Increased accuracy of the two-point correlation function at no extra cost

Martin Kerscher
Ludwig–Maximilians Universtät München,
Fakultät für Physik, Schellingstr. 4, D-80799 München
martin.kerscher@lmu.de

January 11, 2024

Using the pair-count implementaion from the `Corrfunc` package we show that with a low discrepency sequence we can calculate the two-point correlation function more accurately than with random points at no extra computational cost.

1 Introduction

The `Corrfunc` package (Sinha & Garrison, 2020; Sinha & Garrison, 2019) offers one of the fastest implementation of pair-counts and is frequently used for calculating the two-point correlation function for large-scale-structure statistics in cosmology. We illustrate how one can combine the conceptual improvements suggested by Kerscher (2022) with the blazingly fast pair-counts from `Corrfunc`[1]. With this approach we are increasing the numerical accuracy of the pair-counts and we are reducing the systematic error in calculations of the two-point correlation function.

The two-point correlation function of the galaxy distribution is often used to constrain models of structure and galaxy formation and to estimate parameters of cosmological models. Observations give us the positions of galaxies in space. The two-point density

$$\varrho_2(\boldsymbol{x}_1,\boldsymbol{x}_2) = \varrho^2\ (1+\xi(|\boldsymbol{x}_1-\boldsymbol{x}_2|)) \tag{1}$$

is the probability of finding two galaxies at $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, where ϱ is the number density and $\xi(r)$ is the two-point correlation function. In a homogeneous and isotropic point process $\xi(r)$ only depends on the separation $r = |\boldsymbol{x}_1 - \boldsymbol{x}_2|$. We use estimators to determine $\xi(r)$ from a galaxy catalogue within a finite domain $W \subset \mathbb{R}^3$. In cosmology estimators based on random point sets are most commonly used. These rely on the data–data DD, data–random DR, and random–random RR pair-counts.

With DR and RR one performs a Monte-Carlo integration using random points (Kerscher, 2022, 1999). Replacing these random point sets with a low-discrepancy sequence of points results in a quasi Monte-Carlo integration. This leads to an improved scaling of the error that is almost proportional to $1/N_\mathrm{q}$, where N_q is the number of points from this low-discrepancy sequence.

In Sec. 2 we give the relevant definitions, in Sec. 3 we illustrate the preferable properties of the quasi Monte-Carlo integration with a numerical example, and in Sec. 4 we provide details on the implementation[2] and give some further comments.

[1] `https://github.com/manodeep/Corrfunc`
[2] see `https://github.com/makerscher/corracc` for the code.

2 Pair-counts and estimators

Given the data points $\{\boldsymbol{x}_i\}$, $i = 1,\dots,N$ within the observational window $\boldsymbol{x}_i \in W$ the

$$\mathsf{DD}(r) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} \mathbb{1}_{[r,r+\delta]}(|\boldsymbol{x}_i-\boldsymbol{x}_j|) \tag{2}$$

is the normalised number of data–data pairs with a distance of $r = |\boldsymbol{x}_i - \boldsymbol{x}_j|$ in the interval $[r, r+\delta]$. The indicator function $\mathbb{1}_A$ of the set A is defined as $\mathbb{1}_A(q) = 1$ if $q \in A$ and 0 for $q \notin A$. Now we consider N_r random points $\{\boldsymbol{r}_i\}$, $\boldsymbol{r}_i \in W$, $i = 1,\dots,N_\mathrm{r}$ within the same sample geometry W as the data and define the normalised number of data-random pairs

$$\mathsf{DR}(r) = \frac{1}{NN_\mathrm{r}}\sum_{i=1}^{N}\sum_{j=1}^{N_\mathrm{r}} \mathbb{1}_{[r,r+\delta]}(|\boldsymbol{x}_i-\boldsymbol{r}_j|), \tag{3}$$

and similarly

$$\mathsf{RR}(r) = \frac{1}{{N_\mathrm{r}}^2}\sum_{i=1}^{N_\mathrm{r}}\sum_{j=1,j\neq i}^{N_\mathrm{r}} \mathbb{1}_{[r,r+\delta]}(|\boldsymbol{r}_i-\boldsymbol{r}_j|), \tag{4}$$

the normalised number of random–random pairs. The Landy & Szalay (1993) estimator is then given by

$$\widehat{\xi}(r) = \frac{\mathsf{DD}(r) - 2\mathsf{DR}(r) + \mathsf{RR}(r)}{\mathsf{RR}(r)}. \tag{5}$$

As an alternative to the random points we consider a low discrepancy sequence of points. We use two distinct low discrepancy sequence $\{\boldsymbol{q}_i\}$ and $\{\boldsymbol{s}_i\}$, with $\boldsymbol{q}_i, \boldsymbol{s}_i \in W$, $i = 1,\dots,N_\mathrm{q}$ (see also Dávila-Kurbán et al. 2021). With one sequence $\{\boldsymbol{q}_i\}$ we can define DQ in full analogy to DR. To define QQ we need both sequences,

$$\mathsf{QQ}(r) = \frac{1}{{N_\mathrm{q}}^2}\sum_{i=1}^{N_\mathrm{q}}\sum_{j=1}^{N_\mathrm{q}} \mathbb{1}_{[r,r+\delta]}(|\boldsymbol{q}_i-\boldsymbol{s}_j|). \tag{6}$$

QQ is the quasi Monte-Carlo integration scheme, and correspondingly, RR is the standard Monte-Carlo integration scheme for the same six-dimensional volume integral (see

eq. (15) and (16) in Kerscher 2022). Now we are set to define a Landy & Szalay (1993) type estimator using a low discrepancy sequence instead of random points:

$$\widehat{\xi}_{\rm qmc}(r) = \frac{\mathrm{DD}(r) - 2\mathrm{DQ}(r) + \mathrm{QQ}(r)}{\mathrm{QQ}(r)}. \qquad (7)$$

3 Comparison

For the comparison of $\widehat{\xi}$ and $\widehat{\xi}_{\rm qmc}$ we choose the example data set `gals_Mr19.ff` from the `Corrfunc` distribution as our test data set[3]. This simulated galaxy sample is inside a rectangular box, and we can calculate the exact reference value $\Xi(r)$ for the Landy & Szalay (1993) estimator in this simple window (Kerscher, 2022). We generate $M = 500$ random point sets and also 500 randomized low-discrepancy sets and calculate $\widehat{\xi}^{(l)}$ and $\widehat{\xi}^{(l)}_{\rm qmc}$ for each of the $l = 1, \ldots, M$ point sets. We use the same number of points $N_{\rm r} = N_{\rm q}$ ranging from 10^4 to 10^7. To quantify the deviation from the exact value $\Xi(r)$ we use

$$\sigma_{\rm qmc}(N_{\rm q}, r)^2 = \frac{1}{M^2} \sum_{l=1}^{M} \left(\widehat{\xi}^{(l)}_{\rm qmc}(r) - \Xi(r) \right)^2 . \qquad (8)$$

and similarly $\sigma(N_{\rm r}, r)^2$.

In the following figures we compare the estimated standard errors $\sigma(N_{\rm r}, r)$ and $\sigma_{\rm qmc}(N_{\rm q}, r)$. We see that with $\widehat{\xi}_{\rm qmc}$ we gain accuracy in all the situations. For small radii the estimated σ can be reduced by a factor of 2-3 using a low discrepancy sequence (see Fig. 1). Unfortunately the scaling of $\sigma_{\rm qmc}$ with $N_{\rm q}$ is only slightly steeper than the scaling of σ with $N_{\rm r}$ in the standard estimator. Hence for small radii the gain in accuracy is not really convincing.

As can be seen from Fig. 2 this changes on large scales. By using a low discrepancy sequence we gain accuracy up to a factor of 10. To obtain a factor of 10 in accuracy with random points requires 100 times the number of random points. The error for the standard estimator using random points $\sigma(N_{\rm r}, r)$ scales proportional to $1/\sqrt{(N_{\rm r})}$, whereas the error $\sigma_{\rm qmc}(N_{\rm q}, r)$ for a low-discrepancy sequence is almost proportional to $1/N_{\rm q}$. The implementation of $\widehat{\xi}_{\rm qmc}$ follows closely the implementation of the standard Landy & Szalay (1993) estimator $\widehat{\xi}$ (see also the next section). For the same number of points we expect a similar runtime. For a more examples and a detailed comparison of run-times see Kerscher (2022).

[3] `https://github.com/manodeep/Corrfunc/blob/master/theory/tests/data/gals_Mr19.ff`

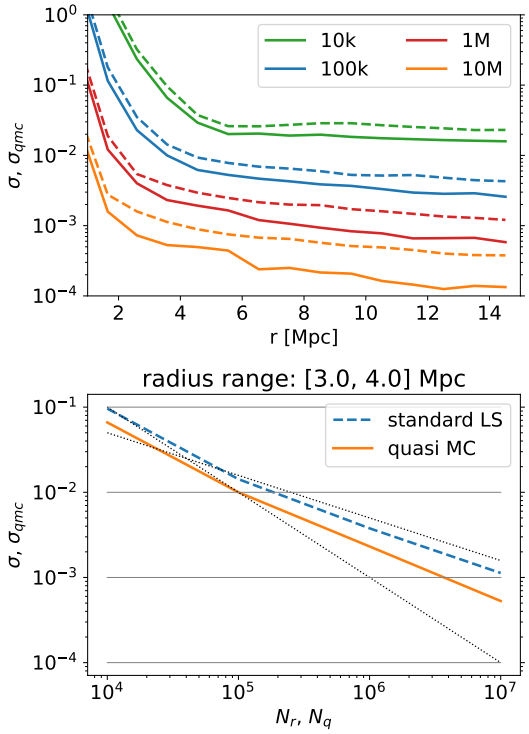

Figure 1: In the top plot we show the estimated error $\sigma_{\rm qmc}(N_{\rm q}, r)$ (solid line, quasi MC) and $\sigma(N_{\rm r}, r)$ (dashed line, standard) for different radii depending on the number of points used. In the lower plot we show the scaling of the error with the number of points for one radius range. The two dotted lines are proprtional to $1/\sqrt{N}$ and $1/N$.

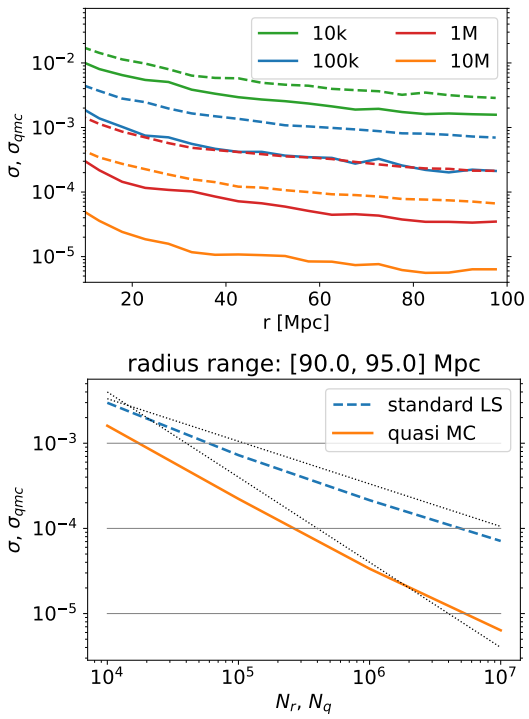

Figure 2: The same quantities as in Fig. 1 are shown for larger radii.

4 Some notes

The code: We provide a sample implementation[4] to illustrate how one can use the pair-count functions as provided in `Corrfunc` with the low discrepancy sequences from SciPy. For calculating the auto- or cross-correlation function of point sets the package `Corrfunc` provides the function `Corrfunc.theory.DD`. We use this function to calculate DD, RR, DR, DQ and QQ.

To generate random points we use the random number generator from NumPy (Harris et al., 2020). To generate low discrepancy sequences we use the function `Halton` as provided in SciPy `scipy.stats.qmc` (starting with version ≥`1.7.0`, Virtanen et al. 2020). This function allows us to generate randomised Halton sequences (see Owen 2017, Owen & Rudolf 2021).

Special care has to be taken in the calculation of QQ. As one can see from eq. (6) we need two distinct low discrepancy sequences (see also eq. (15) and (16) in Kerscher 2022). We start with a six dimensional Halton sequence inside $W \times W$ and split it into two three dimensional sequences.

Exact $\Xi(r)$ As already mentioned the DR, RR and also DQ, QQ are (quasi) Monte-Carlo schemes for special volume integrals which can be expressed in geometrical terms for a box (Kerscher, 2022, 1999). We use these terms to calculate the exact estimator $\Xi(r)$ for a box. Still a numerical integration remains to be performed. We are using the integration routine `quad` from `scipy.integration`. This calculation can take up to several hours, but for the comparison we only need this once.

Beyond a box For more general sampling areas we are not able to calculate the exact reference value $\Xi(r)$. Hence we have to compare to the empirical mean

$$\overline{\xi}(N_\mathrm{q}, r) = \frac{1}{M}\sum_{l=1}^{M} \widehat{\xi}^{(l)}_\mathrm{qmc}(r), \tag{9}$$

and calculate the sample error

$$\widehat{\sigma}_\mathrm{qmc}(N_\mathrm{q}, r)^2 = \frac{1}{M^2}\sum_{l=1}^{M} \left(\widehat{\xi}^{(l)}_\mathrm{qmc}(r) - \overline{\xi}(N_\mathrm{q}, r)\right)^2. \tag{10}$$

In the comparions in Sect. 3 we also calculated $\widehat{\sigma}_\mathrm{qmc}$ and saw that it is almost indistinguishable from σ_qmc. Hence we expect that $\widehat{\sigma}_\mathrm{qmc}$ is also a good proxy for the expected error σ_qmc in more general situations. This allows the quantification of systematic errors for a window W beyond a box.

The periodic box: We compared the $\widehat{\xi}_\mathrm{qmc}(r)$ and $\widehat{\xi}(r)$ for data in a box, which we saw as an example for more general windows W. Hence we ignored the fact that the data came from a simulation with periodic boundaries.

Estimating $\xi(r)$ from data with periodic boundaries is much simpler. The pair-counts $\mathsf{DD}(r)$ have to be calculated respecting the periodic boundary conditions which can be done also with `Corrfunc.theory.DD`. The geometric factors for the estimation of $\xi(r)$ are well known and specifically simple. If we assume a quadratic box $[0, L]^3$ with periodic boundaries we get

$$\widehat{\xi}_\mathrm{per}(r) = \frac{L^3}{\frac{4\pi}{3}\left((r+\delta)^3 - r^3\right)}\mathsf{DD}(r) - 1 \tag{11}$$

More details are given in Appendix A.1 of Kerscher (2022). No random or low discrepancy point set appears, only the pair-counts $\mathsf{DD}(r)$ are used, see also `Corrfunc.theory.xi`.

A special DR and DQ Kerscher (2022) suggests a special scheme for calculating DR and DQ based on ideas of Rivolo (1986). Since this requires a different algorithm than the pair-counts we do not cover it here. Together with another (slower) pair-count implementation such a special DR, DQ implementation is provided at the following link[5] .

References

Dávila-Kurbán, F., Sánchez, A. G., Lares, M., & Ruiz, A. N. 2021, Mon. Not. Roy. Astron. Soc., 506, 4667

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Nature, 585, 357

Kerscher, M. 1999, Astron. Astrophys., 343, 333

Kerscher, M. 2022, Astron. Astrophys., 666, A181

Landy, S. D. & Szalay, A. S. 1993, ApJ, 412, 64

Owen, A. B. 2017, arXiv e-prints, arXiv:1706.02808

Owen, A. B. & Rudolf, D. 2021, SIAM Rev., 63(2), 360

Rivolo, A. R. 1986, ApJ, 301, 70

Sinha, M. & Garrison, L. 2019, in Software Challenges to Exascale Computing, ed. A. Majumdar & R. Arora (Singapore: Springer Singapore), 3–20

Sinha, M. & Garrison, L. H. 2020, MNRAS, 491, 3022

Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, Nature Methods, 17, 261

[4] `https://github.com/makerscher/corracc`

[5] `https://homepages.physik.uni-muenchen.de/~Martin.Kerscher/software/accuratexi/`