (10)$$

$$\kappa = \frac{M_0}{M_{\text{min}}} \quad (11)$$

with M_{cut}, M_1 and α remaining the same.

When populating halo catalogues with galaxies, several observational corrections are made to the true position/velocity data. The velocity bias is corrected by randomly drawing the peculiar velocity from a normal distribution to account for the rms dispersion within the halo,

$$v_{\text{pec}} \sim N(0, \frac{v_{\text{rms}}}{\sqrt{3}}\alpha_c) \quad (12)$$

$$v'_{\text{los}} = v_{\text{los}} + v_{\text{pec}} \quad (13)$$

assuming $v_{\text{rms}} = \sqrt{v_x^2 + v_y^2 + v_z^2}$ and $v_{\text{los}} = v_z$. Redshift-space distortion seen by an observer is just modifying the z-coordinate of the halo/galaxy, analytically

$$x'_z = x_z + \frac{v_z}{aH(a)} = x_z + \frac{v_z}{\frac{H(z)}{1+z}} = x_z + \frac{1+z}{H(z)}v_z \quad (14)$$

as derived in halotools documentation. This formula a very good approximation. However, halotools assumes $h = 1$ and $H_0 = 100h = 100\,\text{km}\,\text{s}^{-1}\,\text{Mpc}^{-1}$, whereas our simulations are more realistic and were run with $h \approx 0.67$. There are two ways to manually add RSD, bypassing relevant halotools functionalities. One is to use Astropy's cosmology class, which comes as `halocat.cosmology` and includes the $E(z)$ function for the particular cosmology. With $H(z) \equiv H_0E(z)$, RSD can be implemented as

$$x'_z = x_z + \frac{1+z}{H_0E(z)}v_z \quad (15)$$

The other way is to use the conversion factor in `halocat.header['VelZSpace_to_kms']`, defined as

$$\beta \equiv aH(z)L_{\text{box}} = \frac{H(z)}{1+z}L_{\text{box}} \quad (16)$$

where β is in $\text{km}\,\text{s}^{-1}$ and L_{box} in Mpc. RSD is then applied as

$$x'_z = x_z + \frac{L_{\text{box}}}{\beta}v_z . \quad (17)$$

where x_z and v_z are in default catalogue units of Mpc and $\text{km}\,\text{s}^{-1}$. These two methods are numerically equivalent up to 1 in 10^{15}.

The following is yet to be implemented. For each of the 16 phase boxes, the random number generator is seeded such that the DM particles each receive the same random number in a fixed order regardless of HOD parameters. To further increase the signal-to-noise in the cluster- ing statistics, 16 realisations of a given HOD is generated with different initial seeds.

2.3 Galaxy Pair Counting

All counting is done in fine (s, μ) bins: s bin edges are from 0 to $150\,\text{h}^{-1}$Mpc at $1\,\text{h}^{-1}$Mpc steps, and $\mu = \cos\theta$ bin edges are from 0 to 1 at 0.01 steps, meaning a total of (150, 100) bins.

We need the auto-counts for each box to calculate the auto-correlation for the whole box, and the cross-counts between the box and a subvolume of itself to calculate cross-correlation and estimate the covariance matrix. All these counts are saved to disk. Raw counts are saved as `paircount-DD.npy` in the original Currfunc count format, i.e. a structured array. Note that `c_api_timer` needs to be turned off in Corrfunc for this file to be saved and recovered properly and not as "object" type.

The optimal s bin size and range for fitting were studied in BOSS DR12 analysis Ross et al. (2017). Based on these results, we choose the bin size $5\,\text{h}^{-1}$Mpc and the range $50\,\text{h}^{-1}\text{Mpc} < s < 150\,\text{h}^{-1}$Mpc. Also we use a μ bin size of 0.05 to be consistent with BOSS DR12. The raw counts were re-binned into (30, 20) coarse bins and total pair counts verified.

2.4 Auto-correlation Functions

The PH natural estimator of the auto-correlation requires DD and RR counts for each phase box. DD is calculated using the re-binned DD counts for (30, 20) bins. RR pair counts are calculated for the same bins analytically as given in Appendix A.

The auto-correlation function ξ is decomposed into the monopole component ξ_0 for $\ell = 0$ and the quadrupole component ξ_2 for $\ell = 2$ in the Legendre spherical harmonics basis using

$$\xi(s, \mu) = \sum_{\ell} \xi_\ell(s) L_\ell(\mu) \quad (18)$$

$$\xi_\ell(s) = \frac{2\ell + 1}{2} \int_{-1}^{1} \xi(s, \mu) L_\ell(\mu)\,\text{d}\mu \quad (19)$$

as shown in Fig. 1. 16 ξ samples from 16 realisations are averaged and become a single auto-correlation result for the box. This averaging step can be done before or after multipole decomposition which is a linear operation.

Then we jackknife co-add the 16 boxes, dropping one box at a time and taking the mean of the other 15. This yields 16 jackknifed correlation function samples, and each one is passed on to the fitter for BAO fitting.

2.5 Covariance Matrix

The BAO fitter takes correlation functions and covariance matrix as input. The covariance is empirically estimated. Each simulation box is divided into $\text{sub}^3 = 3^3$ subvolumes. The cross-correlation function between one subvolume and the entire box is calculated for all 16 phases, resulting in $27\times16 = 432$ correlation function samples (each one is an average of 16 realisations). With (30, 20) total bins for ξ_0 and ξ_2, there are $30 \times 20 \times 2 = 1200$ bins, monopole and quadrupole included. The covariance between every two (s, μ) bins is calculated from 432 samples, including the covariance between monopole and quadrupole. Thus resulting covariance matrix has shape 1200×1200.

Finally this covariance matrix needs to be rescaled to account for our volume and averaging tricks. The covariance scales inversely with volume, so the covariance derived from the subvolume (1/27 of the box's volume) is much larger than the actual. In addition, if we run 16 realisations and average over all of them, there is another factor of 16. Also due to jackknife, each sample is an average of 15

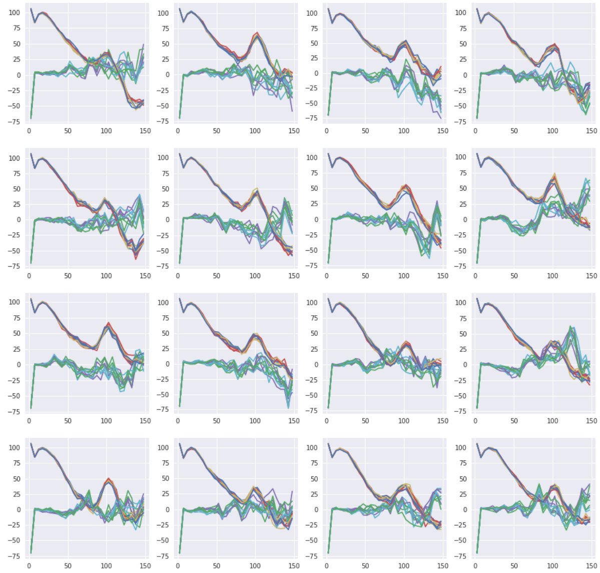

Figure 1. Monopole and quadrupole of auto-correlation functions for 16 phase boxes. Each box has 16 realisations overplotted, which are averaged to give one result.

boxes, thus a factor of 15. Therefore the true covariance matrix is the one obtained from subvolume divided by (27 × 16 × 15).

3 BAO FITTER

A recent, functional fitter is one by Ross https://github.com/ashleyjross/LSSanalysis used for BOSS DR12 analysis. The fitter takes correlation function ξ_0, ξ_2 in all (s, μ) bins and the covariance between all bins as input, and evaluates χ^2 on the $(\alpha_{/\!/}, \alpha_\perp)$ parameter grid.

The log likelihood ratio follows a χ^2 distribution. In the $(\alpha_{/\!/}, \alpha_\perp)$ two-parameter plane, we may find the confidence region for any confidence level σ by calculating the constant $\Delta\chi^2$ contour. The probability corresponding to confidence level σ is $P = \mathrm{Erf}(\sigma/\sqrt{(2)})$, and the contour can be found by

$$\Delta\chi^2 = Q(P) \tag{20}$$

where Q is the quantile function (percent-point function) for a χ^2 distribution with dof = 2.

With 16 jackknife ξ samples and a covariance matrix for each HOD model, we obtain 16 $(\alpha_{/\!/}, \alpha_\perp)$ values by finding the global χ^2 minimum. Then for every pair of HOD models M_i and M_j, we calculate 16 phase-matched differences $\alpha_{ij} = \alpha_i - \alpha_j$, plotted in Fig. 2.

We will rewrite the fitter in a modern style in Python conforming to PEP standards and optimising for performance.

REFERENCES

Ross A. J., et al., 2017, MNRAS, 464, 1168

Wang Y., Yang X., Mo H. J., van den Bosch F. C., Weinmann S. M., Chu Y., 2008, ApJ, 687, 919

White M., et al., 2011, ApJ, 728, 126

Zheng Z., Coil A. L., Zehavi I., 2007, ApJ, 667, 760

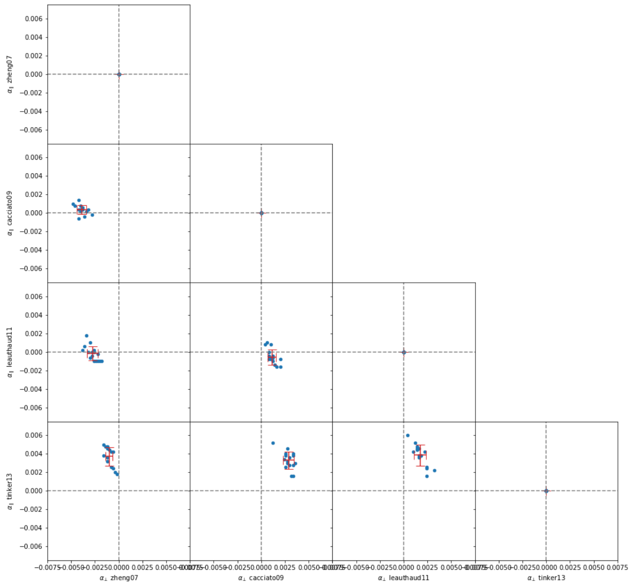

Figure 2. Comparison of $(\alpha_{/\!/}, \alpha_\perp)$ for four HOD models done in halotools. vertical axis is $\alpha_{/\!/}$, horizontal $\alpha_\perp$.

APPENDIX A: AUTO-CORRELATION FUNCTION ESTIMATOR

In this implementation, raw pair counts are saved, and $N_{\rm D}$, $N_{\rm R}$ normalisation is done only at the estimator step.

For auto-correlation of a sample in periodic box, the Peebles & Hauser (1974) estimator (natural estimator) is

$$\xi = \frac{N_{\rm R}(N_{\rm R}-1)}{N_{\rm D}(N_{\rm D}-1)}\frac{DD}{RR} - 1 \tag{A1}$$

where DD is the auto-correlation data-data pair count and RR is the random-random pair count.

One way to obtain RR is to generate a random sample of particle number $N_{\rm R}$ in the same volume and calculate its auto-correlation pair counts. Alternatively, RR can be calculated analytically as follows. Let $\mathrm{d}V$ be the volume of the (s, μ) bin and $V = L_{\rm box}^3$ be the volume the data sample occupies. The random sample must have the same number density as the data sample, $n_{\rm R} = n_{\rm D}$. For simple survey geometry, such as a cube, we may well let the random sample have the same number count, volume, and number density as the data sample.

$$RR(s,\mu) = \frac{N_{\rm R}(N_{\rm R}-1)}{V}\,\mathrm{d}V(s,\mu) \tag{A2}$$

where $\mathrm{d}V(s, \mu)$ is the volume of the (s, μ) bin.

APPENDIX B: CROSS-CORRELATION FUNCTION ESTIMATOR

For cross-correlation between a sample D_1 in the periodic simulation box and a subsample D_2 in its subvolume, the Davis & Peebles (1983) estimator is

$$\xi = \frac{\bar{n}_{\rm R1}}{\bar{n}_{\rm D1}}\frac{D_1 D_2}{R_1 D_2} - 1 \tag{B1}$$

where $\bar{n}_{\rm D1}$ is the mean number density of data sample D_1, $\bar{n}_{\rm R1}$ is the mean number density of R_1, the random sample corresponding to D_1, $D_1 D_2$ is the cross-correlation pair count between two data samples, and $R_1 D_2$ is the cross-correlation pair count between a random and a data sample. Usually a random sample is about 7 times the size of the corresponding data sample (Wang et al. 2008).

Again one may generate a random sample R_1 in the same volume and do the cross counting, or alternatively calculate it analytically,

$$R_1 D_2 = \frac{N_{\rm R1} N_{\rm D2}}{V_1}\, {\rm d}V(s,\mu) = \bar{n}_{\rm R1} N_{\rm D2}\, {\rm d}V(s,\mu). \tag{B2}$$

This paper has been typeset from a TEX/LATEX file prepared by the author.