Emulation and Compression for Weak Lensing Cosmology

Imperial College London

Arrykrishna Mootoovaloo

October 2021

A thesis submitted for the degree of

Doctor in Philosophy

Department of Physics

Imperial College London

Supervisors: Prof. Alan Heavens, Prof. Andrew Jaffe, Dr. Florent Leclercq

DECLARATION

I hereby declare that the content of this thesis is my own work. I have acknowledged and referenced all other sources of materials, published or unpublished to the best of my ability. A large component of the thesis, as elaborated below, is based on my own published paper, submitted paper for publication and work in progress, along with my collaborators Prof. Alan Heavens, Prof. Andrew Jaffe and Dr. Florent Leclercq. In particular,

- Chapters 1, 2 and 3 correspond to reviews on Weak Lensing, Bayesian Statistics and Kernel Methods and Gaussian Process respectively. All sources of information are appropriately referenced.
- Chapters 4 and 5 are precursors for Chapter 6. In particular, the former two chapters entail a detailed exploratory analysis of how we can develop algorithms for emulation and compression.
- Chapter 6 is based on:

  Paper 1

  Mootoovaloo, A., Heavens, A. F., Jaffe, A. H., & Leclercq, F., Parameter inference for weak lensing using Gaussian Processes and MOPED. 2020, MNRAS, 497, 2213

- Chapter 7 is a precursor for Chapter 8. In particular, we develop and test semi-parametric Gaussian Process model on MOPED coefficients.
- Chapter 8 is based on:

  Paper 2

  Mootoovaloo, A., Jaffe, A. H., Heavens, A. F., & Leclercq, F., Kernel-Based Emulator for the 3D Matter Power Spectrum from CLASS. 2021, arXiv e-prints, arXiv:2105.02256

  and is submitted to Astronomy & Computing journal for publication.

- Chapter 9 presents some new mathematical derivations which can be used in a weak lensing analysis.
- Chapter 10 is still work in progress and the objective is to extend and apply the tools we have developed in this thesis to the KiDS+VIKING-450 and KiDS-1000 data.
- Chapter 11 is the conclusion of the work covered in this thesis. We also briefly touch upon the different possibilities of extending the work for upcoming weak lensing surveys.

Conference Paper

International Conference on Learning Representations (ICLR) is one of the top Computer Science conferences and our work from Chapter 6 has been accepted for a poster presentation at this venue. In particular, the paper, video and code are available at

Paper 3

Gaussian Processes and MOPED Compression for Weak Lensing (,  ,)

A. Mootoovaloo, A. Heavens, A. Jaffe and F. Leclercq, ICLR Conference

Softwares

- From Chapter 5, we have built an interactive web application to illustrate the combination of compression and emulation and the application is available at:

  https://jlamopedgp.herokuapp.com/

- From Chapter 6, the code is distributed on Github at

  https://github.com/Harry45/gp_emulator

- The new semi-parametric GP technique developed in Chapter 7 can be accessed at:

  https://github.com/Harry45/semi_gp

- For Chapter 8, the code is on Github at

  https://github.com/Harry45/emuPK

  and the corresponding documentation is available at

  https://emupk.readthedocs.io/

Co-author Publications

- This work is based on using different Machine Learning algorithms, including random forest, to perform classification in a hierarchical way. This leads to an overall gain in performance compared to solving the multi-class problem directly.

  Paper 4

  Hosenie, Z., Lyon, R. J., Stappers, B. W., & **Mootoovaloo, A.**, Comparing Multi-class, Binary, and Hierarchical Machine Learning Classification schemes for variable stars. 2019a, MNRAS, 488, 4858

- One of the biggest challenges in Machine Learning is imbalanced dataset. This paper uses different techniques to show that careful data augmentation can lead to a better classification result.

  Paper 5

  Hosenie, Z., Lyon, R., Stappers, B., **Mootoovaloo, A.**, & McBride, V., Imbalance learning for variable star classification. 2020, MNRAS, 493, 6050

Declaration of Copyright

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisors: Alan, Andrew and Florent for their constant support, consistent weekly meeting, continual guides and most importantly, for understanding my thorough, meticulous approach and my reluctance to just accept any proposed methodologies. Needless to say that the COVID-19 caused a massive shift in the way we normally work and multiple lockdowns in the UK compelled us to work from home. My supervisors have made sure that I have access to all my research material, including the purchase of new computing equipment and office chair to ensure that I have a proper working environment. I am, without doubt, very fortunate to be in their midst.

Next, I would like to thank Imperial College London, for supporting me financially through the Imperial President's PhD scholarship. Due to the pandemic, my funding was also extended for three additional months and I am forever obliged to the *all* parties involved in the funding committee. I would like to thank my colleagues in ICIC, the Imperial Centre for Inference and Cosmology. I have had the opportunity to discuss Statistics and Cosmology with staffs namely Boris Leistedt, Roberto Trotta, Daniel Mortlock, Marc Deisenroth, Bruce Bassett, Martin Kunz and Jonathan Pritchard. Within the ICIC group, I have had the opportunity to discuss various topics with my friends, in no particular order, Claude Schmidt, Tai-An Cheng, Joshua Greenslade, Wahid Rahman, George Kyriacou, Adélie Gorce, Luke Johnson, Daniel Jones, Tom Binnie, Hikmatali Shariff, Ian Hothi and Lena Lenz.

This work would not have been possible without the support of my family members - my mother, father, elder brother, little sister and little brother who have all made a huge contribution, especially in the last 10 years. They have all given me the freedom to pursue my own career path and I am always grateful to them. It was very difficult to leave behind my closely-knit family, especially my little brother and sister to pursue my MSc at the University of Cape Town, followed by this PhD in London. I was also very lucky to meet my life partner during my undergraduate study in Mauritius and she followed a similar career path as mine. As I complete this PhD, she is also completing her PhD in Astrophysics (and Machine Learning) from the University of Manchester. I would like to thank her for the endless support, encouragement, kind understanding and ability to cope with ever-changing work and living environments, especially during the pandemic.

ABSTRACT

Future surveys of the Universe face the dual challenge of data size and data statistics. The non-Gaussianity induced by gravity presents severe difficulties to robust data analysis, as the sampling distributions are unknown. In this landscape, machine learning and extreme data compression will play an essential role in being able to handle the data size and complexity, in order to reach the scientific goals such as finding the driver of the accelerated expansion of the Universe and resolving the tensions between early- and late-Universe observations.

This thesis consists of three main parts. As a first application, we show that we can recover robust parameter constraints by combining emulation and compression for a challenging data-set such as KiDS-450, which consists of only 24 band powers. In particular, we build a Gaussian Process (GP) emulator for two different test cases, the first one being at the level of the band powers and the second one for the coefficients of the summary data massively compressed with the MOPED algorithm. In the former, cosmological parameter inference is accelerated by a factor of $\sim 10 - 30$ compared to the Boltzmann solver, CLASS, and this factor depends on whether we want to include the GP uncertainty in the inference mechanism. Importantly, with future surveys, the gain can be up to $\sim 10^3$ when the Limber approximation is used. The GP formalism, along with the MOPED compression algorithm, provides us with the option of dropping the Limber approximation, without which each forward simulation is inconveniently very slow. By comparing the Kullback-Leibler divergence between the emulator likelihood and the CLASS likelihood, along with the uncertainty analysis on the inferred parameters, including the GP uncertainty does not justify the additional computational cost. In fact, the mean predictor from the GP is faster and requires a smaller memory footprint. Importantly, the number of summary statistics will be large $(\sim 10^4)$ and the speed of the emulated MOPED coefficients depend on the number of parameters and not on the number of summary. Hence the gain in speed is very large. In the non-Limber case, this speed-up factor can be $\sim 10^5$.

While the above pipeline elegantly provides a solution for speeding up computing via the MOPED algorithm, an important ingredient which we would like to propagate in the analysis is the $n(z)$ uncertainty. Starting with the $n(z)$ distributions and the 3D matter power spectrum,

$P_\delta(k, z)$, to the calculation of the weak lensing and intrinsic alignment power spectra, followed by the calculation of the band powers (and perhaps the MOPED coefficients if we want to), the most expensive part is the calculation of $P_\delta(k, z)$. The latter can be computed either using a Boltzmann solver such as CLASS or using large-scale N-body simulations. Hence, in the second part of this thesis, we develop, document and share an emulator for $P_\delta(k, z)$ using a semi-parametric Gaussian Process. Our code enables the calculation of the following quantities: the non-linear matter power spectrum with/without baryon feedback, the linear matter power spectrum at a fixed redshift, the weak lensing intrinsic alignment power spectra. In addition, the first and second derivatives of the 3D matter power spectrum with respect to the input parameters can also be calculated. The emulator is accurate when tested on an independent set of parameters, drawn from the prior. The fractional uncertainty, $\Delta P_\delta / P_\delta$ is centered on zero. The emulator is also $\sim$ 300 times faster compared to CLASS, hence opening up the possibility of sampling cosmological and nuisance parameters in a Monte Carlo routine. The software (`emuPK`) is distributed with a set of pre-trained Gaussian Process (GP) models, based on 1000 Latin Hypercube (LH) samples, which are roughly distributed according to priors used in current weak lensing analysis.

In the third part, we use the KiDS+VIKING-450 dataset to test our emulator, `emuPK`. We also introduce two new components in the weak lensing likelihood analysis, namely a double sum approach (essentially re-casting the standard approach of numerically performing integration as a double sum) to calculate the weak lensing and intrinsic alignment power spectra. Moreover, we also include a novel Bayesian Hierarchical method for estimating the $n(z)$ distributions. Early results using the novel $n(z)$ distribution along with the emulator indicate promising avenue.

Abbreviations

2PCFs	Two-point Correlation Functions
BHM	Bayesian Hierarchical Model
COSEBIs	Complete Orthogonal Sets of E/B-Integrals
GAN	Generative Adversarial Network
GI	Interference Alignment Effects
GP	Gaussian Process
HMC	Hamiltonian Monte Carlo
II	Intrinsic Alignment Effects
JLA	Joint Light Curve Analysis
KiDS	Kilo-Degree Survey
KV-450	KiDS+VIKING-450
LH	Latin Hypercube
LSS	Large Scale Structures
MCMC	Markov Chain Monte Carlo
MH	Metropolis-Hastings
ML	Machine Learning
PICO	Parameters for the Impatient Cosmologist
SOM	Self-organising Map
WFIRST	Wide-Field Infrared Survey Telescope
WL	Weak Lensing

CONTENTS

List of Figures

List of Tables

Chapter 1

WEAK LENSING COSMOLOGY

> Measure what can be measured, and make measurable what cannot be measured.
>
> *Galileo Galilei*

Gravitational lensing, the bending of light from a source to an observer, is an effect which arises due to the intervening matter, for example, clusters of galaxies. It is a consequence of general relativity and the amount of bending depends on the gravitational force caused by the massive object. There are three types of lensing phenomena (Dodelson, 2017). In particular,

1. *strong lensing* is a phenomenon which occurs in the case where the deflection caused by the foreground galaxies or cluster galaxies is large enough that multiple images (arcs or Einstein rings) or at least major distortions of images will be produced (Treu, 2010). It is an interesting proxy for studying galaxies and black holes, determining the content of the universe and understanding the spatial distribution of mass (Nightingale et al., 2019).

2. *microlensing* is similar to strong and weak lensing except that the mass of the lens mass is quite small, for example, mass of a planet or star. The shape changes are not visible and are typically unresolved. Moreover, the magnifications (and hence the brightness) change with time since the relative position of the lens and source changes with time (Dodelson, 2017). This technique is often used to detect very small and faint objects.

3. *weak lensing* is a powerful technique for probing cosmological models. It essentially maps the original galaxies to new positions on the sky. In general, the lensing effect for an individual galaxy is barely detectable, hence *weak* gravitational lensing (Kilbinger, 2015). Lensing by Large Scale Structures (LSS) results in very small distortion of images and is dubbed as cosmic shear. The latter is a promising research area for LSS.

In this work, we shall focus on the last. In §1.1, we touch upon the basics of cosmology be-

fore elaborating on the weak lensing formalism in §1.2. Since this thesis takes a fully Bayesian approach to understanding weak lensing, it is important to understand the main observables of weak lensing, which we discuss in §1.3. Concomitantly, we then briefly cover the theoretical model(s) in §1.3 which play an important role in constraining cosmological and nuisance parameters. In §1.4, we elaborate on various challenges in a typical weak lensing routine, starting from images to deriving constraints on cosmological parameters. In §1.5, we discuss briefly past, current and future surveys and future data analysis, including Machine Learning (ML) topics which provide an alternative approach to performing data analysis in Cosmology. Finally, in §1.7, we recapitulate on the important topics covered in this chapter.

1.1 Cosmology Background

In this section, we will briefly cover basics of the cosmology, focusing mainly on distances in cosmology in §1.1.2, the Friedmann equation in §1.1.1 and structure formation in §1.1.3. We refer the reader to Cosmology textbooks (Liddle, 1998; Dodelson, 2003; Mukhanov, 2005) which cover these topics in further details.

1.1.1 Friedmann Equation

Before going through the Friedmann equations, it is worth highlighting the cosmological principle which is based on isotropy and homogeneity, that is, over a sufficiently large scale, all observable properties are expected to be isotropic since there exists an average motion of radiation and matter in the Universe. Moreover, all observers experience the same mean motion and history of the Universe, under the assumption that they suitably set their clocks.

On a sufficiently large scale, the Friedmann-Lemaître-Robertson-Walker (FLRW) metric is an appropriate space-time metric to describe the line element $\mathrm{d}s$ of a homogeneous and isotropic 3D space. Such a metric in spherical coordinate system, with radial coordinate r and two angles (θ, φ) is

$$\mathrm{d}s^2 = -c^2\mathrm{d}t^2 + a^2(t)[\mathrm{d}r^2 + S_k^2(r)(\mathrm{d}\theta^2 + \sin^2\theta\mathrm{d}\varphi^2)] \tag{1.1.1}$$

where $a(t)$ is the scale factor which has dimensions of length and a radial light propagates according to $|c\,\mathrm{d}t| = a\,\mathrm{d}r$. S_k depends on the geometry of the Universe and constitutes of various possibilities such as

$$S_k(r) = \begin{cases} \sinh r & \text{if } k < 0 \\ r & \text{if } k = 0 \\ \sin r & \text{if } k > 0 \end{cases}$$

and $k = -1, 0, 1$ corresponds to open, flat and closed Universe respectively. At this stage, the evolution of the FLRW Universe can be derived from Einstein equation,

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu} \tag{1.1.2}$$

where $T_{\mu\nu}$ is the stress-energy tensor of the matter, which describes the energy density and pressure. Λ is a cosmological constant, $R_{\mu\nu}$ is the Ricci curvature tensor and $g_{\mu\nu}$ describes the structure of space-time. The field equations result in two independent equations:

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3c^2}\sum_\alpha \rho_\alpha - \frac{c^2 K}{a^2} + \frac{c^2\Lambda}{3} \tag{1.1.3}$$

and

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3c^2}\sum_\alpha (\rho_\alpha + 3p_\alpha) + \frac{c^2\Lambda}{3} \tag{1.1.4}$$

where a dot refers to derivative with respect to time t and α refers to each fluid component. These two equations are referred to as the *Friedmann equations*. Equation 1.1.4 can be re-written in terms of the effective energy density and pressure, which are defined as

$$\rho' \equiv \rho + \frac{\Lambda c^2}{8\pi G},$$

$$p' \equiv p - \frac{\Lambda c^2}{8\pi G}$$

and

$$w = \frac{p'}{\rho'} \tag{1.1.5}$$

For a single fluid, the acceleration equation, in term of w, is

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3c^2}(1 + 3w)\rho' \tag{1.1.6}$$

and for non-relativistic matter, $w = 0$ and in an expanding universe, the total energy corresponding to non-relativistic matter is conserved. In a radiation dominated universe, $w = 1/3$. In general, the expansion of the universe is accelerating when $w < -1/3$. This can be seen by setting Equation 1.1.6 greater than zero. The equation of state for the cosmological constant is $w = -1$ and this value was indeed inferred from observations. Some important definitions and parameters in a typical cosmological analysis include:

Hubble parameter

$$H = \frac{\dot{a}}{a} \tag{1.1.7}$$

Its value at present time is denoted by $H(t = t_0) = H_0$ and is referred to as the Hubble constant. It is customary to also write it as $H_0 = 100h$ km s^{-1} Mpc^{-1} and its value is not fully known. In fact, it is currently a topic of extended debate the cosmology community. *Planck* estimates the value at $H_0 = 67.4 \pm 0.5$ km s^{-1} Mpc^{-1} (Planck Collaboration et al., 2020) while using distance ladder method, Riess et al. (2019) determined the value of $H_0 = 74.03 \pm 1.42$ km s^{-1} Mpc^{-1}, hence a 9.4% difference between these two values, leading to a tension of 4.4σ. On the other hand, Freedman et al. (2020) reported a value of $H_0 = 69.6 \pm 0.8$ km s^{-1} Mpc^{-1} based on the Tip of the Red Giant (TRGB) method.

Critical Density of the Universe

$$\rho_c = \frac{3H^2}{8\pi G} \tag{1.1.8}$$

and in terms h, the critical density of the Universe is, $\rho_c \approx 1.88 \times 10^{-29} h^2$ g cm^{-3}.

Density Parameter

$$\Omega_\alpha = \frac{\rho_\alpha}{\rho_c} \tag{1.1.9}$$

that is, the density parameter is a dimensionless quantity defined in terms of the critical density of the Universe. This also implies that the Friedmann equation satisfies

$$\sum_\alpha \Omega_\alpha = 1. \tag{1.1.10}$$

For a Universe with radiation density, Ω_r, matter density, Ω_m, curvature density parameter, $\Omega_k := -\frac{c^2k^2}{a^2H^2}$ and cosmological constant, Ω_Λ, the Friedmann equation in terms of the current values of the density of the Universe can be written as

$$H^2(t) = H_0^2[\Omega_r a^{-4}(t) + \Omega_m a^{-3}(t) + \Omega_K a^{-2}(t) + \Omega_\Lambda a^{-3(1+w)}] \tag{1.1.11}$$

and for the cosmological constant, we have $w = -1$. Note that, density parameters refer to the values at present time, that is, the subscript 0 is omitted here. Hence, an FLRW model of the Universe can be characterised by 4 parameters, namely, the Hubble constant, H_0, and the present density parameters for radiation, matter and the cosmological constant. Note that, under closure, we have $\Omega_K = 1 - \Omega_r - \Omega_m - \Omega_\Lambda$. $\Omega_r \ll \Omega_m$ at high redshift, hence we often write, $\Omega_K = 1 - \Omega_m - \Omega_\Lambda$. The matter density parameter can further be expressed in terms of cold dark matter (CDM), baryonic matter and heavy neutrinos, that is, $\Omega_m = \Omega_c + \Omega_b + \Omega_\nu$.

1.1.2 Distances

In this section, we will cover the different types of distances we often encounter in cosmological data analysis. Unlike typical Euclidean space, the notion of distance does not hold the same entity in a curved space-time. Different prescriptions will result in various distance measures. Here, we cover *proper distance*, *comoving distance*, *angular diameter distance* and *luminosity distance*. We will denote an emission and an observation event to occur at times t_2 and t_1 corresponding to redshifts z_2 and z_1 respectively. We will also denote the scale factors by $a_2 = a(t = t_2)$ and $a_1 = a(t = t_1)$.

Proper Distance

The proper distance is defined as the distance covered by a light ray as it propagates from position z_2 to position z_1 where the observer is. It is defined as $\mathrm{d}D_\mathrm{p} = -c\,\mathrm{d}t$. Hence, in terms of the Hubble constant, the density parameters and scale factor, the proper distance is

$$D_\mathrm{p} = \frac{c}{H_0}\int_{a_2}^{a_1} \mathrm{d}a\ [\Omega_m a^{-1} + \Omega_K + \Omega_\Lambda a^2]^{-1/2}. \tag{1.1.12}$$

Comoving Distance

The comoving distance is a constant distance between two points on the spatial hyper-surface, comoving with the cosmic flow. It is strictly the coordinate distance between the two events at t_2 and t_1. Hence, the comoving distance is simply, $\mathrm{d}D_\mathrm{c} = dr = -ca^{-1}\,\mathrm{d}t$ and is also related to the proper distance via the scale factor a by $D_\mathrm{c} = aD_\mathrm{p}$. The comoving distance is

$$D_\mathrm{c} = \frac{c}{H_0}\int_{a_2}^{a_1} \mathrm{d}a\ [\Omega_m a + \Omega_K a^2 + \Omega_\Lambda a^4]^{-1/2}. \tag{1.1.13}$$

Angular Diameter Distance

The angular diameter distance is the distance subtended by the physical cross section of an object, δA at z_2 to the solid angle $\delta\omega$ at z_1. Hence,

$$\begin{aligned} D_{\mathrm{a}}^2 &= \frac{\delta A}{\delta \omega} \\ &= \frac{4\pi^2 a^2(z_2) S_k^2[r(z_1, z_2)]}{4\pi} \\ &= a^2(z_2) S_k^2[r(z_1, z_2)] \end{aligned}$$

Hence,

$$D_{\mathrm{a}} = a(z_2) S_k[r(z_1, z_2)] \tag{1.1.14}$$

Luminosity Distance

Suppose the luminosity, defined as the energy emitted per unit time, of an object at a distance, D, is L and we measure the flux, F on Earth, then the relationship between the luminosity and flux is simply, $F = \frac{L}{4\pi D^2}$. The luminosity distance, D_{L} in a curved and expanding spacetime can be defined as $F = \frac{L}{4\pi D_L^2}$. Due to the expansion of the Universe and the redshift, the photons emitted from an object are doppler shifted and the number of photons is reduced, each effect contributing to a decrease of $a_0/a_e = 1 + z_e$. a_0 is the present day scale factor, a_e and z_e are the scale factor and redshift of the object respectively. The relation between the observed flux and the luminosity in terms of the luminosity distance is then:

$$F = \frac{L}{4\pi a_0^2 S_k^2 (1+z)^2} \tag{1.1.15}$$

and $D_L = a_0 S_k(1+z)$. The luminosity distance, D_L of an object at z_2, with reference to the flux as received by an observer at z_1 satisfies the following relations in terms of the angular diameter distance, D_{a}

$$\begin{aligned} D_L &= \left(\frac{a_1}{a_2}\right)^2 D_a \\ &= \frac{a_1^2}{a_2} S_k[r(z_1, z_2)]. \end{aligned} \tag{1.1.16}$$

1.1.3 Structure Formation

In the previous section, we looked into the Friedmann equations under the assumption that our Universe follows the cosmological principle, that is, that of homogeneity and isotropy.

However, one can argue that this assumption is only valid at large scales.

Note 1.1: Important Concepts for Linear Perturbation Theory

Continuity Equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0$$

The continuity equation expresses the fact that matter is conserved.

Euler Equation

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} = -\frac{\nabla P}{\rho} - \nabla \Phi$$

The Euler equation describes the conservation of momentum and the influence of force on a fluid.

Poisson Equation

$$\nabla^2 \Phi = 4\pi G \rho$$

The Poisson equation gives the relationship between the gravitational potential and the matter density.

The Universe is rather inhomogeneous at small scales, which is justified by the formation of stars and galaxies. In addition to this argument, the question of whether the Universe is homogeneous at large scales is also another subject of debate, the reason being that surveys' maps show that while there are regions of over-densities, there are also empty regions which are referred to as *voids*. This then raises the question of whether we have a proper scale definition to treat the Universe as being homogeneous.

In order to explain a few quantities in this section, we introduce a few terms, which will enable us to derive other important quantities such as the power spectrum. For an in-depth discussion on structure formation, we refer the reader to the excellent textbook by Schneider (2006). We first define the *relative density contrast* or *fractional density perturbation* as

$$\delta(\boldsymbol{r}, t) \equiv \frac{\delta\rho(\boldsymbol{r}, t)}{\rho} = \frac{\rho(\boldsymbol{r}, t) - \bar{\rho}(t)}{\bar{\rho}(t)} \tag{1.1.17}$$

where $\bar{\rho}(t)$ refers to the mean density at time t. At $z = 1000$, the amplitude of the density fluctuations was very small, $\sim 10^{-5}$ compared to the amplitude of the density inhomogeneities

today. Following Equation 1.1.17, it is also important to realise that $\delta(\boldsymbol{r}, t) \geq -1$. In general, density fluctuations increase as a function of time. Regions of low density will decrease they density contrast while over-dense regions will increase their density contrast. This evolution of structure can be explained by a model of gravitational instability.

When the density fluctuations are small enough (at very early times), *linear perturbation theory* can be used to understand the evolution of structures. Once these fluctuations are large enough, we have to consider the non-linear evolution of structures, hence requiring higher-order perturbation theory or numerical simulation. Here, we summarise the results from linear perturbation theory. We consider pressure-free matter, that is, dust particles only which can be treated using fluid approximation. The velocity field of this fluid is $\boldsymbol{v}(\boldsymbol{r}, t)$.

The three equations provided in Note 1.1 cannot be solved analytically, unless some approximations are considered. Approximate solutions can be derived for $|\delta| \ll 1$ by linearisation. It turns out that the evolution of the density contrast can be described by a second order differential equation of the form

$$\frac{\partial^2 \delta}{\partial t^2} + \frac{\dot{a}}{a}\frac{\partial \delta}{\partial t} = 4\pi G \bar{\rho} \delta \tag{1.1.18}$$

and crucially, we do not have derivatives with respect to the spatial part but rather only the derivatives with respect to time only. The above differential equation has solutions of the form $\delta(\boldsymbol{r}, t) = D(t)\tilde{\delta}(\boldsymbol{r})$. The solution to the differential equation, for the density contrast, can be expressed as

$$\delta(\boldsymbol{r}, t) = D_+(t)\delta_0(\boldsymbol{r}) \tag{1.1.19}$$

where $D_+(t)$ is referred to as the *growth factor*. An important observation from linear perturbation theory is that the spatial shape of the density fluctuations is fixed and that their amplitude increases as a result of an increase in the growth factor. The latter as a function of the scale factor, a reads

$$D_+(a) \propto \frac{H(a)}{H_0} \int_0^a \mathrm{d}a' \, \left[\Omega_m a'^{-1} + \Omega_\Lambda a'^2 + \Omega_K\right]^{-3/2}$$

Note that there are multiple limitations in this formalism but the linear perturbation theory here is used as an illustration to the evolution of structures in the Universe. Importantly, for $\delta > 1$, it becomes more challenging to solve the equations analytically and we can no longer integrate various assumptions as in the linear perturbation theory.

1.2 Weak Lensing Formalism

Weak lensing is a very promising probe for improving our understanding of Physics of the Universe and importantly, it sensitive of the mass of the foreground, due to the alignment of background around the lensing mass. As a result, weak lensing is strictly a statistical measurement and is a proxy to measure masses of astronomical objects. It is the central focus of this thesis and we will show how we can apply techniques such as data compression and Machine Learning in the analysis of weak lensing cosmological data.

Weak lensing is the prime technique to understand the mass distribution of individual objects such as galaxy clusters. The distortion of background images are generally very small in the context of weak lensing, and this in itself poses a challenge to the weak lensing. The latter is usually analysed in a statistical way.

Central to weak lensing is the change in shape of images of distant objects in the Universe. The intermediate clumpy matter distribution changes the trajectories of photon paths, from the source to the observer. This results in distorting the images we observe. These changes are manifested in terms of the size and magnitude, as well as, the change in shape of the source. In general, both can be used as a proxy for the study of cosmological weak lensing but the change in shape is typically better since the signal to noise ratio is superior. The change in shape will be referred to as *shear* and the effect due to change in size and magnitude is often referred to as *magnification* and *amplification*.

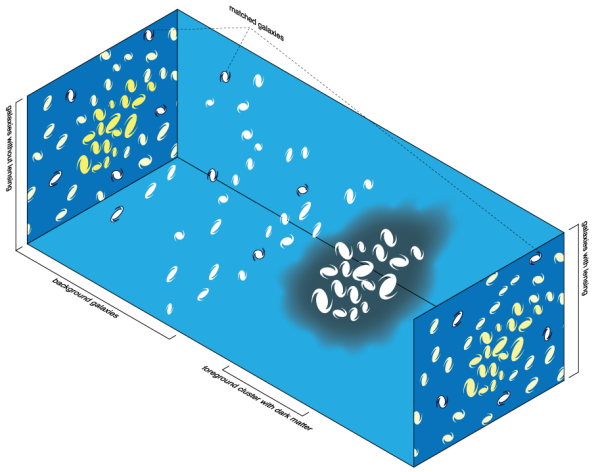


Figure 1.1 – Light (photon) trajectories from distant objects, such as galaxies, being deflected as a result of intermediate foreground matter between the sources and the observer. Usually, the changes in shapes of the distorted objects are very tiny, hence weak lensing. Image credit: Wikipedia.

The observed shapes of distant objects which are close to each other are generally correlated and this correlation drops with separation on the sky (Mandelbaum, 2018).

1.2.1 The Lensing Potential

In this section*, we will derive the relationship between the cosmological potential and the lensing potential. We will assume a flat universe but an extension to non-flat universe is straightforward and strictly, the distortion of a photon has to be treated using General Relativity (GR) but one could also adopt a simpler approach such as Newtonian Physics.

Before sketching the derivations, it is useful to define a few symbols. $\mathbf{x}$ is the comoving coordinate and η is the *conformal time*, defined as $\mathrm{d}\eta = {}^{c}/_{a(t)}\mathrm{d}t$, where $a(t)$ is the scale factor of the Universe. The equation of motion of a photon is

$$\frac{\mathrm{d}^2 x_i}{\mathrm{d}\eta^2} = -\frac{1}{c^2}\frac{\partial(\Phi+\Psi)}{\partial x_i} \quad i = 1, 2 \tag{1.2.1}$$

where Φ and Ψ are the two scalar (cosmological or Bardeen) potentials. For small scalar perturbations, the interval in a Friedmann-Robertson-Walker (FRW) metric may be written as

$$\mathrm{d}s^2 = \left(1+\frac{2\Phi}{c^2}\right)c^2\mathrm{d}t^2 - \left(1-\frac{2\Psi}{c^2}\right)a^2(t)\left[\mathrm{d}r^2 + S_k^2(r)(\mathrm{d}\theta^2 + \sin^2\theta\mathrm{d}\varphi)\right] \tag{1.2.2}$$

where $\boldsymbol{r} \equiv (r, \theta, \varphi)$ is the comoving spherical coordinate system. Using the conformal time and assuming a flat Universe, Equation 1.2.2 simplifies to

$$\mathrm{d}s^2 = a^2(t)\left[\left(1+\frac{2\Phi}{c^2}\right)\mathrm{d}\eta^2 - \left(1-\frac{2\Psi}{c^2}\right)(\mathrm{d}r^2 + r^2\mathrm{d}\theta^2 + r^2\sin^2\theta\mathrm{d}\varphi^2)\right]. \tag{1.2.3}$$

The unperturbed radial path is $\mathrm{d}s^2 = 0$, that is, $\mathrm{d}\eta^2 - \mathrm{d}r^2 = 0$. Thus, for a radial incoming ray,

$$\frac{\mathrm{d}r}{\mathrm{d}\eta} = -1. \tag{1.2.4}$$

The equation of motion can then be derived using variational principle. The Euler-Lagrangian equation is

$$\frac{\partial L^2}{\partial x^\mu} - \frac{\mathrm{d}}{\mathrm{d}p}\left(\frac{\partial L^2}{\partial \dot{x}^\mu}\right) = 0, \tag{1.2.5}$$

where p is a parameter which increases monotonically along the world line and $L = c\frac{\mathrm{d}\tau}{\mathrm{d}p}$ and τ is just an element of the proper time elapsed between two space-time points A and B. In terms of the comoving radial coordinates, $\mathbf{x} = (r\theta\cos\varphi, r\theta\sin\varphi)$ and after some algebra, the equation of motion is

*based on the notes from Prof. Alan Heavens

$$\frac{\mathrm{d}^2\mathbf{x}}{\mathrm{d}\eta^2} = -\frac{1}{c^2}\nabla_{\mathbf{x}}(\Phi + \Psi). \tag{1.2.6}$$

In GR, $\Phi = \Psi$ and hence,

$$\boxed{\frac{\mathrm{d}^2\mathbf{x}}{\mathrm{d}\eta^2} = -\frac{2}{c^2}\nabla_{\mathbf{x}}\Phi(\boldsymbol{r})} \tag{1.2.7}$$

The gravitational potential $\Phi(\boldsymbol{r})$ is related to the matter over-density $\delta(\boldsymbol{r}) \equiv \frac{\delta\rho(\boldsymbol{r})}{\bar{\rho}}$, that is,

$$\nabla^2_{\boldsymbol{r}}\Phi(\boldsymbol{r}) = \frac{3\Omega_m H_0^2}{2a(t)}\delta(\boldsymbol{r}) \tag{1.2.8}$$

where H_0 is the present Hubble constant in $\mathrm{km\,s^{-1}\,Mpc^{-1}}$ and Ω_m is the present day total matter density. The solution to Equation 1.2.7 can be found by integrating (and reversing the order of integration) it twice with respect to r (recall $\mathrm{d}\eta = -\mathrm{d}r$), resulting in

$$x_i = r\theta_i - \frac{2}{c^2}\int_0^r (r - r')\frac{\partial\Phi}{\partial x_i'}\,\mathrm{d}r' \tag{1.2.9}$$

By performing a Taylor expansion of $\frac{\partial\Phi}{\partial x_i'}$, the separation between two light light rays is

$$\Delta x_i = r\Delta\theta_i - \frac{2}{c^2}\Delta\theta_j\int_0^r r'(r - r')\frac{\partial^2\Phi}{\partial x_i\partial x_j}\,\mathrm{d}r' \tag{1.2.10}$$

which can be re-written as

$$\Delta x_i = r\Delta\theta_j(\delta_{ij} - \phi_{ij}) \tag{1.2.11}$$

where δ_{ij} is the Kronecker delta and ϕ_{ij} is given by

$$\phi_{ij} \equiv \frac{2}{c^2}\int_0^r \frac{r - r'}{rr'}\frac{\partial^2\Phi}{\partial\theta_i\partial\theta_j}\,\mathrm{d}r'. \tag{1.2.12}$$

There are two important takeaways from this derivation. First, the mapping between the source plane and the image plane is

$$\boxed{\Delta\vartheta_i = \Delta\theta_j(\delta_{ij} - \phi_{ij}).} \tag{1.2.13}$$

We will delve into more details on this relationship in the next section, where we discuss convergence and shear. In the second place, since we have an expression of $\phi_{ij} \equiv \frac{\partial^2\phi}{\partial\theta_i\partial\theta_j}$, we can also

write the cosmological lensing potential as

$$\phi(\boldsymbol{r}) \equiv \frac{2}{c^2}\int_0^r \frac{r-r'}{rr'}\Phi(\boldsymbol{r}')\,\mathrm{d}r' \tag{1.2.14}$$

Throughout this work, we have assumed a flat Universe. In the case of non-flat Universe, the lensing potential can be written in terms of $S_k(r)$, where $r \to S_k(r)$. Moreover, the integration is performed in the radial direction, but in reality, the path of the photon is not quite radial. The assumption here is that the path is unperturbed by the lens. This approximation is referred to as the *Born approximation*. One can also summarise Equation 1.2.14 as the lensing potential being a 2D projection of the 3D gravitational potential $\Phi(\boldsymbol{r})$. Strictly, the lensing potential is a 3D quantity. One typically average over the redshift distribution, $n(z)$ of the source galaxies. In particular,

$$\phi(\boldsymbol{r}) = \frac{2}{c^2}\int_0^r \frac{g(r)\Phi(\boldsymbol{r}')}{r'}\,\mathrm{d}r' \tag{1.2.15}$$

where

$$g(r) \equiv \int_r^\infty n(r')\frac{r'-r}{r'}\,\mathrm{d}r'. \tag{1.2.16}$$

Estimating the redshift distribution is another topic on its own, which we discuss in §1.4.1.3. Existing techniques involve estimating the distances to the source galaxies using photometric redshifts. The expression for the lensing potential will be useful when we derive the weak lensing statistics in §1.3.

1.2.2 Convergence and Shear

The distortion of an image can be described as a linear transformation between the unlensed (source) plane, $\boldsymbol{\vartheta}$ and the lensed (image) plane, $\boldsymbol{\theta}$, that is,

$$\boldsymbol{\vartheta} = \mathbf{A}\boldsymbol{\theta}. \tag{1.2.17}$$

The *amplification* matrix, also referred to as the *distortion* matrix, $\mathbf{A}_{ij} = \delta_{ij} - \phi_{ij}$ can be parametrised by an isotropic expansion term, the convergence field and a shear field. This matrix is given by:

$$\mathbf{A} = \begin{pmatrix} 1-\kappa & 0 \\ 0 & 1-\kappa \end{pmatrix} + \begin{pmatrix} -\gamma_1 & -\gamma_2 \\ -\gamma_2 & \gamma_1 \end{pmatrix}. \tag{1.2.18}$$

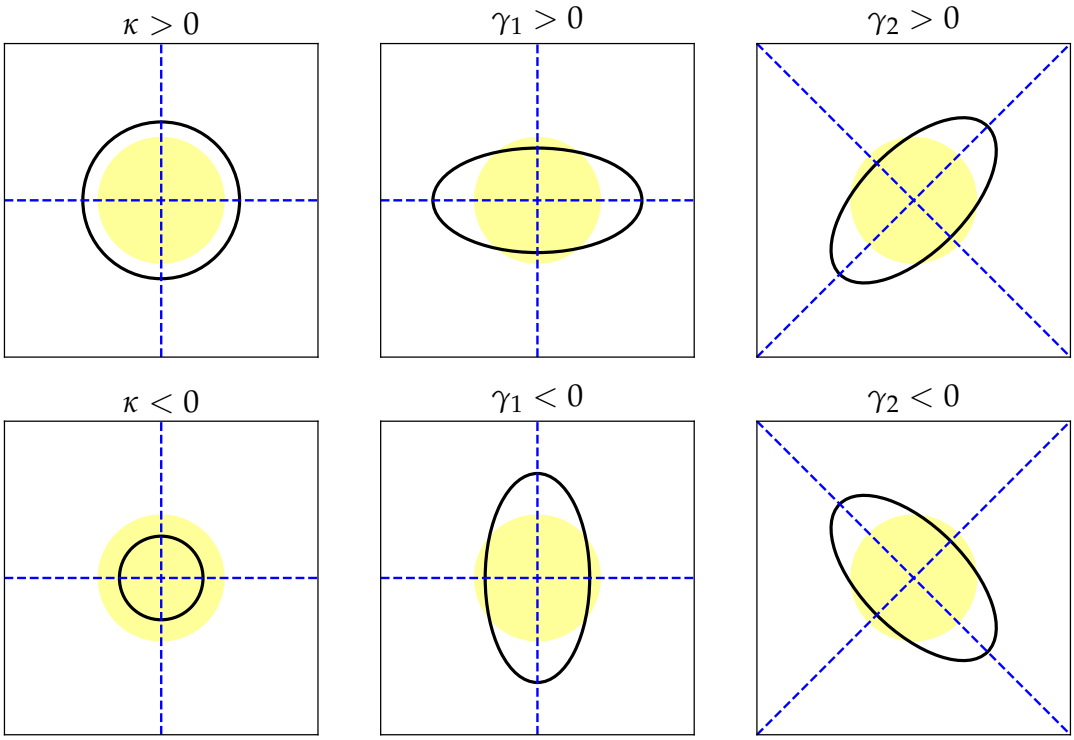


Figure 1.2 – Figure showing the various possibilities for convergence, κ and shear, γ. In particular, an increase and decrease in κ results in an isotropic expansion and contraction respectively. On the other hand, a positive change in γ_1 results in elongation along the x-axis but contraction along the y-axis, whereas a positive change in γ_2 leads to contraction along the line $y = -x$ but elongation along the line $y = x$. A similar explanation applies for the second row in the figure, with the different options for κ, γ_1 and γ_2.

$\mathbf{A}$ contains meaningful information for a weak lensing analysis. κ is referred to as the *convergence* field while $\gamma = \gamma_1 + i\gamma_2$ is the complex shear field. The values of these quantities have an overall effect on the final shape of a source, as shown in Figure 1.2. For example, a positive change in γ_1 leads to an elongation along the x-axis but a contraction along the y-axis while a positive change in γ_2 results in an elongation along the line $y = x$, but a contraction along the line $y = -x$.

These quantities, κ, γ_1 and γ_2 can be decomposed in terms of the lensing potential, ϕ, such that

$$\begin{aligned} \kappa &= \frac{1}{2}\left(\phi_{11} + \phi_{22}\right), \\ \gamma_1 &= \frac{1}{2}\left(\phi_{11} - \phi_{22}\right), \\ \gamma_2 &= \phi_{12} \end{aligned} \tag{1.2.19}$$

In addition to the above, the *magnification factor*, μ is defined as the inverse of the determinant

of the magnification matrix, $\mathbf{A}$, that is, $\mu = |\mathbf{A}|^{-1} = [(1-\kappa)^2 - |\gamma|^2]^{-1}$. For weak lensing, both $|\kappa|$ and $|\gamma|$ are $\ll 1$ and hence, $\mu \approx 1 + 2\kappa$. The magnification matrix can also be written as

$$\mathbf{A} = (1-\kappa) \begin{pmatrix} 1-g_1 & -g_2 \\ -g_2 & 1+g_1 \end{pmatrix} \tag{1.2.20}$$

where $g_i \equiv \gamma_i/1-\kappa$ is referred to as the *reduced shear*. Recall that $|\kappa| \ll 1$ and the fact that surface brightness is preserved under lensing, an actual measurement of the shear does not correspond to the shear itself but in fact to the reduced shear. We have elaborated on the shear, convergence, reduced shear and magnification factor, we will now cover briefly the approach taken to infer the shear.

1.2.3 Shear Measurements

The major contributing effect that lensing has on the observed galaxies is a change in their shapes. This then begs the question of how we can use the distorted shapes as a proxy to determine an estimator for the shear field. One approach as developed by Kaiser et al. (1995) is to derive their ellipticity statistics, ϵ, also sometimes referred to as polarisation, from the quadrupole moments of their images. In particular, in the absence of lensing, this statistics vanishes since it averages to zero because the objects are statistically isotropic, that is, they are randomly oriented. However, in the weak lensing regime, this statistics depends mildly on the gravitational shear.

These ellipticity statistics, $\boldsymbol{\epsilon}$ are generated using the moments of the surface brightness distribution of the source, that is,

$$\mathbf{Q}_{ij} \equiv \frac{\int \mathrm{d}^2\boldsymbol{\Theta}\, \theta_i \theta_j I(\boldsymbol{\Theta}) W(\boldsymbol{\Theta})}{\int \mathrm{d}^2\boldsymbol{\Theta}\, I(\boldsymbol{\Theta}) W(\boldsymbol{\Theta})} \tag{1.2.21}$$

where I is the surface brightness profile of the the galaxy on the sky and W is a weighting function. Note that we have assumed that the source is centred at the origin, but this can dealt with by replacing $\theta_i \to \theta_i - \bar{\theta}_i$. The complex ellipticity, $\epsilon = \epsilon_1 + i\epsilon_2$ can then be defined as

$$\epsilon_1 = \frac{\mathbf{Q}_{11} - \mathbf{Q}_{22}}{\mathbf{Q}_{11} + \mathbf{Q}_{22}} \qquad \epsilon_2 = \frac{2\mathbf{Q}_{12}}{\mathbf{Q}_{11} + \mathbf{Q}_{22}} \tag{1.2.22}$$

and sometimes another definition of the ellipticity can be derived by replacing the denominator by $\mathbf{Q}_{11} + \mathbf{Q}_{22} + 2\sqrt{|\mathbf{Q}|}$, where $|\mathbf{Q}| = \mathbf{Q}_{11}\mathbf{Q}_{22} - \mathbf{Q}_{12}^2$ is the determinant of the matrix $\mathbf{Q}$. These methods for deriving ellipticities via moments can also take into account the effect of the point

spread function (PSF) due to the telescope and the atmosphere (Mandelbaum, 2018). We will briefly discuss PSF in §1.4.1.2.

Defining r as the ratio of the minor axis, b to the major axis, a, for an image with elliptical isophotes, the complex ellipticity can be written as

$$\epsilon = \frac{1-r^2}{1+r^2}\exp(2i\vartheta) \tag{1.2.23}$$

where $r \leq 1$. Importantly, the factor 2 in the exponential term enforces that the complex ellipticity remains unchanged if the image is rotated by a factor of 180°.

Moreover, the moments of the source is related to the observed (image) moments via the magnification matrix, $\mathbf{A}$ by

$$\mathbf{Q}_s = \mathbf{AQA} \tag{1.2.24}$$

From Equation 1.2.24, Schneider & Seitz (1995) derived the transformation between the source ellipticity, ϵ_s and the observed ellipticity ϵ in terms of the convergence, κ and shear γ as

$$\epsilon_s = \frac{(1-\kappa)^2\epsilon - 2(1-\kappa)\gamma + \gamma^2\epsilon^*}{(1-\kappa)^2 + |\gamma|^2 - 2(1-\kappa)\Re(\gamma\epsilon^*)} \tag{1.2.25}$$

where $\gamma = \gamma_1 + i\gamma_2$ and $*$ implies the complex conjugate. Dividing the numerator and the denominator by $(1-\kappa)^2$, the above equation can be re-written in terms of the reduced shear, g such that

$$\epsilon_s = \frac{\epsilon - 2g + g^2\epsilon^*}{1 + |g|^2 - 2\Re(g\epsilon^*)}. \tag{1.2.26}$$

Moreover, the inverse transformation can be obtained by either replacing g by $-g$ or one can also start from Equation 1.2.24, that is, $\mathbf{Q} = \mathbf{A}^{-1}\mathbf{Q}_s\mathbf{A}^{-1}$ and

$$\epsilon = \frac{\epsilon_s + 2g + g^2\epsilon_s^*}{1 + |g|^2 + 2\Re(g\epsilon_s^*)}. \tag{1.2.27}$$

Seitz & Schneider (1997) extended the above derivations by considering cases where $|g| \leq 1$ and $|g| \geq 1$ and hence the transformation between the observed and the source image in terms of the ellipticity, ϵ is:

$$\epsilon_s = \begin{cases} \dfrac{\epsilon - g}{1 - g^*\epsilon} & \text{for } |g| \leq 1 \\ \dfrac{1 - g\epsilon^*}{\epsilon^* - g^*} & \text{for } |g| \geq 1 \end{cases} \tag{1.2.28}$$

If we want the inverse transformation, that is, the relation between the observed ellipticity and the source ellipticity, then $g \to -g$. For weak lensing, since $\kappa \ll 1$, $|\gamma| \ll 1$ and $|g| \ll 1$, $\epsilon \approx \epsilon_s + \gamma$.

Under the assumption that source ellipticities have no preferred orientation, $\langle \epsilon_s \rangle = 0$ and the observed ellipticity turns out to be an unbiased estimator of the (reduced) shear, that is, $\langle \epsilon \rangle = g$. However, in the presence of intrinsic alignment, which is discussed in §1.4.2.1, the relation between the observed ellipticity and the reduced shear does not hold.

In a weak lensing analysis, recall that the shear is very small and noisy. The intrinsic ellipticity dispersion, σ_ϵ is about 0.3. Therefore measurement from a single galaxy is not entirely reliable. Instead, one would need to average it over many galaxies, N, which leads to a better signal-to-noise ratio, $S/N = \gamma\sqrt{N}/\sigma_\epsilon$, equivalently to a reduced error on the average ellipticity.

1.2.4 Convergence and Shear as Spin Weight Objects

The complex shear, γ is an example of a spin-weight field. Under an anticlockwise rotation in a fixed coordinate system, the shear transforms as $\gamma \to \gamma e^{-is\psi}$, where $s = 2$ is the spin weight and ψ is the angle of rotation. There are two main key takeaways from this notation, the first is that the shear field is invariant over a rotation over π radians and the second is that under a rotation of $\psi = \pi/4$ radians, one component transforms into another, that is, $\tilde{\gamma}_1 = -\gamma_2$ and $\tilde{\gamma}_2 = \gamma_1$.

The shear field can be described using a geometrical differential operator called *edth* and is denoted by $\eth$ (Newman & Penrose, 1966; Goldberg et al., 1967; Castro et al., 2005). In general, any spin-weight function can be decomposed in a scalar part part, referred to as the electric or even or 'E-component' and a scalar curl, referred to as the magnetic or odd or 'B-component'. The shear can be written as a second *edth*-derivative of lensing potential

$$\gamma(\boldsymbol{r}) = \frac{1}{2}\eth\eth\phi(\boldsymbol{r}) \tag{1.2.29}$$

If we introduce two components, ϕ_E and ϕ_B corresponding to the even and odd parts of the lensing potential, the shear field can be written as

$$\gamma(\boldsymbol{r}) \equiv \gamma_1(\boldsymbol{r}) + i\gamma_2(\boldsymbol{r}) = \frac{1}{2}\eth\eth\left[\phi_{\rm E}(\boldsymbol{r}) + i\phi_{\rm B}(\boldsymbol{r})\right] \tag{1.2.30}$$

Note that in weak lensing, it is expected that $\phi_{\rm E}(\boldsymbol{r}) = \phi(\boldsymbol{r})$ and $\phi_{\rm B} = 0$. Therefore, the odd part, that is, the B-component can be used as a test for systematics in a weak lensing analysis. Examples of E and B patterns are shown in Figure 1.3. On the other hand, the convergence field, κ is a spin-weight 0 object and can be written in terms of the $\eth$ operators as

$$\kappa(\boldsymbol{r}) = \frac{1}{4}\left[\eth\bar{\eth} + \bar{\eth}\eth\right]\phi(\boldsymbol{r}) \tag{1.2.31}$$

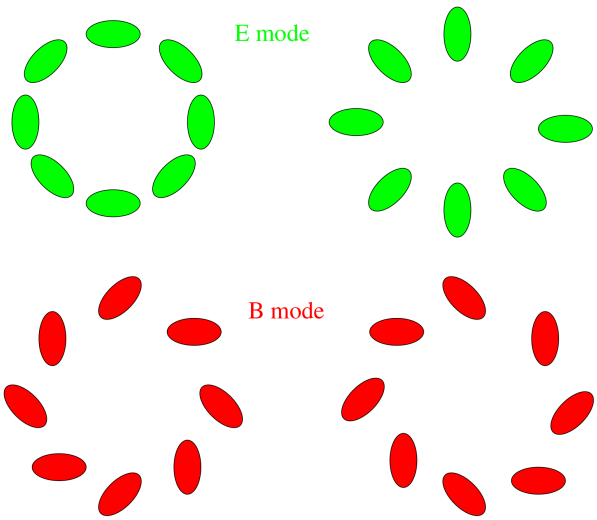


Figure 1.3 – E and B modes scenarios in weak lensing (Figure from Van Waerbeke & Mellier (2003)). In weak lensing, E-mode is the most dominant effect but the presence of B-modes is manifested as systematics in the analysis.

A detailed analysis of spin-weight objects in the weak lensing context has been performed by Heavens (2003) and extended by Castro et al. (2005). These works elegantly present a detailed and well-crafted set of information on weak lensing in 3 dimensions.

1.3 Weak Lensing Statistics

Following §1.2.1, the lensing potential, $\phi(\boldsymbol{r})$ is related to the gravitational potential, $\Phi(\boldsymbol{r})$ via a lensing kernel and involves an integral with respect to the radial distance. As a result, the statistics of the lensing potential will be very important in order to investigate the evolution and growth of structure in the Universe. Strictly, we are interested in constraining cosmological parameters given a model of the Universe.

In particular, the average shear is zero and the most common statistics in cosmology is the

two-point statistics, although higher order statistics can also be used but is quite cumbersome. In this section, we will focus only on two-point statistics. One has to choose an appropriate basis when dealing with quadratic measures, for example, if we choose to work in the pixel space, then two-point statistics of the lensing field is the real-space correlation function while the two-point function is the power spectrum if we choose the harmonic space. The following content has been adapted from Castro et al. (2005) and we refer the reader to this paper for a detailed overview.

1.3.1 Spherical Coordinates and Spherical Bessel Functions

Since we are interested in computing lensing power spectra, this will require spectral expansion of the lensing field. A natural choice is the spherical coordinate system for various reasons, despite the fact that sky coverage is small. Recall that the lensing potential is a radial integral and distance measurements via photometric redshifts are also radial errors. In general, if we are working with Fourier transforms in Cartesian coordinates, the eigenfunctions are the exponential functions. However, in spherical coordinates, the product of the spherical harmonics and the spherical Bessel functions, $Y_{\ell m}(\theta,\varphi)j_\ell(kr)$, turns out to be the eigenfunctions of the Laplacian operator. Consider a scalar field $f(\boldsymbol{r})$ with a flat background geometry. The 3D spherical harmonic transform is defined as

$$f_{\ell m}(k) \equiv \sqrt{\frac{2}{\pi}} \int \mathrm{d}^3\boldsymbol{r}\, f(\boldsymbol{r}) k j_\ell(kr) Y^*_{\ell m}(\theta,\varphi) \tag{1.3.1}$$

and its inverse transform is

$$f(\boldsymbol{r}) = \sqrt{\frac{2}{\pi}} \int k\mathrm{d}k \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} f_{\ell m}(k) j_\ell(kr) Y_{\ell m}(\theta,\varphi) \tag{1.3.2}$$

where $j_\ell(kr)$ is the spherical Bessel function, Y is a spherical harmonic and k is a wavenumber. Using Equation 1.2.14 and Equation 1.3.1 the relation between the lensing and gravitational potential coefficients is

$$\phi_{\ell m}(k) = \frac{4k}{\pi c^2} \int_0^\infty \mathrm{d}k'\, k' \int_0^\infty \mathrm{d}r\, r j_\ell(kr) \int_0^r \mathrm{d}r' \left(\frac{r-r'}{r'}\right) j_\ell(k'r') \Phi_{\ell m}(k';r'). \tag{1.3.3}$$

On the other hand, one can straightforwardly express the coefficients of the gravitational potential and the over-density, which are related via the Poisson's equation - see Equation 1.2.8, that is,

$$\Phi_{\ell m}(k;r) = -\frac{3\Omega_m H_0^2}{2k^2 a(r)}\,\delta_{\ell m}(k,r). \tag{1.3.4}$$

Note 1.2: Spin-Weight Spherical Harmonics

A spin-weight s function, ${}_sf(\boldsymbol{x})$ can be expressed in terms of the its spin spherical harmonics as

$${}_sf(\boldsymbol{x}) = \int_0^\infty \mathrm{d}k \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell} \left[a_{s,\ell m}(k)\right]\, {}_sZ_{klm}(x,\theta,\varphi) \tag{A1}$$

and its inverse transform is

$$a_{s,\ell m}(k) = \int \mathrm{d}^3\boldsymbol{x}\, \left[{}_sf(\boldsymbol{x})\right]\, {}_sZ^*_{klm}(x,\theta,\varphi) \tag{A2}$$

where

$${}_sZ_{klm}(x,\theta,\varphi) = \sqrt{\frac{2}{\pi}} k j_\ell(kx) {}_sY_{\ell m}(\theta,\varphi) \tag{A3}$$

and ${}_sY_{\ell m}(\theta,\varphi)$ corresponds to the spin-weight spherical harmonics. The latter also satisfy the *orthogonality* relation

$$\int_0^{2\pi}\mathrm{d}\varphi \int_{-1}^{1} \mathrm{d}\cos\theta\; {}_{s'}Y^*_{\ell' m'}(\theta,\varphi)\, {}_sY_{\ell m}(\theta,\varphi) = \delta_{\ell\ell'}\delta_{mm'}\delta_{ss'},$$

whereas the basis functions ${}_sZ_{k\ell m}(x,\theta,\varphi)$ are *orthonormal*, that is,

$$\int \mathrm{d}^3x\; {}_sZ_{k\ell m}(x,\theta,\varphi)\, {}_{s'}Z^*_{k'\ell' m'}(x,\theta,\varphi) = \delta_D(k-k')\delta_{\ell\ell'}\delta_{mm'}\delta_{ss'}.$$

Note that

$$\int x^2 \mathrm{d}x \left[\sqrt{\frac{2}{\pi}} k j_\ell(kx)\right]\left[\sqrt{\frac{2}{\pi}} k' j_\ell(k'x)\right] = \delta_D(k-k')$$

In the same spirit, the expansion of other field can be performed in a similar fashion, that is, we can establish relations between ϕ, Φ and δ interchangeably in terms of the harmonic coefficients. Note that, unlike Φ and δ, ϕ is not a homogeneous and isotropic field in 3D, that is, we will strictly focus on writing the coefficients as a function of the radial distance, for example, $\Phi_{\ell m}(k,r)$ and $\delta_{\ell m}(k,r)$. In the next section, we will look into spin-weight spherical harmonics

because the observable shear field is not a scalar field but rather a spin-weight 2 field.

1.3.2 Spin-Weight Spherical Harmonics

Before delving into the statistics of the shear field, following Equation 1.2.29, the complex conjugate of the shear field is

$$\gamma^*(\boldsymbol{r}) = \frac{1}{2}\bar{\eth}\bar{\eth}\phi(\boldsymbol{r}) \tag{1.3.5}$$

and each orthogonal component, $\gamma_1(\boldsymbol{r})$ and $\gamma_2(\boldsymbol{r})$ can be expressed in a similar way in terms of the $\eth$ operators, that is,

$$\begin{aligned}\gamma_1(\boldsymbol{r}) &= \frac{1}{4}(\eth\eth + \bar{\eth}\bar{\eth})\phi(\boldsymbol{r}),\\ \gamma_2(\boldsymbol{r}) &= \frac{i}{4}(\eth\eth - \bar{\eth}\bar{\eth})\phi(\boldsymbol{r}).\end{aligned} \tag{1.3.6}$$

The shear field can be decomposed into two orthogonal components as $\gamma(\boldsymbol{r}) = \gamma_1(\boldsymbol{r}) + i\gamma_2(\boldsymbol{r})$. As a result, we can also introduce two scalar fields, $\phi_{\rm E}(\boldsymbol{r})$ for the even part and $\phi_{\rm B}(\boldsymbol{r})$ for the odd part, that is, we will re-write the lensing potential as $\phi(\boldsymbol{r}) = \phi_{\rm E}(\boldsymbol{r}) + i\phi_{\rm B}(\boldsymbol{r})$. In the context of weak lensing, there should be no B-component and this serves as a systematic test when constraining cosmological parameters. In short, we have the following:

$$\gamma(\boldsymbol{r}) = \frac{1}{2}\eth\eth[\phi_{\rm E}(\boldsymbol{r}) + i\phi_{\rm B}(\boldsymbol{r})] \qquad \gamma^*(\boldsymbol{r}) = \frac{1}{2}\bar{\eth}\bar{\eth}[\phi_{\rm E}(\boldsymbol{r}) - i\phi_{\rm B}(\boldsymbol{r})]. \tag{1.3.7}$$

Note 1.3: Properties of Spin Weight Spherical Harmonics

Some important properties of spin-weight spherical harmonics are summarised below:

$$\begin{aligned}\eth\,{}_sY_{\ell m} &= [(\ell - s)(\ell + s + 1)]^{1/2}\,{}_{s+1}Y_{\ell m}\\ \bar{\eth}\,{}_sY_{\ell m} &= -[(\ell + s)(\ell - s + 1)]^{1/2}\,{}_{s-1}Y_{\ell m}\\ \bar{\eth}\eth\,{}_sY_{\ell m} &= -(\ell - s)(\ell + s + 1)\,{}_sY_{\ell m}\\ \eth\bar{\eth}\,{}_sY_{\ell m} &= -(\ell + s)(\ell - s + 1)\,{}_sY_{\ell m}\end{aligned}$$

and

$$\begin{aligned}\eth\eth Y_{\ell m} &= \sqrt{\frac{(\ell+2)!}{(\ell-2)!}}\,{}_2Y_{\ell m}\\ \bar{\eth}\bar{\eth}Y_{\ell m} &= \sqrt{\frac{(\ell+2)!}{(\ell-2)!}}\,{}_{-2}Y_{\ell m}\end{aligned} \tag{B1}$$

Using Note 1.2 and Note 1.3, we can express $\eth\eth\phi_{\mathrm{E}}(\boldsymbol{r})$ as

$$\begin{aligned}\eth\eth\phi_{\mathrm{E}}(\boldsymbol{r}) &= \int_0^\infty \mathrm{d}k \sum_{\ell=2}^{\infty}\sum_{m=-\ell}^{\ell}\sqrt{\frac{(\ell+2)!}{(\ell-2)!}}\phi_{\mathrm{E},\ell m}(k)\sqrt{\frac{2}{\pi}}kj_\ell(kr)\,{}_2Y_{\ell m} \\ &= \int_0^\infty \mathrm{d}k \sum_{\ell=2}^{\infty}\sum_{m=-\ell}^{\ell}\sqrt{\frac{(\ell+2)!}{(\ell-2)!}}\phi_{\mathrm{E},\ell m}(k)\,{}_2Z_{k\ell m}(r,\theta,\varphi)\end{aligned} \tag{1.3.8}$$

In a similar way, other expressions for $\eth\eth\phi_{\mathrm{B}}(\boldsymbol{r})$, $\bar{\eth}\bar{\eth}\phi_{\mathrm{E}}(\boldsymbol{r})$ and $\bar{\eth}\bar{\eth}\phi_{\mathrm{B}}(\boldsymbol{r})$. In summary, the shear field and its complex conjugate, in terms of the spin-2 spherical harmonics are:

$$\begin{aligned}\gamma(\boldsymbol{r}) &= \int_0^\infty \mathrm{d}k \sum_{\ell=2}^{\infty}\sum_{m=-\ell}^{\ell}{}_2\gamma_{\ell m}(k)\,{}_2Z_{k\ell m}(r,\theta,\varphi) \\ \gamma^*(\boldsymbol{r}) &= \int_0^\infty \mathrm{d}k \sum_{\ell=2}^{\infty}\sum_{m=-\ell}^{\ell}{}_{-2}\gamma_{\ell m}(k)\,{}_{-2}Z_{k\ell m}(r,\theta,\varphi)\end{aligned} \tag{1.3.9}$$

Multiple results follow from the derivations above. In the case of weak lensing, the B-component is 0 and hence,

$${}_{\pm2}\gamma_{\ell m}(k) = \frac{1}{2}\sqrt{\frac{(\ell+2)!}{(\ell-2)!}}\phi_{\ell m}(k) \tag{1.3.10}$$

and the orthogonal components, γ_1 and γ_2, of the shear field in terms of the coefficients of the lensing potential are:

$$\begin{aligned}\gamma_{1,\ell m}(k) &= \frac{1}{2}\sqrt{\frac{(\ell+2)!}{(\ell-2)!}}\phi_{\ell m}(k) \\ i\gamma_{2,\ell m}(k) &= \frac{1}{2}\sqrt{\frac{(\ell+2)!}{(\ell-2)!}}\phi_{\ell m}(k)\end{aligned} \tag{1.3.11}$$

Note that, new bases are defined to obtain the latter, that is,

$$X_{1,k\ell m} = \frac{1}{2}({}_2Z_{k\ell m} + {}_{-2}Z_{k\ell m}) \qquad X_{2,k\ell m} = \frac{1}{2}({}_2Z_{k\ell m} - {}_{-2}Z_{k\ell m})$$

Furthermore, we can repeat this process to find the expansion coefficients of the convergence field. Recall that the convergence field is a spin-0 object. From Note 1.3, we have $\bar{\eth}\eth Y_{\ell m} + \eth\bar{\eth}Y_{\ell m} = -2\ell(\ell+1)Y_{\ell m}$ and from Equation 1.2.31, the expansion coefficients of κ are

$$\kappa_{\ell m}(k) = -\frac{1}{2}\ell(\ell+1)\phi_{\ell m}(k) \tag{1.3.12}$$

Now that we have obtained the coefficients of the shear and convergence field in terms of the coefficients of the lensing potential, one could also derive other expressions in terms of the expansion coefficients of the gravitational potential, Φ and eventually in terms of the over-density field via the Poisson's equation. Along these lines, the observable shear field and the over-density field, which is a function of cosmological parameters enable us to constrain these parameters, for example, using a Bayesian formalism. Next, we look into computing weak lensing power spectra.

1.3.3 Weak Lensing Power Spectra

In this section, we use the coefficients derived in the previous section to obtain expressions for the weak lensing power spectrum. In particular, we are interested in relating the 3D shear and convergence power spectra with the 3D gravitational potential power spectra $C_\ell^{\Phi\Phi}$. For a statistically homogeneous and isotropic field $f(\boldsymbol{r}; r)$ at radial distance r (note that r and time t can be used interchangeably since they are physically equivalent) with coefficients $f_{\ell m}(k; r)$, we have

$$\langle f_{\ell m}(k;r) f^*_{\ell' m'}(k';r)\rangle = C_\ell(k;r)\delta_D(k-k')\delta_{\ell\ell'}\delta_{mm'} \tag{1.3.13}$$

where $\delta_{\alpha\beta}$ is the Kronecker delta function and $\delta_D(x)$ is a 1D Dirac delta function. In the same spirit, for the lensing power spectrum, we have

$$\langle \phi_{\ell m}(k)\phi^*_{\ell' m'}(k'\rangle = C_\ell^{\phi\phi}\delta_{\ell\ell'}\delta_{mm'}. \tag{1.3.14}$$

where $C_\ell^{\phi\phi}$ is the 3D lensing potential power spectrum. However, recall that the lensing potential ϕ is not homogeneous and isotropic in 3D. The same property holds for the shear, convergence and lensing potential as they share the same statistical property. Following the derivations in §1.3.2, the following expressions for power spectra can be obtained straightforwardly:

$$C_\ell^{\gamma\gamma}(k_1,k_2) = \frac{1}{4}\frac{(\ell+2)!}{(\ell-2)!}C_\ell^{\phi\phi}(k_1,k_2) \tag{1.3.15}$$

$$C_\ell^{\gamma_1\gamma_1}(k_1,k_2) = \frac{1}{4}\frac{(\ell+2)!}{(\ell-2)!}C_\ell^{\phi\phi}(k_1,k_2) \tag{1.3.16}$$

$$C_\ell^{\gamma_2\gamma_2}(k_1,k_2) = \frac{1}{4}\frac{(\ell+2)!}{(\ell-2)!}C_\ell^{\phi\phi}(k_1,k_2) \tag{1.3.17}$$

$$C_\ell^{\kappa\kappa}(k_1,k_2) = \frac{1}{4}\ell^2(\ell+1)^2 C_\ell^{\phi\phi}(k_1,k_2) \tag{1.3.18}$$

While the above expressions relate the weak lensing power spectra in terms of the lensing potential power spectrum, a final step is to express the lensing potential spectrum itself in terms of the gravitational lensing potential power spectrum, $C_\ell^{\Phi\Phi}$. Using Equation 1.3.3, we have

$$C_\ell^{\phi\phi}(k_1,k_2) = \frac{16}{\pi^2 c^4}\int_0^\infty k^2 \mathrm{d}k\; I_\ell(k_1,k) I_\ell(k_2,k) \tag{1.3.19}$$

where

$$I_\ell(k_i,k) \equiv k_i \int_0^\infty \mathrm{d}r\, r j_\ell(k_i r) \int_0^r \mathrm{d}r' \left(\frac{r-r'}{r'}\right) j_\ell(kr')\sqrt{P_\Phi(k;r')}$$

Note that we have introduced the usual 3D power spectrum of the gravitational potential, that is, $C_\ell^{\Phi\Phi}(k;r,r') = P_\Phi(k;r,r')$ and we have also assumed that $P_\Phi(k;r,r') \simeq \sqrt{P_\Phi(k;r)\,P_\Phi(k;r')}$.

1.3.4 Approximations

The weak lensing power spectra can further be simplified using various approximations. The main reasons for further simplifications include numerical stability, computational time an various others. In this section, we will cover three different approximations, namely flat-sky approximation, tomographic approximation and Limber approximation. Some of these can also be used concurrently.

Flat Sky Approximation

In the flat sky approximation limit, instead of the combination (spin-weight) spherical harmonics and Bessel functions, one can use a combination of Fourier modes and Bessel functions to perform the forward and inverse transformation of a 3D field f, that is,

$$f(r,\vec{\theta}) = \sqrt{\frac{2}{\pi}}\int_0^\infty k\,\mathrm{d}k \int_0^\infty \frac{\mathrm{d}^2\vec{\ell}}{(2\pi)^2} f(k,\vec{\ell}) j_\ell(kr) e^{i\vec{\ell}\cdot\vec{\theta}} \tag{1.3.20}$$

$$f(k,\vec{\ell}) = \sqrt{\frac{2}{\pi}}\int_0^\infty r^2\,\mathrm{d}r \int_0^\infty \mathrm{d}^2\theta\, f(r,\vec{\theta}) k j_\ell(kr) e^{-i\vec{\ell}\cdot\vec{\theta}} \tag{1.3.21}$$

where $\vec{\ell} = (\ell\cos\varphi_\ell, \ell\sin\varphi_\ell)$ is a 2D angular wavenumber and $\vec{\theta} = (\theta\cos\varphi, \theta\sin\varphi)$. See Appendix C from Hu (2000) or Appendix A2 from Santos et al. (2003) for further details. Following Castro et al. (2005) and Heavens et al. (2013), the spherical harmonic expansion of the shear and convergence in terms of the coefficients of the lensing potential can be approximated as:

$$\kappa_{\ell m} \approx -\frac{1}{2}\ell^2\phi_{\ell m}, \tag{1.3.22}$$

$$\gamma_{1,\ell m} \approx \frac{1}{2}\ell^2\phi_{\ell m}, \tag{1.3.23}$$

$$i\gamma_{2,\ell m} \approx \frac{1}{2}\ell^2\phi_{\ell m}. \tag{1.3.24}$$

As a result, the flat-sky approximated shear and convergence power spectra are:

$$\hat{C}_\ell^{\gamma\gamma} = \frac{1}{4}\ell^4 C_\ell^{\phi\phi}, \tag{1.3.25}$$

$$\hat{C}_\ell^{\kappa\kappa} = \frac{1}{4}\ell^4 C_\ell^{\phi\phi}. \tag{1.3.26}$$

Once these expressions are derived, various work-streams have used them for forecasting cosmological parameter constraints, see for example Heavens et al. (2006). If surveys have large opening angles on the sky, the combination of spherical Bessel functions and (spin-weight) spherical harmonics remains the obvious choice. However, for small angle surveys, most of the information lies in the signal with the high-ℓ mode. $\ell \gtrsim 100$ is a good criterion where most of the cosmological information is. Computing spherical harmonics at large ℓ can pose computational challenges. Instead, a flat-sky expansion as explained in this section can be very useful.

Limber Approximation

For angular modes, $\ell \gtrsim 100$, one can use the *Limber approximation* (Limber, 1953; Loverde & Afshordi, 2008; Lemos et al., 2017) to approximate the integration over the spherical Bessel functions when computing power spectra. In general, the exact computation of the power spectrum, including the integration over the spherical Bessel functions is very time consuming as a result of the high oscillation of the Bessel functions at large multipoles. Instead, the Limber approximation implies,

$$j_\ell(kr) \to \sqrt{\frac{\pi}{2\nu}}\delta_D(\nu - kr) \tag{1.3.27}$$

that is, the spherical Bessel function is replaced by a delta function. $\nu = \ell + 1/2$ and the radial distance and the wavenumber k are related by $\nu = kr$. Therefore,

$$\int \mathrm{d}r\, f(r) j_\ell(kr) \to \sqrt{\frac{\pi}{2\nu}}\, f\left(\frac{\nu}{k}\right) \tag{1.3.28}$$

and this approximation means that the following integration can effectively be replaced by the expression on the right. In Chapter 5, we will look into how we can use Gaussian Process as a technique to accelerate inference of cosmological parameters. This method can also be applied to the case where we would like to accelerate power spectra computation whilst retaining full formalism without the Limber approximation.

Tomographic Weak Lensing Power Spectra

Lensing is strictly a 3D effect but performing a full statistical analysis in 3D can pose significant challenges. Recall that it can also be mathematically summarised as an integrated (radial) effect along the line of sight. Instead, we can opt to perform an analysis by congregating observed galaxies into redshift slices, α (hence the notion of tomography), with minimal loss of information compared to the full 3D statistical analysis. Each group of galaxies follows a specific redshift distribution where

$$n_\alpha(z)\,\mathrm{d}z = n_\alpha(r)\,\mathrm{d}r \tag{1.3.29}$$

and $\int n_\alpha(r)\,\mathrm{d}r = 1$. As a result, the lensing field, $\phi(\boldsymbol{r})$ averaged over the redshift distribution $n_\alpha(r)$ can be written as

$$\phi_\alpha(\theta, \varphi) = \int_0^\infty \phi(\boldsymbol{r}) n_\alpha(r)\,\mathrm{d}r. \tag{1.3.30}$$

Before expanding on the details of weak lensing tomography, it is easier to use the convergence as an example to illustrate the concept behind. The convergence in terms of the matter over-density field, $\delta(\boldsymbol{r})$ via the Poisson equation is

$$\kappa(\boldsymbol{r}) = \frac{3H_0^2\Omega_m}{2c^2} \int_0^r \mathrm{d}r'\, \frac{r'(r-r')}{r}\, \frac{\delta(\boldsymbol{r}')}{a(r')} \tag{1.3.31}$$

The above equation tells us that the convergence is effectively an integrated effect of the matter over-density weighted by the comoving angular diameter distance and the scale factor. However, while this is for a single, fixed source, the (mean) convergence for a multiple sources can be calculated by averaging over the normalised source distribution, $n(r)$, that is,

$$\bar{\kappa}(\boldsymbol{r}) = \int_0^\infty \mathrm{d}r\, n(r)\kappa(\boldsymbol{r}) \tag{1.3.32}$$

We can further simplify the expression for the convergence and introduce the weighting function, $g(r)$ (given by Equation 1.2.16), that is,

$$\bar{\kappa}(\boldsymbol{r}) = \frac{3H_0^2\Omega_m}{2c^2}\int_0^\infty \mathrm{d}r\, \frac{rg(r)}{a(r)}\delta(\boldsymbol{r}). \tag{1.3.33}$$

The convergence power spectrum, in terms of the three-dimensional matter power spectrum reads

$$C_\ell^{\kappa\kappa} = \left(\frac{3H_0^2\Omega_m}{2c^2}\right)^2 \int_0^\infty \mathrm{d}r\, \left[\frac{g(r)}{a(r)}\right]^2 P_\delta\left(k = \frac{\ell + 1/2}{r}; r\right) \tag{1.3.34}$$

Note that the Limber approximation is assumed in this procedure. If we have photometric redshift information of source galaxies, we can introduce the notion of tomography where galaxies are subdivided into redshift bins, such that the number of density of galaxies in redshift bin α is defined to be between r_i and r_{i+1}, that is,

$$n_\alpha = \int_{r_i}^{r_{i+1}} \mathrm{d}r\, n(r)$$

As a result, the auto- and cross- convergence power spectrum between bins α and β is

$$C_{\ell,\alpha\beta}^{\kappa\kappa} = \left(\frac{3H_0^2\Omega_m}{2c^2}\right)^2 \int_0^\infty \mathrm{d}r\, \frac{g_\alpha(r)g_\beta(r)}{a^2(r)} P_\delta\left(k = \frac{\ell + 0.5}{r}; r\right). \tag{1.3.35}$$

For N redshift bins, we have $\frac{1}{2}N(N+1)$ auto- and cross- power spectra to compute. Current weak lensing surveys employ 3 to 5 redshift bins but it is anticipated that this number will increase to 10 in future surveys. In most weak lensing analysis, we rather work with summary statistics, which essentially involve compression of the 2-point statistics. In Chapter 9, we will cover correlation functions, as well as, summary statistics such as band powers and COSEBIs.

1.4 Challenges

Performing a weak lensing analysis is not a straightforward process. From the observation of million of galaxies to constraining cosmological parameters, including testing different cosmological models, we also have to account for various systematics and challenges in the pipeline. In this section, we cover some of these scientific and observational challenges, which include modelling the point spread function, intrinsic alignment, baryon feedback and many others. For a thorough understanding of this topic, we refer the reader to well-crafted reviews by Kilbinger (2015) and Mandelbaum (2018).

1.4.1 Observational Challenges

In this section, we will cover some of the observational challenges faced from the stage of observing a source to the stage where we have a compressed format of the data (usually the ellipticities) which can be used to perform inference.

1.4.1.1 Shape Measurements

In §1.2.2, we discussed how the ellipticities of observed galaxies are crucial to estimate the shear, which is eventually used in a cosmological data analysis pipeline. In particular, the very first step of accurately measuring the shape of galaxies poses a challenge and an attempt to improve over existing techniques has been investigated in various papers, see for example early work by Kaiser et al. (1995).

A common approach is via forward modelling technique, that is, the procedure involves fitting a model consisting of the ellipticity parameters and surface brightness to the observed image. In particular, the effect of the PSF can be accounted for by first convolving the model with the PSF before fitting it to the data. However, the main challenge is that images are usually from multiple exposures. One workaround, as developed in *lensfit* (Miller et al., 2007; Kitching et al., 2008; Miller et al., 2013), is to employ a fully Bayesian framework to find the posterior distribution of the ellipticities of galaxies, with minimal loss of information. However, one challenge resides in determining the number of free parameters in a model. A model with only a few parameters can lead to model bias, under-fitting while a model with too many parameters tend to fit the noise and may result in over-fitting.

1.4.1.2 Point Spread Function

The correction for the PSF is a main challenge in inferring galaxy shapes. The PSF impact the shear estimate, resulting in a multiplicative bias and also contributes to an additive bias in the galaxy ellipticity values. If ground-based telescopes are used for observing, the observed images are further deteriorated due to atmospheric PSF. Accurate determination of the PSF is a critical challenge for a weak lensing analysis. PSF model misspecification can trivially lead to multiplicative and additive biases.

Typical technique for modelling PSF includes two steps. In the fist instance, a bright star is used a reference to model the PSF and the second is to interpolate to other positions in order to measure galaxy photometry and shapes. In the past, it has been found that the choice of

the interpolation scheme may affect the measured cosmic signal (Hoekstra, 2004). As a result, different interpolating schemes have been investigated with a view to improve PSF modelling.

1.4.1.3 Redshift Distribution

As described in §1.3.4, the convergence (or shear) power spectrum is an integral weighted by the source galaxy distribution, $n(z)$ via the weighting function $g(r)$. Hence, for robust cosmological parameter inference, not only the redshift of the source galaxies have to be determined accurately but also the full $n(z)$ distribution. For a tomographic weak lensing analysis, Huterer et al. (2006) estimated that the centroids of each bin have to be calculated to better than a per cent to improve the accuracy of a dark energy model by almost 50%.

In a typical weak lensing survey, millions of galaxies are observed and it is an arduous task, both in terms of cost and time, to determine the redshift of these galaxies via spectroscopic method. As a consequence, *photometric redshifts* are estimated from broad-band photometry. There are various techniques for estimating photometric redshifts. Template-based method is a common approach where templates of Spectral Energy Distribution (SED) are used to perform a χ^2 fit to the flux in the observed bands. However, as anticipated by the reader, there will be a fixed set of models and it is important to design robust algorithms to prevent model-misspecification. One possibility is to adopt a Bayesian approach. Indeed, *Bayesian Photometric Redshift Estimation* (BPZ) is a common tool for estimating the mean redshift, along with finding the full posterior distribution function of the redshift (Benítez, 2000).

Moreover, ML-based approaches have also been used to improve upon existing methods. The idea behind ML is to learn a non-linear function which maps the fluxes to redshifts by using a set of spectroscopic redshifts. Once the function is learnt, one could use it to predict the redshift at any test fluxes. For example, Collister & Lahav (2004) developed a neural network method, ANNz which is deemed to be competitive compared to template-based method. Recently, Jones & Heavens (2019a,b) improved upon the existing BPZ method to estimate photometric of blended sources, where the latter itself is another challenge for weak lensing. On the other hand, Leistedt et al. (2016) developed a Hierarchical Bayesian Inference routine to infer the redshift distribution of galaxies using photometric redshifts and is actually what we need in a weak lensing analysis.

1.4.2 Scientific Challenges

Once we have a compressed dataset, the standard approach is to fit a model to the data. However, this is not a trivial process in a weak lensing analysis since we have to account for various systematics effects. In other words, we have to marginalise over the nuisance parameters.

1.4.2.1 Intrinsic Alignment

Intrinsic alignment is a major theoretical concern for weak lensing. In short, the intrinsic alignment of galaxies refers to preferential and coherent orientation of galaxy shapes as a result of physical effects apart from lensing alone. In general, one would assume that the shape alignments are due to lensing only, but such an assumption can weaken a weak lensing analysis. For example, Singh & Mandelbaum (2016) found that intrinsic alignment is a major bottleneck that future surveys such as the Vera C. Rubin Observatory (previously referred to as the Large Synoptic Survey Telescope, LSST) must mitigate.

There are two main mechanisms which are believed to contribute to intrinsic alignment. The observed ellipticity of a galaxy can be summarised as

$$\epsilon_i = \gamma_i^{\mathrm{G}} + \gamma_i^{\mathrm{I}} + \epsilon_i^{R} \tag{1.4.1}$$

for a photo-z bin i. G, I and R refers to gravitational shear, intrinsic shear and random unlensed ellipticity respectively. The observed ellipticity is modelled as a combination of a gravitational shear component, γ^{G}, an intrinsic component, γ^{I} as a result of alignment of a galaxy in its local environment and an uncorrelated component, ϵ^{R} for the random intrinsic orientations. In terms of power spectra, the tomographic two point observables can be summarised by

$$C_{\ell,ij}^{\mathrm{EE}} = C_{\ell,ij}^{\mathrm{GG}} + C_{\ell,ij}^{\mathrm{GI}} + C_{\ell,ij}^{\mathrm{II}} \tag{1.4.2}$$

Note that the C^{IG} term is negligible since the background (G) and foreground (I) do not correlate. In real data analysis, this term will not be zero due to photometric redshift contamination but this is negligible in the different analyses we perform in this thesis. The shear signal (GG) is the main and clean proxy for constraining cosmological parameters. The additional components GI and II may also contribute to the final signal but is poorly understood. The first tern, GI, arises as a result of cross-correlation between intrinsic ellipticity and gravitational shear. The second term, II, arises due to correlation of ellipticities of nearby galaxies since the shapes and orientations are affected by the local tidal gravitational field, thus giving rise to a

preferential orientation. The first term (GI) subtracts from the measured signal while the second term (II) adds positively to it (Hirata & Seljak, 2004; Joachimi & Bridle, 2010). These effects can be modelled when constraining cosmological parameter in a likelihood analysis - see Chapters 4, 5 and 7 where we incorporated intrinsic alignment effects in our likelihood analysis. We will discuss the models for intrinsic alignments in the these chapters later. See §4.2.1 for further details.

1.4.2.2 Baryon Feedback

Baryon feedback is of the various astrophysics systematics that needs to be accounted for when constraining cosmological parameters from shear power spectra measurement. This process is poorly understood and depends on high-resolution N-body simulation.

Active galactic nuclei (AGN) feedback changes the matter distribution at small scales, leading to a modification of the dark matter power spectrum at large scales. Indeed, Semboloni et al. (2011) argued that strong feedback is required at scales relevant for weak lensing analysis. In fact, they argue that baryon feedback may lead to significant biases in the inferred cosmological parameters. Moreover, van Daalen et al. (2011) found that one should not ignore baryon processes, particularly AGN feedback in the calculation of theoretical power spectra for $k \gtrsim 0.3\, h\,\mathrm{Mpc}^{-1}$. The effect of baryons is typically quantified via a bias function

$$b^2(k,z) \equiv \frac{P_\delta^{\mathrm{mod}}(k,z)}{P_\delta^{\mathrm{ref}}(k,z)} \tag{1.4.3}$$

where $P_\delta^{\mathrm{mod}}(k,z)$ and $P_\delta^{\mathrm{ref}}(k,z)$ refer to the power spectra with and without baryon feedback respectively. Baryon feedback leads to a significant of power at large multipoles. For weak lensing surveys, in order to mitigate the bias on inferred cosmological parameters, a recommended approach is to marginalise over the feedback parameters which is linked to the effects of baryonic processes. See Chapter 4 where we implemented a fitting formula for baryon feedback to model the power spectrum in our likelihood analysis.

1.5 Surveys

In this era of big data, current and future-planned cosmological surveys are becoming more and more data-intensive, with the view to understanding better the Universe. Data from existing surveys have been used to constrain cosmological parameters and they are also being used as a proxy to improve our understanding of the various theoretical and observational challenges

explained in the previous section. In this section, we highlight briefly some surveys (past, current and future). We focus on the most popular surveys and the surveys elaborated below is not an exhaustive list of *all* surveys.

1.5.1 CFHTLenS

The Canada-France-Hawaii Telescope Lensing Survey (CFHTLenS) determines the weak gravitational lensing signal from data obtained from the CFHT Legacy Survey (CFHTLS). It is a 154 deg^2 survey using five optical bands, namely, *ugriz*. The data extend across four different fields denoted by W1 ($\sim$ 63.8 deg^2), W2 ($\sim$ 22.6 deg^2), W3 ($\sim$ 44.2 deg^2) and W4 ($\sim$ 23.3 deg^2). Following the redshift distribution estimation, the galaxy sample has a median redshift of 0.70 and a mean redshift of 0.75 (Erben et al., 2013).

The tomographic weak lensing analysis as performed by Heymans et al. (2013) for a flat ΛCDM cosmology, inferred the normalisation of the matter power spectrum, $\sigma_8 = 0.799 \pm 0.015$ and the matter density parameter, $\Omega_\mathrm{m} = 0.271 \pm 0.010$. Note that the final cosmology data product consisted of 21 sets of shear correlation functions as a result of 6 redshift bins, for an angular range of $1.5 < \theta < 35$ arcmin. Moreover, for a wCDM cosmology, the dark energy equation of state parameter for this particular dataset is inferred to be , $w = -1.02 \pm 0.09$. In addition to the results obtained from this analysis, the main takeaways are to develop new statistical techniques which are less sensitive to intrinsic alignment, and to explore the possibility of including more reliable and robust photometric redshifts and complementary spectroscopic data.

1.5.2 KiDS

The Kilo Degree Survey (KiDS) is a 1500 deg^2 optical survey in four bands *ugri* (de Jong et al., 2013). It is expected that KiDS will detect around 10^8 galaxies and 2×10^4 clusters. The KiDS catalogue will consist of around 10^5 sources per deg^2. The main scientific goals of this surveys is to elucidate the nature of dark energy by mapping the distribution of matter through techniques such as gravitational weak lensing.

Here, we focus on the tomographic weak lensing analysis performed by Köhlinger et al. (2017). At this stage, only 450 deg^2 of imaging data from KiDS was used in this process, hence referred to as KiDS-450. This final cosmological data product consists of 6 sets of band powers, corresponding to 3 redshift bins. The analysis is restricted to a multipole range of $76 \leq \ell \leq 1310$ and the analysis is performed in a fully Bayesian framework with a ΛCDM model with a flat

geometry.

The main cosmological result is expressed in terms of the parameter combination, S_8 defined as $S_8 \equiv \sigma_8\sqrt{\Omega_m/0.3}$ and is determined to be $S_8 = 0.759^{+0.024}_{-0.021}$. This value is at 3σ tension with the constraints derived by *Planck* for the CMB (Asgari et al., 2021). The KiDS-450 data set is at the centre of this thesis and we will elaborate more on it in Chapters 4, 5 and 7.

1.5.3 DES

The Dark Energy Survey (DES) is a near-infrared and visible survey (central wavelengths of roughly 472 nm to 1 μm) with the aim of understanding the Physics behind the large scale structure of the Universe. It is a 5000 deg^2 survey in the southern sky in five photometric bands *grizY*. DES has observed the sky for 6 years and consisted of 758 observing nights. In particular, the first data release consists of more than 400 million objects, many of them believed to be galaxies. This gigantic survey has 4 scientific probes, namely, weak gravitational lensing, Type 1a supernovae, number of galaxy clusters and the baryon acoustic oscillation (BAO).

Here, we focus on the early results on weak lensing. Abbott et al. (2016) used the 139 deg^2 Science Verification (SV) data, which is just less than 3% of the full DES data and derived constraint on the S_8 parameter as $S_8 \equiv \sigma_8\sqrt{\Omega_m/0.3} = 0.81 \pm 0.06$. They performed a 3 redshift bins tomographic analysis using the shear two-point correlation functions, marginalising over 7 systematic parameters. In this study, they also concluded that their constraints were in agreement with the Planck and CFHTLenS results. However, note that this was just a preliminary data analysis based on the SV data.

On the other hand, the DES Year 1 results, based on a combined analysis of galaxy clustering and weak lensing with 1321 deg^2 of imaging data, inferred the parameters $S_8 \equiv \sigma_8(\Omega_\mathrm{m}/0.3)^{0.5} = 0.773^{+0.026}_{-0.020}$ and $\Omega_\mathrm{m} = 0.267^{+0.030}_{-0.017}$ for the ΛCDM model. For a wCDM model, Abbott et al. (2018) reports $S_8 = 0.782^{+0.036}_{-0.024}$, $\Omega_\mathrm{m} = 0.284^{+0.033}_{-0.030}$ and $w = -0.82^{+0.21}_{-0.20}$, all values quoted at 68% credible interval. Recently, Amon et al. (2021) performed an analysis on the DES Year 3 data, spanning 4143 deg^2 with four tomographic redshift bins. Using the ΛCDM model, they constrain $S_8 = 0.759^{+0.025}_{-0.023}$ and for a ΛCDM-optimised analysis, which includes smaller scale information, $S_8 = 0.772^{+0.018}_{-0.017}$. These latest results are lower by 2.3σ and 2.1σ respectively compared to the Planck CMB result.

1.5.4 *Euclid*

Euclid will be a visible to near infra-red space telescope (from 550 nm to 2μm), to be launched in 2022 (Laureijs et al., 2011). The main scientific goals of this instrument is to understand better the expansion of the Universe and determine the source of acceleration of this phenomenon which is referred to as dark energy. *Euclid* is expected to generate insightful information up to redshift $z \gtrsim 2$, which corresponds roughly to the evolution of the Universe during the past 10 billion years. The survey will generate petabytes of data and it is estimated that more than 10 billion sources will be observed by *Euclid* and 1 billion will be used in the context of weak lensing. *Euclid* is a larger survey survey compared to the two previous telescopes described above. It will observe 15 000 deg^2 on the sky. Moreover, a resolution of 0.2″ makes it especially good for weak lensing.

1.5.5 Vera C. Rubin Observatory

The Vera C. Rubin Observatory, previously known as the Legacy Survey of Space and Time (LSST), is an optical and near-infrared telescope. The camera will have six filters, *ugrizY*, from 330 nm to 1.1 μm. It is expected that it will start observing in the year 2021 and will cover 18 000 deg^2 of the sky. In particular, it is also expected to yield around 37 billion sky objects on a yearly basis. Moreover, this ambitious project will require roughly 250 teraflops of computational power and 100 petabytes of storage.

Some of the scientific goals of LSST include those already covered by DES. In particular, we still have to understand better the Physics of the Universe and hence, some of the scientific probes include Type 1a supernovae, BAO and weak gravitational lensing. In addition, other scientific aims include but are not limited to mapping the Milky Way, generating catalogues of objects in our Solar System and detecting transients.

1.5.6 Subaru Hyper Suprime-Cam

The Hyper Suprime-Cam (HSC) is a wide-field imaging telescope in Mauna Kea in Hawaii which will cover 1400deg^2 in five bands, *grizy* (Aihara et al., 2018). One of its fundamental scientific goals is use weak lensing measurements to elucidate the distribution of dark matter in the universe. Tanaka et al. (2018) recently performed an analysis to compute the photometric redshifts and the tomographic redshift distributions, $n(z)$. On the other hand, Hikage et al. (2019) determined the value of $S_8 = 0.780^{+0.030}_{-0.033}$ using weak lensing shear power spectra, with

multipoles $300 \leq \ell \leq 1900$, with HSC data. The photometric redshift range adopted in the analysis was $0.3 \leq z \leq 1.5$, with four tomographic redshift distributions.

1.5.7 Nancy Grace Roman Space Telescope

The Nancy Grace Roman Space Telescope, previously known as the Wide-Field Infrared Survey Telescope (WFIRST), is an infrared space telescope which is expected to be launched in May 2027. It will cover a survey area of 2000 deg^2 (Eifler et al., 2021). The scientific goals of this experiment is to measure different cosmological probes such as weak lensing, galaxy clustering, redshift space distortion and baryon acoustic oscillations. These will enable us to answer questions about the nature of dark energy, which is complementary to the scientific goals of the *Euclid* mission too. Other scientific missions will be to find exoplanet systems to understand more about the potential for life in the universe.

1.6 Type Ia Supernovae

Another cosmological probe to understand the universe is the Type Ia supernovae (SNe Ia). The latter has been instrumental in the discovery of the accelerating expansion of the universe (Riess et al., 1998; Perlmutter et al., 1999). Type Ia supernovae remains a strong candidate to elucidate the nature of dark energy. In brief, Type Ia supernova occur as a result of the thermonuclear explosion of a white dwarf in a binary system, exceeding the Chandrasekhar limit of $1.44\,\mathrm{M}_{\odot}$. Hence, they can be used as distance indicators after correcting for the systematics in the lightcurve shape and colour.

A recent analysis was done by Abbott et al. (2019) using the DES data, where 327 SNe Ia were employed. For a flat ΛCDM model, $\Omega_{\mathrm{m}} = 0.331 \pm 0.038$ and for a flat wCDM model combined with the CMB, $w = -0.978 \pm 0.059$ and $\Omega_{\mathrm{m}} = 0.321 \pm 0.018$. In Chapter 5, we will use the SNe Ia JLA data (Betoule et al., 2014) to test the idea behind compression and emulation, which are central to the different weak lensing analysis performed in this thesis.

1.7 Summary

In this chapter, we have provided an in-depth review on weak lensing cosmology and these concepts will be very important in the forthcoming chapters. In particular, we have discussed briefly the basics of cosmology, which can be found in most cosmology textbooks. This is followed by a detailed description of weak lensing essentials before elaborating on the weak

lensing statistics, for example, power spectrum and correlation functions. A weak lensing analysis is also susceptible to multiple observational and scientific challenges and we highlight them in this chapter. Moreover, we briefly cover Type Ia supernovae, which is another probe to elucidate the nature of dark energy. Finally, we discuss some of the past, ongoing and future weak lensing surveys.

Chapter 2

BAYESIAN STATISTICS

> Remember that using Bayes' Theorem does not make you a Bayesian. Quantifying uncertainty with probability makes you a Bayesian.
>
> *Michael Betancourt*

In this chapter, we will provide an overview of Bayesian Statistics, which is becoming increasingly relevant in many ML applications. One of the main motivations for taking a probabilistic modelling approach is for uncertainty quantification. Probabilistic ML also provides an elegant framework for designing machines, capable of learning from data through experience (Ghahramani, 2015).

With the rise of computational power, adopting a Bayesian approach is now central in scientific data analysis, ML and automation. It leads to a broad spectrum of new topics such as data compression, Bayesian optimisation, probabilistic programming, approximate inference and so forth. The development of techniques such as automatic differentiation (Baydin et al., 2017) has further bolster the field of probabilistic ML.

We will also often encounter problems where a full analytical treatment is not possible at all. In these situations, techniques such as Monte Carlo (MC) methods and Variational Inference (VI), where both fall under the category of approximate inference, are often adopted. We will cover some of these approximation techniques in this chapter. In particular, in §2.1 we discuss the concepts behind probability and in §2.2, we cover briefly the normal distribution which is commonly used in Cosmology. In §2.3, we explicitly look into Bayes's theorem which is now ubiquitous for almost any cosmological data analysis. In §2.4, we discuss different types of priors, to motivate the application of certain prior in a specific problem and and in §2.5, we briefly cover directed acyclic graphs (DAG), which is a common technique to show relationships among random variables. We touch briefly about Bayesian model comparison in §2.6

and in §2.7, we highlight a few sampling algorithms commonly used to infer the distribution of parameters which are of key interest to us.

2.1 Probability

We first start with the definition of probability, which in itself is a topic of heated debate. For example, De Finetti made a provocative statement, *probability does not exist*, strictly referring to probability in the objective sense. This brings us to two types of probability, often referred to *objective probability* and *subjective probability* (Nau, 2001). In other aspects, they may also be referred to as the *classical* versus the *modern* views of statistics. In particular, the most common comparison is between the *frequentist* and *Bayesian* statistics, where adopting a Bayesian approach to a particular problem implies a modern and subjective probabilistic methodology. We briefly cover these two types of probability in this section and we refer the reader to the review by Trotta (2008) for an in-depth overview on this topic.

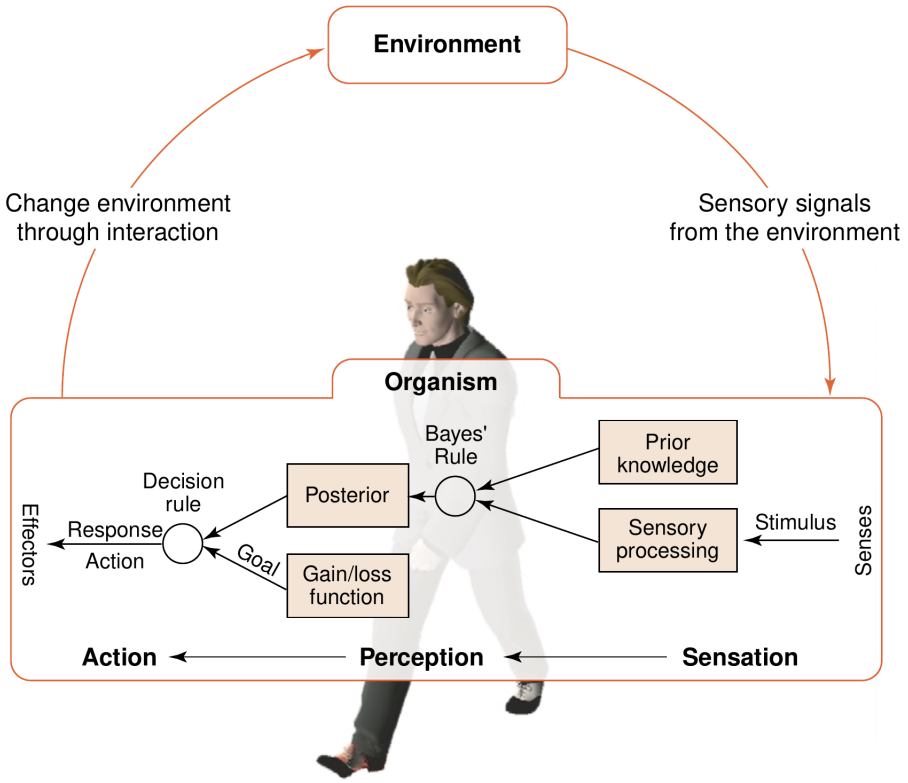


Figure 2.1 – The human being is considered as a Bayesian thinker. Generally, we obtain information from the environment and based on the prior information, we update the information via the sensory processing unit (likelihood) to the posterior. The final step is to take a decision and act upon the environment. An analogous comparison can be made with the Bayesian Optimisation algorithm (see Chapter 3 for a brief overview on this topic).

The *frequentist* (classical/objective) view of probability relies on the notion of randomness. Moreover, this probability is based on an existing set of recorded information or a long history

of collected information. This clearly raises some profound question such as, 'what if we only have the resources to perform a single experiment?'. This is very common in the Cosmology community where, perhaps in a single decade, we will have only one CMB experiment.

Frequentist Probability

Probability is defined as the ratio of the number of successes to the number of trials, in the limit of an infinite number of trials, that is,

$$p = \lim_{n \to \infty} \frac{s}{n}$$

where n is the number of trials and s is the number of successes.

As argued by Trotta (2008), this definition of probability is circular, that is, each experiment performed assumes a fixed probability of success while it is the very same quantity that we want to estimate. Moreover, this treatment of probability cannot deal with unrepeatable experiments since the definition itself is based on the number of trials. In the same spirit, even if we were able to perform repeatable experiments, this raises the question of number of trials that one should perform to obtain a sound estimate of p. In the following discussion, we will see that this is easily dealt with the Bayesian framework through the concept of *marginalisation*.

On the other hand, the *Bayesian* (modern/subjective) approach deals with probability distribution to quantify the uncertainty of an event happening. In particular, we can now allude to the possibility of learning the probability distribution of a parameter. This concept is very important in inverse problems, where we want to learn the distribution of a set of parameters which explain the data we have.

Bayesian Probability

Probability in the Bayesian context is interpreted as a measure of the degree of belief about a hypothesis. Unlike the frequentist approach, we now have a *prior*, $p(\theta)$, which encodes our knowledge about a specific parameter and this prior gets updated to the posterior distribution, $p(\theta|\boldsymbol{x})$ via a likelihood function, $p(\boldsymbol{x}|\theta)$. $\boldsymbol{x}$ refers to a set of (observed) data points. In addition, we instead refer to credible intervals, rather than confidence intervals. The former gives a more robust interpretation of probability. For example, q % credible interval implies that there is q % probability of finding that specific parameter, θ, within that range.

The Bayesian approach is a neat and elegant way to interpret probability. Most importantly, it has the advantage of incorporating a prior on the parameters and this is crucial in almost any inference engine in Cosmology. For example, $0 < \Omega_{\rm cdm} < 1$, and this information can trivially be included in a sampling scheme. Moreover, one can easily deal with nuisance parameters in a Bayesian framework by marginalising over them. See next section for further details on marginalisation. It is customary to ignore or perhaps fix nuisance parameters to certain values, but this can erroneously lead to model misspecification, that is, this procedure can result in overestimating or underestimating the uncertainty quantification of the parameters of interest. In Note 2.2, we provide the basic ingredients when working with probability.

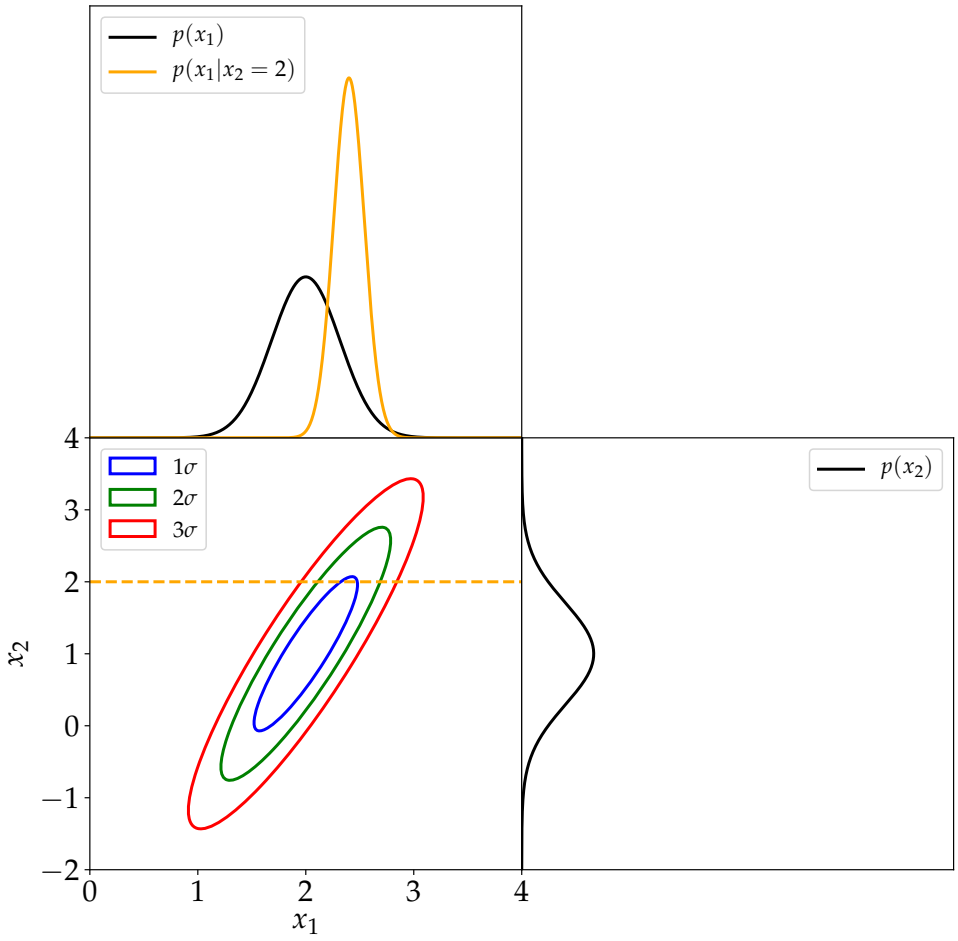


Figure 2.2 – Figure showing a 2D Gaussian distribution in the lower left panel, with the blue, green and red contours showing the 1σ, 2σ and 3σ countours respectively. The marginal distributions, $p(x_1)$ and $p(x_2)$ are shown in the upper left and lower right panel respectively. Moreover, an example of the conditional distribution, $p(x_1|x_2)$ at $x_2 = 2$ is shown in orange in the upper left panel.

2.2 Normal Distribution

Before delving into the detail of inference mechanisms, it is important to discuss the properties of the normal distribution, which arises in many applications in Cosmology. It will also be used

extensively in Chapter 3 in the context of Gaussian Processes. A one dimensional Gaussian distribution, also known as the normal distribution, can be written as

$$p(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right] \tag{2.2.1}$$

and a multivariate Gaussian distribution is

$$p(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{2.2.2}$$

where $\boldsymbol{\mu}$ is a d dimensional vector and $\boldsymbol{\Sigma}$ is a $d \times d$ covariance matrix. An important property of a multivariate Gaussian distribution is that both the conditional distribution and the marginal distribution are Gaussian distributions. To understand this better, let us consider a 2-dimensional multivariate normal distribution with mean and covariance

$$\boldsymbol{\mu} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 0.1 & 0.2 \\ 0.2 & 0.5 \end{pmatrix}$$

Note 2.1: Conditional and Marginal distributions of a Gaussian

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix}$$

with mean and covariance given by

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} \qquad \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}$$

The conditional distribution, $p(\boldsymbol{x}_1|\boldsymbol{x}_2)$ is another Gaussian distribution with mean and covariance

$$\begin{aligned} \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \end{aligned} \tag{2.2.3}$$

The marginal distribution of $\boldsymbol{x}_1$ is simply another Gaussian distribution with mean and covariance, $\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_{11}$ respectively.

Following Note 2.1, the marginal distributions, $p(x_1)$ and $p(x_2)$ are 1D Gaussian distributions, that is, $x_1 \sim \mathcal{N}(2, 0.1)$ and $x_2 \sim \mathcal{N}(1, 0.5)$. Note that the formula also apply to the multivariate case. Moreover, the conditional distribution $p(x_1|x_2)$ at $x_2 = 2$ is a Gaussian distribution, $x_1|x_2 = 2 \sim \mathcal{N}(2.4, 0.02)$. Note that the notation $x \sim \mathcal{N}(\mu, \sigma^2)$ implies a normal distribution with mean μ and variance σ^2. In Figure 2.2, we show the 2D multivariate normal, conditional and marginal distributions for this particular case.

2.3 Bayes' Theorem

Bayes' theorem is fundamental to deal with inverse problems in Cosmology. Strictly, the Bayesian viewpoint asserts that if two persons are given the same data and they make exactly similar assumptions, they will draw similar conclusions. Making assumptions in a Bayesian analysis should not be deemed as being too subjective or a weakness of the procedure. Instead, advocates of the Bayesian methodology suggest that *one cannot do inference without making assumptions* (MacKay, 2003). In general, as shown in Figure 2.1, we, humans are considered as Bayesian thinkers, since we constantly update the information we receive from the environment to turn it into meaningful action onto the environment.

In the following, we will stick to the following notations: $\boldsymbol{\theta}$ refers to a vector of parameters which are of key interest to us, for example, it might refer to the cosmological parameters. $\boldsymbol{\beta}$ is a vector of nuisance (systematic) parameters. The variable $\boldsymbol{z}$ refers to the set of latent (hidden/unobserved) variables $(\boldsymbol{\theta}, \boldsymbol{\beta})$. $\boldsymbol{x}$ refers to the observed data vector. Note that, we are assuming a single model, often denoted by $\mathcal{M}$. This is implicitly assumed throughout all equations below.

In the very first instance, we have to define a distribution of the data, that is, normally one defines the conditional distribution, $p(\boldsymbol{x}|\boldsymbol{z})$ which is referred to as the *likelihood function*. Importantly, it should be called the *likelihood of the parameters*, $\boldsymbol{z}$ and it is <u>not</u> a (normalised) probability distribution. Once we have the observed data $\boldsymbol{x}$ and assuming a model $\mathcal{M}$, which is a function of $\boldsymbol{z}$, the obvious question to ask is, 'what are the likely values of $\boldsymbol{z}$?' This is very common in many applications, where we want to learn $\boldsymbol{z}$, hence an inverse problem. We will also assume some prior distributions, $p(\boldsymbol{z})$ for the latent variables. Note that in practice, the choice of the likelihood depends on the assumption of the data and the choice of the prior may depend on the results from past experiments.

Following the rules of probability, in particular, the *product rule*, along with the symmetric property, $p(x, y) = p(y, x)$, we have the following relationship between the two conditional

probabilities, $p(\boldsymbol{x}|\boldsymbol{z})$ and $p(\boldsymbol{z}|\boldsymbol{x})$:

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z})\, p(\boldsymbol{z})}{p(\boldsymbol{x})} \tag{2.3.1}$$

which is in fact *Bayes' theorem*. $p(\boldsymbol{z}|\boldsymbol{x})$ is the *posterior distribution* or *posterior belief* of $\boldsymbol{z}$ and the denominator, $p(\boldsymbol{x})$ is referred to as the *evidence* or *marginal likelihood*. The important aspect of Bayes' theorem is that it separates inference from modelling. At the heart of an exact Bayesian analysis lies *integration*.

Note 2.2: Rules of Probability

Consider two continuous random variables X and Y. We can write the following rules (and properties) of probability (MacKay, 2003).

Marginal Probability

$$p(x) = \int p(x, y)\, \mathrm{d}y$$

Sum rule

Involves re-writing the marginal probability definition in another way.

$$\begin{aligned} p(x) &= \int p(x, y)\, \mathrm{d}y \\ &= \int p(x|y)\, p(y)\, \mathrm{d}y \end{aligned}$$

Product rule

$$p(x, y) = p(x|y)p(x)$$

This is also known as the chain rule.

Independence

The two random variables X and Y are said to be independent (also written as $X \perp Y$) if

$$p(x, y) = p(x)\, p(y)$$

Marginal Likelihood

Since we are dealing with continuous variables $\boldsymbol{z}$, the marginal likelihood is

$$p(\boldsymbol{x}) = \int \mathrm{d}\boldsymbol{z}\, p(\boldsymbol{x}, \boldsymbol{z}) = \int \mathrm{d}\boldsymbol{z}\, p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}). \tag{2.3.2}$$

Marginalisation

The latent variables consist of both the parameters which are of interest to us, $\boldsymbol{\theta}$ and the nuisance parameters, $\boldsymbol{\beta}$. If we want to learn the distribution of the $\boldsymbol{\theta}$, we have to integrate out (marginalise over) the nuisance parameters, that is,

$$p(\boldsymbol{\theta}|\boldsymbol{x}) \propto \int \mathrm{d}\boldsymbol{\beta}\, p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\beta}) p(\boldsymbol{\theta}, \boldsymbol{\beta}) \tag{2.3.3}$$

and we have to include a normalisation factor, so the posterior distribution of $\boldsymbol{\theta}$ is properly normalised.

Posterior Predictive Distribution

Suppose we already have the full posterior distributions of the latent variables $\boldsymbol{z}$. We might also be interested in making predictions, $\boldsymbol{x}_*$ of the data at test points. Hence, we need to evaluate the predictive distribution

$$\begin{aligned} p(\boldsymbol{x}_*|\boldsymbol{x}) &= \int \mathrm{d}\boldsymbol{z}\, p(\boldsymbol{x}_*, \boldsymbol{z}|\boldsymbol{x}) \\ &= \int \mathrm{d}\boldsymbol{z}\, p(\boldsymbol{x}_*|\boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{x}) \end{aligned} \tag{2.3.4}$$

where the first term on the right is simply the likelihood of the parameters $\boldsymbol{z}$ in light of the new data and the second term is simply the posterior distribution of $\boldsymbol{z}$.

Summary Statistics

The latent variables $\boldsymbol{z}$ are typically high-dimensional vectors and we cannot visualise high dimensional posterior distributions. We are rather interested in the statistics of the posterior, for example, the expected mean, $\boldsymbol{\mu}$ and variance, $\boldsymbol{\Sigma}$ respectively are:

$$\boldsymbol{\mu} = \int \mathrm{d}\boldsymbol{z}\, \boldsymbol{z} p(\boldsymbol{z}|\boldsymbol{x}) \tag{2.3.5}$$

$$\boldsymbol{\Sigma} = \int \mathrm{d}\boldsymbol{z}\, (\boldsymbol{z} - \boldsymbol{\mu})(\boldsymbol{z} - \boldsymbol{\mu})^{\mathrm{T}} p(\boldsymbol{z}|\boldsymbol{x}) \tag{2.3.6}$$

All the important quantities which we want to evaluate involve, in some way, the integral of a function $f(\boldsymbol{z})$ multiplied by some probability distribution $p(\boldsymbol{z}|\cdot)$, where the $\cdot$ represents a particular set of variable depending on the operation being performed. However, it is very unlikely that we will be able to perform these integrations analytically for various reasons. Doing high-dimensional integration is very challenging and in most cases, we are dealing with non-linear models. Moreover, inference also depends on the choice of the prior. Even if we

had a linear model, if the functional form of the prior is very different, for example, in the case of non-conjugate priors, an analytical approach to deriving the posterior distribution is not possible. See next section for a discussion on priors.

Case Study: Coin Toss

We are given a coin (and we do not know if it is a fair or biased coin) and we are asked to find the value of θ, that is, the probability of getting heads on any flip. The coin is tossed 20 times and we obtain 15 heads. What would be our conclusion based on this observation?

Let us first take a *Bayesian* approach to this problem. The quantity we are interested in is $p(\theta|x)$, that is, the posterior distribution of θ given the data. Using Bayes' theorem,

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} \tag{2.3.7}$$

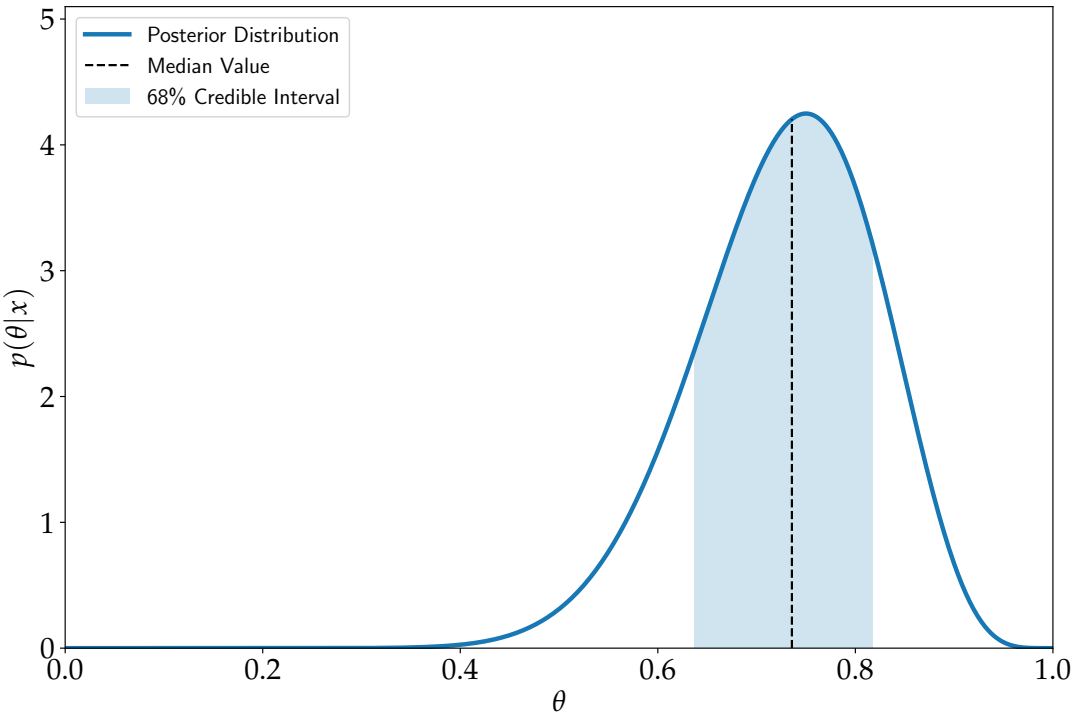


Figure 2.3 – The full posterior distribution of the parameter θ. Unlike the frequentist probability, a Bayesian approach allows us to quantify the uncertainty associated with the estimate of θ in a principled way. If we had coin which is biased towards heads, this information can further be incorporated via, for example, a $\beta-$distribution.

Note that the denominator does not depend on θ, so we can safely ignore it. Hence, $p(\theta|x) \propto p(x|\theta)p(\theta)$. For the likelihood, $p(x|\theta)$, we can assume a Binomial distribution and we will assume a non-informative prior on θ. Here, we assume a uniform prior, that is, $p(\theta) = \mathcal{U}[0,1]$. It is also safe to assume that the full probability density lies within this interval. The likelihood function is

$$p(x|\theta) = \begin{pmatrix} n \\ x \end{pmatrix} \theta^x (1-\theta)^{n-x}$$

where $n = 20$ and $r = 15$. For this particular case, $p(\theta|x) \propto p(x|\theta)$ since we are assuming a uniform prior on θ. In Figure 2.3, we show the normalised posterior distribution of θ and the median value of θ is 0.74 and at 68% credible interval, $\theta = 0.74^{+0.08}_{-0.10}$.

In the *frequentist* approach, we will typically assume $\theta = 0.75$, although this is not quite true, and in this case, an error estimate is can be found by approximation. For large n, a binomial distribution can be approximated as a normal distribution with mean θ/n and variance $\theta(1-\theta)/n$. In short, $\theta = 0.75 \pm 0.10$, where the estimator for the confidence interval is $z\sqrt{\theta(1-\theta)/n}$ and for a $1\sigma-$confidence interval, $z = 1$. Note that, this is an ad hoc procedure and the approximation of a Binomial distribution is valid when n is very large. Moreover, as discussed previously, the frequentist approach does not allow one to encode prior information and it does not yield a full probability distribution for the parameter θ.

Case Study: COVID Test

In this case study, we will calculate the probability of someone being infected by COVID-19 given that the test is positive. In general, medical tests are accurate enough that the tests can be deemed to be above 90% correct. In Figure 2.4, we show all the possibilities with the associated probabilities, (c_1, p_1, p_2). C denotes that someone has the virus (probability c in Figure 2.4) and P denotes that the test is positive. We introduce two terms, namely the *sensitivity*, which is the probability of someone being tested positive given they have the virus, that is, $p(\mathrm{P}|\mathrm{C})$ (p_1 in Figure 2.4). On the other hand, *specificity* is the probability some being tested negative given she does not have the virus, that is, $p(\bar{\mathrm{P}}|\bar{\mathrm{C}})$ (p_2 in Figure 2.4).

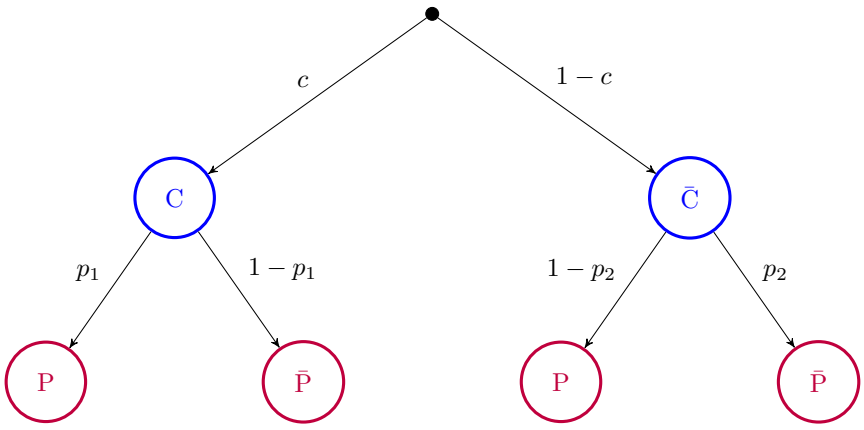


Figure 2.4 – Probability tree diagram for a patient being tested positive or negative if she is infected by the COVID virus. It also shows all the possibilities for the test results. We can expect, false positives (FP), that is, a patient who does not have the virus, yet tested positive.

Our goal is to find, $p(\mathrm{C}|\mathrm{P})$. Using the *product rule*, we have

$$p(\mathrm{C}|\mathrm{P}) = \frac{p(\mathrm{C},\mathrm{P})}{p(\mathrm{P})} \tag{2.3.8}$$

Using Bayes' theorem, this equation can further be written as

$$p(\mathrm{C}|\mathrm{P}) = \frac{p(\mathrm{P}|\mathrm{C})p(\mathrm{C})}{p(\mathrm{P}|\mathrm{C})p(\mathrm{C}) + p(\mathrm{P}|\bar{\mathrm{C}})p(\bar{\mathrm{C}})} \tag{2.3.9}$$

and in terms of the sensitivity, p_1 and specificity, p_2 and noting $p(\mathrm{C}) = c$,

$$p(\mathrm{C}|\mathrm{P}) = \frac{p_1 c}{p_1 c + (1-p_2)(1-c)}. \tag{2.3.10}$$

Plugging in some numbers, for example, $p_1 = 0.95$, $p_2 = 0.90$ and $c = 0.01$, $p(\mathrm{C}|\mathrm{P}) = 0.088$. This is quite unintuitive. Conditional probabilities are generally counter-intuitive. If a test is positive, this would immediately lead us to the conclusion that the person is infected by the virus. However, the probability of this happening is quite small, as per our example. Hence, it is a good idea to perhaps have a second test carried out to draw an accurate conclusion.

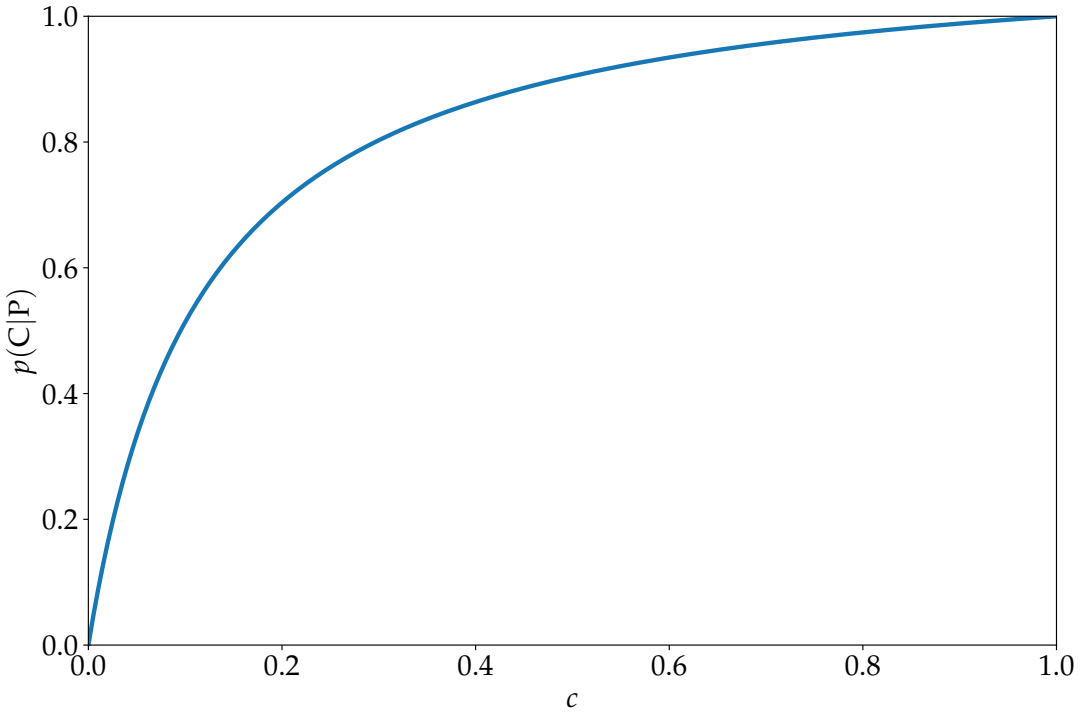


Figure 2.5 – The probability of having the virus given the test outcome is positive as a function of c, the probability of someone having the virus. As more and more people get infected, it makes sense that $p(\mathrm{C}|\mathrm{P})$ should increase.

In Figure 2.5, we fix the sensitivity and the specificity at the values used in the previous example, but we now vary c, that is, $p(\mathrm{C})$. It is expected that $p(\mathrm{C}|\mathrm{P})$ should increase as c in-

creases. Intuitively, if the whole population is infected by COVID, then $p(\mathrm{C}|\mathrm{P}) \to 1$, which is effectively shown in Figure 2.5.

2.4 Priors

In this section, we cover briefly the different types of priors which are used in this thesis and the motivation for using them. The cornerstone of a Bayesian analysis is not only the prior but also the notion of averaging over many different possibilities. In the limit of sufficient data, the likelihood function is the most dominant term, relative to the prior and hence similar posterior distributions are obtained irrespective of the choice of the prior. In most common cases, there are generally two schools of thoughts when it comes to the choice of priors: objective versus subjective priors, which we discuss next.

2.4.1 Objective Priors

In the absence of prior knowledge about the parameters of interest, the choice is to adopt non-informative priors that encode ignorance and share frequentist properties as well.

An example of a non-informative prior is the **reference prior**. The idea is to measure the information gain from the data by a divergence measure between the prior and the posterior distributions. A significant information gain corresponds to a large divergence measure between the prior and the posterior. Under a given model, the information gain can be written as

$$I(p(\boldsymbol{\theta})) = \int \mathrm{D}_{\mathrm{KL}} \left[p(\boldsymbol{\theta} \,\|\, p(\boldsymbol{\theta}|\boldsymbol{x})) \right] \, p(\boldsymbol{x}|\boldsymbol{\theta}) \, p(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \tag{2.4.1}$$

where in this case, the divergence measure is the Kullback-Leibler divergence, D_{KL} and

$$\mathrm{D}_{\mathrm{KL}} \left[p(\boldsymbol{\theta} \,\|\, p(\boldsymbol{\theta}|\boldsymbol{x})) \right] = \int p(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{x})} \, \mathrm{d}\boldsymbol{\theta}.$$

A reference prior, $p_r(\boldsymbol{\theta})$ then corresponds to finding the prior that maximises the information gain, that is,

$$p_r(\boldsymbol{\theta}) = \arg\max_{p(\boldsymbol{\theta})} I(p(\boldsymbol{\theta})) \tag{2.4.2}$$

However, it is hard to compute the reference prior since it involves high-dimensional integration schemes. In the same spirit, another example of an objective prior is the **Jeffreys' prior**. It is motivated by the invariance principle, that is, re-parametrisation of the prior does not matter:

$$p(\phi) = p(\theta) \left| \frac{\mathrm{d}\theta}{\mathrm{d}\phi} \right|. \tag{2.4.3}$$

Jeffreys (1946) proposed the following form of prior

$$p_J(\boldsymbol{\theta}) = c \left| I^{\mathrm{F}}(\boldsymbol{\theta}) \right|^{1/2} \tag{2.4.4}$$

where $|\cdot|$ denotes the determinant, c is a constant factor and I^{F} refers to the Fisher information and is defined as

$$\begin{aligned} I^{\mathrm{F}}(\boldsymbol{\theta}) &:= -\mathbb{E}_{[\boldsymbol{x}|\boldsymbol{\theta}]} \frac{\partial^2}{\partial \boldsymbol{\theta}^2} \log p(\boldsymbol{x}|\boldsymbol{\theta}) \\ &= -\int p(\boldsymbol{x}|\boldsymbol{\theta}) \frac{\partial^2}{\partial \boldsymbol{\theta}^2} \log p(\boldsymbol{x}|\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{x} \end{aligned} \tag{2.4.5}$$

It is worth noting that the Jeffreys' prior is invariant under re-parametrisation. While objective priors are certainly commonly adopted in a Bayesian analysis, they are not without criticisms. For example, for an unbounded uniform prior, $p(\theta) =$ constant or even the Jeffrey prior, the problem lies in finding an appropriate normalisation factor. These are also often referred to as **improper priors** and they are used as long as the posterior distribution, $p(\boldsymbol{\theta}|\boldsymbol{x}) \propto p(\boldsymbol{x}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})$ is proper.

2.4.2 Subjective Priors

Another school of thought argues that a Bayesian analysis should capture our beliefs as much as possible. This subjective view of priors stems from the fact that *knowledge is objective while belief is subjective*. In practice, there is no universal rule for choosing a prior. Generally, the subjective belief of $\boldsymbol{\theta}$ will differ from one expert to another and this is totally acceptable as long as we can convince others that our belief is meaningful based on the current (and past) knowledge.

However, our beliefs of the prior are governed by problem definition. Moreover, it is only possible to do *an* analysis and not *the* analysis. As discussed previously, we cannot perform inference without making assumptions about the data. If the answer to a problem changes as a result of a change in, for example, the prior, it is crucial to acknowledge this change.

2.4.3 Hierarchical Priors

Hierarchical priors are viewed as priors on priors and are also referred to as *hyper-priors*. In essence, these hyper-priors are only applied to a handful number of parameters. For example, if we are specifying a Gaussian prior on θ, we can have two hyper-priors on the mean, μ and standard deviation σ of the distribution. Hence, we can write

$$p(\theta) = \int p(\theta|\alpha)\, p(\alpha)\, \mathrm{d}\alpha \tag{2.4.6}$$

and if α relies on another parameter, β, then

$$p(\theta) = \int p(\theta|\alpha)\, p(\alpha|\beta)\, p(\beta)\, \mathrm{d}\alpha \mathrm{d}\beta \tag{2.4.7}$$

resulting in a hierarchy of distributions. An easy way to visualise this is via directed acyclic graphs which we discuss in §2.5.

2.4.4 Empirical Priors

Another type of prior is the empirical prior and refers to the case where one learns some of the parameters of the prior using the data. This procedure is also referred to as *Empirical Bayes*. Let us consider a hierarchical model, where the prior distribution of θ relies on some parameter α. Since we are using the data to estimate the hyper-parameter, we can write

$$p(\boldsymbol{x}|\alpha) = \int p(\boldsymbol{x}|\theta)\, p(\theta|\alpha)\, \mathrm{d}\theta \tag{2.4.8}$$

and the hyper-parameter α can be estimated from the data by maximising $p(\boldsymbol{x}|\alpha)$, that is,

$$\hat{\alpha} = \arg\max_{\alpha} p(\boldsymbol{x}|\alpha). \tag{2.4.9}$$

While on one hand this technique attempts to overcome misspecification of the prior by finding a suitable set of hyper-parameters, it can also lead to over-fitting. Moreover, in this procedure, the data is being used twice, that is, first for setting α and second, for finding the posterior distribution of θ. Using the data more than once in a Bayesian analysis is not deemed as an elegant approach of inference.

2.4.5 Conjugate Priors

The idea behind adopting a conjugate prior is to obtain a posterior distribution which is in the same family as the prior. For example, for a Gaussian Linear Model (GLM), $\boldsymbol{y} = \Phi\boldsymbol{\theta}$, if we use a Gaussian prior for the parameters, $\boldsymbol{\theta}$, the posterior distribution for $\boldsymbol{\theta}$ will also be Gaussian. Depending on the problem, the motivation for using a conjugate prior is to obtain the posterior update (see Equation 2.3.1) in a closed form. See §2.7 for a discussion on approximating the posterior via sampling techniques in the case where prior-to-posterior update cannot be obtained in a closed form.

In many Bayesian analyses, an exponential family of distributions are used for the likelihood and the prior, the motivation being that that these exponentials are easily integrated and hence, computationally more convenient. $p(\boldsymbol{x}|\boldsymbol{\theta})$ is in the exponential family if it can be written as

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = f(\boldsymbol{x})g(\boldsymbol{\theta})\exp[\Phi^{\mathrm{T}}(\boldsymbol{\theta})s(\boldsymbol{x})] \tag{2.4.10}$$

where $\Phi(\boldsymbol{\theta})$ is a vector, which is a function of the parameters, $s(\boldsymbol{x})$ is a vector of summary statistics, often referred to as the data vector in Cosmology, f and g are positive functions of $\boldsymbol{x}$ and $\boldsymbol{\theta}$ respectively. To understand this better, let us consider a Gaussian likelihood with a data vector $\boldsymbol{x}$, forward model, $\boldsymbol{\phi}$ (which is a function of the parameters $\boldsymbol{\theta}$) and a covariance matrix, Σ. The likelihood can be written as

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{1}{\sqrt{|2\pi\Sigma|}}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\phi})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\phi})\right] \tag{2.4.11}$$

and the exponential term can further be simplified as

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = c\,f(\boldsymbol{x})\,g(\boldsymbol{\theta})\exp\left[-\boldsymbol{\phi}^{\mathrm{T}}\tilde{\boldsymbol{x}}\right]$$

where $f(\boldsymbol{x}) = \exp[-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x}]$, $g(\boldsymbol{\theta}) = \exp[-\frac{1}{2}\boldsymbol{\phi}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{\phi}]$, $\tilde{\boldsymbol{x}} = \boldsymbol{\Sigma}^{-1}\boldsymbol{x}$ and c is just a constant. Note the similarity between the Gaussian likelihood and the standard form for a distribution from an exponential family. A conjugate prior in this case is of the form

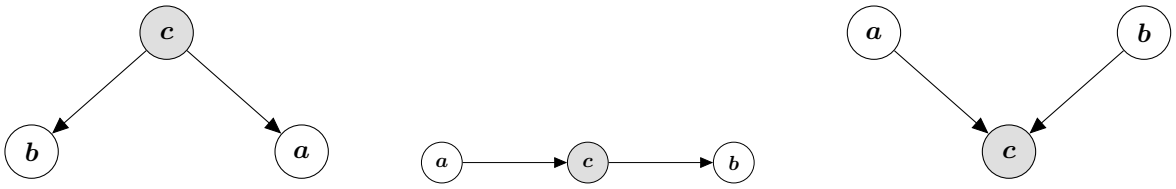
$$p(\boldsymbol{\theta}) = q\,g(\boldsymbol{\theta})\exp[\Phi(\boldsymbol{\theta})] \tag{2.4.12}$$

and q is just a normalisation constant. Despite the nice property of a conjugate prior, from an objective perspective, it might not be an optimal approach since the goal of an objective prior

is such that the posterior distribution contains maximum information from the data. From a subjective approach, a conjugate prior may have more influence on the posterior and this can be a favourable view in a Bayesian analysis.

2.5 Directed Acyclic Graphs

A crucial probabilistic concept when dealing with multiple variables is *conditional independence*. It is best understood via Directed Acyclic Graphs (DAGs), which we will cover briefly in this section. For an in-depth review on this topic, we refer the reader to the textbook by Bishop (2006). Throughout this section, we will use three variables a, b and c to illustrate the different test cases.



(a) Tail-Tail Node **(b)** Head-Tail Node **(c)** Head-Head Node

Figure 2.6 – Example of different DAG graphs for different conditional independence cases. In particular, the left, middle and right panels show the graphs for the tail-tail, head-tail and head-head test cases. Each of these graphs can be used to derive the conditional probability of the parameters of interest when doing parameter inference in Cosmology.

If a does not depend on b, the conditional probability

$$p(a|b,c) = p(a|c) \tag{2.5.1}$$

and the joint probability of a and b can be written as

$$\begin{aligned} p(a,b|c) &= p(a|b,c)\, p(b|c) \\ &= p(a|c)p(b|c) \end{aligned} \tag{2.5.2}$$

factors into the product of two probability distributions, hence, the two variables a and b are statistically independent given the variable c. This can be written in shorthand format as

$$a \perp\!\!\!\perp b|c \tag{2.5.3}$$

Importantly, once we have a graphical model, the conditional independence can be inferred directly from the graph, without the need for additional analytical manipulation. We will con-

sider a few examples below.

Tail-Tail Node

Let us consider the first example, a tail-tail node as shown in panel (a) in Figure 2.6. The joint distribution, $p(a, b, c)$ can be written as

$$p(a, b, c) = p(a|c)\, p(b|c)\, p(c) \tag{2.5.4}$$

and if we choose to condition on c, the joint distribution, $p(a, b|c)$ can be written as

$$p(a, b, c) = p(a, b|c)\, p(c) \tag{2.5.5}$$

and hence using Equation 2.5.4, the joint distribution, $p(a, b|c)$ is:

$$p(a, b|c) = p(a|c)\, p(b|c) \tag{2.5.6}$$

which implies, that in a tail-tail scenario, conditioning on c yields conditional independence, that is, $a \perp\!\!\!\perp b|c$.

Head-Tail Node

Next, we consider a head-to-tail scenario as shown in the middle panel of Figure 2.6. The joint distribution, $p(a, b, c)$ can be written as

$$p(a, b, c) = p(a)\, p(c|a)\, p(b|c) \tag{2.5.7}$$

and conditioning on c, we have

$$\begin{aligned} p(a, b|c) &= \frac{p(a, b, c)}{p(c)} \\ &= \frac{p(a)\, p(c|a)\, p(b|c)}{p(c)} \\ &= p(a|c)\, p(b|c) \end{aligned} \tag{2.5.8}$$

where the last line is obtained using Bayes' theorem, that is, $p(a|c) = \dfrac{p(c|a)\, p(a)}{p(c)}$. Hence, in this case, as in the previous case, we obtain the conditional independence property in a head-to-tail scenario, that is, $a \perp\!\!\!\perp b|c$.

Head-Head Node

Finally, we consider a head-head scenario as shown in the right panel of Figure 2.6. In this case, the joint distribution, $p(a, b, c)$ is:

$$p(a, b, c) = p(a)\, p(b)\, p(c|a, b) \tag{2.5.9}$$

and conditioning on c, we can write

$$\begin{aligned} p(a, b|c) &= \frac{p(a, b, c)}{p(c)} \\ &= \frac{p(a)\, p(b)\, p(c|a, b)}{p(c)} \end{aligned} \tag{2.5.10}$$

and this cannot be simplified further as in the product of $p(a|c)\, p(b|c)$. Hence, in this scenario, conditioning on c in a head-head case leads to conditional dependence, that is, $a \not\perp\!\!\!\perp b|c$. These probabilistic techniques have paved their way in several cosmological data analysis problems, for example, in cosmic shear power spectrum inference (Alsing et al., 2016). We will look into two cases below, first a standard map making processing, involving Wiener filter equations and one which includes a novel technique, referred to as messenger field (Elsner & Wandelt, 2013; Jasche & Lavaux, 2015).

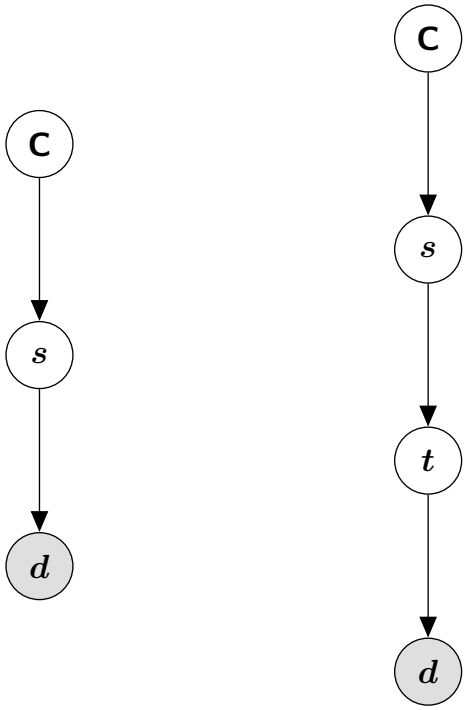


Figure 2.7 – Two different inference mechanisms for inferring the signal, $\boldsymbol{s}$. On the left, we have the standard approach of learning the full posterior distribution of the different quantities, $\mathbf{C}$ and $\boldsymbol{s}$ given the fixed data, $\boldsymbol{d}$. However, this is generally a challenging task and the introduction of the messenger field, $\boldsymbol{t}$, in the right panel, simplifies the task, at the cost of inferring the $\boldsymbol{t}$ field in an iterative scheme.

We will first look into the DAG on the left of Figure 2.7. Let us assume a fixed noise covariance matrix, $\mathbf{N}$. The joint density $p(\boldsymbol{d}, \boldsymbol{s}, \mathbf{C}|\mathbf{N})$ can be written as:

$$p(\boldsymbol{d}, \boldsymbol{s}, \mathbf{C}|\mathbf{N}) = p(\boldsymbol{d}|\boldsymbol{s}, \mathbf{N})\, p(\boldsymbol{s}|\mathbf{C})\, p(\mathbf{C}) \tag{2.5.11}$$

and the distributions for the data and signal are given by:

$$p(\boldsymbol{d}|\boldsymbol{s}, \mathbf{N}) = \frac{1}{\sqrt{|2\pi\mathbf{N}|}} \exp\left[-\frac{1}{2}(\boldsymbol{d}-\boldsymbol{s})^{\mathrm{T}}\mathbf{N}^{-1}(\boldsymbol{d}-\boldsymbol{s})\right] \tag{2.5.12}$$

and

$$p(\boldsymbol{s}|\mathbf{C}) = \frac{1}{\sqrt{|2\pi\mathbf{C}|}} \exp\left[-\frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\mathbf{C}^{-1}\boldsymbol{s}\right]. \tag{2.5.13}$$

A Jeffreys prior is normally assumed for the covariance $\mathbf{C}$. Note that $\mathbf{C}$ is (block-) diagonal in harmonic space but is a very dense matrix is real space. Hence, in this map-power spectrum inference process, it is customary to switch between two bases (harmonic/Fourier and real space) to infer the signal. We are interested in finding the posterior distribution of the signal, $\boldsymbol{s}$, conditioned on all other variables, that is, using Bayes' theorem, we can write

$$p(\boldsymbol{s}|\boldsymbol{d}, \mathbf{C}, \mathbf{N}) = \frac{p(\boldsymbol{d}|\boldsymbol{s}, \mathbf{N})\, p(\boldsymbol{s}|\mathbf{C})}{p(\boldsymbol{d}|\mathbf{N})} \tag{2.5.14}$$

and this can further be simplified as

$$p(\boldsymbol{s}|\boldsymbol{d}, \mathbf{C}, \mathbf{N}) \propto p(\boldsymbol{d}|\boldsymbol{s}, \mathbf{N})\, p(\boldsymbol{s}|\mathbf{C}) \tag{2.5.15}$$

since the denominator is just a constant term, independent of the signal, $\boldsymbol{s}$. The maximum a posteriori probability (MAP) estimate of the above is a multivariate normal distribution:

$$p(\boldsymbol{s}|\boldsymbol{d}, \mathbf{C}, \mathbf{N}) = \frac{1}{\sqrt{|2\pi\mathbf{S}|}} \exp\left[-\frac{1}{2}(\boldsymbol{s}-\hat{\boldsymbol{s}})^{\mathrm{T}}\mathbf{S}^{-1}(\boldsymbol{s}-\hat{\boldsymbol{s}})\right] \tag{2.5.16}$$

with mean and covariance given by: $\hat{\boldsymbol{s}} = (\mathbf{C}^{-1} + \mathbf{N}^{-1})^{-1}\mathbf{N}^{-1}\boldsymbol{d}$ and $\mathbf{S} = (\mathbf{C}^{-1} + \mathbf{N}^{-1})^{-1}$. The mean signal obtained from this approach is referred to as the Wiener filter of the data (Alsing et al., 2016).

While the above formalism nicely describes the inference technique for estimating the signal, a major challenge is that the noise matrix is generally not sparse in the real space, because the pixel noise is not homogeneous and isotropic. Had it been the case, the noise matrix would simply be a diagonal matrix. Hence, the noise matrix can easily be inverted in pixel space and the signal covariance matrix, $\mathbf{C}$ can be inverted in harmonic space (recall it is sparse in

this basis). To alleviate this issue, an auxiliary field, referred to as the messenger field, $\boldsymbol{t}$ is introduced, as shown in the right panel of Figure 2.7.

The main idea is to split the noise covariance matrix, $\mathbf{N}$ into two parts, an isotropic part: $\mathbf{M} = \tau\mathbf{I}$, where $\tau \leq \min[\text{diag}(\mathbf{N})]$ and an anisotropic part: $\bar{\mathbf{N}} = \mathbf{N} - \mathbf{M}$. The joint distribution can be written as:

$$p(\boldsymbol{s}, \boldsymbol{d}, \boldsymbol{t}, \mathbf{C}) = p(\boldsymbol{d}|\boldsymbol{t}, \bar{\mathbf{N}})\, p(\boldsymbol{t}|\boldsymbol{s}, \mathbf{M})\, p(\boldsymbol{s}|\mathbf{C})\, p(\mathbf{C}) \tag{2.5.17}$$

The different probability distributions on the right of the above equation are given by:

$$p(\boldsymbol{d}|\boldsymbol{t}, \bar{\mathbf{N}}) = \frac{1}{\sqrt{|2\pi\bar{\mathbf{N}}|}} \exp\left[-\frac{1}{2}(\boldsymbol{d}-\boldsymbol{t})^{\mathrm{T}}\bar{\mathbf{N}}^{-1}(\boldsymbol{d}-\boldsymbol{t})\right], \tag{2.5.18}$$

$$p(\boldsymbol{t}|\boldsymbol{s}, \mathbf{M}) = \frac{1}{\sqrt{|2\pi\mathbf{M}|}} \exp\left[-\frac{1}{2}(\boldsymbol{t}-\boldsymbol{s})^{\mathrm{T}}\mathbf{M}^{-1}(\boldsymbol{t}-\boldsymbol{s})\right] \tag{2.5.19}$$

and

$$p(\boldsymbol{s}|\mathbf{C}) = \frac{1}{\sqrt{|2\pi\mathbf{C}|}} \exp\left[-\frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\mathbf{C}^{-1}\boldsymbol{s}\right]. \tag{2.5.20}$$

We are interested in finding the posterior of $\boldsymbol{t}$ and the signal $\boldsymbol{s}$. In the first case, we can write the posterior distribution of $\boldsymbol{t}$ using Bayes' theorem or repeated application of the product rule, that is,

$$p(\boldsymbol{t}|\boldsymbol{d}, \boldsymbol{s}, \mathbf{C}, \mathbf{M}, \bar{\mathbf{N}}) \propto p(\boldsymbol{d}|\boldsymbol{t}, \bar{\mathbf{N}})\, p(\boldsymbol{t}|\boldsymbol{s}, \mathbf{M}) \tag{2.5.21}$$

and the posterior distribution of $\boldsymbol{t}$ is a Gaussian distribution with mean $\hat{\boldsymbol{t}}$ and covariance $\mathbf{T}$,

$$p(\boldsymbol{t}|\boldsymbol{d}, \boldsymbol{s}, \mathbf{C}, \mathbf{M}, \bar{\mathbf{N}}) = \frac{1}{\sqrt{|2\pi\mathbf{T}|}} \exp\left[-\frac{1}{2}(\boldsymbol{t}-\hat{\boldsymbol{t}})^{\mathrm{T}}\mathbf{T}^{-1}(\boldsymbol{t}-\hat{\boldsymbol{t}})\right] \tag{2.5.22}$$

where $\hat{\boldsymbol{t}} = (\bar{\mathbf{N}}^{-1} + \mathbf{M}^{-1})^{-1}(\bar{\mathbf{N}}^{-1}\boldsymbol{d} + \mathbf{M}^{-1}\boldsymbol{s})$ and $\mathbf{T} = (\bar{\mathbf{N}}^{-1} + \mathbf{M}^{-1})^{-1}$. In a similar way, the posterior distribution of the signal $\boldsymbol{s}$ is:

$$p(\boldsymbol{s}|\boldsymbol{d}, \boldsymbol{t}, \mathbf{C}, \mathbf{M}) \propto p(\boldsymbol{t}|\boldsymbol{s}, \mathbf{M})\, p(\boldsymbol{s}|\mathbf{C}) \tag{2.5.23}$$

and the resulting distribution is another multivariate normal distribution with mean $\hat{\mathbf{s}}$ and $\mathbf{S}$,

$$p(\boldsymbol{s}|\boldsymbol{d},\boldsymbol{t},\mathbf{C},\mathbf{M}) = \frac{1}{\sqrt{|2\pi\mathbf{S}|}}\exp\left[-\frac{1}{2}(\boldsymbol{s}-\hat{\boldsymbol{s}})^{\mathrm{T}}\mathbf{S}^{-1}(\boldsymbol{s}-\hat{\boldsymbol{s}})\right] \tag{2.5.24}$$

where $\hat{\boldsymbol{s}} = (\mathbf{M}^{-1}+\mathbf{C}^{-1})^{-1}\mathbf{M}^{-1}\boldsymbol{t}$ and $\mathbf{S} = (\mathbf{M}^{-1}+\mathbf{C}^{-1})^{-1}$. Note that finding the posterior of $\mathbf{C}$ involves another step, which deals with the inverse-Wishart distribution. We refer the reader to Alsing et al. (2016) for further technical details.

2.6 Bayesian Model Comparison

In Cosmology, we often have to deal with competing models. For example, a naive question might be, does introducing an extra parameter in my model makes it better or worse? In another word, is an extra parameter warranted by the data? A conventional (and frequentist) approach of just finding the minimum χ^2 does not help us answer this question. Instead, adding additional parameters lead to over-fitting and χ^2 values may be misleading.

Using the same notations as in §2.3, where $\boldsymbol{x}$ is the data vector, $\boldsymbol{z} = \{\boldsymbol{\theta}, \boldsymbol{\beta}\}$ is the set of latent variables ($\boldsymbol{\theta}$ is a vector of parameters of interest and $\boldsymbol{\beta}$ is a vector of nuisance parameters), recall that the first level of inference (parameter inference) is:

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z})\,p(\boldsymbol{z})}{p(\boldsymbol{x})} \tag{2.6.1}$$

where the denominator is ignored since it is independent from the model's parameters. $p(\boldsymbol{x})$ is the marginal likelihood or model evidence and is obtained by marginalising over all latent variables, that is,

$$p(\boldsymbol{x}) = \int \mathrm{d}\boldsymbol{z}\; p(\boldsymbol{x}|\boldsymbol{z})\,p(\boldsymbol{z}) \tag{2.6.2}$$

Suppose we have two competing models, $\mathcal{M}_1$ and $\mathcal{M}_2$, the posterior probability of model i is:

$$p(\mathcal{M}_i|\boldsymbol{x}) \propto p(\boldsymbol{x}|\mathcal{M}_i)\,p(\mathcal{M}_i). \tag{2.6.3}$$

The prior on each model, $p(\mathcal{M}_i)$ enables us to weigh the relative preference for a particular model. In the case where we assume all the models being considered are equally likely, then the posterior model probability is just equal to the model evidence. One can therefore calculate the ratio of the two model probabilities as:

Table 2.6.1 – The Jeffrey's scale for assessing the strength of a model. The middle column gives the relative odds of $\mathcal{M}_1$ against $\mathcal{M}_2$ and the last column provides a qualitative description for assessing the strength of a model over the other. Note also, that the two models, $\mathcal{M}_1$ and $\mathcal{M}_2$ form part of an exhaustive set, that is, $p(\mathcal{M}_1) + p(\mathcal{M}_2) = 1$.

$\lvert\ln B_{12}\rvert$	Odds	Strength of evidence
< 1.0	$\lesssim 3:1$	Inconclusive
1.0	$\sim 3:1$	Weak evidence
2.5	$\sim 12:1$	Moderate evidence
5.0	$\sim 150:1$	Strong evidence

$$\begin{aligned} B_{12} &= \frac{p(\mathcal{M}_1|\boldsymbol{x})}{p(\mathcal{M}_2|\boldsymbol{x})} \\ &= \frac{p(\boldsymbol{x}|\mathcal{M}_1)}{p(\boldsymbol{x}|\mathcal{M}_2)}\,\frac{p(\mathcal{M}_1)}{p(\mathcal{M}_2)} \end{aligned} \tag{2.6.4}$$

and this ratio is referred to as the *Bayes factor*. If the two prior model probabilities are the same, then this ratio is just equal to the ratio of the marginal likelihood of the data under the two different models. A value of $B_{12} > 1$ implies support for $\mathcal{M}_1$ compared to $\mathcal{M}_2$. The preference of one particular model over another is usually assessed using the Jeffrey's scale, which is an empirical scale.

A common scenario which arises often in Cosmology is when we have nested models. For simplicity, let us consider two models for fitting a straight line to a data. If m and c are the slope and the $y-$intercept of straight line, we have under $\mathcal{M}_1$, $y = mx + c$ and under $\mathcal{M}_2$, $y = mx$. Then, $\mathcal{M}_2$ is nested in M_1 at $c = 0$. In this particular case, we do not need to evaluate the marginal likelihood for both models to do model comparison. Instead, we can use the more complex model ($\mathcal{M}_1$ in this case) to find the posterior distribution of c and the Bayes factor is readily given by

$$B_{21} = \left.\frac{p(c|\boldsymbol{x}, \mathcal{M}_1)}{p(c|\mathcal{M}_1)}\right|_{c=0}. \tag{2.6.5}$$

This is know as the Savage-Dickey Density Ratio (SDDR) and it is useful to compare models with an additional parameter at a time (Dickey, 1971). Interestingly, the posterior distribution of the additional parameter is just a 1-dimensional quantity and this makes it easy to find an estimate for the Bayes factor. There exist various other techniques for estimating the marginal likelihood and we refer the reader to Trotta (2008) for an overview on these techniques.

Note 2.3: Learning Graphs

Let us consider two set of parameters, $\boldsymbol{z} = \{\boldsymbol{\theta}, \boldsymbol{\beta}\}$, where $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ correspond to the parameters of interest and a set of nuisance parameters respectively. These parameters are referred to as *latent* (hidden) variables and are typically inferred from the **observed** data, $\boldsymbol{x}$ (hence shaded in the graph below). A common graphical representation is as follows:

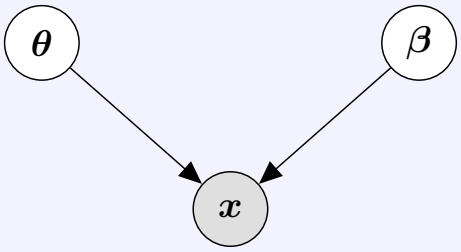


Figure 2.8 – An example of a graph structure for inferring $\boldsymbol{\theta}$, the set of parameters which is of interest to us. $\boldsymbol{x}$ is the observed data and $\boldsymbol{\beta}$ is a set of nuisance parameters.

As discussed in this section, the marginal likelihood is computed by marginalising over **all** latent variables, that is,

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\beta}) \, \mathrm{d}\boldsymbol{\theta} \, \mathrm{d}\boldsymbol{\beta}.$$

Often, the goal is not to just learn the posterior distributions of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ but to also learn the graph structure from the data. As a result, there is a large of possible structures and we need a measure to score each structure. If $p(\mathcal{M})$ is the prior on each possible graph structure, then, the posterior distribution of a graph is:

$$p(\mathcal{M}|\boldsymbol{x}) \propto p(\boldsymbol{x}|\mathcal{M}) \, p(\mathcal{M}) \tag{2.6.6}$$

where $p(\boldsymbol{x}|\mathcal{M})$ is the marginal likelihood under graph structure $\mathcal{M}$. Learning the score for each graph is a daunting task for two reasons. First, it is not trivial to do high dimensional integration and second, the number of different graph structures grows exponentially with the number of nodes. Note that in Figure 2.8, we have used a vector notation for $\boldsymbol{\theta} \in \mathbb{R}^m$ and $\boldsymbol{\beta} \in \mathbb{R}^n$, which means that there is a large possible number of combinations for $\boldsymbol{z}$.

Tensions in Cosmology

Different thought-provoking questions on tensions have been raised in the last decade in Cos-

mology. The highly debated ones include the H_0 tension between the distance ladder approach and *Planck* and the S_8 parameter between a weak lensing analysis and *Planck* (see Chapter 1 for a brief discussion).

As in any Bayesian analysis (or in fact, any Scientific process), the two main ingredients are the model (hypothesis) and the data. In an attempt to explain the source of the tensions, most of the debates are centred around the fact that the systematics are not being modelled properly and this can arise in any experiment. Another school of thought argues that perhaps, we do not fully understand the underlying Physics model.

As explained in this section, the marginal likelihood (and hence the Bayes factor between two competing models) is a proxy for assessing if a particular model is a good fit to the data. Recently, Joachimi et al. (2021a) argued that tension might just be a fluctuation, because a different data realisation can lead to a sampling distribution of the marginal likelihood. To elaborate on the exact meaning the marginal likelihood, we discuss how the latter is computed from a graphical perspective in Note 2.3.

2.7 Sampling Techniques

In this section, we will cover briefly different sampling algorithms which are used to perform Bayesian parameter inference in Cosmology. During the past 15 years or so, there has been a significant development of tools for performing sampling. The most common ones include Metropolis-Hastings, Gibbs and Hamiltonian Monte Carlo sampling schemes. Codes which are based on these techniques include `emcee` (Foreman-Mackey et al., 2013), `pyro` (Bingham et al., 2019), `pystan` (Riddell et al., 2021) and `pymc` (Salvatier et al., 2016). Since the development of sampling algorithm such as nested sampling, which also outputs an estimate for the marginal likelihood, there have been other variants such as `polychord` (Handley et al., 2015a,b) and `dynesty` (Higson et al., 2019). `Multinest` is another sampler, which not only provides samples for the posterior distributions but also an estimate of the Bayesian evidence (Feroz et al., 2009).

2.7.1 Metropolis-Hastings Sampling

The Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) is probably one of the most common Markov Chain Monte Carlo sampling algorithms used in different branches of Science. The aim is to obtain random samples from a distribution from which it is difficult to sample directly. Sampling techniques such as MH are generally used for multivariate analysis, where the dimensionality of the problem is high. Conventional grid-based

approach can be inefficient since it will require a large number of forward evaluations to accurately model the distribution.

A pseudo-algorithm for MH is given in Algorithm 2.1. It can be summarised as follows. Suppose we want to sample the full posterior distribution, $p(\boldsymbol{\theta}|\boldsymbol{x})$. A starting point, $\boldsymbol{\theta}$ and a proposal distribution, also known as the candidate generating density or jump distribution, $q(\boldsymbol{\theta})$ are first specified. An arbitrary step $\boldsymbol{u}$ is then taken and this step is accepted based on the following probability:

$$\min\left\{1, \frac{p(\mathbf{u}|\boldsymbol{x})q(\boldsymbol{\theta}|\boldsymbol{u})}{p(\boldsymbol{\theta}|\boldsymbol{x})q(\boldsymbol{u}|\boldsymbol{\theta})}\right\} \tag{2.7.1}$$

Algorithm 2.1 Random Walk Metropolis-Hastings

Initiate $\boldsymbol{\theta}$
for $i = 1, 2, \ldots, N_{\text{steps}}$ **do**
  Sample $\Delta\boldsymbol{\theta}$ from a proposal distribution $q(\Delta\boldsymbol{\theta}|\boldsymbol{\theta})$.
  $\boldsymbol{u} = \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$
  Draw $\alpha \sim \mathcal{U}[0, 1]$
  if $\alpha < \min\left\{1, \frac{p(\boldsymbol{u}|\boldsymbol{x})\, q(\boldsymbol{\theta}|\boldsymbol{u})}{p(\boldsymbol{\theta}|\boldsymbol{x})\, q(\boldsymbol{u}|\boldsymbol{\theta})}\right\}$ **then**
    $\boldsymbol{\theta}(i+1) = \boldsymbol{u}$
  else
    $\boldsymbol{\theta}(i+1) = \boldsymbol{\theta}(i)$
  end if
end for

The proposal distribution plays an important role in the sampling process. If a large proposal distribution is specified, most of the steps will be rejected and most of the distribution remains unexplored. On the other hand, if a small proposal distribution is used, the algorithm will take long to converge to a stationary distribution. A common practice is to approximate and update the proposal distribution based on the covariance matrix of the random samples. Fortunately, advanced samplers such as `emcee` allows for multiple `walkers` and the chains can be run in parallel and we do not have to specify for a proposal distribution.

2.7.2 Gibbs Sampling

Another technique for sampling is the Gibbs method (Geman & Geman, 1984). The advantage here is that the specification of a proposal distribution is not required. However, it requires analytic expression for the conditional distributions of each random variable. Depending on the problem, it is also possible to have another sampler in the Gibbs sampling scheme, for example, using an HMC to sample one of the conditional distributions.

For example, if we have two random variables, $\{\theta_1, \theta_2\}$ and our goal is to find the joint posterior of $p(\theta_1, \theta_2|\boldsymbol{x})$, we would require the following conditional distributions: $p(\theta_1|\boldsymbol{x}, \theta_2)$ and $p(\theta_2|\boldsymbol{x}, \theta_1)$. The sampling then procedes in block, that is, we sample θ_1 whilst conditioning on θ_2 and we sample θ_2 whilst conditioning on θ_1. This repetitive procedure allows for sampling the full joint posterior distribution. A pseudo-algorithm for the Gibbs sampling procedure is shown in Algorithm 2.2.

Algorithm 2.2 The Gibbs Sampler

Initialise $x^{(0)}$
for iteration $i = 1$ to N_{samples} **do**
$x_1^{(i)} \sim p(X_1 = x_1|X_2 = x_2^{(i-1)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$
$x_2^{(i)} \sim p(X_2 = x_2|X_1 = x_1^{(i)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$
$\vdots$
$x_D^{(i)} \sim p(X_D = x_D|X_1 = x_1^{(i)}, X_2 = x_3^{(i)}, \ldots, X_{D-1} = x_{D-1}^{(i)})$
end for

2.7.3 Hamiltonian Monte Carlo Sampling

Hamiltonian Monte Carlo, also referred to as Hybrid Monte Carlo (HMC) is an advanced sampling algorithm which is based on Hamiltonian dynamics (Duane et al., 1987; Neal, 2011; Betancourt, 2017). Coupled with the position variables in MH algorithms, HMC consists of another set of variables, called the *momentum* variables. These generally are independent normal distributions and the HMC alternates between the position and momentum variables. Unlike the simple MH formalism, because of the Hamiltonian dynamics, the HMC overcomes the slow exploration of the distribution we want to sample from and thus leads to a high acceptance ratio, that is, the number of accepted steps to the pre-defined number of steps.

Before delving into the details of HMC, it is important to understand the basics of Hamiltonian dynamics. We denote the position as $\boldsymbol{\theta}$ and the momentum as $\boldsymbol{p}$. The Hamiltonian of a system is given by the sum of kinetic energy, $K(\boldsymbol{p})$ and potential energy $U(\boldsymbol{\theta})$, that is,

$$\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{p}) = U(\boldsymbol{\theta}) + K(\boldsymbol{p}). \tag{2.7.2}$$

The negative log-posterior is chosen as the potential energy in the sampling procedure, that is,

$$U(\boldsymbol{\theta}) = -\ln[p(\boldsymbol{x}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})] \tag{2.7.3}$$

and the kinetic energy is given in terms of the momentum, that is,

$$K(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^{\mathrm{T}}\mathbf{M}^{-1}\boldsymbol{p} \tag{2.7.4}$$

Algorithm 2.3 Hamiltonian Monte Carlo. See Hajian (2007) for further details.

Initialise $\boldsymbol{\theta}_0$
for $i = 1$ to N_{samples} **do**
  $\boldsymbol{p} \sim \mathcal{N}(0, \mathbf{M})$
  $(\boldsymbol{\theta}^*_{(0)}, \boldsymbol{p}^*_{(0)}) = (\boldsymbol{\theta}_{(i-1)}, \boldsymbol{p})$
  for $j = 1$ to L **do**
    Make a leapfrog move: $(\boldsymbol{\theta}^*_{(j-1)}, \boldsymbol{p}^*_{(j-1)}) \to (\boldsymbol{\theta}^*_{(j)}, \boldsymbol{p}^*_{(j)})$
  end for
  $(\boldsymbol{\theta}^*, \boldsymbol{p}^*) = (\boldsymbol{\theta}_{(L)}, \boldsymbol{p}_{(L)})$
  Draw $\alpha \sim \mathcal{U}[0, 1]$
  if $\alpha < \min\{1, e^{-[\mathcal{H}(\boldsymbol{\theta}^*, \boldsymbol{p}^*) - \mathcal{H}(\boldsymbol{\theta}, \boldsymbol{p})]}\}$ **then**
    $\boldsymbol{\theta}_{(i)} = \boldsymbol{\theta}^*$
  else
    $\boldsymbol{\theta}_{(i)} = \boldsymbol{\theta}_{(i-1)}$
  end if
end for

$\mathbf{M}$ is the mass matrix and is generally chosen to be diagonal. The pdf for the kinetic energy correspond to a d dimensional multivariate normal distribution centred on zero and covariance $\mathbf{M}$. The choice of the mass matrix is important to ensure good performance of the sampling procedure. The partial derivatives of the Hamiltonian with respect to time are given by the Hamilton's equations:

$$\begin{aligned} \frac{d\boldsymbol{\theta}_i}{dt} &= \frac{\partial \mathcal{H}}{\partial \boldsymbol{p}_i} \\ \frac{d\boldsymbol{p}_i}{dt} &= -\frac{\partial \mathcal{H}}{\partial \boldsymbol{\theta}_i} \end{aligned} \tag{2.7.5}$$

where the index i corresponds to the i^{th} dimension. An important property of the dynamics is that the total energy is conserved, that is, the Hamiltonian of the system is invariant. The two equations above form a set of differential equations, which need to be solved to obtain approximate solution for $\boldsymbol{\theta}$ and $\boldsymbol{p}$. This can be achieved numerically by discretising time. While *Euler's method* is best known for solving differential equations numerically, an alternative approach is the *Leapfrog method*:

$$\begin{aligned} \boldsymbol{p}_i(t + \epsilon/2) &= \boldsymbol{p}_i(t) - \frac{\epsilon}{2}\left(\frac{\partial U}{\partial \boldsymbol{\theta}_i}\right)\Bigg|_t \\ \boldsymbol{\theta}_i(t + \epsilon) &= \boldsymbol{\theta}_i(t) + \epsilon\frac{\boldsymbol{p}_i(t + \epsilon/2)}{m_i} \\ \boldsymbol{p}_i(t + \epsilon) &= \boldsymbol{p}_i(t + \epsilon/2) - \frac{\epsilon}{2}\left(\frac{\partial U}{\partial \boldsymbol{\theta}_i}\right)\Bigg|_{t+\epsilon} \end{aligned} \tag{2.7.6}$$

The HMC algorithm can be understood as follows. It has two main steps, where in the first, only the momentum changes while in the second both the momentum and the position are evolved. The momentum is typically randomly drawn from a multivariate normal distribution, the mass matrix for this step. In the second stage, a step analogous to the Metropolis update is performed and the Hamiltonian dynamics is also simulated by defining the stepsize, ϵ and the number of steps, L for the Leapfrog method. Once the step for the Leapfrog integrator is completed, the proposed state is accepted with probability

$$\min\left[1, \exp(-\mathcal{H}(\boldsymbol{\theta}_*, \boldsymbol{p}_*) + \mathcal{H}(\boldsymbol{\theta}, \boldsymbol{p}))\right] \tag{2.7.7}$$

With a good specification of the stepsize, ϵ, the number of Leapfrog moves, L and the mass matrix, $\mathbf{M}$, the HMC can depict significant performance over other sampling algorithms. For example, HMC typically generates chains with fewer correlated steps and results in better acceptance rate and convergence. However, the derivatives of the potential energy (the log-posterior) can be an expensive quantity to compute. Strictly, HMC allows solving for problems that cannot be solved otherwise, using other samplers.

2.7.4 Nested Sampling

Another sampler which is commonly used in cosmology is `multinest` (the `Python` wrapper being `PyMultinest`), the algorithm behind being the nested sampling originally developed by Skilling (2004, 2006). As discussed in §2.6, an important quantity in a Bayesian analysis is the Bayesian evidence, which is a difficult quantity to compute since it involves an integration over the whole set of parameters.

Algorithm 2.4 Nested sampling algorithm (from Wikipedia).

N points are drawn from the prior volume.
for $i = 1$ to N_{iter} **do**
  $L_i :=$ min(current likelihoods of the points)
  $X_i := \exp(-i/N)$
  $w_i := X_{i-1} - X_i$
  $Z := Z + L_i \cdot w_i$
  The point with the least likelihood is saved as a sample point, the weight w_i.
  The point with least likelihood is updated via MCMC step.
  Steps which are above L_i are kept.
end for
return Z

Nested sampling not only outputs an estimate for the Bayesian evidence but also returns posterior samples. The choice of nested sampling is motivated in cases where the posterior

distribution is believed to have several peaks, with separated modes and also in examples where the joint posterior has peculiar shapes such as bananas. The algorithm is summarised in Algorithm 2.4. Among the first nested sampling algorithms is `multinest` (Feroz et al., 2009). Other variants based on the idea of nested sampling are `polychord` (Handley et al., 2015a,b) and `dynesty` (Higson et al., 2019). `UltraNest` is another sampler based on nested sampling (Buchner, 2014, 2019).

2.8 Summary

Bayesian Statistics is the cornerstone to almost any scientific data analysis. In this chapter, we start with the very basics of probability, followed by describing the normal distribution which is central in this thesis. Next, we dive deep into Bayes' theorem and we provide two examples, illustrating the application of Bayes' theorem in two different simple contexts. The choice of priors is a topic of hot debate when adopting a Bayesian approach in any data analysis problem. We therefore discuss the different types of priors and the motivation for choosing one type of prior distribution over another. Strictly, it depends very much on the problem we want to solve. In many cases, the problem also compels us to think of it as a series of computations, occurring in a hierarchical fashion. Bayesian hierarchical methods naturally allow one to learn about the latent variables by specifying hierarchical building blocks. We also cover Bayesian model comparison which deals with quantifying whether a particular model is a better fit compared to another. We finally discuss some of the sampling methods which can be used to sample the joint posterior distribution of cosmological and nuisance parameters.

Chapter 3

KERNEL METHODS AND GAUSSIAN PROCESS

> Causal interpretation of the results of regression analysis of observational data is a risky business. The responsibility rests entirely on the shoulders of the researcher, because the shoulders of the statistical technique cannot carry such strong inferences.
>
> *Jan de Leeuw*

As discussed in Chapter 2, (probabilistic) modelling is central in any field of Science and Engineering. The Bayesian modelling approach uses the rules of probability to learn parameters of a model (Bayesian parameter inference), compare models (Bayesian model comparison) and make predictions (Bayesian posterior predictive distribution). This framework becomes more powerful when combined with flexible approach such as Bayesian non-parametric methods (Ghahramani, 2013).

A model can be thought of as a representation of the data that one can observe. Often, we are interested in making predictions or forecasts at points where no data has been observed. In some applications, deterministic forecasts can easily be falsified, for example, a statement such as, '*the temperature will be* 15^oC' is too fragile to be accepted. Hence, uncertainty quantification plays a major role in Bayesian non-parametric.

In practice, we also want to design learning algorithms which are adaptive to new data, that is, we would expect the predictive probability to improve in the regime of an increased amount of new data, although in some cases, this might not hold. Bayesian modelling is not a trivial task. It can be quite challenging to deal with data corrupted by noise. While uncertainty quantification remains an important ingredient in this framework, there can be various sources of uncertainty which are complicated to be accounted for in the modelling process. For example, we might not fully understand the noise properties in the measurement process. Fortunately, the Bayesian framework provides an elegant, natural and coherent approach to represent all

forms of uncertainty in the model. Essentially, starting from the data to making predictions, all uncertainties can be encapsulated in the Bayesian modelling framework.

In this chapter, we will specifically look into Bayesian non-parametric, involving kernel methods. In particular, we will cover the concepts behind Gaussian Processes (GP) which are crucial to the applications in Chapters 4, 5 and 7. In §3.1, we highlight the importance of kernel methods and in §3.2, we elaborate on the theory behind GPs. Finally, in §3.3, we provide a brief summary of this chapter.

3.1 Kernels

In this section, we will elaborate on how kernels can be used in various Machine Learning algorithms. In short, kernels allow one to construct models of the data by encoding structure such as additivity, interaction between variables and periodicity. Importantly, the fact that kernels can be added and multiplied allows us to build better and more representative models of the data. We will often treat kernel from a GP perspective but note that kernels are extensively used in different Machine Learning algorithms such as Support Vector Machines (SVM), Principle Component Analysis (PCA) and many more.

3.1.1 Definition and Examples

A kernel, also referred to as the kernel function or covariance function, is simply a positive-definite function between two points (vectors), $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ in Euclidean space. It is generally deemed as a measure of *similarity* between any two vectors. For example, in Gaussian process models (see §3.2), a kernel is used to define the prior covariance between any two function values, f and f', that is, $\text{cov}[f(\boldsymbol{\theta}), f'(\boldsymbol{\theta}')] = k(\boldsymbol{\theta}, \boldsymbol{\theta}')$. A common assumption is that points which are close to $\boldsymbol{\theta}$ are likely to have similar values of the target and hence in a GP model, training points which are close to the test point are informative about the prediction at that test point.

From a different perspective, we can also think of kernel as a scalar product of feature transformations, $\boldsymbol{\theta} \rightarrow \phi(\boldsymbol{\theta})$, that is

$$\langle \boldsymbol{\theta}, \boldsymbol{\theta}' \rangle \rightarrow \langle \phi(\boldsymbol{\theta}), \phi(\boldsymbol{\theta}') \rangle = k(\boldsymbol{\theta}, \boldsymbol{\theta}'). \tag{3.1.1}$$

For example, if we consider a polynomial kernel, of order 2, where the inputs are of dimension 2, we have

$$
\begin{aligned}
k(\boldsymbol{\theta}, \boldsymbol{z}) &= (\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{z})^2 \\
&= (x_1 z_1 + x_2 z_2)^2 \\
&= x_1^2 z_1^2 + x_2^2 z_2^2 + 2 x_1 z_1 x_2 z_2 \\
&= \phi^{\mathrm{T}}(\boldsymbol{\theta})\phi(\boldsymbol{z})
\end{aligned}
\tag{3.1.2}
$$

where $\phi(\boldsymbol{\theta}) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$. For example, if we consider the XOR classification problem, we cannot find a separable plane which separates the X and the O. This is evident when we look at the left panel in Figure 3.1.

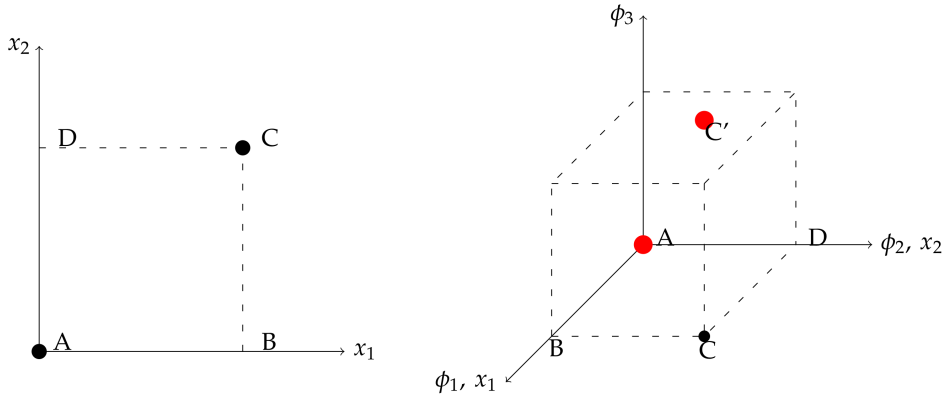


Figure 3.1 – The left figures shows the standard XOR classification problem, where the goal is to find a separation line which will separate (A,C) with (B,D). Unfortunately, in the original space, it is impossible to find such a line, but as shown in the figure on the right hand side, after applying the (polynomial) kernel trick, one can define a separation plane which separates (A,C) with (B,D).

In Table 3.1.1, we give the positions of the points A, B, C and D and their respective labels, denoted by y. For example, we can interpret 0 as being 'O' while 1 being 'X'. If we use the polynomial kernel of order 2 where $\phi(\boldsymbol{\theta}) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$, the different positions in this new frame are shown in Table 3.1.2.

Table 3.1.1 – XOR classification example

Points	x_1	x_2	y
A	0	0	0
B	1	0	1
C	1	1	0
D	0	1	1

In the original frame, we cannot determine a separation line to successfully perform the classification problem. However, when we focus on the right panel of Figure 3.1, we can cer-

tainly find a separation plane which separates 'O' from 'X'. In fact, there is exists a large number of possible solutions in this case.

Table 3.1.2 – XOR example using polynomial kernel

Points	$\phi_1 = x_1^2$	$\phi_2 = x_2^2$	$\phi_3 = \sqrt{2}x_1x_2$	y
A	0	0	0	0
B	1	0	0	1
C′	1	1	$\sqrt{2}$	0
D	0	1	0	1

This simple classification example motivates the adoption of kernel methods in many Machine Learning algorithm today. In fact, in some cases, kernel methods can outperform techniques such as deep learning. One of the problems which kernel method faces is scalability since it is based on the dimensions of data space, which can be very large. However, significant progress has been made in this field of research. Before elaborating on Gaussian Process, which is based on a kernel, we will first provide a formal definition of a kernel below.

Mercer's Theorem

A function $k(\theta, \theta')$ is a kernel if it is symmetric, that is, $k(\theta, \theta') = k(\theta', \theta)$ and is positive-definite, that is, for every function $g : \theta \to \mathbb{R}$,

$$\int \int k(\theta, \theta')\, g(\theta)\, g(\theta')\, \mathrm{d}\theta d\theta' \geq 0$$

and similarly, for the positive-definite property, given a finite set $(\theta_1, \theta_2, \ldots \theta_N)$, the Gram matrix

$$\{\mathbf{K}(\theta_i, \theta_j)\}_{i,j=1}^{N} \succcurlyeq 0. \tag{3.1.3}$$

A kernel normally has a set of parameters, also referred to as *hyper-parameters*, which define its overall shape. For example, for a Gaussian kernel, also referred to as the Squared-Exponential (SE) kernel,

$$k(\theta, \theta') = A \exp\left[-\frac{1}{2}\frac{(\theta - \theta')^2}{\lambda^2}\right]$$

the set of hyper-parameters is $\{A, \lambda\}$. λ defines the width of the kernel and controls the

smoothness property of the function being modelled while A is the amplitude of the kernel. To understand this better, let us look into one example of the Gaussian kernel. In Figure 3.2, we fix $A = 1$ and $\lambda = 1$ and the functions drawn from the prior, are shown in the solid line. We also show a second set of samples (the dotted curves) in Figure 3.2 where the hyper-parameters, $A = 1$ and $\lambda = 0.1$.

Kernels can be classified into two main groups, namely *stationary* and *non-stationary* kernels. A stationary kernel is one whose value depends only on the difference between the two points, that is, $\boldsymbol{\theta} - \boldsymbol{\theta}'$. Changing the values of the $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ by the same amount will not affect the value of the kernel function. Hence, it is *invariant* to *translations*. If the kernel function is a function of the magnitude of the difference, $|\boldsymbol{\theta} - \boldsymbol{\theta}'|$, then it *isotropic*. Moreover, if k is a function of only $|\boldsymbol{\theta} - \boldsymbol{\theta}'|$, then it is referred to as a *radial basis function* (RBF).

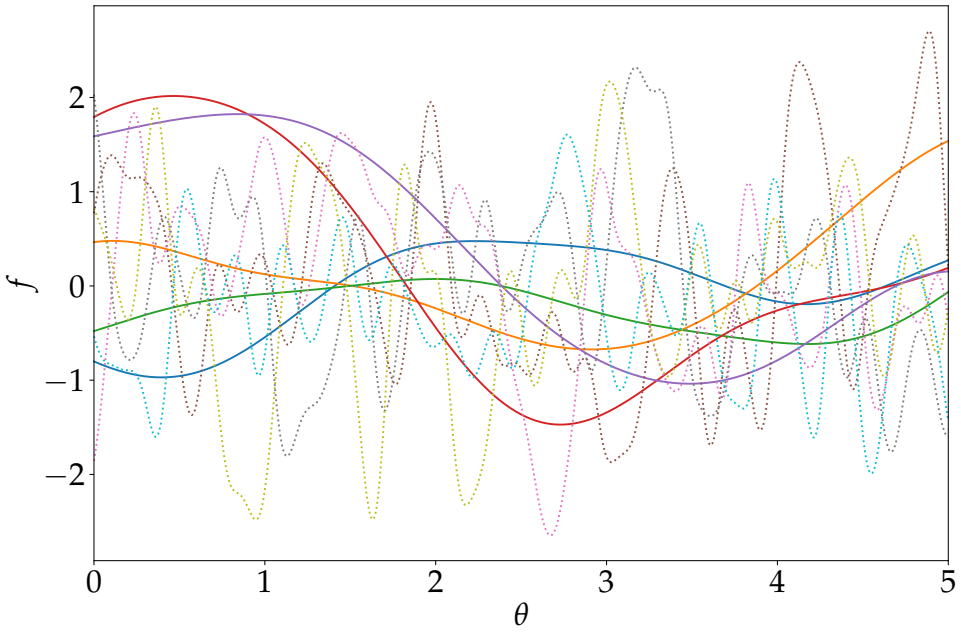


Figure 3.2 – In this figure, we show two sets (solid and dotted curves) of 5 samples of function drawn from a multivariate normal prior, $\mathcal{N}(\mathbf{0}, \mathbf{K})$. For the solid curves, the hyper-parameters for the Gaussian function are $A = 1$ and $\lambda = 1$ while for the dotted curves, $A = 1$ and $\lambda = 0.1$. Hence, the parameter λ controls the smoothness of the function.

In contrast, a non-stationary kernel is one which will cause the prediction, for example in a GP model, to change if the position of the points are moved whilst keeping the kernel parameters fixed.

3.1.2 Constructing Kernels

An important property of kernels is that they can be combined or modified to generate new kernels from old ones. For example, if we have a function $f(\boldsymbol{\theta}) = f_1(\boldsymbol{\theta}) + f_2(\boldsymbol{\theta})$, where $f_1(\boldsymbol{\theta})$ and $f_2(\boldsymbol{\theta})$ are independent, then $k(\boldsymbol{\theta}, \boldsymbol{\theta}') = k_1(\boldsymbol{\theta}, \boldsymbol{\theta}') + k_2(\boldsymbol{\theta}, \boldsymbol{\theta}')$. In short, kernels can be added.

In the same spirit, if we have a function $f(\boldsymbol{\theta}) = f_1(\boldsymbol{\theta})\, f_2(\boldsymbol{\theta})$, then $k(\boldsymbol{\theta}, \boldsymbol{\theta}') = k_1(\boldsymbol{\theta}, \boldsymbol{\theta}')\, k_2(\boldsymbol{\theta}, \boldsymbol{\theta}')$, that is, a kernel can be constructed by multiplying two different kernels.

Crucially, the addition and/or multiplication of two positive-definite kernels lead to another positive-definite kernel. Below, we provide a list of how kernels can be constructed. If $k_1(\boldsymbol{\theta},\, \boldsymbol{\theta}')$ and $k_2(\boldsymbol{\theta},\, \boldsymbol{\theta}')$ are two arbitrary kernels, $h(\boldsymbol{\theta})$ and $\varphi(\boldsymbol{\theta})$ are arbitrary functions, then

1. $k(\boldsymbol{\theta},\, \boldsymbol{\theta}') = c k_1(\boldsymbol{\theta},\, \boldsymbol{\theta}')$
2. $k(\boldsymbol{\theta},\, \boldsymbol{\theta}') = k_1(\boldsymbol{\theta},\, \boldsymbol{\theta}') k_2(\boldsymbol{\theta},\, \boldsymbol{\theta}')$
3. $k(\boldsymbol{\theta},\, \boldsymbol{\theta}') = k_1(\boldsymbol{\theta},\, \boldsymbol{\theta}') + k_2(\boldsymbol{\theta},\, \boldsymbol{\theta}')$
4. $k(\boldsymbol{\theta},\, \boldsymbol{\theta}') = k_1\left(\varphi(\boldsymbol{\theta}),\, \varphi(\boldsymbol{\theta}')\right)$
5. $k(\boldsymbol{\theta},\, \boldsymbol{\theta}') = h(\boldsymbol{\theta}) k_1(\boldsymbol{\theta},\, \boldsymbol{\theta}') h(\boldsymbol{\theta}')$
6. $k(\boldsymbol{\theta},\, \boldsymbol{\theta}') = e^{k_1(\boldsymbol{\theta}, \boldsymbol{\theta}')}$

where $c > 0$.

Kernels are also used to build multi-dimensional models, that is, functions with more than 1 input. For example, if we consider the squared-exponential kernel with different dimensions, we can define a characteristic length-scale, λ_d for each dimension and the resulting kernel is often referred to as the SE-ARD kernel, where ARD refers to *automatic relevance determination*. Hence, using the multiplication property of kernels, we have

$$\begin{aligned} k(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \prod_{d=1}^{D} A_d \exp\left[-\frac{1}{2}\frac{(\theta_d - \theta_d')^2}{\lambda_d^2}\right] \\ &= A \exp\left[-\frac{1}{2}\sum_{d=1}^{D}\frac{(\theta_d - \theta_d')^2}{\lambda_d^2}\right] \end{aligned} \tag{3.1.4}$$

and the fact that we have a length-scale for each dimension, implies that the magnitude of λ_d will determine the relevance for a particular dimension. In other words, if the length-scale, λ_i is large, the overall contribution of dimension i is small to the overall kernel function evaluation. The SE-ARD remains the popular choice for high-dimensional modelling for various reasons. It is interpretable and consists of very few kernel hyper-parameters. Moreover, it can model any continuous function given sufficient amount of data. However, since we are effectively computing pairwise distance, this means that as the dimensionality of the problem increases, $|\boldsymbol{\theta} - \boldsymbol{\theta}'| \to 0$ and the kernel fails to capture the behaviour of the function. In short, the kernel can be slow to learn due to the curse of dimensionality.

3.2 Gaussian Processes

Before elaborating on the technical details of GPs, it is worth reminding ourselves what parametric Bayesian modelling is. In Chapter 2, we have seen that a parametric model is governed by a finite set of parameters, $\boldsymbol{\theta}$ and often the goal is to infer the posterior distribution of these parameters, that is, we want $p(\boldsymbol{\theta}|\boldsymbol{x})$. Once the posterior distributions are learned, they capture all the information we have to know about the data, and hence the data becomes irrelevant when making predictions, that is, if we want the learn $p(x_*|\boldsymbol{\theta})$ where x_* is a point we want to predict, then,

$$p(x_*|\boldsymbol{\theta}) = \int p(x_*|\boldsymbol{\theta})\, p(\boldsymbol{\theta}|\boldsymbol{x})\, \mathrm{d}\boldsymbol{\theta}. \tag{3.2.1}$$

On the other hand, a Bayesian non-parametric approach assumes that we cannot model the data using a finite set of parameters, $\boldsymbol{\theta}$. Non-parametric models are defined with an infinite dimensional $\boldsymbol{\theta}$ and the latter is represented by a function. Unlike parametric methods where most of the information is retained by a small set of parameters, $\boldsymbol{\theta}$, non-parametric methods are generally memory-based, that is, all information about the training data must be stored in order to make predictions.

GPs are deemed as simple probabilistic models of functions. In particular, a GP is a distribution over the functions such that any finite set of functions have a joint multivariate Gaussian distribution. In the absence of observed data, a GP is fully specified by its mean function

$$\mathbb{E}[\boldsymbol{f}] = \boldsymbol{\mu} \tag{3.2.2}$$

and covariance

$$\mathrm{cov}[f(\boldsymbol{\theta}), f(\boldsymbol{\theta}')] = k(\boldsymbol{\theta}, \boldsymbol{\theta}') \tag{3.2.3}$$

via the kernel function, k. Hence, the GP prior (before seeing any data), can be written as, $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$, where the kernel matrix $\mathbf{K}$ is constructed by evaluating the kernel function for every pair of inputs, $\boldsymbol{\theta}$. It is customary to assume a zero mean GP in most applications. In this work, we will adopt a zero mean GP in Chapters 4 and 5 but we will relax this assumption in Chapter 7 where we will assume an explicit mean function, which depends on the inputs to the model.

GPs are widely used in the field of spatial statistics, where one wants to model, for example,

the temperature as a function of spatial location. In the field of Machine Learning (ML), they have been used extensively to do regression and classification. Related research topics which apply GPs include bandit optimisation, Bayesian optimisation and emulation (Desautels et al., 2012; Slivkins, 2019; Mootoovaloo et al., 2020). We focus entirely on regression because this is central to the research carried out in this work.

3.2.1 Regression - Weight Space

Let us consider a simple non-linear regression of the form

$$y_i = f(\boldsymbol{\theta}_i) + \epsilon_i \tag{3.2.4}$$

where we have n observed data points, that is, $\{(\boldsymbol{\theta}_1, y_1), (\boldsymbol{\theta}_2, y_2) \ldots, (\boldsymbol{\theta}_n, y_n)\}$. Let us also assume that the noise covariance matrix is $\boldsymbol{\Sigma}$. If we choose to model this data by a polynomial function, using vector and matrix notations, we can re-write Equation 3.2.4 as

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3.2.5}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{n \times d}$ is a design matrix whose columns contain the different basis functions, then we can analytically derive an expression for the posterior distributions of the regression coefficients, $\boldsymbol{\beta}$ if we assume a Gaussian prior of mean zero and covariance $\mathbf{C}$. This modelling approach is referred to as a Gaussian Linear Model, since the model is linear in the regression coefficients $\boldsymbol{\beta}$ and their posterior distributions turn out to be Gaussian. Following Note 3.1, the posterior distribution of $\boldsymbol{\beta}$ is given by:

$$p(\boldsymbol{\beta}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\Omega}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{y}, \boldsymbol{\Omega}) \tag{3.2.6}$$

where $\boldsymbol{\Omega} = (\mathbf{C}^{-1} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi})^{-1}$. Suppose, we want to predict the function at $\boldsymbol{\theta}_*$, this is straightforwardly given by:

$$\begin{aligned} y_* &= \Phi_* \boldsymbol{\beta}_{\text{map}} \\ &= \Phi_* \boldsymbol{\Omega}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{y} \end{aligned} \tag{3.2.7}$$

where $\boldsymbol{\beta}_{\text{map}}$ is the mean of the posterior distribution of the $\boldsymbol{\beta}$ and Φ_* is the design matrix computed at the test point. Instead of working in the $\boldsymbol{\beta}$ space, we will now work in the data

space. Defining $\mathbf{K} = \mathbf{\Phi C \Phi}^{\mathrm{T}}$ and $k_*^{\mathrm{T}} = \mathbf{\Phi}_* \mathbf{C \Phi}^{\mathrm{T}}$ and using the Woodbury identity*, we have

$$y_* = k_*^{\mathrm{T}} (\mathbf{K} + \mathbf{\Sigma})^{-1} \boldsymbol{y}. \tag{3.2.8}$$

Note 3.1: Marginal and Conditional Gaussian distributions

Following Bishop (2006), if we have a distribution (prior) for $\boldsymbol{\theta}$ and a conditional distribution (likelihood) for $\boldsymbol{y}$ given x,

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} | \boldsymbol{\mu}, \mathbf{\Lambda}^{-1})$$

$$p(\boldsymbol{y} | \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y} | \mathbf{A}\boldsymbol{\theta} + \boldsymbol{b}, \mathbf{L}^{-1})$$

the conditional distribution (posterior distribution), $p(\boldsymbol{\theta}|\boldsymbol{y})$ and the marginal distribution (evidence), $p(\boldsymbol{y})$ are respectively given by:

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\theta} | \mathbf{\Sigma}[\mathbf{A}^{\mathrm{T}} \mathbf{L} (\boldsymbol{y} - \boldsymbol{b}) + \mathbf{\Lambda}\boldsymbol{\mu}], \mathbf{\Sigma})$$

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y} | \mathbf{A}\boldsymbol{\mu} + \boldsymbol{b}, \mathbf{L}^{-1} + \mathbf{A}\mathbf{\Lambda}^{-1}\mathbf{A}^{\mathrm{T}})$$

where $\mathbf{\Sigma} = (\mathbf{\Lambda} + \mathbf{A}^{\mathrm{T}} \mathbf{L} \mathbf{A})^{-1}$.

Importantly, this equation is in fact the mean of the predictive distribution if we were to use a GP model with prior mean zero. The matrix, $\mathbf{\Phi C \Phi}^{\mathrm{T}}$ is referred to as a *Gram* matrix and is a valid kernel (see §3.1 for further details).

3.2.2 Regression - Function Space

Instead of working in the weight space ($\boldsymbol{\beta}$), we will now model directly using a set of functions, $\boldsymbol{f}$. As in parametric methods, we can write the posterior distribution of $\boldsymbol{f}$ as:

$$p(\mathbf{f} | \mathbf{y}, \boldsymbol{\theta}) = \frac{p(\mathbf{y} | \mathbf{f}, \boldsymbol{\theta}) \; p(\mathbf{f} | \boldsymbol{\theta})}{p(\mathbf{y} | \boldsymbol{\theta})}$$

In Figure 3.3, we show the graphical model for such a regression task. If we assume a zero mean

*$(\mathbf{A} + \mathbf{CBC}^{\mathrm{T}})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{C}(\mathbf{B}^{-1} + \mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{C})^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}$

GP prior, then, the joint distribution of the training points (the outputs) and the prediction, f_* is:

$$\begin{pmatrix} \boldsymbol{y} \\ f_* \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}+\boldsymbol{\Sigma} & \boldsymbol{k}_* \\ \boldsymbol{k}_*^{\mathrm{T}} & k_{**} \end{bmatrix}\right) \tag{3.2.9}$$

where $\boldsymbol{k}_* \in \mathbb{R}^n$ is a vector of the kernel values computed between the test point and each of the training point. Similarly, k_{**} is just the kernel value at the test point. Using the properties of the conditional distribution from Note 2.1, the conditional (predictive) distribution of f_* is another normal distribution with mean and variance

$$\begin{aligned} \mathbb{E}[f_*] &= \boldsymbol{k}_*^{\mathrm{T}}(\mathbf{K}+\boldsymbol{\Sigma})^{-1}\boldsymbol{y} \\ \mathrm{var}[f_*] &= k_{**} - \boldsymbol{k}_*^{\mathrm{T}}(\mathbf{K}+\boldsymbol{\Sigma})^{-1}\boldsymbol{k}_* \end{aligned} \tag{3.2.10}$$

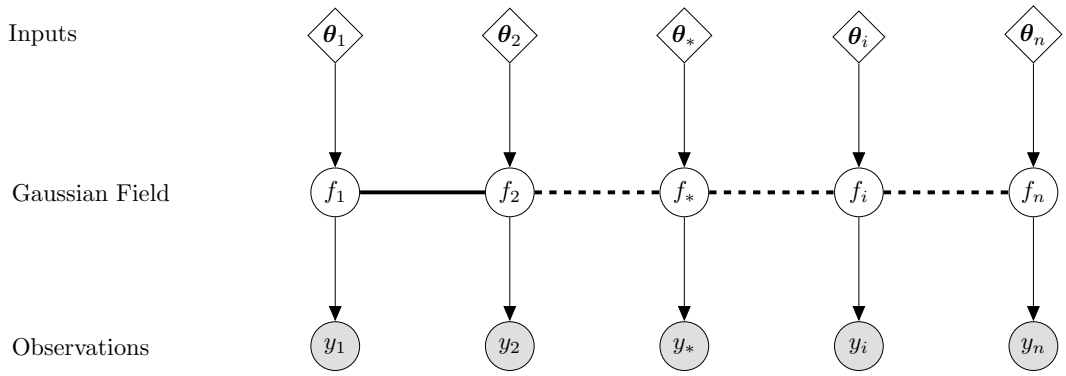


Figure 3.3 – Graphical model using Gaussian Process for a regression problem. The shaded nodes represent the observed variables, y, the middle row represent the Gaussian field, that is, the latent variables, $\boldsymbol{f}$ and in the top row, we have the inputs, $\boldsymbol{\theta}$. Importantly, the addition of other observables, y, inputs, $\boldsymbol{\theta}$ and latent function, f does not change the distribution of the existing variables because of the marginalisation property of GPs.

We can draw various conclusions from Equations 3.2.10. For example, if we were dealing with noise-free regression, then, $\boldsymbol{\Sigma} \to 0$. However, a jitter term is often added for solving a linear system of equations, $\mathbf{A}\boldsymbol{\theta} = \boldsymbol{b}$, that is, the kernel $\mathbf{K} \to \mathbf{K} + \sigma^2\mathbb{I}$, where σ^2 is set to a very small value, usually $\sim 10^{-10}$. Moreover, once $\boldsymbol{\alpha} = (\mathbf{K}+\boldsymbol{\Sigma})^{-1}\boldsymbol{y}$ is computed and stored, the mean prediction can be calculated very quickly (since there is no further matrix inversion). In other words, the mean prediction is a linear predictor since it is a linear combination of the observations, $\boldsymbol{y}$. On the other hand, an important observation is that the variance calculation always involve an $\mathcal{O}(n^2/2)$ operation, assuming the Cholesky factor, $\mathbf{L}$ is computed and stored.

While the predictive distribution is important for making predictions, as well as, quantifying the uncertainty at test points, the marginal likelihood $p(\boldsymbol{y}|\boldsymbol{\theta})$ is crucial for model selection.

In general, the kernel function, $k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ is also a function of other hyper-parameters, $\boldsymbol{\eta}$. For example, let us consider the Squared-Exponential kernel,

$$k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = A \exp\left[-\frac{1}{2}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)^{\mathrm{T}} \boldsymbol{\Lambda}^{-1} (\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)\right] \tag{3.2.11}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1^2, \lambda_2^2, \ldots \lambda_d^2)$. Hence, $\boldsymbol{\eta} = \{A, \boldsymbol{\Lambda}\}$ is a vector of the hyper-parameters A and λ_i where i refers to each dimension. Once observations are made, these kernel hyper-parameters are optimised by maximising the marginal likelihood which is given by:

$$\log p(\boldsymbol{y}|\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}(\mathbf{K} + \boldsymbol{\Sigma})^{-1}\boldsymbol{y} - \frac{1}{2}\log|\mathbf{K} + \boldsymbol{\Sigma}| + \text{constant}. \tag{3.2.12}$$

The above expression for the marginal likelihood has two terms, which take into account the model fit and the model complexity. The first term encourages the fit to the data and depends on the data while the second term, involves a complexity penalty, $\log|\mathbf{K} + \boldsymbol{\Sigma}|$ and is referred to as the *Occam factor*. An important observation from Equation 3.2.12 is that the fact that the matrix, $\mathbf{K}_y = \mathbf{K} + \boldsymbol{\Sigma}$ is of size $n \times n$, where n is the number of training points, implies that training a GP model involves an $\mathcal{O}(n^3)$ operation at each step of the optimisation procedure, when we are learning the kernel hyper-parameters, $\boldsymbol{\eta}$.

For numerical stability, we first compute the Cholesky factor, $\mathbf{L}$, of $\mathbf{K}_y \equiv \mathbf{L}\mathbf{L}^{\mathrm{T}}$, solve for $\boldsymbol{u}$ in the linear system $\mathbf{L}\boldsymbol{u} = \boldsymbol{y}$ followed by solving for $\boldsymbol{\alpha}$ in $\mathbf{L}^{\mathrm{T}}\boldsymbol{\alpha} = \boldsymbol{u}$. The marginal likelihood is then given by

$$\log p\left(\boldsymbol{y} \mid \boldsymbol{\theta}\right) = -\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{\alpha} - \sum_i \log \mathbf{L}_{ii} + \text{constant}. \tag{3.2.13}$$

Moreover, the partial derivatives of equation (3.2.12) with respect to the kernel hyper-parameters, $\boldsymbol{\eta} = \{A, \boldsymbol{\lambda}\}$ can be computed in closed form

$$\frac{\partial}{\partial \boldsymbol{\eta}_i} \log p\left(\boldsymbol{y} \mid \boldsymbol{\theta}\right) = \frac{1}{2}\mathrm{tr}\left[\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{T}} - \mathbf{K}^{-1}\right)\frac{\partial \mathbf{K}}{\partial \boldsymbol{\eta}_i}\right] \tag{3.2.14}$$

and $\boldsymbol{\alpha} = \mathbf{K}_y^{-1}\boldsymbol{y}$. The gradients are useful when maximising the marginal likelihood when using gradient-based optimisation.

One can in fact adopt a fully Bayesian approach and marginalise over the kernel hyper-parameters, $\boldsymbol{\eta}$, that is, we can write the posterior distribution of $\boldsymbol{\eta}$ as

$$p(\boldsymbol{\eta}|\boldsymbol{\theta}, \boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{\eta})\, p(\boldsymbol{\eta}) \tag{3.2.15}$$

where $p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{\eta})$ is given by Equation 3.2.12 and the kernel hyper-parameters are marginalised over in this procedure. While this would be a preferred approach to learn the kernel hyper-parameters, the $\mathcal{O}(n^3)$ computational cost compels us to stick with the optimisation procedure, especially when we have multiple functions to learn as in this work. See Chapters 4, 5 and 7.

Moreover, a completely different branch of research in the GP community involves devising techniques to deal with the most expensive part of the procedure, that is, training the GP model, which involves an $\mathcal{O}(n^3)$ cost and predicting the uncertainty, which involves an $\mathcal{O}(n^2)$ cost. However, these are usually approximate techniques because not all the training points are used at once. For example, sparse GP methods use inducing variables, that is, a subset of the training points to improve the time complexity in training from $\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2)$, where m is the number of inducing variables (Hensman et al., 2013). In Chapter 4, we will look into a different approach of partitioning the training set before learning the kernel hyper-parameters.

3.2.3 Useful Properties and Limitations

There are multiple reasons why one prefers to use a GP to model the data, for example, in a regression analysis. Importantly, it lends itself nicely to analytic inference, where the predictive distribution and the marginal likelihood can be computed exactly in closed form. In a Bayesian analysis, this is generally rare (except if we are working with conjugate priors and Gaussian Linear Models, for example). The freedom for choosing a kernel function enables us to improve expressivity of the function being learnt. In fact, we have seen that kernels can be added and/or multiplied together and the resulting kernel function is still a valid kernel. Unlike ad-hoc methods for choosing a model, the marginal likelihood of a GP gives us a principled way to choose a model by marginalising over all the latent variables. In particular, unlike neural network techniques which require optimising for millions of parameters, a GP has just a few hyper-parameters to be estimated, hence not requiring advanced optimisation schemes.

However, one of the main challenges is the slow inference (training) of GP because of the $\mathcal{O}(n^3)$ cost at each step in the optimisation procedure. This therefore limits the application of GP to a few thousands training points, if we choose not to use approximate techniques such as sparse GP explained in the previous section. Furthermore, GP has analytic forms of the predictive distribution and the marginal likelihood, in the case of a Gaussian likelihood. If we have non-Gaussian likelihood, this can be more complicated and might have to resort to numerical methods. While learning the kernel hyper-parameters can be set by maximising the marginal likelihood, another challenge is the need to choose a kernel function, for which there

are many possibilities.

3.3 Summary

In this chapter, we have looked at kernel methods and explained why it is an important branch in Machine Learning. Both regression and classification can be tackled using kernels and in this thesis, since our main focus is on regression, we then elaborate on Gaussian Processes. We first discuss how we would do parametric Bayesian linear regression before motivating the use of Gaussian Process, where we work in function/data space. We also discuss the advantages and disadvantages of using kernel methods. We cover briefly the recent development in this area of research, in particular, scaling Gaussian Process to millions (and possibly billions) of training points.

Chapter 4

SCALABLE EMULATING METHODS FOR KIDS-450

> I have a simple algorithm, which is, wherever you see paid researchers instead of grad students, that's not where you want to be doing research.
>
> *Larry Page*

Emulation is increasingly becoming an important tool used as part of parameter inference in cosmology. In the mid 2000s, Fendt & Wandelt (2007b) used polynomial regression techniques to interpolate CMB power spectra and ever since, we have seen more advanced techniques such as neural network techniques and Gaussian Processes emerging in the cosmology literature. We will cover these techniques in further details in this chapter. Recently, more complicated emulation techniques, based on Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) have been devised for accelerating cosmological simulations (Rodríguez et al., 2018; Mustafa et al., 2019). All these techniques contribute to the application of approximate inference, the main reason being that the likelihood is never exact since the model/theory evaluation itself is not accurate. However, this should not be criticised since there are compelling reasons for using approximate inference in cosmology:

1. the model is very expensive,

2. the model is not totally understood and/or is quite complicated.

The conventional approach is to sample the full posterior distribution of cosmological and nuisance parameters using MCMC-based techniques (as discussed in §2.7) and this requires an accurate evaluation of the model/likelihood which is often an expensive process. In general, a large number of MCMC samples is required to ensure convergence. If the model evaluation itself is costly, then this results in a major computational bottleneck for the overall sampling scheme.

Hence, it is common practice to try various models (and algorithms) and find the one which works best for that very specific problem. Indeed, data analysis in cosmology requires careful selection of models (see Trotta (2008) for further details) and also the choice of the sampling algorithm when dealing with parameter inference. It strictly depends on various trade-off such as speed, complexity and accuracy.

In cosmology and indeed in any other branch of science, it is customary to design *accurate* forward model (simulators) to interpret the data. This often requires painstaking effort to encode all our knowledge about the cosmological model, as well as models for the systematics (which we might not have full control of). The natural question to ask in this scenario is: *'what if we had an approximate model of the world, which is less expensive to compute, to explain the data?'*.

Monte Carlo is one of the approximate techniques used in cosmology. It is argued to be exact in the limit of an infinite number of samples. However, this cannot happen in practice and there Monte Carlo methods are deemed as approximate methods. Moreover, with the goal of limiting the number of forward simulations, *expansion* methods, which depends on techniques such as Taylor expansion, have also been developed. These are generally local procedures, meaning the expansion is performed at a given point in parameter space. Another option to accelerate parameter inference is via *variational inference*. In this case, an optimisation approach is adopted, with the goal being to minimise a distance metric (in particular, the Kullback-Leibler divergence) between the prior and the posterior. Another emerging technique in the Cosmology literature is *likelihood-free* inference, which attempts to minimise the number of forward simulations and to also mitigate the need to define an explicit likelihood function.

In this chapter, we will explore another branch of approximate inference, which deals with scalable emulating methods. All techniques have their own pros and cons and also depend on specific application. We will discuss some of these pros and cons of emulation. In §4.1 we explain the KiDS-450 data. We then cover three algorithms in §4.3.1, the PICO algorithm based on polynomial regression, §4.5, Bayesian Committee Machine based on Gaussian Processes and neural networks in §4.4 respectively. In §4.6, we discuss the results obtained using these three algorithms. In §4.7 we discuss how the algorithms described can be improved and accommodate for scalable inference in future surveys. Finally, in §4.8, we provide a short summary of the different algorithms we have developed in this chapter. This chapter makes use of the KiDS-450 likelihood code*.

*https://bitbucket.org/fkoehlin/kids450_qe_likelihood_public/

4.1 Data

In this section, we briefly cover the data and the model we will use for parameter inference. The detailed explanation for the data reduction process is found in Köhlinger et al. (2017). The final KiDS (Kilo Degree Survey) will cover 1350 deg^2 in four different bands, namely u, g, r and i. For this particular application, the KiDS-450 data consists of around 450 individual tiles each of $\sim$ 1 deg^2. In particular, the positions of the galaxies are given in the usual spherical astronomical coordinate system, right ascension, α and declination, δ. In the data reduction process, instead of using the spherical coordinates, a tangential plane projection, also referred to as gnomonic projection is applied.

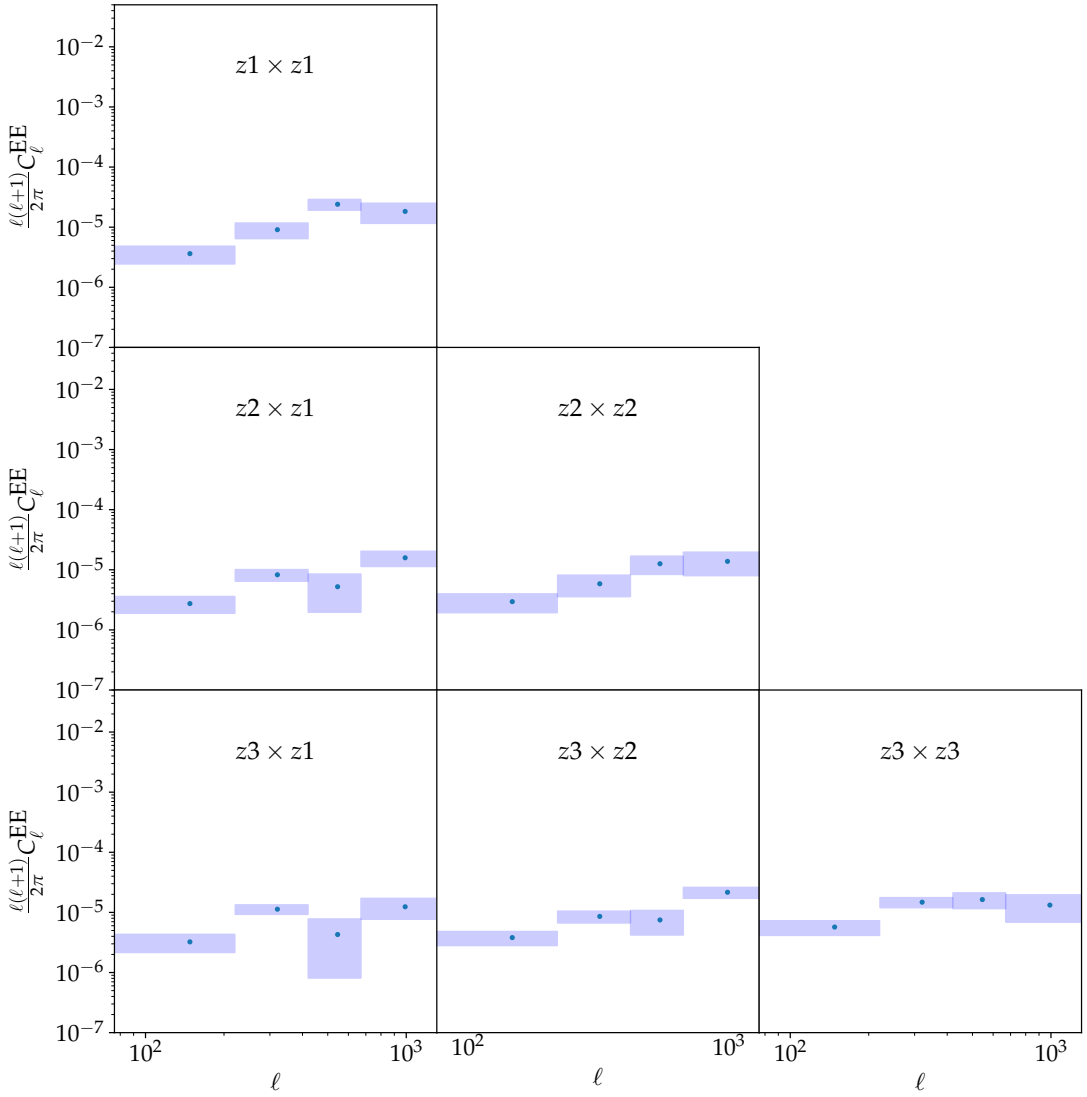


Figure 4.1 – The E-mode band powers (data) used in our inference scheme, similar to the KiDS-450 analysis (Köhlinger et al., 2017). The $\ell-$ranges are as follows: $76 \leq \ell < 220$, $221 \leq \ell < 420$, $421 \leq \ell < 670$ and $671 \leq \ell < 1310$. In particular, the auto-correlation band powers are along the main diagonal ($z1 \times z1$, $z2 \times z2$ and $z3 \times z3$) for the 3 redshift bins $0.10 < z_1 \leq 0.30$, $0.30 < z_2 \leq 0.60$ and $0.60 < z_3 \leq 0.90$. The off-diagonal blocks show the unique cross-correlation band powers. The blue shaded regions indicate the 1σ level errors from the covariance matrix.

Once each sub-patch is pixelized into shear pixels using the plane projection, the shear components per pixel are estimated as

$$g_a(x_c, y_c) = \frac{\sum w_i e_{a,i}}{\sum_i w_i}, \tag{4.1.1}$$

where the label c refers to the centre of each pixel, the index a correspond the shear and ellipticity components, i refers to the i^{th} object in that pixel and the weights, w and ellipticities, e are obtained from the ellipticity measurement using *lensfit* (Miller et al., 2007; Kitching et al., 2008; Miller et al., 2013). Moreover, the distances $r_{ij} = |\boldsymbol{n}_i - \boldsymbol{n}_j|$ and the angles $\varphi = \arctan(\Delta y/\Delta x)$ for each pair of pixels i and j are used in a quadratic estimator algorithm (Hu & White, 2001) to optimise for the band powers, $\mathcal{B}$. The angular position of a pixel is given by $\boldsymbol{n}$.

From Chapter 1, the shear field is a spin-weight 2 quantity with components $\gamma(\boldsymbol{n}, z_\mu) = \gamma_1(\boldsymbol{n}, z_\mu) + i\gamma_2(\boldsymbol{n}, z_\mu)$ corresponding to photometric redshift bin, z_μ. In the flat-sky limit, the Fourier transform of this shear field can be written as

$$\gamma_1(\boldsymbol{n}, z_\mu) \pm i\gamma_2(\boldsymbol{n}, z_\mu) = \int \frac{\mathrm{d}^2\ell}{(2\pi)^2} W_{\text{pix}}(\boldsymbol{\ell}) \left[\kappa^{\text{E}}(\boldsymbol{\ell}, z_\mu) \pm i\kappa^{\text{B}}(\boldsymbol{\ell}, z_\mu)\right] \exp(\pm 2i\varphi_\ell) \exp(i\boldsymbol{\ell}\cdot\boldsymbol{n}), \tag{4.1.2}$$

where φ_ℓ is the angle between the $x-$axis and the two-dimensional vector $\boldsymbol{\ell}$. In the above equation, the shear field is explicitly written in two components, namely the curl-free (E) and the divergence-free (B) components. In the absence of any systematics, most of the cosmological information is contained in the convergence field, κ^{E}. Moreover, the Fourier transform of the pixel window function, W_{pix}, can be written as

$$W_{\text{pix}}(\boldsymbol{\ell}) = j_0\left(\frac{\ell\sigma_{\text{pix}}}{2}\cos\varphi_\ell\right) j_0\left(\frac{\ell\sigma_{\text{pix}}}{2}\sin\varphi_\ell\right), \tag{4.1.3}$$

where σ_{pix} is the side length in radian of a square pixel and $j_0 = \sin x/x$ is the $0^{\text{th}}-$order spherical Bessel function. Given the above formalism, the shear correlation matrix between pixels $\boldsymbol{n}_i$ an $\boldsymbol{n}_j$ and tomographic bins μ and ν can be written as:

$$\mathbf{C}^{\text{signal}} = \langle \gamma_a(\boldsymbol{n}_i, z_\mu)\gamma_b(\boldsymbol{n}_j, z_\nu)\rangle. \tag{4.1.4}$$

Under the assumption that the shear field is Gaussian, the log-likelihood can be written as

$$\log\mathcal{L} = -\frac{1}{2}\boldsymbol{d}^{\text{T}}[\mathbf{C}(\mathcal{B})]^{-1}\boldsymbol{d} - \frac{1}{2}|\mathbf{C}(\mathcal{B})| + \text{constant}, \tag{4.1.5}$$

where the data vector contains elements with $d_{ai\mu} = \gamma_a(\boldsymbol{n}_i, z_\mu)$. The covariance matrix is $\mathbf{C} = \mathbf{C}^{\rm signal} + \mathbf{C}^{\rm noise}$, where $\mathbf{C}^{\rm signal}$ is given by Equation 4.1.4. This quantity depends on the shear power spectra which are approximated by piece-wise constant band powers, $\mathcal{B}$. For a pedagogical treatment of the noise covariance matrix, we refer the reader to Köhlinger et al. (2017). From Equation 4.1.5, an optimisation procedure, such as Newton-Raphson method is adopted to find the root of the equation, that is, $\frac{{\rm d}\ln\mathcal{L}}{{\rm d}\mathcal{B}} = 0$ until $\mathcal{B}_{i+1} = \mathcal{B}_i + \delta\mathcal{B}$ converges to the maximum-likelihood solution.

This set of generated band powers and the noise covariance matrix are publicly available and are used in this analysis. We use 3 tomographic redshift bins, namely, $0.10 < z < 0.30$, $0.30 < z < 0.60$ and $0.60 < z < 0.90$ and the convergence power spectrum is computed in the range $10 < \ell < 4000$. Moreover, we follow Köhlinger et al. (2017) and drop the first, second-to-last and last band powers in our analysis, that is, we use only the band powers corresponding to the following ℓ-ranges: $76 \leq \ell < 220$, $221 \leq \ell < 420$, $421 \leq \ell < 670$ and $671 \leq \ell < 1310$. For a 3-bin tomographic analysis, we have 6 auto- and cross- tomographic power spectra to calculate. The data and covariance matrix for this problem are shown in Figures 4.1 and 4.2 respectively.

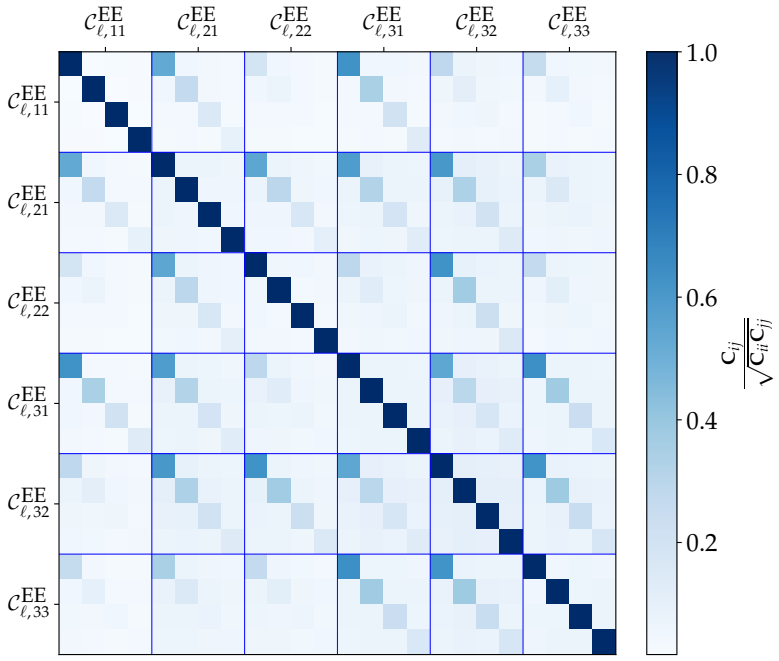


Figure 4.2 – The data correlation matrix for the KiDS-450 analysis. We have ordered the covariance matrix in order of the tomographic labelling ij. Note that we have 4 band powers per tomographic bin, hence 6×4 blocks in the covariance matrix.

There are multiple ways to build an emulating scheme to accelerate parameter inference. The emulator can be built at the level of the power spectra or the band powers. Here we choose to build a GP for each band power, giving 24 GPs. Alternatively, one can emulate the likelihood

directly using the GPs (see Leclercq (2018) and Fendt & Wandelt (2007a)). For power spectrum reconstruction, one can use the PICO method or an alternative, but constrictive, stance is to adopt the approach taken by Habib et al. (2007) to first learn a set of basis functions via Singular Value Decomposition (SVD) and model the resulting weights by a Gaussian Process. However, building an emulator for weak lensing analysis needs to account for systematic effects, but some of these can be included analytically without emulation, resulting in an 8-dimensional GP (6 cosmological parameters, 1 parameter due to baryon feedback, $A_{\rm bary}$ and 1 parameter due to intrinsic alignment, $A_{\rm IA}$) rather than 12 (the eight parameters plus an additional set of 4 nuisance parameters) if we were to emulate the likelihood. See Table 4.2.1 for further details. In the following sections, we will investigate three different algorithms which are used to emulate the band powers.

4.2 Model

The shear field can be decomposed into E and B modes corresponding to the curl-free and divergence-free components. In particular, the convergence field, $\kappa^{\rm E}$ contains most of the cosmological information since $\kappa^{\rm B}$ is negligible in the absence of systematics (Castro et al., 2005). Under this condition, the E-mode lensing power spectrum between tomographic bins i and j is equal to the convergence power spectrum, that is, $C^{\rm EE}_{\ell,ij} = C^{\kappa\kappa}_{\ell,ij}$ and is given, in the Limber approximation (Limber, 1953; Loverde & Afshordi, 2008) by

$$C^{\rm EE}_{\ell,ij} = \int_0^{\chi_H} {\rm d}\chi \, \frac{w_i(\chi) w_j(\chi)}{\chi^2} \, P_\delta \left(k = \frac{\ell + 1/2}{\chi}; \chi \right), \tag{4.2.1}$$

where χ is the comoving radial distance and χ_H is the comoving distance to the horizon. Without the Limber approximation, the integrals can be slow to compute, although faster methods are being developed (Fang et al., 2020). Crucially, the tomographic convergence power spectrum is sensitive to the background geometry and the growth of structure. It depends on the the three-dimensional matter power spectrum, $P_\delta(k;\chi)$ which is a function of redshift (Weinberg et al., 2013). The weight function w_i is

$$w_i(\chi) = \frac{3\Omega_{\rm m} H_0^2}{2c^2} \chi (1+z) \int_\chi^{\chi_H} {\rm d}\chi' n_i(\chi') \left(\frac{\chi' - \chi}{\chi'} \right), \tag{4.2.2}$$

which depends on the lensing kernel. $\Omega_{\rm m}$ is the present matter density, H_0 is the Hubble constant and c is the speed of light. An important quantity is the redshift distribution, $n_i\,(z)\,dz = n_i\,(\chi)\,d\chi$ which is normalised such that

$$\int n_i(\chi) d\chi = 1. \tag{4.2.3}$$

For a weak lensing survey, the data vector consists of the measured shear per pixel for each redshift bin. At this point, in order to extract the shear power spectrum, one can either take a quadratic estimator approach using a maximum-likelihood technique (Bond et al., 1998) or employ, for example, a pseudo-$C(\ell)$ approach (Hinshaw et al., 2007). Alternatively, one can also build a full Bayesian hierarchical model, to infer the full shear power power spectrum (Alsing et al., 2016, 2017). Here, we focus on the tomographic band power spectra, as determined by Köhlinger et al. (2017).

4.2.1 Astrophysical Systematics

Coupled to the E-mode power spectrum are various systematics which we should consider. For example, baryon feedback results in altering the power at high k. Although feedback is not fully understood, it is often parametrized through the bias function, $b^2(k,z)$, such that the modified power spectrum is

$$P_\delta^{\rm mod}\,(k,z) = b^2\,(k,z)\;P_\delta\,(k,z)\,. \tag{4.2.4}$$

As an example, for the KiDS-450 analysis, the following fitting formula from van Daalen et al. (2011) was used

$$b^2\,(k,z) = 1 - A_{\rm bary}\left[A_z e^{(B_z x - {\rm C}_z)^3} - D_z x e^{E_z x}\right], \tag{4.2.5}$$

where $x = \log_{10}(k/1\,{\rm Mpc}^{-1})$ and the other parameters A_z, B_z, C_z, D_z and E_z depend on the scale factor a. Moreover, we must account for intrinsic alignment effects which give rise to a preferred ellipticity orientation. The total lensing power spectrum between two redshift slices is a linear combination of the gravitational lensing (EE), intrinsic alignment (II) and interference (GI) power spectra. Specifically, the II effect is due to correlation of ellipticities in the local environment and contributes positively towards the total lensing spectrum. The second effect, GI, is due to correlation between tidally-stretched foreground galaxies and the shear of background galaxies. The GI term subtracts from the total lensing spectrum. We model the power spectrum, following Köhlinger et al. (2017), as

$$C^{\rm tot}_{\ell,ij} = C^{\rm EE}_{\ell,ij} + A^2_{\rm IA} C^{\rm II}_{\ell,ij} - A_{\rm IA} C^{\rm GI}_{\ell,ij}, \tag{4.2.6}$$

where the II power spectrum, $C^{\rm II}_{\ell,ij}$ and the GI power spectrum, $C^{\rm GI}_{\ell,ij}$ respectively are

$$C^{\rm II}_{\ell,ij} = \int_0^{\chi_H} {\rm d}\chi \, \frac{w_i(\chi) w_j(\chi)}{\chi^2} \, P_\delta \left(k = \frac{\ell + 1/2}{\chi}; \chi \right) \, F^2 \left(\chi \right), \tag{4.2.7}$$

and

$$C^{\rm GI}_{\ell,ij} = \int_0^{\chi_H} {\rm d}\chi \, \frac{w_i(\chi) n_j(\chi) + w_j(\chi) n_i(\chi)}{\chi^2} \, P_\delta \left(k = \frac{\ell + 1/2}{\chi}; \chi \right) \, F(\chi) \tag{4.2.8}$$

where

$$F(\chi) = C_1 \rho_{\rm crit} \frac{\Omega_m}{D_+(\chi)} \tag{4.2.9}$$

and $A_{\rm IA}$ is a free parameter to be inferred during sampling. This allows for the flexibility of rescaling the otherwise fixed normalisation value, $C_1 = 5 \times 10^{-14} \; h^{-2} \, {\rm M}_\odot^{-1} \, {\rm Mpc}^3$. $\rho_{\rm crit}$ is the critical density of the Universe while $D_+(\chi)$ refers to the linear growth factor normalized to unity today.

4.2.2 Priors

The priors adopted for sampling the posterior is given in Table 4.2.1 and we use the limits to define the bounds of our emulation scheme. In order to build the emulator, we focus on the most expensive part of the likelihood evaluation. Hence, we choose to emulate the band powers directly. To be more specific, we emulate the band powers arising due to the 3 different types of weak lensing power, EE, GI and II as explained in §4.2.

The observed shear is generally a biased estimator of the true shear, γ and is parametrised in terms of the multiplicative bias correction, m and the additive bias, c as

$$\gamma_{\rm obs} = (1 + m)\gamma + c, \tag{4.2.10}$$

and the multiplicative bias arises mainly due to the effect of the pixel noise in the measurement of the galaxy ellipticities. In order to account to the $m-$correction, it is applied to both the shear power spectrum calculation and the covariance matrix and m is marginalised over in the likelihood analysis. c is usually very tiny and is fixed to zero. Hildebrandt et al. (2017) also found that the multiplicative $m-$ correction to be small and they reported

$$m = [-0.0131, \, -0.0107, \, -0.0087, \, -0.0217] \pm 0.01.$$

Table 4.2.1 – Set of cosmological and systematic parameters which are used in the emulating scheme. The first set will be referred to as $\boldsymbol{\theta}$ and the remaining ones as $\boldsymbol{\beta}$ and we include $A_{\rm bary}$ as part of the emulating scheme. The prior range is also shown in the last column.

Definition	Symbol	Prior
CDM density	$\Omega_{\rm cdm}h^2$	$\mathcal{U}[0.01, 0.50]$
Baryon density	$\Omega_{\rm b}h^2$	$\mathcal{U}[0.019, 0.026]$
Scalar spectrum amplitude	$\ln(10^{10}A_{\rm s})$	$\mathcal{U}[1.70, 5.00]$
Scalar spectral index	n_s	$\mathcal{U}[0.70, 1.30]$
Hubble parameter	h	$\mathcal{U}[0.64, 0.82]$
Neutrino mass (eV)	Σm_ν	$\mathcal{U}[0.06, 10.0]$
Free amplitude baryon feedback parameter	$A_{\rm bary}$	$\mathcal{U}[0.0, 10.0]$
Intrinsic alignment parameter	$A_{\rm IA}$	$\mathcal{U}[-6.0, 6.0]$
Free amplitude (bin 1)	A_1	$\mathcal{U}[-1.0, 1.0]$
Free amplitude (bin 2)	A_2	$\mathcal{U}[-1.0, 1.0]$
Free amplitude (bin 3)	A_3	$\mathcal{U}[-1.0, 1.0]$
Multiplicative bias	m	$\mathcal{U}[-0.033, 0.007]$

We take a similar approach as Köhlinger et al. (2017) and use a flat prior, $2\sigma_m$ centred on the fiducial value $m_{\rm fid(z_1)}$ for the first redshift bin, where $\sigma_m = 0.01$. In short, $-0.0131 - 0.02 = -0.033$ and $-0.0131 + 0.02 = 0.007$ are the lower and upper limits of the prior on the parameter m (see Table 4.2.1 for the prior range adopted).

Moreover, to account for other systematics, the following technique was adopted by Köhlinger et al. (2017). It is argued that the quadratic estimator algorithm, covered in §4.1 requires precise and accurate characterisation of the noise calculation in the data. To take this into consideration in the likelihood analysis, the following parametric model was adopted:

$$p_{\rm noise}(\ell, z_i) = A_i \frac{\sigma^2_{\hat{\gamma}(z_i)}}{n_{\rm eff}(z_i)}, \tag{4.2.11}$$

where $n_{\rm eff}$ is the effective number of galaxies per arcmin^2 and $\sigma_{\hat{\gamma}(z_i)}$ is the dispersion of the intrinsic ellipticity distribution. We refer the reader to Table 2 in Köhlinger et al. (2017) for further details on these values. The amplitude, A_i determines the strength of the excess noise in the autocorrelation power spectra and are also marginalised over in the likelihood analysis.

4.3 Polynomial Regression

In the very simple case, one can define a set of basis functions and fit these functions to the observed data. Suppose we have a training set $\{\boldsymbol{x}_i, y_i\}_{i=1:N}$ where $\boldsymbol{x}$ refers to the inputs and y refers to the response or target. We can exploit this training set to learn a fitting function which will enable us to make prediction at a given test point, $\boldsymbol{x}_*$. In very simple case, we can consider polynomial curve fitting:

$$\begin{aligned} f(\boldsymbol{x}, \boldsymbol{\beta}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_d x_d \\ + \beta_{d+1} x_{d+1}^2 + \beta_{d+2} x_{d+2}^2 + \ldots + \beta_{2d} x_{2d}^2 \\ + \ldots + \beta_{Md} x_{Md}^M, \end{aligned} \tag{4.3.1}$$

where M is the order of the polynomial and β_j refers to the regression coefficient. The above equation can be neatly written in matrix format as:

$$f(\boldsymbol{x}, \boldsymbol{\beta}) = \mathbf{A}\boldsymbol{\beta} \tag{4.3.2}$$

where $\mathbf{A} \in \mathbb{R}^{N \times (1+Md)}$ is a design matrix whose columns and rows contain each basis function evaluated at each input. For example, if we have a 1D fitting function and a single input x, then the design matrix $\mathbf{A}$ is

$$\mathbf{A} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}.$$

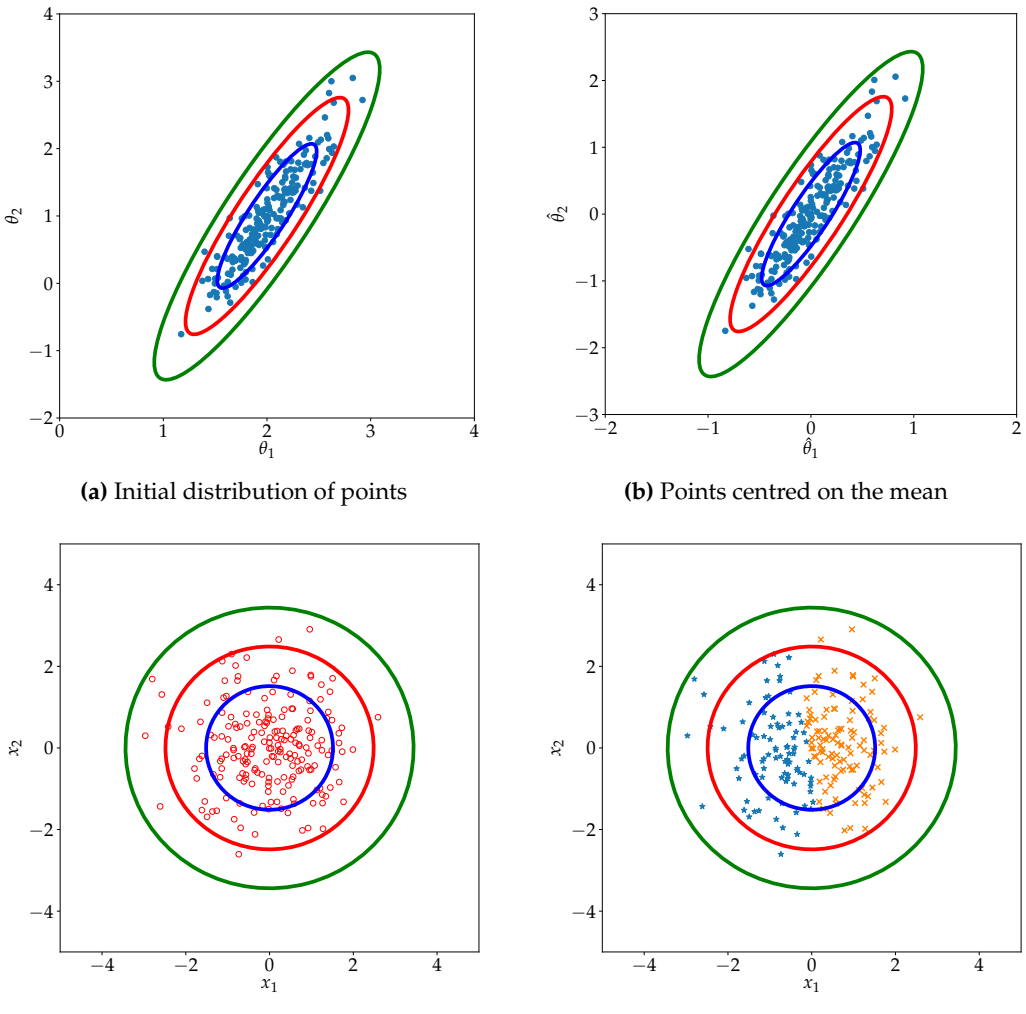
For simplicity, we will also assume a noise covariance matrix, $\boldsymbol{\Sigma}$, and in the case of noise-free regression, a small jitter term can be assumed, that is, $\boldsymbol{\Sigma} = \sigma^2 \mathbb{I}$ such that the fitting model can be re-written as $f(\boldsymbol{x}, \boldsymbol{\beta}) = \mathbf{A}\boldsymbol{\beta} + \boldsymbol{n}$, where $\boldsymbol{n}$ is the noise term.

There are different methods for learning the regression coefficients, $\boldsymbol{\beta}$. If we take a standard frequentist approach (see Chapter 2 for further details), we then obtain the Maximum Likelihood Estimate (MLE) as

$$\begin{aligned} \boldsymbol{\beta}_{\text{MLE}} &= (\mathbf{A}^{\text{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A})^{-1}\mathbf{A}^{\text{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{y} \\ \text{cov}(\boldsymbol{\beta}_{\text{MLE}}) &= (\mathbf{A}^{\text{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A})^{-1}. \end{aligned} \tag{4.3.3}$$

On the other hand, we can also adopt a Bayesian approach and place a prior on the regression coefficients, $\boldsymbol{\beta}$. In order to derive an expression for the Maximum a Posteriori (MAP), we assume a Gaussian prior on the regression coefficients, $\boldsymbol{\beta}$, that is, $p(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}, \mathbf{C})$. We obtain a slightly modified expression for the MAP compared to the MLE:

$$\begin{aligned} \boldsymbol{\beta}_{\text{MAP}} &= (\mathbf{C}^{-1} + \mathbf{A}^{\text{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A})^{-1}(\mathbf{A}^{\text{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{y} + \mathbf{C}^{-1}\boldsymbol{\mu}) \\ \text{cov}(\boldsymbol{\beta}_{\text{MAP}}) &= (\mathbf{C}^{-1} + \mathbf{A}^{\text{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A})^{-1} \end{aligned} \tag{4.3.4}$$



(a) Initial distribution of points

(b) Points centred on the mean

(c) Pre-whitened input parameters

(d) $k-$means with 2 clusters

Figure 4.3 – Illustration of the various steps in the pre-whitening procedure. Let us assume we have a 2D input parameter space, shown in the upper left panel. It is first centred on the mean, as shown in the upper right panel, followed by transforming the inputs such that they are uncorrelated, shown in the lower left panel. Finally, depending on the number of clusters specified (in this example, 2, shown by $\times$ and $\star$), the set of input parameters can be partitioned into different disjoint regions.

Placing a prior on the regression coefficients is analogous to using *regularisation* in machine learning concepts. The regulariser, $\frac{\lambda}{2}\|\boldsymbol{\beta}\|^2$, is used to control over-fitting, that is, we do not wish to consider models with a large number of basis functions. Hence, the regulariser is used to penalise extra terms in the model. If we consider minimising the *error function* only, this is given by

$$J(\boldsymbol{\beta}) = \frac{1}{2}\sum_{n=1}^{N}(f(\boldsymbol{x}_n, \boldsymbol{\beta}) - y_n)^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|^2 \tag{4.3.5}$$

In the next section, we will look into one application of the polynomial regression used as an emulator in cosmology. It was used in the context of CMB data analysis and was among the first techniques to introduce machine learning in the cosmology community. We first elaborate on this technique, before using it to emulate weak lensing band powers for the KiDS-450 data.

4.3.1 The PICO algorithm

PICO (Parameters for the Impatient Cosmologist) is one amongst the early techniques to accelerate parameter inference in the analysis of CMB and since then, has led to the emancipation of emulating techniques in Cosmology (Fendt & Wandelt, 2007b). PICO was initially used to emulate the CMB power spectra and it was argued to be 3000 times faster than CAMB and hence 2000 times faster than the WMAP 3 likelihood code.

The training set of PICO consisted of 60000 8D models from a converged MCMC run of the WMAP first-year. The parameters included were the baryon density, $\Omega_{\rm b}$, the cold dark matter density, $\Omega_{\rm cdm}$, the dark energy density, Ω_Λ, the Hubble's constant, H_0, the scalar spectral index, n_s, the optical depth since reionisation, τ and the normalisation of the power spectra, A_s.

At the heart of the PICO algorithm is polynomial regression, that is, to learn a function that maps the cosmological parameters, $\boldsymbol{\theta}$ to their respective power spectra, $\boldsymbol{y}$. Note that for CMB, we have three different power spectra, namely, the scalar TT, TE and EE power spectra. The goal is to find the regression coefficients, $\boldsymbol{\beta}$ which will minimise the squared error over the training set, that is,

$$R^2 = \sum_{n=1}^{N}[f(\boldsymbol{\theta}_n) - y_n]^2 \tag{4.3.6}$$

and writing the polynomial function in matrix format, that is, $\boldsymbol{f} = \mathbf{A}\boldsymbol{\beta}$ where $\mathbf{A}$ contains basis functions of $\boldsymbol{\theta}$, the solution to the above equation is analytic and the regression coefficients are given by $\boldsymbol{\beta}_{\rm sol} = (\mathbf{A}^{\rm T}\mathbf{A})^{-1}\mathbf{A}^{\rm T}\boldsymbol{y}$. One can substitute the polynomial functions by other basis

functions such as Chebyshev or Legendre polynomial functions.

While the above procedure is simple and straightforward, Fendt & Wandelt (2007b) found that the interpolation scheme fails to model the power spectra over the whole parameter space, the curse of dimensionality being the reason behind. Moreover, polynomial fitting procedures are know to be be *global* fitting methods and are susceptible to over-fitting. To alleviate these issues, two tricks were adopted:

1. a pre-whitening step is applied to the input cosmological parameters, $\boldsymbol{\theta}$ and
2. the training set was divided into M local, disjoint clusters using a clustering algorithm such as $k-$means algorithm.

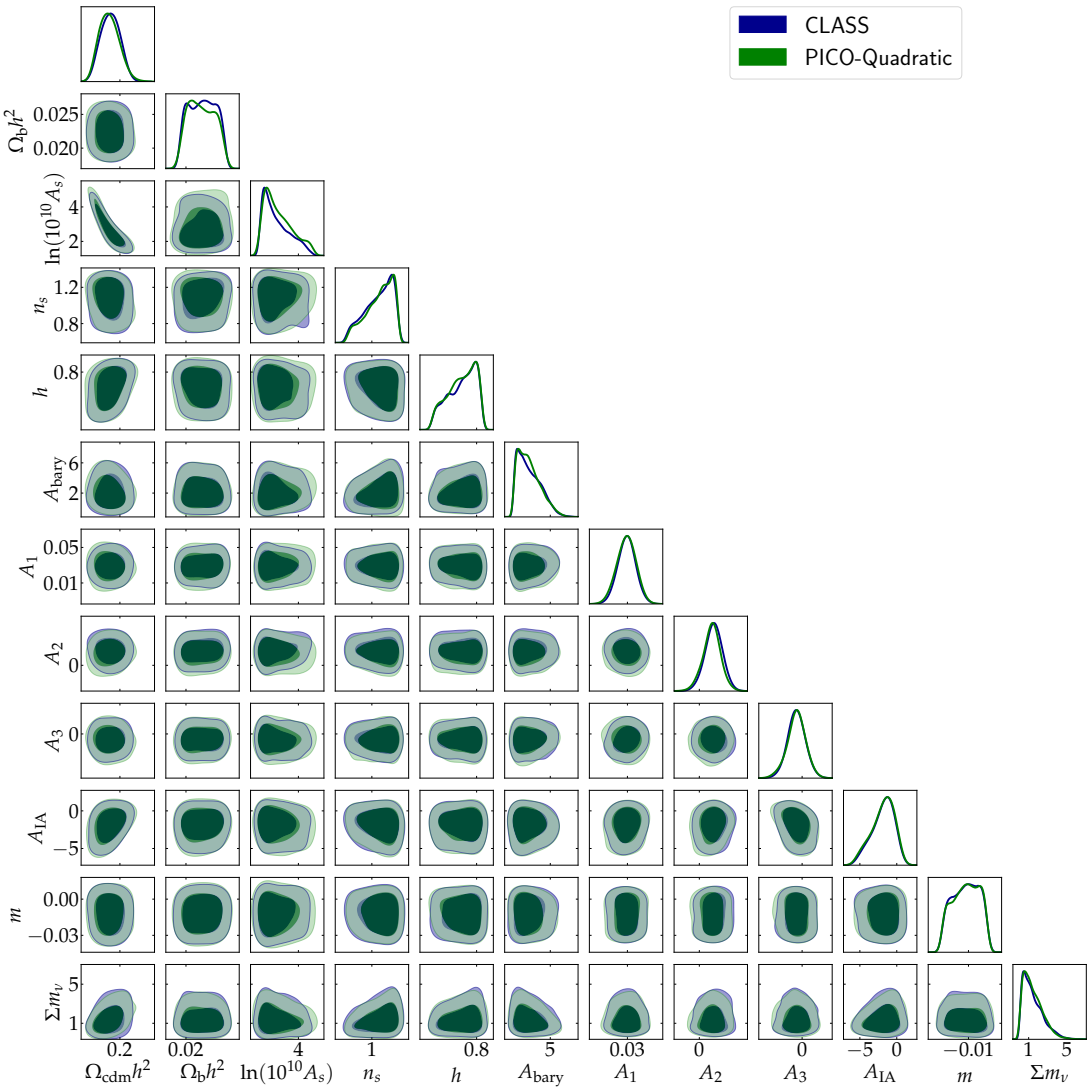


Figure 4.4 – The marginalised posterior distribution of the cosmological and nuisance parameters using the PICO algorithm. In particular, for the PICO algorithm, we used 18000 training points, 120 clusters and a quadratic function to interpolate each band power.

The pre-whitening step can be understood as follows. The $N \times d$ matrix of the (input) training set are first centred on zero. The sample covariance, $\mathbf{C}_{\boldsymbol{\theta}}$ of this translated training set

can be diagonalised as: $\mathbf{C}_{\boldsymbol{\theta}} = \mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}$. $\mathbf{U}$ is a $d \times d$ orthonormal matrix and $\mathbf{D}$ is a diagonal $d \times d$ matrix consisting of the (necessarily positive) eigenvalues. The transformation matrix which whitens $\boldsymbol{\theta}$ is then $\mathbf{U}\mathbf{D}^{\frac{1}{2}}$, such that the transformed input covariates are $\boldsymbol{x} = \mathbf{U}\mathbf{D}^{\frac{1}{2}}\boldsymbol{\theta}$ and the covariance of $\boldsymbol{x}$ is the identity matrix.

Next, the points in the pre-whitened new basis are partitioned into different disjoint regions using a clustering algorithm. For example, if we choose the $k-$means algorithm, it can be intuitively understood as follows. An initial set of k mean vectors are randomly generated and each point within the training set is associated with the nearest mean. The mean is updated by the centroid of each of the k clusters. The former and latter are repeated until convergence is achieved. An illustration of the pre-whitening (and the clustering) step for a 2D input parameter is shown in Figure 4.3.

4.3.2 Application to the KiDS-450 Data

For the KiDS-450 analysis, we generate $N = 18000$ training points (see Table 4.2.1) with the following input parameters:

$$\boldsymbol{\theta} = \left[\Omega_{\mathrm{cdm}}h^2,\ \Omega_{\mathrm{b}}h^2,\ \ln(10^{10}A_{\mathrm{s}}),\ n_s,\ h,\ A_{\mathrm{bary}},\ \Sigma m_\nu\right]$$

and their prior range is given in Table 4.2.1. The input dimension is $d = 7$. In particular, we first run a short MCMC chain consisting of 15000 MCMC samples to learn an approximate Gaussian posterior distribution of the parameters. The training points are sampled from this Gaussian distribution, centred on the estimated mean and the covariance is set to 4 times the approximate covariance matrix estimated from the MCMC run. Note that this step can be substituted by an iterative optimisation algorithm, hence the number of forward simulations would be much less than 15000. However, we found that the fact that the KiDS-450 data is not very informative of the parameters, the iterative algorithm we tried failed to converge to the maximum likelihood. Moreover, the samples (training points) which lie outside the uniform prior range (see Table 4.2.1) are rejected.

Next, we run the forward model at these 18000 training points, recording the values of the different band powers (EE, GI and II), giving us a set of 24×3 band powers at each training point. With a standard polynomial fitting approach, we have $\boldsymbol{y} = \mathbf{A}(\boldsymbol{\theta})\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{y} \in \mathbb{R}^N$ represent one set of band powers, $\boldsymbol{\theta} \in \mathbb{R}^{N\times d}$, $\boldsymbol{\epsilon}$ is the noise term (in this case, we are considering noise-free regression, therefore $\boldsymbol{\epsilon} = 0$) and $\mathbf{A}$ contains the basis functions of $\boldsymbol{\theta}$.

We now apply the PICO algorithm described previously to the band powers. In particular,

we first pre-whiten the input parameters and we choose $M = 120$ clusters for the partitioning procedure. We use a $k-$mean algorithm to perform the clustering. We then fit a quadratic function, with basis functions, $[1, \boldsymbol{\theta}, \boldsymbol{\theta}^2]$ to the generated band powers. The PICO solutions, that is, calculating the regression coefficients, $\boldsymbol{\beta}$, took approximately 30 seconds. This is quick because the inverse $(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}$ is done in weight space and the resulting matrix from $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is just of size 15.

The solutions, $\boldsymbol{\beta}_{\mathrm{sol}}$ are stored and predictions can be made for a given test point, $\boldsymbol{\theta}_*$, using $y_* = \boldsymbol{A}_*\boldsymbol{\beta}_{\mathrm{sol}}$, where $\mathbf{A}_*$ is the new set of basis function at the test point. Hence, this module can be connected to an MCMC sampler to make predictions for each MCMC sample. We use EMCEE (Foreman-Mackey et al., 2013) to sample the full posterior distribution using CLASS and the PICO algorithm. For 24 independent chains, each with 15 000 MCMC samples, the time taken by CLASS is $\sim$ 44 hours while PICO took $\sim$ 150 minutes to generate the same number of MCMC samples. The marginalised 1D and 2D distribution is shown in Figure 4.4. The posterior result obtained from CLASS is shown in blue while the posterior using the PICO algorithm is shown in green and they agree quite well with each other.

4.4 Neural Network

Another class of algorithm which is able to learn from data is neural network (NN). In this case, the number of basis functions are fixed but they are allowed to be adaptive, that is, during the learning procedure, the basis function evolves as the parameter values are updated during the training phase. Neural networks have many different applications, with regression and classification being the most common ones. In the following sections, we will briefly highlight the main concepts behind the *feed-forward neural network*, which we use for the emulation scheme.

4.4.1 Introduction to Neural Networks

In the previous section, we have looked at polynomial regression and it can be summarised as

$$f(\boldsymbol{x}, \boldsymbol{w}) = g\left(\sum_{j=1}^{M} w_j \phi_j(\boldsymbol{x})\right) \tag{4.4.1}$$

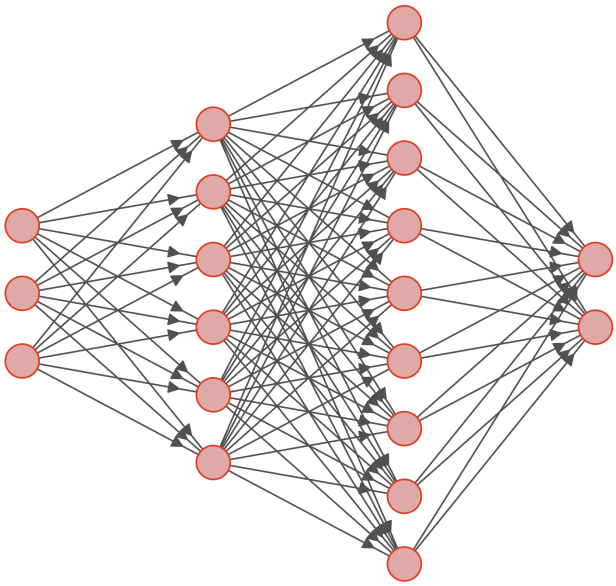


Figure 4.5 – Example of a feed-forward neural network architecture consisting of 3 inputs, 6 nodes in the first layer, 9 nodes in the second layer and 2 outputs. Each intermediary node between the inputs and outputs represent a neuron (following a brain model) and each connection (line joining any two nodes) carries a certain weight, w. These weights are learnt (optimised) depending on the loss function defined, which in itself depends on the problem we want to solve.

where we are using $\boldsymbol{w}$ to denote the regression coefficients (instead of $\boldsymbol{\beta}$) and for polynomial regression, $g(\cdot)$ is just identity. The neural network methodology extends this concept and makes the basis function more flexible since they are now dependent on parameters which are adjusted when training the network. Each basis function is a result of the application of a non-linear function (activation function) on a linear combination of the inputs.

Hence, a neural network can be summarised as a model with a series of non-linear transformations between the inputs and the outputs. For example, in Figure 4.7, we have 2 inputs and 2 outputs, with three *layers* consisting of 6 and 9 nodes respectively in between. There is an additional input (hence a total of 3 inputs) whose value is set to $x_0 = 1$. This then allows one to model for a fixed offset in the data, w_0, and is often referred to as the *bias* weights (parameters).

Before we move to the Mathematical details of the neural network, we will use the following indices to denote important quantities throughout this short neural network explanation. n denotes the n^{th} training point. In total, we have N training points. For the first layer, we can write the linear combination of the inputs to the node, a_j, as

$$a_j = \sum_{i=0}^{D} w_{ji} x_i, \tag{4.4.2}$$

followed by the application of a non-linear activation function, $h(\cdot)$

$$z_j = h(a_j), \tag{4.4.3}$$

For the second layer, we have a similar methodology, that is,

$$\begin{aligned} a_k &= \sum_{j=0}^{M} w_{kj} z_j \\ z_k &= h(a_k), \end{aligned} \tag{4.4.4}$$

where $J+1$ is the number of neurons in the first layer, and for the third layer, we have

$$\begin{aligned} a_\ell &= \sum_{k=0}^{Q} w_{\ell k} z_k \\ z_\ell &= \sigma(a_\ell), \end{aligned} \tag{4.4.5}$$

where $Q+1$ is the number of neurons in the second layer. For a regression problem, suppose we have L outputs, each output, y_ℓ is simply given by z_ℓ, that is,

$$f_\ell = z_\ell. \tag{4.4.6}$$

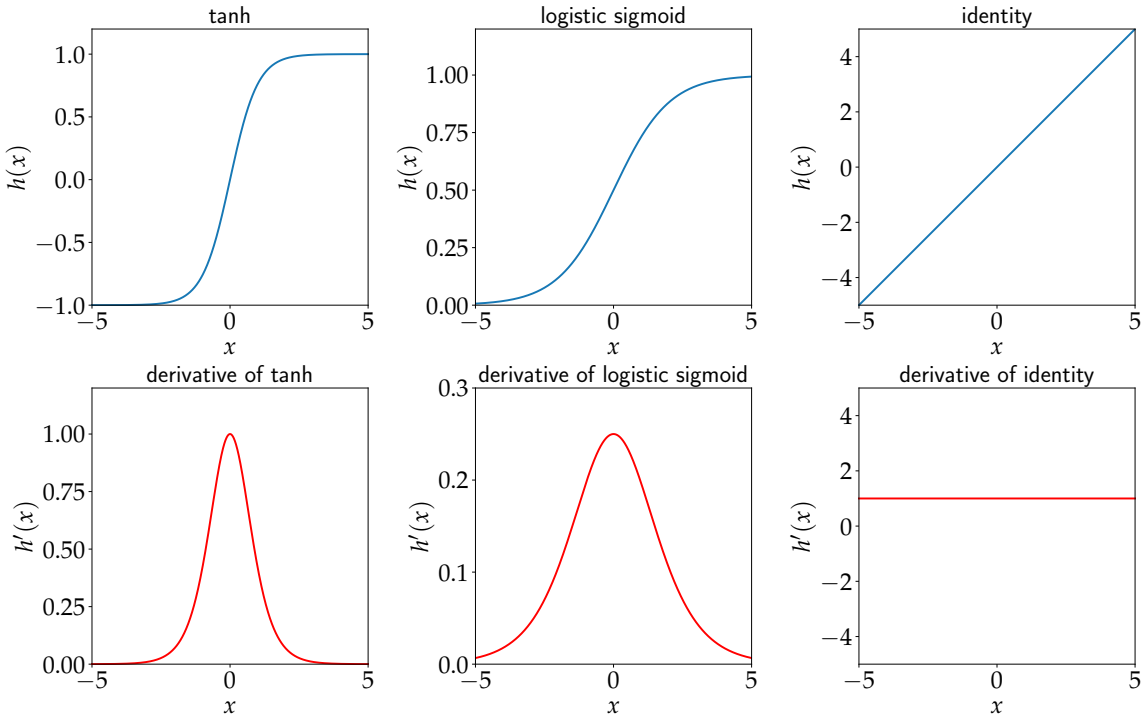


Figure 4.6 – The top panel shows three different types of activation functions, with the upper left and upper middle one showing non-linear activation functions. The bottom panel shows their corresponding gradient function. The `tanh` and `sigmoid` are two common types of activation functions used in the hidden layers of a neural network.

Note that we have two types of activation function used in this process. $h(\cdot)$ is generally a non-linear activation function such as the logistic sigmoid or tanh while the activation function in the last layer, $\sigma(\cdot)$ is simply the identity. The latter differs in different applications,

for example, if we are doing a binary classification, then the activation function is a logistic sigmoid function. This ensures that the output in this case is a probability between 0 and 1. Importantly, the different activation functions have an analytic expression for the gradient calculations. There exist various other activation functions, such as `ReLU`, `ELU`, `Leaky ReLU` and others which are used in building neural network architectures. See Figure 4.6 for an illustration of the different activation functions mentioned.

The feed-forward neural network described above is able to approximate functions. In fact, they are referred to as *universal approximators*. For example, a two-layer neural network is able to approximate any continuous function, given a sufficiently large number of hidden units. Once the network architecture is setup, the next step is to learn (optimise) the parameters (weights and biases) of the neural network. We elaborate more on this in the next section.

4.4.1.1 Training

In the case of regression, where a network is simply learning a mapping between the inputs and the outputs, a straightforward procedure is to define a loss function, which in this case is the error function:

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N} \|\boldsymbol{f}(\boldsymbol{x}_n, \boldsymbol{w}) - \boldsymbol{y}_n\|^2. \tag{4.4.7}$$

As described in §4.3, we can start from defining a Gaussian likelihood and derive an expression for the log-likelihood, which will turn out to be quite similar to the error function above. Moreover, one can also take a Bayesian approach and place a prior on $\boldsymbol{w}$ and this is analogous to using a regulariser on the parameters of the neural network.

In general, there is no close-form solution for $\boldsymbol{w}$ and we have to use some iterative schemes to learn $\boldsymbol{w}$ to the point where the gradient of the error function approximately vanishes, that is, $\nabla J(\boldsymbol{w}) \approx 0$. The common approach is to initialise $\boldsymbol{w}$ to some initial values, $\boldsymbol{w}^{(0)}$ and iteratively update $\boldsymbol{w}$, that is,

$$\boldsymbol{w}^{(p+1)} = \boldsymbol{w}^{(p)} + \Delta\boldsymbol{w}^{(p)}, \tag{4.4.8}$$

where p is the iteration step. For now, if we assume we can compute the gradient of the loss function with respect to $\boldsymbol{w}$, the simplest approach to update the $\boldsymbol{w}$ iteratively is via gradient descent, that is,

$$\boldsymbol{w}^{(p+1)} = \boldsymbol{w}^{(p)} - \eta \nabla J(\boldsymbol{w}^{(p)}), \tag{4.4.9}$$

where $\eta > 0$ is the *learning rate*. Note that in this procedure, we are using the whole training set at once. This might not be an optimal approach and hence, *online* gradient descent methods have been proposed, which have been shown to be more reliable. For example, following the maximum likelihood approach and assuming each data point is independent from each other, we can write the loss function as

$$J(\boldsymbol{w}) = \sum_{n=1}^{N} J_n(\boldsymbol{w}) \tag{4.4.10}$$

and the weight is updated based on each data point at a time, that is,

$$\boldsymbol{w}^{(p+1)} = \boldsymbol{w}^{(p)} - \eta \nabla J_n(\boldsymbol{w}^{(p)}). \tag{4.4.11}$$

Note that we can also update the parameters, $\boldsymbol{w}$, based on batches of the training set. An important aspect of online gradient descent is that it can handle redundancy in the data more efficiently and it can also avoid cases of local minima when training a neural network (Bishop, 2006). In the next section, we will look into an effective technique for updating the gradients of $\boldsymbol{w}$ through each layer of a neural network.

4.4.1.2 Back-propagation

Back-propagation is a technique for calculating the gradient of the loss function, $E(\boldsymbol{w})$ with respect to each parameter of a neural network by alternatively sending information forwards and backwards. This process is known as *back-propagation error* or simply *back-prop*. To illustrate how the back-propagation algorithm works, we will consider a single training point, n, such that the loss due to this training point can be written as:

$$J_n(\boldsymbol{w}) = \frac{1}{2} \sum_{\ell} (f_\ell - y_\ell)^2. \tag{4.4.12}$$

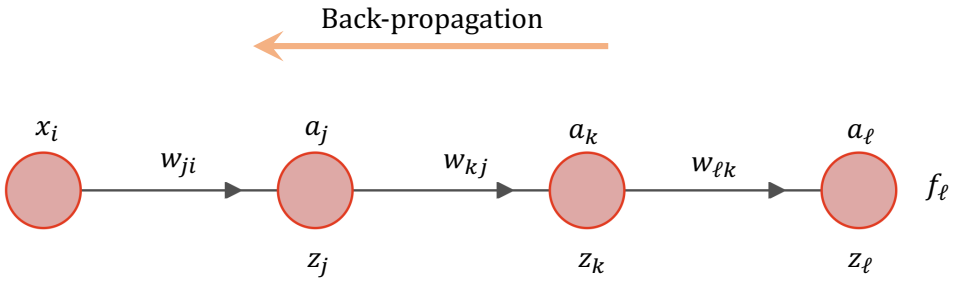


Figure 4.7 – In this figure, we show a sequence of connections from one input,x_i to one output, y_ℓ. In between these two, we have two hidden neurons. The input to the first neuron is a linear combination of the input, x_i and the output from it is a non-linear function, z_j. The same idea applies to the second hidden neuron.

Throughout this section, we will use Figure 4.7 as an example to go through the different steps of the back-propagation procedure. We will assume the `tanh` activation function for the hidden units, that is,

$$\begin{aligned} h(a) &= \tanh(a) \\ h'(a) &= 1 - h^2(a), \end{aligned} \tag{4.4.13}$$

In the general case, suppose we want to calculate the derivative of the loss with respect to a specific parameter, w_{ji}. This can be done using chain rule to give

$$\frac{\partial J_n}{\partial w_{ji}} = \frac{\partial J_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} \tag{4.4.14}$$

and we will introduce the following useful notation:

$$\delta_j = \frac{\partial J_n}{\partial a_j} \tag{4.4.15}$$

and from the second layer onwards, we have

$$\frac{\partial a_j}{\partial w_{ji}} = z_i. \tag{4.4.16}$$

Hence Equation 4.4.14 can be written as

$$\frac{\partial J_n}{\partial w_{ji}} = \delta_j z_i. \tag{4.4.17}$$

We will now use the above formalism to provide an example of how the gradients are calculated for the neural network shown in Figure 4.7. The inputs to each unit and the activations are summarised in §4.4.1. Starting from the last layer, we have

$$\begin{aligned} \delta_\ell &= \frac{\partial J_n}{\partial a_\ell} \\ &= f_\ell - y_\ell \end{aligned} \tag{4.4.18}$$

and this is because, for linear regression, we have $y_\ell = a_\ell = z_\ell$. Next,

$$\begin{aligned} \delta_k &= \frac{\partial J_n}{\partial a_k} \\ &= \sum_\ell \frac{\partial J_n}{\partial a_\ell} \frac{\partial a_\ell}{\partial a_k} \\ &= (1 - z_k^2) \sum_\ell w_{\ell k} \delta_\ell, \end{aligned} \tag{4.4.19}$$

and finally, we have

$$\begin{aligned} \delta_j &= \frac{\partial J_n}{\partial a_j} \\ &= \sum_j \frac{\partial J_n}{\partial a_k} \frac{\partial a_k}{\partial a_j} \\ &= (1 - z_j^2) \sum_\ell w_{kj} \delta_k. \end{aligned} \tag{4.4.20}$$

We can also derive the gradients of the loss function with respect to a parameter in each layer, that is,

$$\begin{aligned} \frac{\partial J_n}{\partial w_{ji}} &= \frac{\partial J_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} \\ &= \delta_j x_i, \end{aligned} \tag{4.4.21}$$

$$\begin{aligned} \frac{\partial J_n}{\partial w_{kj}} &= \frac{\partial J_n}{\partial a_k} \frac{\partial a_k}{\partial w_{kj}} \\ &= \delta_k z_j \end{aligned} \tag{4.4.22}$$

and

$$\begin{aligned} \frac{\partial J_n}{\partial w_{\ell k}} &= \frac{\partial J_n}{\partial a_\ell} \frac{\partial a_\ell}{\partial w_{\ell k}} \\ &= \delta_\ell z_k. \end{aligned} \tag{4.4.23}$$

For batch methods, we can re-write all equations by summing over all patterns, for example,

$$\frac{\partial J}{\partial w_{ji}} = \sum_n \frac{\partial J_n}{\partial w_{ji}}. \tag{4.4.24}$$

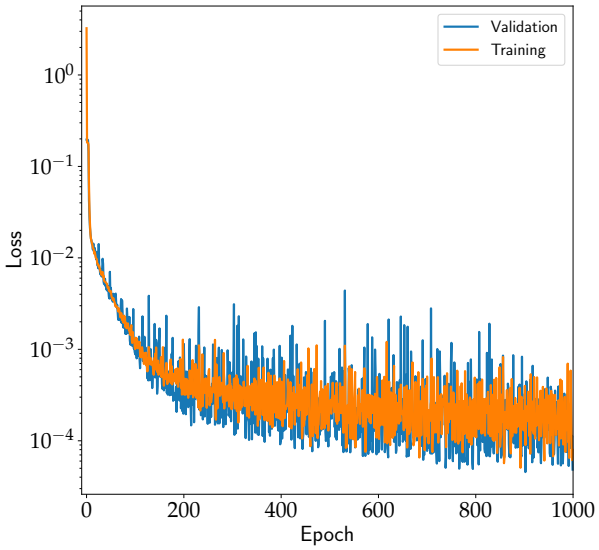


Figure 4.8 – Training and validation loss curve for the neural network employed in this work. It is expected that both loss functions to decrease with the number of epoch. At the beginning of the optimisation procedure, the loss function is high because the weights and biases are not optimised at all. During the optimisation procedure, as the weights and biases are progressively updated via gradient descent, the loss also decreases.

The above formalisms can further be extended in order to obtain the first derivative of the output with respect to the input, that is, the *Jacobian matrix*. Moreover, various approximating schemes have also been developed to estimate the *Hessian matrix* (second derivatives) as well. We refer the reader to Bishop (2006) for further details. Now that we have covered the theory of neural network, in the next section, we show how we can use it to build an emulator for the KiDS-450 data.

4.4.2 Application to KiDS-450 Data

In our application, we use 20 000 points drawn in a similar fashion as in the PICO approach (see §4.3.1). The setup of the neural network is as follows. The number of epochs, which is essentially the number of times that the neural network will pass through the entire dataset, is fixed to 1000 while the fraction of the validation split is 0.1. This then ensures that the network is trained on 18 000 training points, with the remaining set of 2000 points kept as the validation set. The batch size is fixed at 128. The neural network architecture consists of the following number of units [16, 32, 64, 128, 256], organised in a sequential manner. The input and output are of sizes 7 and 72, corresponding to the dimension of the input parameters and number of band powers respectively. We use the Adam optimiser (Kingma & Ba, 2014) with a learning

rate of 0.001. The time taken to train this network is about 1.5 minutes and the behaviour of the loss function is shown in Figure 4.8. The routine is built using `keras` and `tensorflow` (Abadi et al., 2015).

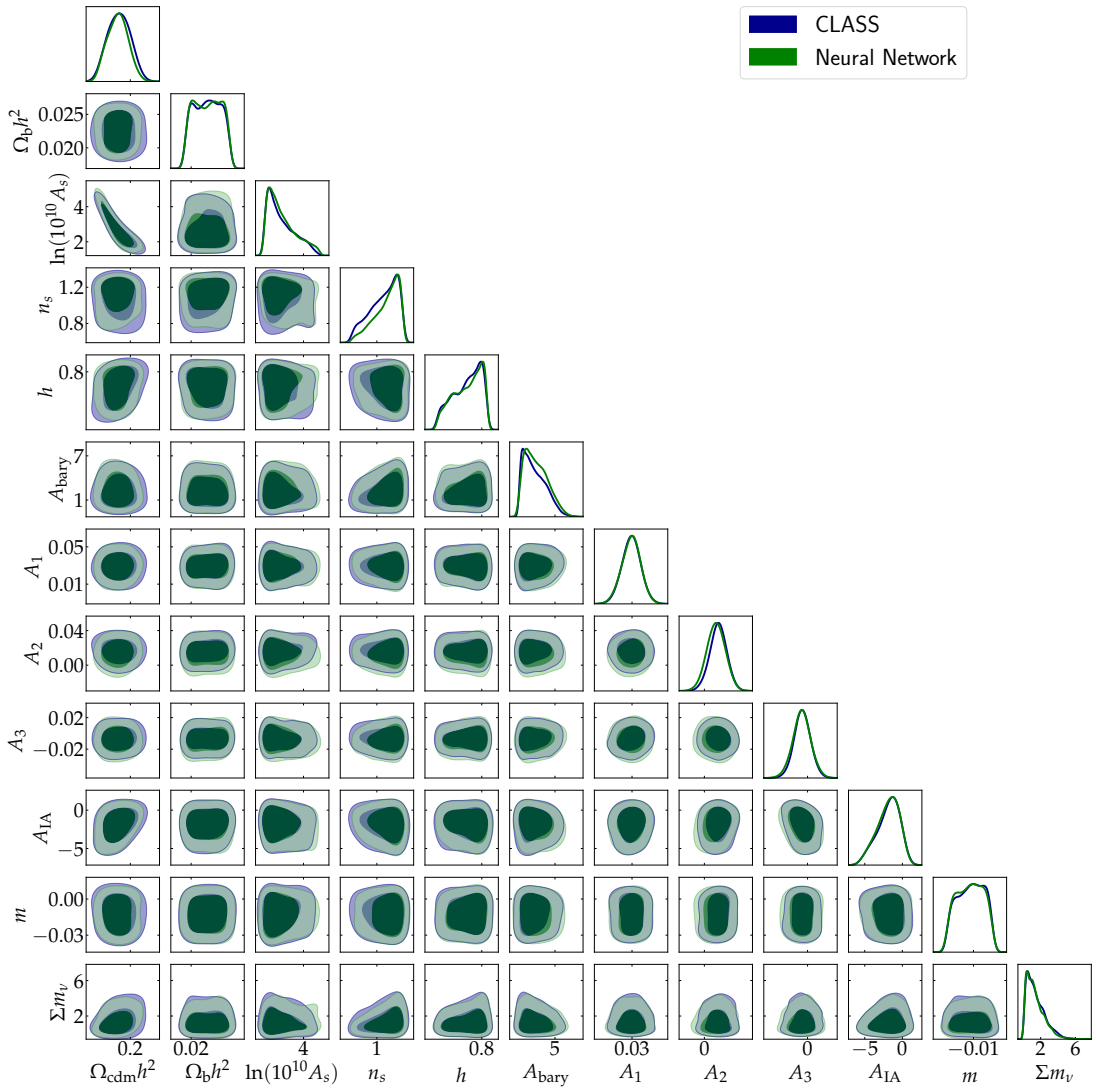


Figure 4.9 – The marginalised posterior distribution of the cosmological and nuisance parameters using neural network as an emulator. The contours in blue and green correspond to the results obtained with CLASS and the neural network respectively. The contours are plotted at 68% and 95% credible intervals. The two results are in good agreement with each other.

Once the neural network is trained, it is stored and can be queried in an MCMC sampler at any time. In this application, we connect it with the EMCEE sampler and we sample the full posterior distribution of the cosmological and nuisance parameters. The neural network is used to predict the band powers at every step in the MCMC routine. The marginalised posterior distribution of all parameters is shown in Figure 4.9.

The neural network emulator is quite fast compared to the accurate solver, CLASS. It takes around 150 minutes to sample the full posterior compared to CLASS, which takes around 44 hours. Importantly, as discussed in §4.6, the fact that the prediction of the band powers is very

accurate, results in the likelihood values, calculated at test points to be quite accurate as well. Despite the fact that the neural network performs so well, we can also probe various steps involved in the process. For example, we do not know exactly what is the right architecture. Hence, neural network strategies are often empirical.

4.5 Scalable Gaussian Process Models

If our aim is to build emulators for cosmological parameter inference, the standard Gaussian Process described in Chapter 3 is optimal for small training sets ($N \sim 1000$). The training step involves an $\mathcal{O}(N^3)$ operation for computing the inverse and determinant while prediction scales as $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$ for computing the mean and variance respectively (assuming we have cached $\mathbf{K}_y^{-1}$). In particular, computing the mean prediction is not a major computational bottleneck. Once, the model is trained, we store $\boldsymbol{\alpha} = \mathbf{K}_y^{-1}\boldsymbol{y}$ and $\mathbb{E}[f(\boldsymbol{x}_*)] = \boldsymbol{k}_*^{\mathrm{T}}\boldsymbol{\alpha}$. Moreover, we also require $\mathcal{O}(N^2 + ND)$ for memory if we have to compute the predictive uncertainty.

If we want *precise* and *accurate* predictions, especially in high dimensions ($d \geq 3$), the only way is to add more and more training points to improve the interpolation scheme. However, as we increase the number of training points, the Gaussian Process becomes prohibitively expensive to train and the computation of the predictive uncertainty is hindered by the $\mathcal{O}(N^2)$ operations. One option to deal with this major limitation is by partitioning the training set before building the GP model.

4.5.1 Product-of-Experts Models

Product-of-Experts (PoE) is a promising candidate for dealing with a large number of training points. Parallel and distributed computations can be fully exploited during training and making prediction.

The full data set, $\mathcal{D} = \{\mathbf{X}, \boldsymbol{y}\}$ is first partitioned into $\mathcal{D}_{(m)} = \{\mathbf{X}_{(m)}, \boldsymbol{y}_{(m)}\}$. Each expert, m is then used to make prediction, which is recombined at the parent node as shown in Figure 4.10. The fact that we have partitioned the full data set leads to the assumption that our kernel matrix is now block diagonal and given the nice properties of the latter, training and prediction can be improved significantly. The inverse of a block diagonal matrix is simply the inverse of each block while the determinant is

$$|\mathbf{K}_y| \approx \prod_{m=1}^{M} |\mathbf{K}_{y(m)}|$$

and the determinant is approximated as the product of the determinant of each block.

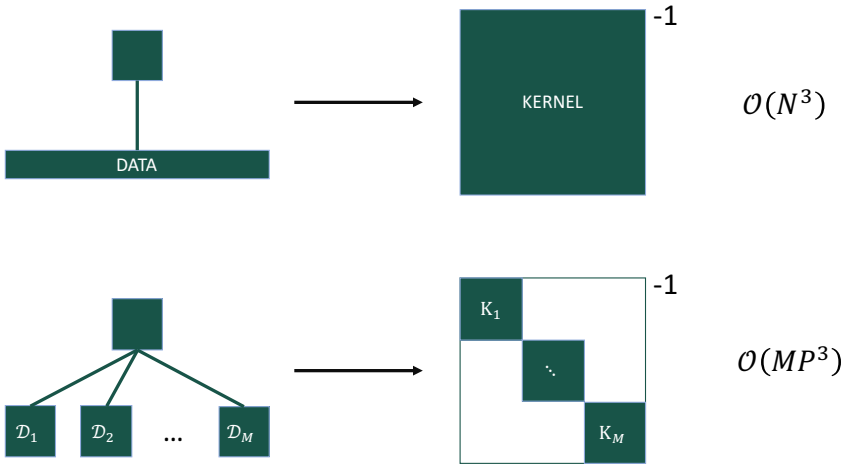


Figure 4.10 – The top panel shows the standard approach for training GPs, which usually involve an $\mathcal{O}(N^3)$ operation at each step in the iterative optimisation scheme. The performance can be improved if we partition the training set, as shown in the bottom panel. If we have M partitions, then this involves only $\mathcal{O}(MP^3)$ computational cost, where P is the number of training point per cluster.

Therefore, the marginal likelihood of this PoE model is:

$$\begin{aligned} \log p(\boldsymbol{y}) &\approx \sum_{m=1}^{M} \log p(\boldsymbol{y}_{(m)}) \\ &= -\frac{1}{2} \sum_{m=1}^{M} \left[\boldsymbol{y}_{(m)}^{\mathrm{T}} \mathbf{K}_{y(m)}^{-1} \boldsymbol{y}_{(m)} + \log |\mathbf{K}_{y(m)}| \right] + \text{constant} \end{aligned} \tag{4.5.1}$$

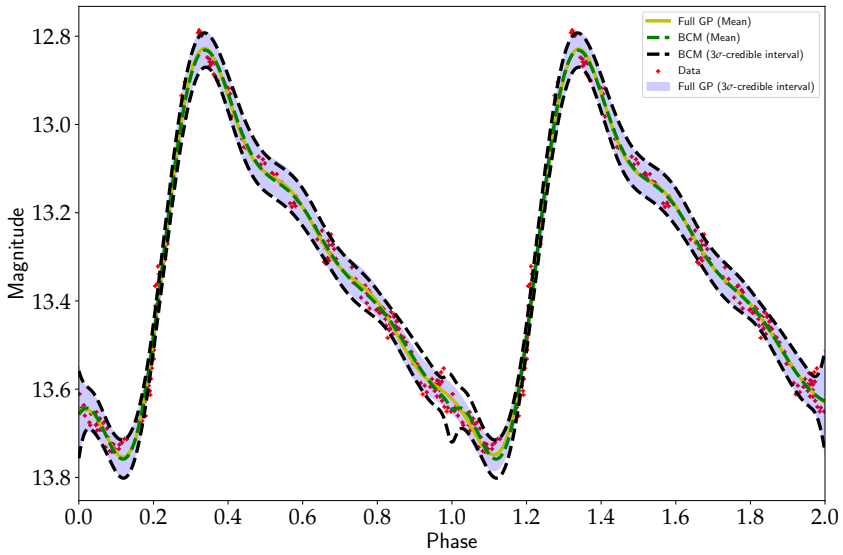


Figure 4.11 – In this figure, we take a noisy data set (a light-curve) with 734 points and we fit a full GP to the data, with the mean and 3σ confidence interval shown in yellow and pale blue respectively. We also compare it to the BCM approach, with 2 clusters and the mean and 3σ confidence interval are shown by the broken green and black curves respectively. The two methods agree quite well, except at $x = 1$, where the data is partitioned.

where $\mathbf{K}_{y(m)}$ is of size $N_m \times N_m$ and $N_m << N$. Training now only involves $\mathcal{O}(N_m)$ opera-

tions and $\mathcal{O}(N_m + N_m D)$ storage. In order to prevent over-fitting, the same single set of kernel hyper-parameters, $\boldsymbol{\eta}$, is used to train the Gaussian Process. In other words, we avoid setting different sets of kernel hyper-parameters for different experts. Another important ingredient is the gradient computation with respect to the marginal likelihood.

$$\frac{\partial}{\partial \boldsymbol{\eta}_i} \log p(\boldsymbol{y}) \approx \frac{1}{2} \sum_{m=1}^{M} \operatorname{tr} \left[\left(\boldsymbol{\alpha}_{(m)} \boldsymbol{\alpha}_{(m)}^{\mathrm{T}} - \mathbf{K}_{y(m)}^{-1} \right) \frac{\partial \mathbf{K}_{(m)}}{\partial \boldsymbol{\eta}_i} \right] \tag{4.5.2}$$

where $\boldsymbol{\alpha}_{(m)} = \mathbf{K}_{y(m)}^{-1} \boldsymbol{y}_{(m)}$. The next step is to predict the function at a given test point, $\boldsymbol{x}_*$ under the new model. One can take various approaches at this level. One can vary from choosing a single unit to including all computational units for predictions.

4.5.1.1 Single Unit Prediction

A single node, corresponding to the region of the parameter space where the test point is, can be used to make prediction. The mean and variance is simply the predictive mean and variance of a standard Gaussian Process, that is,

$$\begin{aligned} \mu_{*(m)} &= \boldsymbol{k}_{*(m)}^{\mathrm{T}} \mathbf{K}_{y(m)}^{-1} \boldsymbol{y}_{(m)} \\ \sigma_{*(m)}^2 &= \boldsymbol{k}_{**(m)} - \boldsymbol{k}_{*(m)}^{\mathrm{T}} \mathbf{K}_{y(m)}^{-1} \boldsymbol{k}_{*(m)} \end{aligned} \tag{4.5.3}$$

Note that this applies if our training set is partitioned via clustering method, hence exploiting locality. This is a quick technique to obtain the full predictive distribution at a test point.

4.5.1.2 PoE Prediction

An alternative option is to recombine the mean and variance from each node. Let us consider two computational units to derive the predictive distribution, that is, we seek the following:

$$\begin{aligned} p(f_* | \mathcal{D}_{(i)}, \mathcal{D}_{(j)}) &\propto p(\mathcal{D}_{(i)}, \mathcal{D}_{(j)} | f_*)\, p(f_*) \\ &= p(\mathcal{D}_{(i)} | f_*)\, p(\mathcal{D}_{(j)} | f_*)\, p(f_*) \\ &= \frac{p(\mathcal{D}_{(i)}, f_*)\, p(\mathcal{D}_{(j)}, f_*)}{p(f_*)} \\ &\propto \frac{p(f_* | \mathcal{D}_{(i)})\, p(f_* | \mathcal{D}_{(j)})}{p(f_*)} \end{aligned} \tag{4.5.4}$$

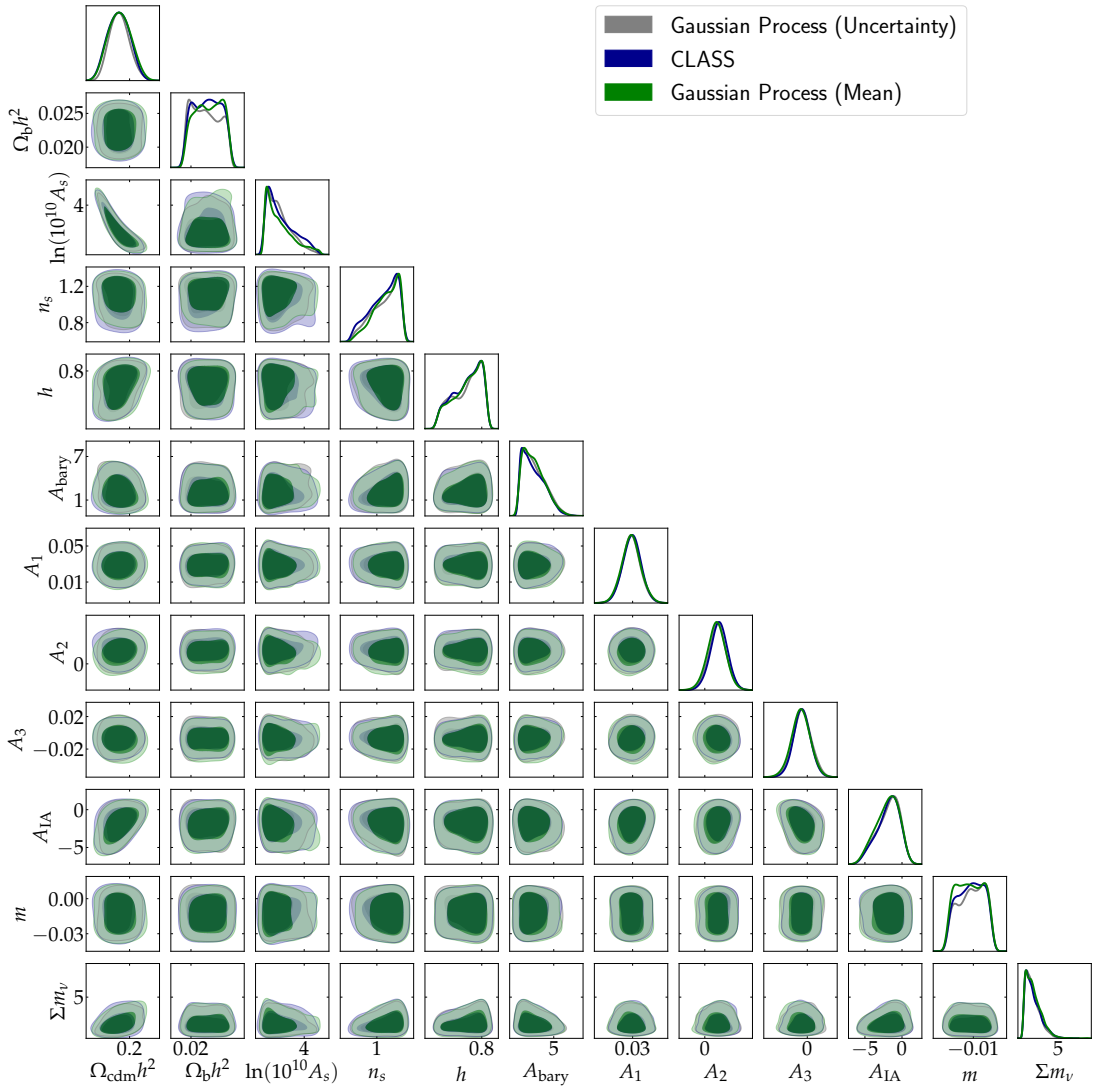


Figure 4.12 – The marginalised posterior distribution of all the cosmological and nuisance parameters using the scalable GP emulator, with 18000 training points and 120 clusters. The blue, green and grey contours correspond to experiments performed using CLASS, GP mean and GP error respectively. All the contours are plotted at 68% and 95% credible interval respectively.

The PoE assumes a flat prior on the predictive distribution (at the parent node) and a recursive application of the above formalism leads to:

$$p(f_*|\mathcal{D}) = \prod_{m=1}^{M} p(f_*|\mathcal{D}_{(m)}) \tag{4.5.5}$$

and the mean and variance are:

$$\begin{aligned} \mu_* &= \sigma_*^2 \sum_{m=1}^{M} \frac{\mu_{*(m)}}{\sigma^2_{*(m)}} \\ \sigma_*^{-2} &= \sum_{m=1}^{M} \sigma^{-2}_{*(m)} \end{aligned} \tag{4.5.6}$$

where $\mu_{*(m)}$ and $\sigma^2_{*(m)}$ are given by Equation 4.5.3, that is, they are predictions from the indi-

vidual child node. The PoE formalism will not be used in this work and we will instead use the technique discussed in the next section.

4.5.1.3 BCM Prediction

On the other hand, the Bayesian Committee Machine (BCM) does not ignore the prior function and the posterior distribution of the function at a given test point is:

$$p(f_*|\mathcal{D}) = \frac{\prod_{m=1}^{M} p(f_*|\mathcal{D}_{(m)})}{p^{M-1}(f_*)} \tag{4.5.7}$$

and the predictive mean and variance due to all the computational units are:

$$\begin{aligned} \mu_* &= \sigma_*^2 \sum_{m=1}^{M} \frac{\mu_{*(m)}}{\sigma^2_{*(m)}} \\ \sigma_*^{-2} &= (1-M)\sigma_{**}^{-2} + \sum_{m=1}^{M} \sigma^{-2}_{*(m)}, \end{aligned} \tag{4.5.8}$$

where σ^2_{**} is variance evaluated at the test point under the prior. the Note the extra correction term σ^{-2}_{**} in computing the variance compared to the PoE method. If partitioning is clustering-based, one can use a few neighbouring experts to recombined the mean and variance (Vijayakumar et al., 2005). The contribution of units which are far from the unit which contains the test point is negligible. An illustration of a noisy 1D regression using the BCM method is shown in Figure 4.11. In particular, the predictive mean and variance agree well with the full Gaussian Process regression.

4.5.2 Application to KiDS-450

The BCM model is very similar to the PICO algorithm described in §4.3.1, except that we are using GP models instead of polynomial regression models. In particular, the first two steps are analogous to the ones adopted when implementing the PICO algorithm, that is,

- the pre-whitening step is applied to the inputs and
- the $k-$means algorithm is used to partition the training set.

We use $N = 18000$ training points and $M = 120$ as in the previous section and we use Equation 4.5.1 to train each GP model. In summary, we have 72 GP models for each band power output. Training all the GP models took approximately 75 minutes.

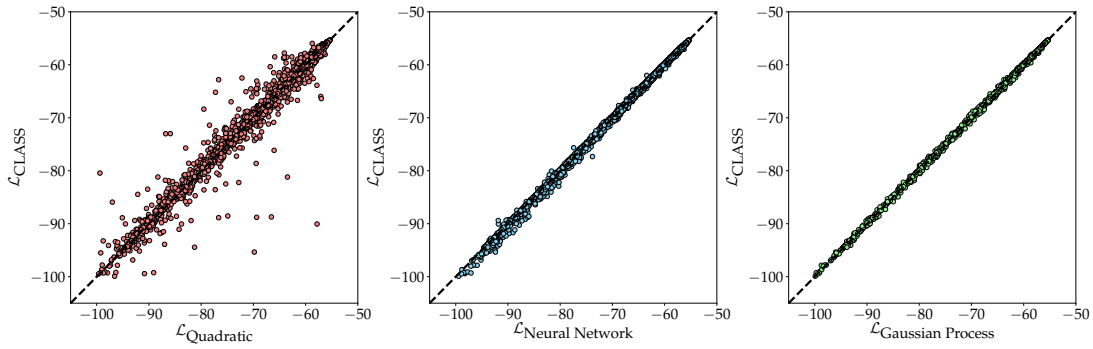


Figure 4.13 – In the three panels, we are comparing the log-likelihood values, as obtained from either method (PICO, NN and BCM) with CLASS. The Neural Network and Scalable GP methods are more robust compared to the PICO approach. This can be explained by the fact that PICO requires specifying the fitting model in the very first place, for which there is a large choice.

At this point, we can choose either of the methods described in §4.5.1.1, §4.5.1.2 or §4.5.1.3 to make predictions at test points in parameter space. We test two methods, namely the single unit prediction and the BCM prediction, if we choose to use more than 1 cluster. The scalable GP emulator is connected to the EMCEE sampler to sample the full posterior distribution of the cosmological and nuisance parameters. If we use a single unit, this takes around 215 and 250 minutes with the GP mean and uncertainty respectively. On the other hand, if we choose the BCM approach and choose 2 units to make predictions, this takes around 250 minutes either with the mean or uncertainty from the GP. In Figure 4.12, we show the marginalised posterior distribution using the single unit approach to make predictions.

4.6 Results

In this section, we highlight the main results obtained from the different emulating methods (PICO algorithm, Neural Network and Scalable GP models) presented in this chapter. Since we are using thousands of training points, one performance test we can do is to check how the log-likelihood, calculated using either method, compares with the one from CLASS. In Figure 4.13, we show the one-to-one relationship between the two log-likelihood calculations at an independent set of 5000 test points. The neural network and the scalable Gaussian Process approach are in robust agreement with the expected log-likelihood as calculated using CLASS.

Moreover, once we sample the full posterior distribution of the cosmological and nuisance parameters, we can check the distribution of the log-posterior values as recorded by the EMCEE sampler. If the function is properly reconstructed, we expect the distribution of the log-posterior using either method (PICO, NN, BCM, CLASS) should follow each other. Indeed, in all the 3 cases, the sampler consistently explore regions of high likelihood (posterior) as shown

in Figure 4.14.

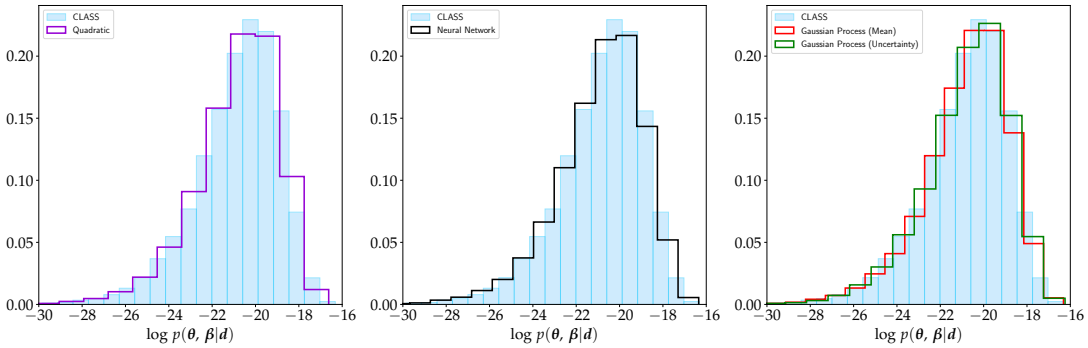
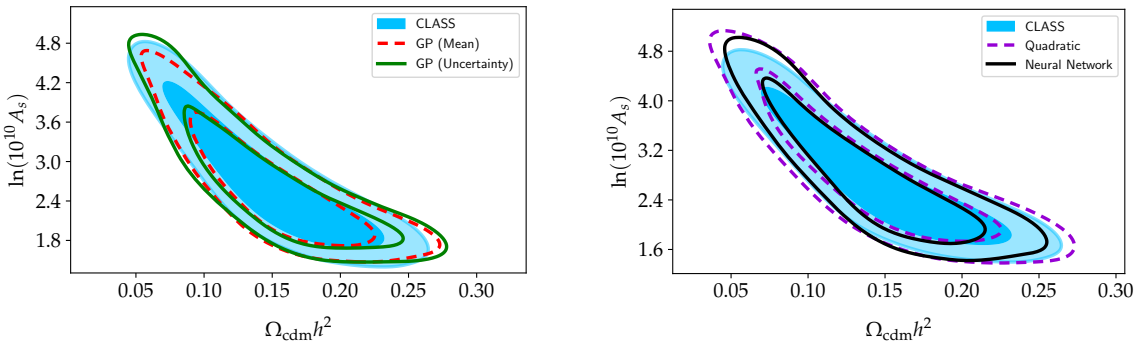


Figure 4.14 – The three panels show the distribution of the log-posterior of the MCMC samples using the four different methods - PICO, NN, BCM and CLASS. In all the panels, the pale blue histogram corresponds to the log-posterior as obtained when running the sampler with CLASS. The 3 different emulators all perform quite well. Importantly, the GP emulator also allows us to sample the posterior by marginalising over the GP uncertainty (shown by the green histogram in the right panel).

The two cosmological parameters which are currently constrained by weak lensing data are $\ln(10^{10}A_s)$ and $\Omega_{\rm cdm}h^2$ or derived versions of them, for example, σ_8 and $\Omega_{\rm m}$. The quantity σ_8 measures the amplitude of the linear matter power spectrum at a scale of $8\,{\rm h}^{-1}\,{\rm Mpc}$ and $\Omega_{\rm m}$ is the matter density of the present day Universe. It incorporates all forms of matter, including baryonic and dark matter. Another parameter combination which is often adopted in many weak lensing data analysis is S_8 defined as:

$$S_8 \equiv \sigma_8\sqrt{\frac{\Omega_{\rm m}}{0.3}}, \tag{4.6.1}$$

which corresponds to the well-measured direction in the $\Omega_{\rm m} - \sigma_8$ plane.



(a) BCM approach compared to CLASS

(b) PICO and NN compared to CLASS

Figure 4.15 – In this figure, we show the marginalised posterior distribution of $\ln(10^{10}A_s)$ and $\Omega_{\rm cdm}h^2$. In the left panel, we compare the scalable GP approach (BCM) with CLASS while in the right panel, we compare PICO and the neural network with CLASS.

In Figure 4.15, we show the marginalised banana-shaped posterior distribution of $\ln(10^{10}A_s)$ and $\Omega_{\rm cdm}h^2$. Similar shapes are expected if we plot σ_8 against $\Omega_{\rm m}$ instead of $\ln(10^{10}A_s)$ and

$\Omega_{\rm cdm}h^2$. The posterior distributions as obtained by the 3 emulating methods are robust when compared to CLASS.

In summary, the number of training points used for the three different methods are 18 000 and this took $\sim$ 3 hours to be generated. The different timings, pros and cons can be summarised in the table below.

Table 4.6.1 – The pros and cons of the different methods investigated in this chapter

Method	Pros	Cons
PICO	• Analytic error estimate • Very fast training	• Specifying the basis functions
NN	• Very fast training	• Choice of a model architecture • No uncertainty estimate on predictions
BCM	• Analytic error estimate • Possibility for calculating analytic derivatives	• Slow training procedure.

The posterior distributions obtained using the different methods, in this exploratory analysis, are good by visual inspections. However, in Chapter 6, we will use very few training points and we will also quantify which technique we will recommend to perform emulation, that is, if when we use the GP approach, one can include or exclude the GP uncertainty. Moreover, despite the fact that the BCM (GP) procedure are slow to train, two important outputs are the uncertainty estimates on the function and the analytical derivatives, which we explore in Chapter 7. Nevertheless, the fact that we are emulating a deterministic function, perhaps the mean function from the GP is more suitable. A summary of the pros and cons of the different methods is presented in Table 4.6.1.

While the above results look promising, there are various issues we encountered when we pursued this work. Some of these limitations are highlighted in the next section and we propose multiple solutions to these limitations and these solutions significantly improve this work. The different solutions proposed are investigated in more details in Chapter 6.

4.7 Possible Improvements

While the above methods work well for deriving constraints on the cosmological and nuisance parameters, they are not without limitations. In the first case, we do not have strong constraints on the parameters for current weak lensing surveys and this is only going to be improved as more data are observed. Therefore, the region encompassed by the parameters is quite broad in parameter space. Running a short MCMC chain, followed by sampling points within 4 times the covariance matrix (and ensuring that the points lie within the pre-defined prior box, for example, by adopting a rejection sampling scheme) is an ad-hoc procedure. That said, in future surveys, one can still substitute the short MCMC run by an iterative scheme (which should converge fast enough) and an emulator can be built based on the maximum likelihood solution.

Next, another school of thought might argue that the partitioning step in PICO and BCM is not an elegant approach since the continuous function which we are emulating is not continuous anymore as a result of the partitioning step. The BCM approach attempts to remedy this by calculating a weighted mean and variance using local clusters.

We also found multiple technical issues with the priors defined in Köhlinger et al. (2017), for example, because the parameter $A_{\rm bary}$ is so broad, for some combination of parameters, the modified 3D matter power spectrum becomes negative. In some cases, large neutrino masses led to `nan` values in the power spectrum calculation. We improve upon all these issues in the next chapter. In short, we introduce the following innovations:

- new priors are chosen such that all forward simulations do not result in `nan` ,
- the number of training points is reduced significantly from 18 000 to just a few thousands,
- the input training points are distributed according to the pre-defined prior range and
- instead of using $\log_{10}$ transformation on the band powers directly, we will use matrix logarithm which ensures that band power matrix remains positive definite.

4.8 Summary

In this exploratory analysis, we have provided an in-depth overview of the different algorithms, which are all scalable, to illustrate the concept behind an emulator. In fact, the three algorithms,

namely, PICO, Bayesian Committee Machines and Neural Networks are not the only set of algorithms which one can use. We first explained the data we have used to do the analysis in this chapter followed by the description of the three different emulators. PICO is based on polynomial regression whereas the Bayesian Committee Machines algorithm is based on Gaussian Processes. The different algorithms are able to recover reliable posterior densities. However, we identified a few limitations in both the likelihood code distributed and in our approach as well, which we highlight in §4.7.

Chapter 5

Data Compression and Emulation

> Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution.
>
> *Albert Einstein*

In this chapter, we use the ideas explored in Chapter 4 to further develop the concept of emulation. In particular, we embark on a completely new approach to perform emulation, which combines the massive optimal compression, the MOPED algorithm developed by Heavens et al. (2000). As cosmological surveys become more data intensive, one can possibly use Machine Learning techniques in different cosmological data analyses.

Crucially, MOPED compresses N data points to just p numbers ($p \ll N$) while preserving the constraints on the inferred parameters if we were to compare the inference when using the uncompressed dataset. However, the theoretical prediction with MOPED can still be slow because it still requires the evaluation of the forward model which can be very expensive. To circumvent this issue, the obvious question to ask is: can we evaluate the MOPED coefficients at some points in parameter space and learn the p different function with a Machine Learning algorithm? This is central to this chapter and we test various scenarios and exploratory analysis to illustrate the robustness of our approach.

This chapter makes use of the JLA data* (Betoule et al., 2014) to develop an inference procedure to illustrate how compression and emulation can be used together. This chapter serves as a precursor to Chapter 6 in which we do a full likelihood analysis in the weak lensing context.

This chapter is organised as follows: in §5.1, we discuss briefly the Fisher information matrix for a Gaussian random field. In §5.2, we discuss briefly the MOPED data compression algorithm and in §5.3, we cover briefly the JLA data, covariance and approximate models which

*http://supernovae.in2p3.fr/sdss_snls_jla/ReadMe.html

we use this this chapter. In §5.4, we discuss how the one can use a simple optimisation procedure to estimate the parameters and in §5.5, we dive into the different inference procedures depending on the test case. We finally discuss the main results of our analysis in §5.6 before providing a brief summary of the chapter in §5.8.

5.1 Fisher Information Matrix for a Gaussian Random Field

Here, we briefly review the equations derived in Tegmark et al. (1997) to obtain analytical expressions for the gradient and Hessian of $\mathcal{L} \equiv -\ln L$, where L is a Gaussian likelihood. An analogous prescription is provided by Alsing & Wandelt (2018). Ignoring any additive constant, the negative log-likelihood, $\mathcal{L}$ is given by

$$\mathcal{L} = \frac{1}{2} (\boldsymbol{d} - \boldsymbol{\mu}) \mathbf{C}^{-1} (\boldsymbol{d} - \boldsymbol{\mu})^{\mathrm{T}} + \frac{1}{2} \ln |\mathbf{C}| . \tag{5.1.1}$$

The data matrix is defined as

$$\mathbf{D} \equiv (\boldsymbol{d} - \boldsymbol{\mu}) (\boldsymbol{d} - \boldsymbol{\mu})^{\mathrm{T}} , \tag{5.1.2}$$

such that $\mathcal{L}$ can be re-written as $\mathcal{L} = \frac{1}{2} \mathrm{tr}(\ln \mathbf{C} + \mathbf{C}^{-1}\mathbf{D})$ using the matrix identity $\ln |\mathbf{C}| = \mathrm{tr} \ln \mathbf{C}$. In the following, a comma denotes the partial derivative with respect to a specific parameter. The first and second derivatives of the data matrix are given by

$$\begin{aligned} \mathbf{D}_{,i} &= 2 (\boldsymbol{\mu} - \boldsymbol{d}) \boldsymbol{\mu}^{\mathrm{T}}_{,i}, \\ \mathbf{D}_{,ij} &= 2 \boldsymbol{\mu}_{,j} \boldsymbol{\mu}^{\mathrm{T}}_{,i} + 2 (\boldsymbol{\mu} - \boldsymbol{d}) \boldsymbol{\mu}^{\mathrm{T}}_{,ij}. \end{aligned} \tag{5.1.3}$$

Using the matrix identities $(\mathbf{C}^{-1})_{,i} = -\mathbf{C}^{-1}\mathbf{C}_{,i}\mathbf{C}^{-1}$ and $(\ln \mathbf{C})_{,i} = \mathbf{C}^{-1}\mathbf{C}_{,i}$, the gradient and Hessian of $\mathcal{L}$ are

$$\mathcal{L}_{,i} = \frac{1}{2} \mathrm{tr} \left[\mathbf{C}^{-1}\mathbf{C}_{,i} - \mathbf{C}^{-1}\mathbf{C}_{,i}\mathbf{C}^{-1}\mathbf{D} + \mathbf{C}^{-1}\mathbf{D}_{,i} \right] , \tag{5.1.4}$$

$$\begin{aligned} \mathcal{L}_{,ij} = \frac{1}{2} \mathrm{tr} \Big[& - \mathbf{C}^{-1}\mathbf{C}_{,i}\mathbf{C}^{-1}\mathbf{C}_{,j} + \mathbf{C}^{-1}\mathbf{C}_{,ij} \\ & + \mathbf{C}^{-1} \left(\mathbf{C}_{,i}\mathbf{C}^{-1}\mathbf{C}_{,j} + \mathbf{C}_{,j}\mathbf{C}^{-1}\mathbf{C}_{,i} \right) \mathbf{C}^{-1}\mathbf{D} \\ & - \mathbf{C}^{-1} \left(\mathbf{C}_{,i}\mathbf{C}^{-1}\mathbf{D}_{,j} + \mathbf{C}_{,j}\mathbf{C}^{-1}\mathbf{D}_{,i} \right) \\ & - \mathbf{C}^{-1}\mathbf{C}_{,ij}\mathbf{C}^{-1}\mathbf{D} + \mathbf{C}^{-1}\mathbf{D}_{,ij} \Big] . \end{aligned} \tag{5.1.5}$$

The two equations above give the gradient and Hessian at *any* point in the parameter space. Equations 5.1.4 and 5.1.5 also take into account the fact the covariance matrix might depends on a subset of the parameters in our model. However, at the maximum likelihood estimate (MLE), we have $\mathcal{L}_{,i} = 0$ and the Fisher information matrix, $\mathbf{F}_{ij}$ is

$$\mathbf{F}_{ij} \equiv \left\langle \mathcal{L}_{,ij} \right\rangle = \frac{1}{2}\mathrm{tr}\left[\mathbf{C}^{-1}\mathbf{C}_{,i}\mathbf{C}^{-1}\mathbf{C}_{,j} + 2\mathbf{C}^{-1}\boldsymbol{\mu}_{,j}\boldsymbol{\mu}_{,i}^{\mathrm{T}}\right] \tag{5.1.6}$$

If the covariance matrix is independent of parameters, the above expression for the Fisher information matrix simplifies to

$$\mathbf{F}_{ij} = \mathrm{tr}\left[\mathbf{C}^{-1}\boldsymbol{\mu}_{,j}\boldsymbol{\mu}_{,i}^{\mathrm{T}}\right]. \tag{5.1.7}$$

5.2 The MOPED algorithm

In this section, we briefly discuss the MOPED algorithm (Heavens et al., 2000) which reduces the number of data points from N to just p numbers. N is the number of data points and p is the number of parameters in our model. MOPED essentially finds some weighing vector, $\boldsymbol{b}$, which encapsulates as much information as possible for a specific model parameter $\boldsymbol{\theta}_\alpha$. This vector is then used to find a linear combination of the data, $\boldsymbol{d}$ such that the compressed data is

$$y_\alpha \equiv \boldsymbol{b}_\alpha^{\mathrm{T}}\boldsymbol{d}. \tag{5.2.1}$$

The first and subsequent MOPED vectors are given respectively by

$$\boldsymbol{b}_1 = \frac{\mathbf{C}^{-1}\boldsymbol{\mu}_{,1}}{\sqrt{\boldsymbol{\mu}_{,1}^{\mathrm{T}}\mathbf{C}^{-1}\boldsymbol{\mu}_{,1}}} \tag{5.2.2}$$

and

$$\boldsymbol{b}_\alpha = \frac{\mathbf{C}^{-1}\boldsymbol{\mu}_{,\alpha} - \sum_{\beta=1}^{\alpha-1}(\boldsymbol{\mu}_{,\alpha}^{\mathrm{T}}\boldsymbol{b}_\beta)\boldsymbol{b}_\beta}{\sqrt{\boldsymbol{\mu}_{,\alpha}^{\mathrm{T}}\mathbf{C}^{-1}\boldsymbol{\mu}_{,\alpha} - \sum_{\beta=1}^{\alpha-1}(\boldsymbol{\mu}_{,\alpha}^{\mathrm{T}}\boldsymbol{b}_\beta)^2}} \quad (\alpha > 1), \tag{5.2.3}$$

where $\mathbf{C}$ is the data covariance matrix and $\boldsymbol{\mu}_{,\alpha}$ is the vector obtained by calculating the gradient of our theoretical model at a fiducial parameter set. If $\mathbf{B} \in \mathbb{R}^{N\times p}$ is the matrix whose columns consist of the MOPED vectors, the compressed data vector is just

$$\boldsymbol{y} = \mathbf{B}^{\mathrm{T}}\boldsymbol{d}. \tag{5.2.4}$$

By construction, the MOPED vectors $\boldsymbol{b}_\alpha$ are orthogonal to each other, that is, $\boldsymbol{b}_\alpha^{\mathrm{T}}\mathbf{C}\boldsymbol{b}_\beta = \delta_{\alpha\beta}$. Therefore, the covariance matrix of $\boldsymbol{y}$, $\mathbf{B}^{\mathrm{T}}\mathbf{C}\mathbf{B} = \mathbb{I}$, the identity matrix, of size $p \times p$. As a result of this orthogonality condition, elements from the compressed data vector are uncorrelated. Hence, the log-likelihood is straightforwardly

$$\log \mathcal{L} = -\frac{1}{2}\sum_{\alpha=1}^{p}(y_\alpha - \boldsymbol{b}_\alpha^{\mathrm{T}}\boldsymbol{\mu})^2 + \text{constant}, \tag{5.2.5}$$

where $\boldsymbol{b}_\alpha^{\mathrm{T}}\boldsymbol{\mu}$ is typically the expensive part (if $\boldsymbol{\mu}$ itself is expensive to compute). The fact that the likelihood of the compressed data involves only $\mathcal{O}(p)$ operation makes parameter inference very fast since the $\mathcal{O}(N^3)$ operation (if the covariance matrix depends on the parameters) in the standard likelihood is completely eliminated, provided $\boldsymbol{b}_\alpha^{\mathrm{T}}\boldsymbol{\mu}$ can be rapidly computed.

5.3 Joint Light Curve Analysis (JLA)

The Joint Lightcurve Analysis (JLA) supernova dataset is a compilation of 740 supernova catalogues from various surveys (for details see Betoule et al., 2014). Relevant quantities crucial for our analysis consist of the apparent magnitudes, m_B, redshifts, z and light curve parameters: colour correction term, C and stretch, x_1. In addition, a 2220×2220 covariance matrix as a function of calibration parameters α and β for the stretch and colour is also provided (see §5.3.1 for a detailed explanation on how the 740×740 covariance matrix is constructed). Figure 5.1 shows the apparent magnitudes and the data covariance matrix, which is constructed at the optimised solution (see §5.3.1 for a detailed explanation).

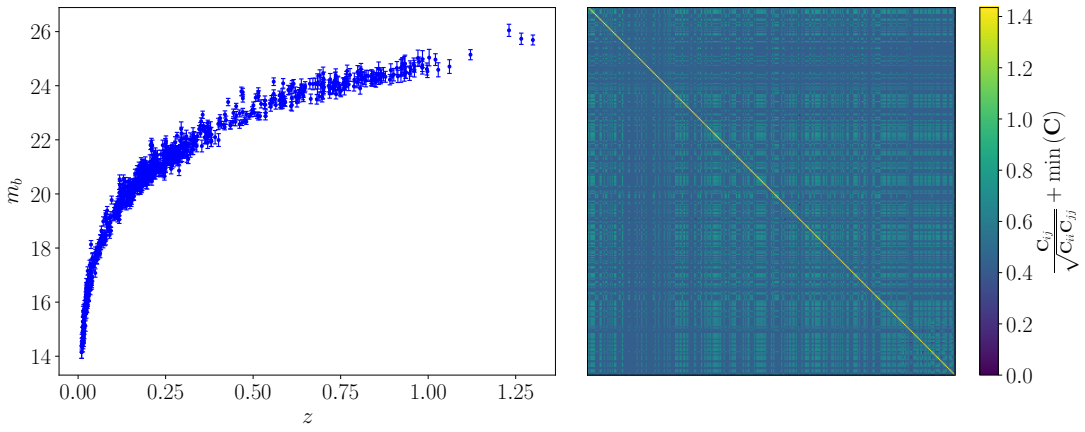


Figure 5.1 – The left panel shows the apparent magnitude, with its associated uncertainty, as a function of redshift and the full data correlation matrix is shown on the right.

5.3.1 Covariance Matrix

In this section, we first explain how the data covariance matrix is constructed before finding the first and second derivatives of the covariance matrix for the JLA dataset. In particular, a covariance matrix which depends on α and β was constructed by Betoule et al. (2014). We are also provided with a matrix of size 2220×2220 and the latter can be interpreted as being 9 blocks each of size 740×740 in the following format

$$\mathbf{R} = \begin{pmatrix} \mathbf{C}_{00} & \mathbf{C}_{01} & \mathbf{C}_{02} \\ \mathbf{C}_{10} & \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{20} & \mathbf{C}_{21} & \mathbf{C}_{22} \end{pmatrix} \tag{5.3.1}$$

and defining the vector $\boldsymbol{\rho} = (1, \alpha, -\beta)$, the final covariance matrix is given by

$$\mathbf{C} = \sum_i \sum_j \boldsymbol{\rho}_i \boldsymbol{\rho}_j \mathbf{R}_{ij} + \mathbf{C}_{\text{diag}}. \tag{5.3.2}$$

It consists of two parts. The first part accounts for the statistical and systematic uncertainties due to the light curve parameters whereas $\mathbf{C}_{\text{diag}}$ considers the errors due to other effects such as variation of magnitudes due to gravitational lensing. We refer the reader to §5.5 in Betoule et al. (2014) for further details on the covariance matrix. The fact that we now have a parameter dependent covariance matrix and that the full Hessian term in Equation 5.1.5 is a function of the partial derivatives of the covariance matrix, the first and second derivatives of the data covariance matrix with respect to α and β are given by

$$\begin{aligned} \mathbf{C}_{,\alpha} &= \mathbf{C}_{01} + \mathbf{C}_{10} + 2\alpha\mathbf{C}_{11} - \beta\mathbf{C}_{12} - \beta\mathbf{C}_{21} \\ \mathbf{C}_{,\beta} &= -\mathbf{C}_{02} - \alpha\mathbf{C}_{12} - \mathbf{C}_{20} - \alpha\mathbf{C}_{21} + 2\beta\mathbf{C}_{22} \\ \mathbf{C}_{,\alpha\alpha} &= 2\mathbf{C}_{11} \\ \mathbf{C}_{,\alpha\beta} &= -\mathbf{C}_{12} - \mathbf{C}_{21} \\ \mathbf{C}_{,\beta\beta} &= 2\mathbf{C}_{22} \end{aligned} \tag{5.3.3}$$

These derivations are useful when finding the set of parameters which maximise the likelihood.

5.3.2 Model

For a comparative study, we assume the same cosmological model used by Alsing et al. (2018) and Leclercq (2018). Type Ia supernova being 'standard candles', the expected apparent mag-

nitude is given by

$$m_B = 5\log_{10} D_L(z) + M_B + \delta M s - \alpha x_1 + \beta C \tag{5.3.4}$$

where D_L is the luminosity distance which depends on the position of the source and is a function of the cosmological parameters. $\tilde{M}_B = M_B + \delta M$ is the absolute magnitude and the extra term, δM is used to model the dependence of the absolute magnitude on the host galaxy properties. In particular, it is given by

$$s = \begin{cases} 1 & \text{if } \log_{10} M_{\text{stellar}} > 10 \\ 0 & \text{otherwise,} \end{cases} \tag{5.3.5}$$

where M_{stellar} is the host-galaxy mass. Through the distance-redshift relation, the cosmological model enters in the analysis, where we assume a flat universe together with cold dark matter and dark energy (wCDM hereafter). The luminosity distance in a wCDM universe is given by

$$D_L = \frac{(1+z)\,c}{H_0} \int_0^z \frac{dz'}{\sqrt{\Omega_m (1+z')^3 + (1-\Omega_m)(1+z')^{3(w_0+1)}}}, \tag{5.3.6}$$

where c is the speed of light, H_0 is the Hubble constant and we fix to $70\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$ and the wCDM universe is described by the two cosmological parameters: the matter density, Ω_m and the equation-of-state, w_0. Hence, our final model, accounting for the nuisance parameters as well, has six parameters in all,

$$\gamma = (\Omega_m,\, w_0,\, M_B,\, \delta M,\, \alpha,\, \beta)$$

.

5.3.3 Dual Compression

The final model in equation 5.3.4 can be viewed as two separate models, consisting of a cosmological model and a Gaussian linear model such that we can write

$$\begin{aligned} \boldsymbol{\mu}(\boldsymbol{\theta}, \boldsymbol{\eta}) &= \boldsymbol{u}(\boldsymbol{\theta}) + \boldsymbol{v}(\boldsymbol{\eta}) \\ \boldsymbol{u}(\boldsymbol{\theta}) &= 5\log_{10} D_L(\boldsymbol{\theta}) \\ \boldsymbol{v}(\boldsymbol{\eta}) &= M_B + \delta M s - \alpha x_1 + \beta C, \end{aligned} \tag{5.3.7}$$

where $\boldsymbol{\theta} = (\Omega_m, w_0)$ and $\boldsymbol{\eta} = (M_B, \delta M, \alpha, \beta)$. In most typical cosmological applications, computing the theoretical model related to the cosmology is the most computationally demanding part of the full analysis. In the same spirit, computing $u(\boldsymbol{\theta})$ (see equations 5.3.6 and 5.3.7) involves an integration for each supernova. As the sample size increases, it becomes prohibitively expensive to evaluate these integrations. It is arguable that these computations are not as expensive as large-scale simulations but we provide only a proof-of-concept analysis in this chapter. Note that the computing the Gaussian Linear model is quick. In large simulation settings, the computational resource required is even more onerous. One would ideally want to reduce the number of simulations to a manageable value whilst still performing a full likelihood analysis, including sampling the full posterior distribution and marginalizing over the nuisance parameters.

Therefore, we propose the following approach. Given N_{train} cosmologies, $\boldsymbol{\theta}_{\text{train}}$, we compute $\boldsymbol{u}$ for each supernova and rescale the training set, $\mathbf{U}\left(\boldsymbol{\theta}_{\text{train}}\right) \in \mathbb{R}^{N \times N_{\text{train}}}$ such that

$$\mathbf{U}_{\text{train}} = \frac{\mathbf{U}\left(\boldsymbol{\theta}_{\text{train}}\right) - \bar{\boldsymbol{u}}}{\sigma}. \tag{5.3.8}$$

$\bar{\boldsymbol{u}}$, of length N, is the mean of the number of forward simulations and σ is the standard deviation of the whole table, $\mathbf{U}\left(\boldsymbol{\theta}_{\text{train}}\right)$. The above is a column-wise operation, The next step involves finding a set of bases which will retain as much information as possible of $\mathbf{U}_{\text{train}}$.

5.3.4 Karhunen-Loève Compression

To this end, we resort to the Karhunen-Loève Compression to transform our training set to a lower and new dimensional space. We first compute the covariance matrix of the scaled training set as in equation 5.3.8, $\mathbf{C}_{\text{train}} = \text{cov}(\mathbf{U}_{\text{train}})$. We then compute the eigen-decomposition of the covariance matrix

$$\boldsymbol{\Lambda}_{\text{train}} = \boldsymbol{\Phi}\mathbf{C}_{\text{train}}\boldsymbol{\Phi}^{\text{T}} \tag{5.3.9}$$

giving a transformation matrix, $\boldsymbol{\Phi}$ which contains the eigenvectors of the covariance matrix $\mathbf{C}_{\text{train}}$ whereas $\boldsymbol{\Lambda}_{\text{train}}$ is a diagonal matrix containing the eigenvalues of $\mathbf{C}_{\text{train}}$. A compression matrix is then built by retaining the eigenvectors with the most significant eigenvalues. This then forms a new set of basis $\tilde{\boldsymbol{\Phi}} \in \mathbb{R}^{M \times N}$, where M is the number of components retained $(M \ll N)$ such that

$$\tilde{\mathbf{U}}_{\text{train}} = \tilde{\mathbf{\Phi}}^{\mathrm{T}}\mathbf{W}, \tag{5.3.10}$$

where $\mathbf{W} = \tilde{\mathbf{\Phi}}\mathbf{U}_{\text{train}}$. Using the compression matrix $\tilde{\mathbf{\Phi}}$ gives a mapping from a very high dimensional space, N to only M space. As we will see in the next section, this step improves the final algorithm to a large extent since we build few emulators only.

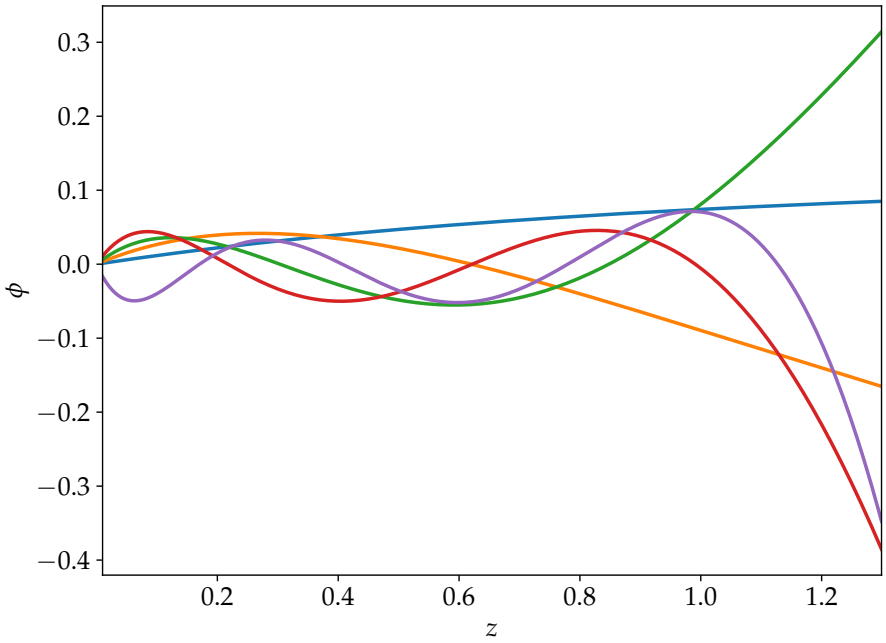


Figure 5.2 – The first five basis functions, as a function of redshift, obtained after performing Karhunen-Loève Compression on our training set.

The weights $\mathbf{W} \in \mathbb{R}^{M \times N_{\text{train}}}$ are now fixed and instead of mapping the cosmological parameters, $\boldsymbol{\theta}_{\text{train}}$ to each of the column of $\mathbf{U}_{\text{train}}$, we will now learn a function which maps $\boldsymbol{\theta}_{\text{train}}$ to each row in $\mathbf{W}$ which we denote as $\boldsymbol{w}_{(m)}$.

5.3.5 Theoretical Prediction

After this point, we now model each of the weight, $\boldsymbol{w}_{(m)}$ by a Gaussian Process. We refer the reader to the brief overview we provided in §3 and standard literature for further details on Gaussian Processes. Each Gaussian Process is trained independently, that is, we have a kernel matrix, $\mathbf{K}_{(m)}$ for each $\boldsymbol{w}_{(m)}$. The predicted mean and variance at a given test point, $\boldsymbol{\theta}_*$ for each weight $w^*_{(m)}$ is given by

$$\begin{aligned} \mathbb{E}\left[w^*_{(m)}\right] &= \boldsymbol{k}^{\mathrm{T}}_{*(m)}\mathbf{K}^{-1}_{(m)}\boldsymbol{w}_{(m)}, \\ \text{var}\left[w^*_{(m)}\right] &= \boldsymbol{k}_{**(m)} - \boldsymbol{k}^{\mathrm{T}}_{*(m)}\mathbf{K}^{-1}_{(m)}\boldsymbol{k}_{*(m)}. \end{aligned} \tag{5.3.11}$$

Given the surrogate models for the $w_{(m)}$, our theoretical model, calculated at a test point, $\gamma_* = (\boldsymbol{\theta}_*, \boldsymbol{\eta}_*)$ is just

$$\boldsymbol{\mu}_* (\boldsymbol{\theta}_*, \boldsymbol{\eta}_*) = \sigma \tilde{\boldsymbol{\Phi}}^{\mathrm{T}} \boldsymbol{w}_* + \boldsymbol{\Psi} \boldsymbol{\eta}_* + \bar{\boldsymbol{u}}, \tag{5.3.12}$$

where $\boldsymbol{\Psi}$ is a design matrix given by $(\mathbf{1}, \boldsymbol{s}, \boldsymbol{x}_1, \boldsymbol{c})$ corresponding to the parameter vector, $\boldsymbol{\eta}$ defined above. Note that the calculation of each mean weight is quick since the prediction from a Gaussian Process is a linear predictor. In other words, $\mathbf{K}_{(m)}^{-1} \boldsymbol{w}_{(m)}$, of size N_{train}, is calculated only once after training the Gaussian Process. Computing the uncertainty associated with the weights can be a computational bottleneck because for each test point, $\boldsymbol{\theta}_*$, we have to compute $\mathbf{K}_{(m)}^{-1} \boldsymbol{k}_{*(m)}$. However, for a small training set ($N_{\text{train}} < 1000$), this is not really an issue and in general, the cost of running the full simulator can be much more expensive than running the emulator with its associated uncertainty. Moreover, in low dimension ($d \leq 3$), one can reconstruct the function almost perfectly with a few hundreds training points. Devising scalable Gaussian Processes is currently an active area of research. For example, the Bayesian Committee Machine first partitions the data into local experts and combine the predictive mean and variance via Bayes' theorem (Cao & Fleet, 2014; Deisenroth & Ng, 2015). Snelson & Ghahramani (2005) developed a method, referred to as the inducing point method, for which the mean and variance involve only $\mathcal{O}(Q)$ and $\mathcal{O}(Q^2)$ computations after training. Q there refers to the number of inducing points, which can either be optimised or specified by the user.

5.4 Implementation

Based on the above description, we now use the JLA data, discussed in §5.3 to illustrate our method. In particular, we will refer to the full forward model (equations 5.3.4 and 5.3.6) as the simulator while equation 5.3.12 will henceforth be referred to as the emulator.

We use only $N_{\text{train}} = 300$ Latin Hypercube samples (Carnell, 2019) to compute $\boldsymbol{u}(\boldsymbol{\theta})$. We found that only 5 components ($M = 5$) were sufficient to accurately reconstruct our table $\mathbf{U}_{\text{train}}$. This compression step circumvents the need to build 740 separate Gaussian Processes for each supernova. Instead, we will model each of the 5 weights,$\boldsymbol{w}_{(m)}$, by an individual Gaussian Process. In Figure 5.2, we show the 5 basis functions[†] which are functions of the redshift. Once

[†]Strictly, there is a small additive error due to the reconstruction which can be propagated in the full statistical framework. However, it is deemed to be very small and has negligible impact on optimization and parameter inference (see Figure 5.4 and panel (a) in Figure 5.10)

we have a surrogate model for each $\boldsymbol{w}_{(m)}$, for each draw of $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$, we can then use equation 5.3.12 to compute the predictive apparent magnitude. The latter also has an uncertainty associated with it, since the predictive weights from the Gaussian Processes are normally distributed. Our analysis is carried out in a fully Bayesian framework, hence propagating the uncertainty obtained from the Gaussian Processes in the likelihood (see §5.5 for a detailed explanation).

Since we are essentially reconstructing a function, which is faster than the original full forward model, we will first show that, the emulator can also be used in an optimization algorithm to give a good approximation to the optimal solution, compared to the full forward model. Of course, the key ingredients for optimization involve the calculation to the gradient and Hessian of the likelihood, which we now discuss.

5.4.1 Optimization

In this section, we discuss how optimization can be used to learn an estimate of the MLE via an iterative procedure. Recall that $\gamma = \{\boldsymbol{\theta}, \boldsymbol{\eta}\}$ is the set of parameters consisting of the cosmological and nuisance parameters. The second order Taylor expansion of $\mathcal{L}$ about an expansion point, $\hat{\gamma}$, in terms of the gradient $\boldsymbol{g} = \nabla\mathcal{L}$ and the Hessian $\mathbf{H} = \nabla^2\mathcal{L}$, yields

$$\mathcal{L}\left(\hat{\gamma} + \delta\gamma\right) \approx \mathcal{L}\left(\hat{\gamma}\right) + \delta\gamma^{\mathrm{T}}\boldsymbol{g}\left(\hat{\gamma}\right) + \frac{1}{2}\delta\gamma^{\mathrm{T}}\mathbf{H}\left(\hat{\gamma}\right)\delta\gamma. \tag{5.4.1}$$

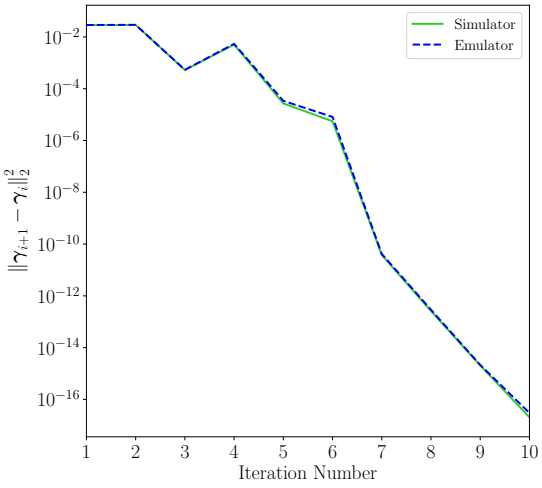


Figure 5.3 – The L_2-norm calculated between the difference vector γ_{i+1} and γ_i (optimisation loss) for 10 iterations. The parameters, $\hat{\gamma}$ converges quickly to the optimal solution using either the simulator or the emulator.

The next question is: how do we choose an optimal $\delta\gamma$ and $\hat{\gamma}$? The latter is obtained in terms of the $\mathbf{H}$ and $\boldsymbol{g}$ by computing the derivatives of the right-hand side of the above equation and setting it to zero such that

$$\delta\gamma = -\mathbf{H}^{-1}(\hat{\gamma})\, g(\hat{\gamma})\,. \tag{5.4.2}$$

Given an initial guess for $\hat{\gamma}_0$, an iterative scheme can be used to find the minimum of $\mathcal{L}$

$$\hat{\gamma}_{n+1} = \hat{\gamma}_n - \mathbf{H}^{-1}(\hat{\gamma}_n)\, g(\hat{\gamma}_n)\,. \tag{5.4.3}$$

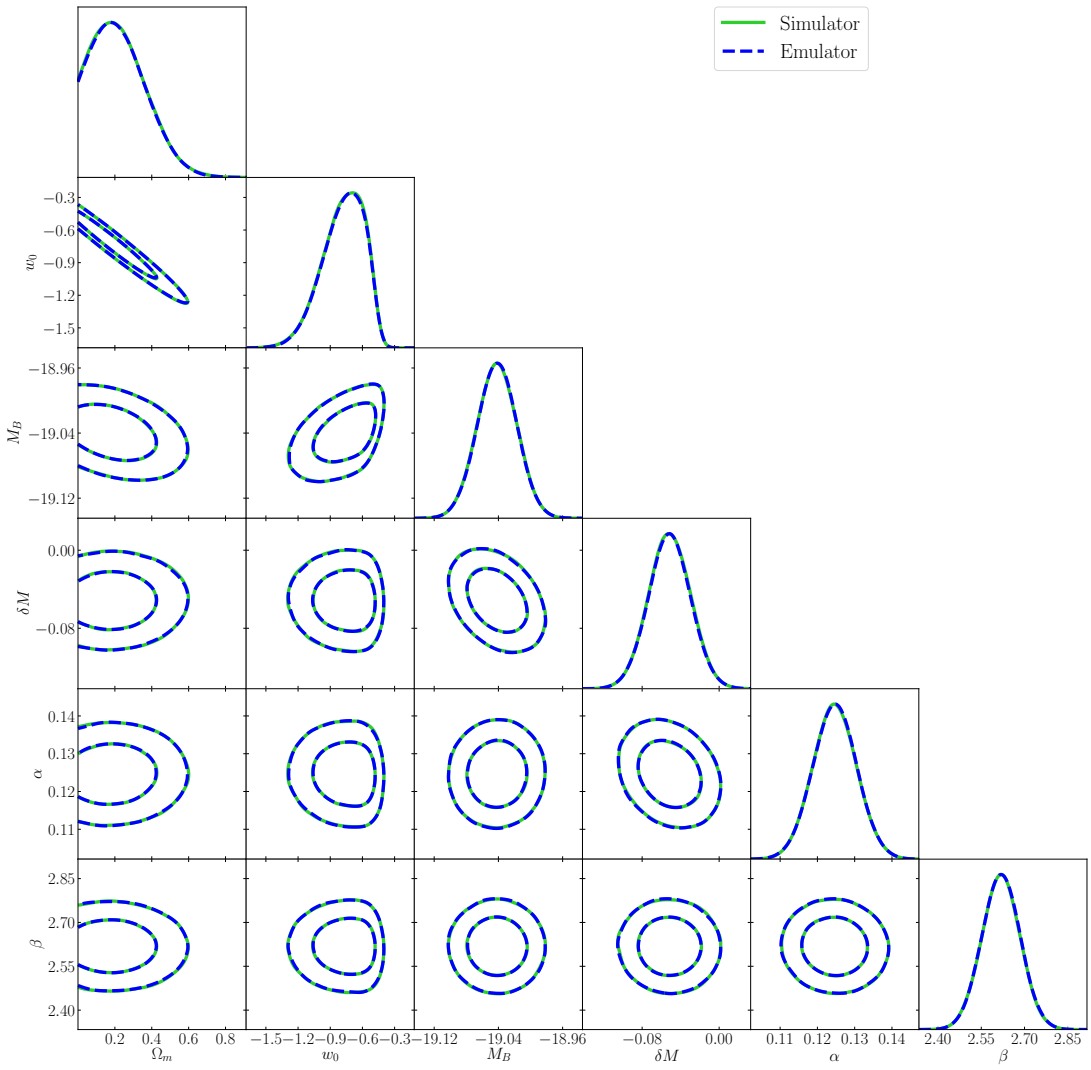


Figure 5.4 – The optimised solutions obtained using the simulator and emulator. Once we obtain the iterative solution, $\hat{\gamma}$ and the inverse of the Hessian matrix for the parameters, $\hat{\mathbf{C}}_\gamma$, we draw 100000 samples from a multivariate normal distribution with mean and covariance, $\gamma \sim \mathcal{N}(\hat{\gamma}, \hat{\mathbf{C}}_\gamma)$. These samples are then used to obtain the above contours (68% and 95% interval) which give a rough idea of the maximum likelihood estimates. Note that both the simulator and the emulator converge to the following solution: $\hat{\gamma} = (0.178, -0.710, -19.039, -0.052, 0.125, 2.618)$.

This is the well-known Newton's method for optimization. The gradient and Hessian are given by Equations (5.1.4) and (5.1.5) respectively. A missing ingredient is the calculation of $\boldsymbol{\mu}_{,i}$ and $\boldsymbol{\mu}_{,ij}$, for which we use the following finite difference formula

$$\begin{aligned}\boldsymbol{\mu}_{,ij} \approx \frac{1}{2\delta\gamma_i\delta\gamma_j}\Big[&\boldsymbol{\mu}\left(\gamma_i+\delta\gamma_i,\gamma_j+\delta\gamma_j\right)-\boldsymbol{\mu}\left(\gamma_i+\delta\gamma_i,\gamma_j\right)\\ &-\boldsymbol{\mu}\left(\gamma_i,\gamma_j+\delta\gamma_j\right)+2\boldsymbol{\mu}\left(\gamma_i,\gamma_j\right)\\ &-\boldsymbol{\mu}\left(\gamma-\delta\gamma_i,\gamma_j\right)-\boldsymbol{\mu}\left(\gamma_i,\gamma_j-\delta\gamma_j\right)\\ &+\boldsymbol{\mu}\left(\gamma_i-\delta\gamma_i,\gamma_j-\delta\gamma_j\right)\Big].\end{aligned} \tag{5.4.4}$$

Note that we can also get $\boldsymbol{\mu}_{,i}$ using central difference method from the above formulae. In particular,

$$\boldsymbol{\mu}_{,i} \approx \frac{\boldsymbol{\mu}\left(\gamma_i+\delta\gamma_i,\gamma_j\right)-\boldsymbol{\mu}\left(\gamma_i-\delta\gamma_i,\gamma_j\right)}{2\delta\gamma_i} \tag{5.4.5}$$

Also, these formula are used only at the level of the cosmological model, $\boldsymbol{u}(\boldsymbol{\theta})$ since the latter is a non-linear function of the cosmological parameters. Moreover, $\boldsymbol{v}(\boldsymbol{\eta})$ is a Gaussian Linear Model and hence all derivatives can be done analytically. Some of the expressions also simplify to a large extent depending on the parameters considered. For example, $\boldsymbol{\mu}_{,\alpha\alpha}=0$. This results in simpler analytical expressions for the gradient and Hessian.

We apply the Cauchy convergence criterion, $\|\gamma_{i+1}-\gamma_i\|_2^2 < \epsilon$ where $\|\cdot\|_2^2$ denotes the L_2 norm and we choose a strict value for $\epsilon = 10^{-16}$. Both the emulator and the simulator converge to the same value after 10 iterations only (see Figure 5.3). As seen in Figure 5.4, both methods (the simulator or the emulator) give an estimate of the uncertainty solely from the optimization procedure depicted above.

At this point, we adopt a simple approach and fix the covariance matrix at the optimised solution for $\alpha = 0.125$ and $\beta = 2.618$ to enable us to use the MOPED algorithm to compress the data. Parameter inference with parameter dependent covariance matrix under the MOPED formalism has recently been explored by Heavens et al. (2017b).

5.5 Inference

From the above demonstration, we now have a robust surrogate model which we can use to do parameter inference. The priors for the cosmological parameters are similar to Alsing et al. (2018) who used hard-cut prior boundaries for Ω_m and w_0, alongside their Gaussian priors. From the optimisation step above, the widths of the priors of the nuisance parameters essentially encompass all of the likelihood. Also, in §5.5.5, we will make use of this prior information to analytically derive the joint posterior distribution of the cosmological parameters, by marginalising over the nuisance parameters, $\boldsymbol{\eta}$. In particular, we assume uniform priors on the

cosmological parameters and the nuisance parameters, we assume Gaussian distribution with mean, corresponding to roughly the optimal solution and unit variance, that is

$$\begin{aligned} \Omega_m &\sim \mathcal{U}\left[10^{-3}, 0.6\right], \\ w_0 &\sim \mathcal{U}\left[-1.5, 0.0\right], \\ M_B &\sim \mathcal{N}\left[-19.0, 1.0\right], \\ \delta M &\sim \mathcal{N}\left[0.0, 1.0\right], \\ \alpha &\sim \mathcal{N}\left[0.10, 1.0\right], \\ \beta &\sim \mathcal{N}\left[2.50, 1.0\right]. \end{aligned} \tag{5.5.1}$$

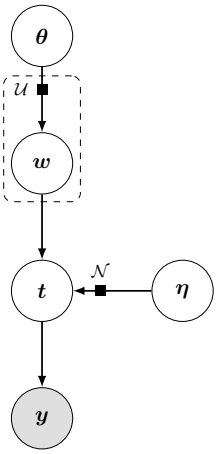


Figure 5.5 – The forward modelling scheme can be understood as follows: we have the first set of parameters $\boldsymbol{\theta}$ which generates probabilistic weights $\boldsymbol{w}_{(m)}$ and a separate independent draw of the nuisance parameters $\boldsymbol{\eta}$. Coupled with the MOPED vectors, the probabilistic theoretical prediction is given by: $\boldsymbol{t} = \mathbf{B}^{\mathrm{T}}\boldsymbol{\mu}$ and $\boldsymbol{y} = \mathbf{B}^{\mathrm{T}}\boldsymbol{d}$ is the fixed MOPED data vector.

Next, we define MOPED data vector as $\boldsymbol{y} = \mathbf{B}^{\mathrm{T}}\boldsymbol{d}$ and the probabilistic prediction $\boldsymbol{t} = \mathbf{B}^{\mathrm{T}}\boldsymbol{\mu}$. The latter is a result of the probabilistic weights $\boldsymbol{w}_{(m)}$. From the directed acyclic graph in Figure 5.5, the joint probability distribution can be written as

$$p\left(\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{t}, \boldsymbol{y}\right) = p\left(\boldsymbol{y}\,|\boldsymbol{t}\right) p\left(\boldsymbol{t}\,|\boldsymbol{\theta}, \boldsymbol{\eta}\right) p\left(\boldsymbol{\theta}\right) p\left(\boldsymbol{\eta}\right). \tag{5.5.2}$$

By product rule, we also have

$$p\left(\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{t}\,|\boldsymbol{y}\right) = \frac{p\left(\boldsymbol{y}\,|\boldsymbol{t}\right) p\left(\boldsymbol{t}\,|\boldsymbol{\theta}, \boldsymbol{\eta}\right) p\left(\boldsymbol{\theta}\right) p\left(\boldsymbol{\eta}\right)}{p\left(\boldsymbol{y}\right)}. \tag{5.5.3}$$

Therefore,

$$p\left(\boldsymbol{\theta}, \boldsymbol{\eta}\,|\boldsymbol{y}\right) = \frac{p\left(\boldsymbol{\theta}\right)p\left(\boldsymbol{\eta}\right)}{p\left(\boldsymbol{y}\right)}\int p\left(\boldsymbol{y}\,|\boldsymbol{t}\right)p\left(\boldsymbol{t}\,|\boldsymbol{\theta}, \boldsymbol{\eta}\right)d\boldsymbol{t} \tag{5.5.4}$$

where the denominator, $p(\boldsymbol{y})$, is simply a constant which does not depend on $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$. It is therefore irrelevant in this case. The integral above is another Gaussian distribution in $\boldsymbol{y}$, that is,

$$p\left(\boldsymbol{\theta}, \boldsymbol{\eta}\,|\boldsymbol{y}\right) \propto \mathcal{N}(\mathbf{B}^{\mathrm{T}}\boldsymbol{\mu}, \mathbb{I} + \sigma^2\mathbf{B}^{\mathrm{T}}\tilde{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Sigma}\tilde{\boldsymbol{\Phi}}\mathbf{B})p\left(\boldsymbol{\theta}\right)p\left(\boldsymbol{\eta}\right) \tag{5.5.5}$$

where $\boldsymbol{\Sigma}$ is a 5×5 diagonal covariance matrix containing the uncertainties of the GP. Therefore, following the above procedures, we can also propagate the GP uncertainty in the analysis. Note that, equation 5.5.5 has an interpretation. If the function is perfectly reconstructed, $\boldsymbol{\Sigma} \to \mathbf{0}$ and we recover the standard MOPED formalism, where the covariance matrix is just a diagonal matrix in the likelihood. This joint posterior distribution can then be sampled using standard MCMC method (see §5.6 for a detailed explanation).

5.5.1 Compression and Emulation Step

In this section, we discuss how the MOPED coefficients can be used to accelerate parameter inference via emulation. Although the MOPED algorithm allows for quick likelihood computation, one would still need to compute the model at a given point in parameter space when running an MCMC. As discussed in §5.3.3, in general, each model evaluation can be quite expensive. However, each MOPED output, t_p is a continuous function over the input domains $(\boldsymbol{\theta}, \boldsymbol{\eta})$. Therefore, we propose the following emulating scheme

1. draw N_{train} parameters from some prior distribution,
2. compute the theoretical model at these points,
3. compress the data/model using the MOPED algorithm and
4. emulate each output with a GP

It is crucial to note that each output is dependent on all the input parameters, that is, $t_p = f(\boldsymbol{\theta}, \boldsymbol{\eta})$. Also note the importance of the compression step, that is, we always have only p Gaussian Processes for the p input parameters.

However, as in *any* Machine Learning algorithm, one question is the optimal location of the training points. Indeed, since the parameter space gets larger and larger with increasing

dimensions, building the emulator with training points generated from the prior volume is challenging in various ways. From an algorithmic perspective, the GP becomes impractical for large number of training points, $\sim \mathcal{O}(1000)$ and might not even be faster than the full forward simulator. However, this also depends on the context we are building the emulator. A single forward simulation might take hours or even days to run. In addition, imagine a scenario where we are distributing $N_{\rm train}$ training points generated from a unit Gaussian distribution in the parameter space while the underlying true posterior distribution of that parameter, assuming it is a Gaussian, has a width 10 times smaller. In other words, we will be placing training points which will be sub-optimal and will lead to a poor function reconstruction with the GP.

To circumvent these potential pitfalls and to simultaneously improve the emulator, we keep the same uniform prior for generating the training points for our cosmology while for the nuisance parameters, we draw training points from an uncorrelated Gaussian distribution with 2 times the covariance of the MLE solution. A similar approach was adopted by Auld et al. (2007) who developed a neural network algorithm for accelerating cosmological parameter inference for the Cosmic Microwave Background, CMB.

In the next section, we illustrate two methods which exploit sophisticated techniques such as experimental design to reduce the number of forward simulations to a manageable number.

5.5.2 Method 1 - LHS (2D)

For this particular problem and using the same idea developed in §5.3.3, the functional form of our model can be written as the sum of two completely disjoint models. In other words, equation 5.3.4 can be written as

$$\boldsymbol{m}_B = \boldsymbol{u}(\boldsymbol{\theta}) + \boldsymbol{v}(\eta), \tag{5.5.6}$$

where $\boldsymbol{u}(\boldsymbol{\theta}) = 5\log_{10} D_L(\boldsymbol{\theta})$ and contains only the cosmology, that is, $\boldsymbol{\theta} = (\Omega_m, w_0)$ while $\boldsymbol{v}(\boldsymbol{\eta}) = M_B + \delta M \boldsymbol{s} - \alpha \boldsymbol{x}_1 + \beta \boldsymbol{C}$ is a function of the nuisance parameters only, that is, $\boldsymbol{\eta} = (M_B, \delta M_B, \alpha, \beta)$. Let us now assume that the MOPED vectors are pre-computed and the compression leads to

$$\begin{aligned} \langle \boldsymbol{y} \rangle &= \mathbf{B}^{\rm T} \boldsymbol{u}(\boldsymbol{\theta}) + \mathbf{B}^{\rm T} \boldsymbol{v}(\eta) \\ &= \tilde{\boldsymbol{u}}(\boldsymbol{\theta}) + \tilde{\boldsymbol{v}}(\eta). \end{aligned} \tag{5.5.7}$$

Note that $\tilde{\boldsymbol{u}}$ is a vector with p numbers but each is a function of just 2 parameters, Ω_m

and w_0. The expensive part of the calculation resides in computing the cosmological model, for example, querying CLASS (Lesgourgues, 2011) at every step in an MCMC for inferring cosmological parameters in a weak lensing analysis.

In short, this is only a 2D problem and does not require many forward simulations. Therefore for this part, we use only 300 Latin Hypercube samples (LHS) to model each y_p by a Gaussian Process.

5.5.3 Method 2 - LHS (6D)

In the previous method, the fact that the final model is just the sum of two functions largely simplify our task since we are able able to focus only on the expensive part of the calculation. However, there are cases where one would need to model the cosmological parameters and the systematics simultaneously. An example of this is the computation of the weak lensing power spectrum where one has an additional nuisance parameter due to baryon feedback (Köhlinger et al., 2017).

Therefore, in this section, we will directly emulate each y_p, as a function of both the cosmological $(\boldsymbol{\theta})$ and nuisance $(\boldsymbol{\eta})$ parameters, using Gaussian Processes. However, the fact that we now have extra dimensions in our emulator compared to the method presented in §5.5.2, there is the need for additional forward simulations to adequately improve the performance of the emulator. In this case, we use a total of 700 forwards simulations to model each y_p and example of the GP emulator for each MOPED coefficient across a slice in parameter space is shown in Figure 5.6.

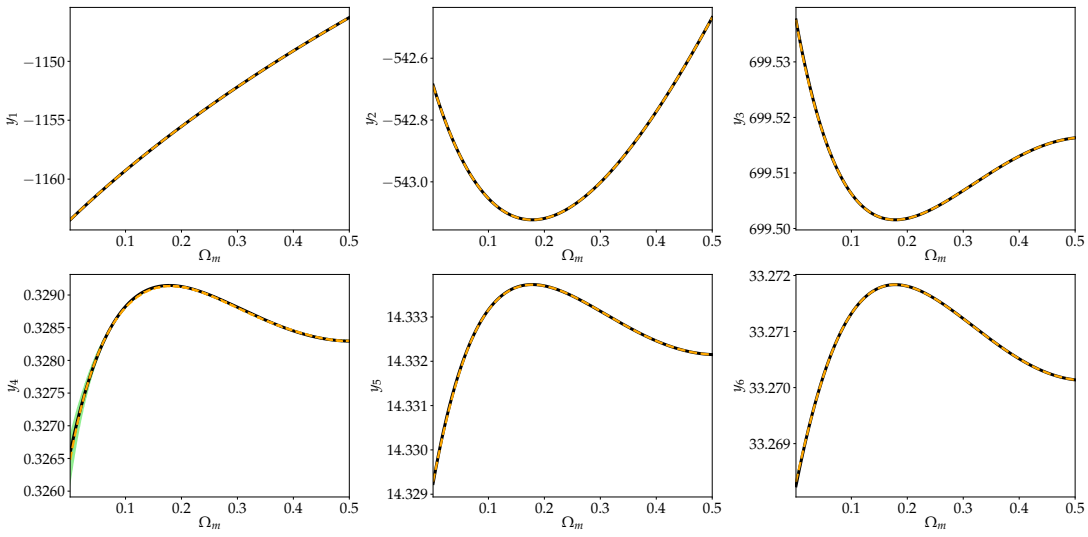


Figure 5.6 – Plot showing the prediction, using the method presented in §5.5.3 from each emulated MOPED output as a function of Ω_m across a slice in parameter space. The orange curves refer to the emulator while the black curves correspond to the output from the full simulator. The green shading represents the 3σ uncertainty from the Gaussian Process.

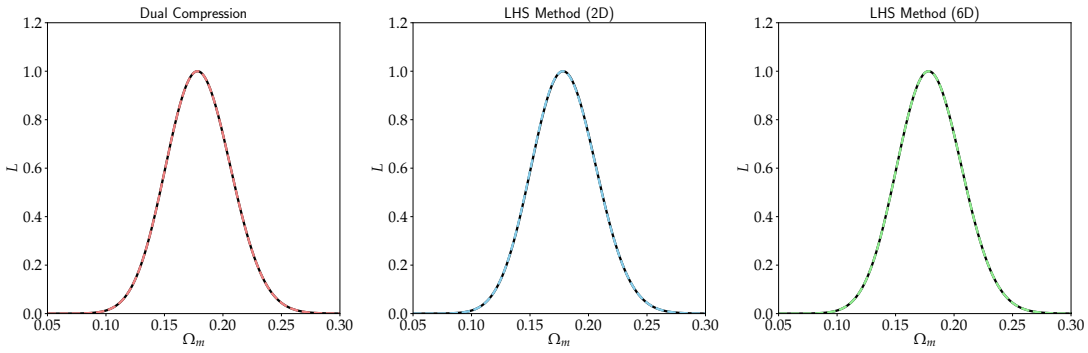


Figure 5.7 – Plot of the likelihood function evaluated across a slice in parameter space for the three different methods presented in §5.3.3, §5.5.2 and §5.5.3. The black curve shows the likelihood calculated using the full simulator while the red, blue and green curves are the likelihoods calculated using the dual compression approach in §5.3.3, the LHS 2D method (§5.5.2) and LHS 6D method (§5.5.3).

5.5.4 Likelihood Regressor Approach

An enticing and emerging technique in cosmology is likelihood-free inference. In this section, we briefly cover this topic and we use the emulator to to instead approximate the log-likelihood, that is, essentially constructing a likelihood regressor. In many Bayesian Inference problems, the full likelihood function might be poorly understood. Approximate Bayesian Computation (ABC), also often referred to as a rejection sampling technique, provides a satisfactory approach to alleviate the issue of intractable likelihood.

In a nutshell, the idea behind most ABC algorithm is to get a representation of the approximate posterior distribution of the parameters by choosing the parameters that produce simulated data to be close enough to the observed data. More explicitly, a sample is drawn from the prior followed by a draw of the simulated data, $\boldsymbol{d}_s$ from the likelihood and the sample is accepted if $\rho(\boldsymbol{d}_s, \boldsymbol{d}_o) < \epsilon$. $\boldsymbol{d}_o$ is the observed data in this case (Alsing et al., 2018).

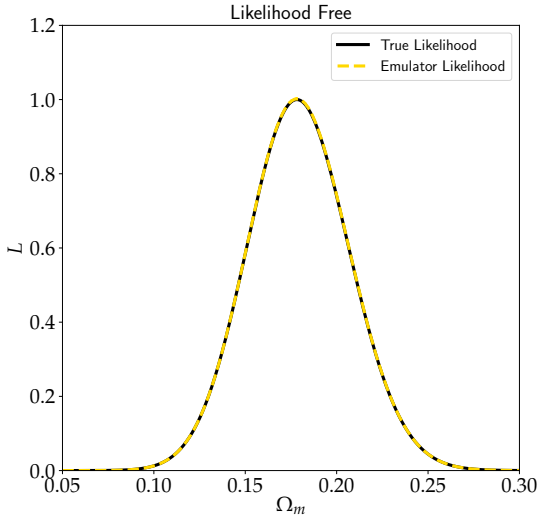


Figure 5.8 – Similar to Figure 5.7, for the same slice in parameter space, we use our emulator for the likelihood regressor approach to compute the likelihood. The true likelihood and emulator likelihood are shown in black and yellow respectively.

As discussed by Leclercq (2018), ABC has various limitations. For example, how do we define the distance metric, ρ between the simulated data and the simulated data, how small should ϵ be? If the latter is too small, many samples are rejected and this is not what we would ideally want because a single realisation of the simulated data might be computationally expensive. In addition to this, ABC does not use prior information about samples which are already accepted according to the distance metric.

While Figure 5.7 compares the likelihood, as calculated across a slice through the $\Omega_{\rm m}$ parameter, between the emulator and the simulator, Figure 5.8 shows the likelihood of the emulator (likelihood regressor) when compared to the exact likelihood using the MOPED compression scheme.

5.5.5 Analytical Marginalisation

If we are solely interested in the cosmological parameters, we can analytically marginalise over the nuisance parameters, $\boldsymbol{\eta}$. In this particular, we exclude the MOPED compression step and the joint posterior of Ω_m and w_0 is given by:

$$p\left(\boldsymbol{\theta}\,|\boldsymbol{d}\right) \propto \exp\left[-\frac{1}{2}(\boldsymbol{f}-\boldsymbol{\mu}_f)^{\mathrm{T}}\mathbf{G}(\boldsymbol{f}-\boldsymbol{\mu}_f)\right]\, p\left(\boldsymbol{\theta}\right) \tag{5.5.8}$$

where

$$\mathbf{G} = (\mathbf{C}+\boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathrm{T}})^{-1}$$
$$\boldsymbol{\mu}_f = \mathbf{G}^{-1}\mathbf{C}^{-1}\boldsymbol{\Psi}(\boldsymbol{\Psi}^{\mathrm{T}}\mathbf{C}^{-1}\boldsymbol{\Psi}+\mathbb{I})^{-1}\bar{\boldsymbol{\eta}}$$

$\bar{\boldsymbol{\eta}}$ is the mean of the Gaussian prior for the nuisance parameters and $\boldsymbol{f}(\boldsymbol{\theta}) = \boldsymbol{d}-\boldsymbol{u}(\boldsymbol{\theta})$. At this point, we can either use the simulator or the emulator to model $\boldsymbol{u}(\boldsymbol{\theta})$. Recall that for the simulator $\boldsymbol{u}(\boldsymbol{\theta}) = 5\log_{10}D_L(\boldsymbol{\theta})$ whereas for the emulator, $\boldsymbol{u}(\boldsymbol{\theta}) = \sigma\tilde{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{w}(\boldsymbol{\theta})+\bar{\boldsymbol{u}}$.

If we use the compressed version of the data, this results in a slightly different formula for the joint posterior. Defining the matrix $\mathbf{A} = \mathbb{I}+\sigma^2\mathbf{B}^{\mathrm{T}}\tilde{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{\Sigma}\tilde{\boldsymbol{\Phi}}\mathbf{B}$:

$$p\left(\boldsymbol{\theta}\,|\boldsymbol{y}\right) \sim \exp\left[-\frac{1}{2}\left(\boldsymbol{h}-\boldsymbol{\mu}_h\right)^{\mathrm{T}}\mathbf{F}\left(\boldsymbol{h}-\boldsymbol{\mu}_h\right)\right]\, p\left(\boldsymbol{\theta}\right) \tag{5.5.9}$$

where

$$\mathbf{F} = (\mathbf{A}+\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathrm{T}}\mathbf{B})^{-1}$$
$$\boldsymbol{\mu}_h = \mathbf{F}^{-1}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}(\mathbb{I}+\boldsymbol{\Psi}^{\mathrm{T}}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\bar{\boldsymbol{\eta}}$$

and $\boldsymbol{h}(\boldsymbol{\theta}) = \boldsymbol{y} - \mathbf{B}^{\mathrm{T}}\boldsymbol{u}(\boldsymbol{\theta})$ and if we exclude the Gaussian Process uncertainty or if we use the simulator directly, then,

$$\mathbf{F} = (\mathbb{I} + \mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathrm{T}}\mathbf{B})^{-1}$$

$$\boldsymbol{\mu}_h = \mathbf{F}^{-1}\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Psi}^{\mathrm{T}}\mathbf{B}\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi} + \mathbb{I})^{-1}\bar{\boldsymbol{\eta}}$$

Equations 5.5.8 and 5.5.9 have the same analytic form but the MOPED formalism really shines if we were to use Equation 5.5.9. $\mathbf{G}$ is of size $N \times N$ whereas $\mathbf{F}$ is just of size $p \times p$. Recall that in our case, $N = 740$ and $p = 6$. While we have provided a full analytical marginalisation for the nuisance parameters here, in the next section we will highlight the main results obtained when doing the marginalisation numerically, via Monte Carlo sampling and we will also discuss the performance of the emulator.

5.6 Results and Performance

We further perform a series of diagnostics in order to understand the performance of various emulators in this study. For the dual compression approach, before sampling the joint posterior distribution, we use the surrogate model to substitute the full simulator in the optimisation procedure.

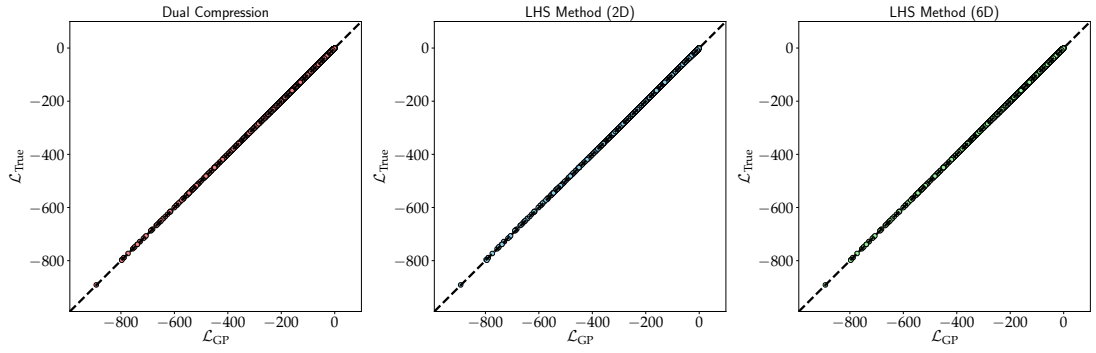
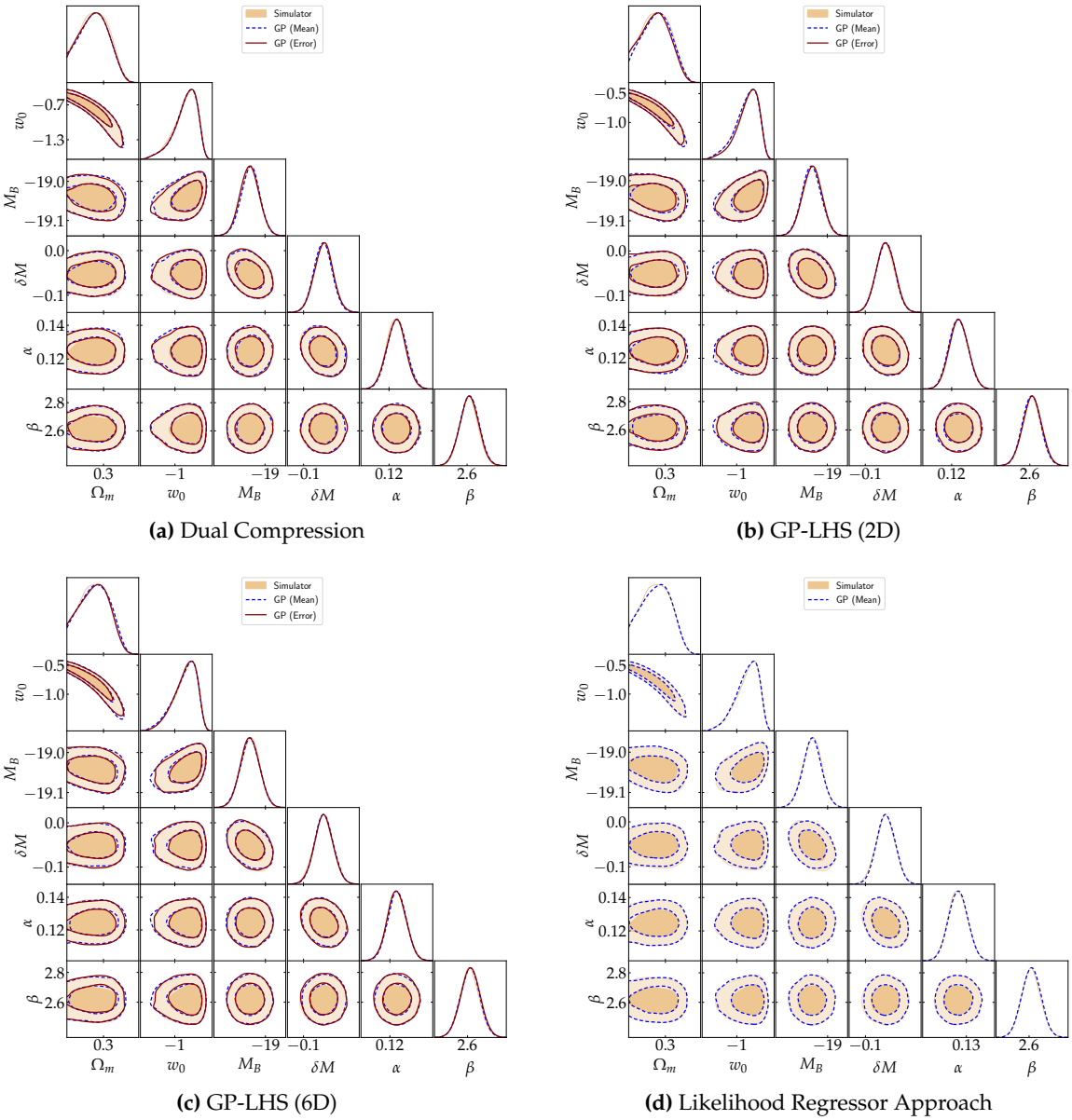


Figure 5.9 – For an independent set of 12000 test points, we also compute the likelihood using the three different methods in §5.3.3, §5.5.2 and §5.5.3 respectively. The fact that the emulator using either method is well reconstructed leads to an almost perfect likelihood as shown in this figure.

We can also test the likelihood values at an independent set of test points. This so-called hold-out test is useful to assess the performance of our emulator models. Hence, we plot the true likelihood against the predicted likelihood using the Gaussian Process emulator in Figure 5.9 for 3 different methods (§5.3.3, §5.5.2 and §5.5.3) we considered.

There are two main computationally expensive components in most inference machinery. The first part is the theory evaluation itself and the second part being the likelihood calculation.

For the particular problem considered in this work, we have introduced a latent probabilistic representation of the function which is quick to compute. Coupled to it, is MOPED which essentially provides a work-around the $\mathcal{O}(N^3)$ cost in the likelihood evaluation. The two pieces combined, makes parameter inference $\sim 23-26$ times faster depending on whether we use the mean only or both the mean and variance from the GP emulator.



(a) Dual Compression

(b) GP-LHS (2D)

(c) GP-LHS (6D)

(d) Likelihood Regressor Approach

Figure 5.10 – The panel in (a) corresponds to the marginalised posterior distributions using the method discussed in §5.3.3 and panel (b) refers to the posterior distribution with the method presented in §5.5.2. Similar plots are shown in panels (c) and (d), corresponding to methods discussed in §5.5.3 and §5.5.4 respectively. In particular, for (a), (b) and (c), it is possible to propagate the Gaussian Process uncertainty in the inference, hence we show two contours, where the solid one refers to the case where the Gaussian Process uncertainty is used and the broken one where only the mean of the Gaussian Process is used. We also compare the posterior obtained with the full simulator and as shown here, the posterior distributions using either emulator are barely discernible relative to the true posterior. The inner and outer contours correspond to the 68% and 95% credible interval respectively.

If one uses prior information, such as the Hessian matrix obtained from a gradient descent

algorithm or previous experiments to place the training points, the number of training points for the emulator can be reduced significantly, hence alleviating the $\mathcal{O}(N^3)$ cost when training the Gaussian Processes. Recall that in our case the training points are distributed across the whole pre-defined prior range for our cosmology.

Table 5.6.1 – Performance of the emulators relative to the simulator

Method	N_{train}	**Training** (s)	**Memory** (MB)	**Inference** (s)	$\log(Z)$	$\lvert\log(B_{01})\rvert$
Simulator (with MOPED)	-	-	-	3900	-19.225	-
Dual Compression (2D): Mean	300	3	0.6	85	-19.240	0.029
Dual Compression (2D): Error	-	-	-	136	-19.216	0.005
GP-LHS (2D): Mean	300	2	0.7	90	-19.236	0.011
GP-LHS (2D): Error	-	-	-	130	-19.255	0.030
GP-LHS (6D): Mean	700	120	24	100	-19.223	0.002
GP-LHS (6D): Error	-	-	-	424	-19.259	0.034
GP (6D): Likelihood Regressor	700	28	4	60	-19.246	0.015

Note: The number of training points/forward simulations is shown in the second column. The third column provides the time taken to train the Gaussian Processes while the fourth column shows the memory consumption for the different types of emulators. In the fifth column, we have the time taken to obtain 120 000 MCMC. Once we have the samples from each method, we use MCEvidence (Heavens et al., 2017a) to compute the log-evidence, shown in the sixth column and we then compute the log-Bayes-Factor relative to the full simulator.

Figure 5.10 shows the marginalised posterior distribution for all the different cases investigated in this chapter. In particular, the top left and right panels show the results obtained using dual compression approach and the case where we emulate only p MOPED coefficients due to the two cosmological parameters, Ω_{m} and w_0. The bottom left and right panels show the marginalised posterior distribution when we emulate the MOPED coefficients directly (a 6D problem) and the case where we build a likelihood regressor.

However, one should not erroneously be led to the conclusion that this can easily be extended to higher dimensions. In the latter, one could presumably use intelligent designs such as the Latin Hypercube sampling, as we do in this work, to generate a few hundreds of training points and build surrogate models with Gaussian Processes but the predictive uncertainty for a test point located far from the training set will in general be large. This arises due to the fact that the distribution of the training points become sparser as the dimensionality of the problem increases. In other words, perfect reconstruction (precise and accurate) of the original function is not a trivial task in higher dimensions because of the *curse of dimensionality*. One way to avoid this is to add more and more training points, but unfortunately, the complexity in terms

of training and storage of the Gaussian Processes increases. Nonetheless, this curse is not solely inherent to Gaussian Processes but to many Machine Learning algorithms.

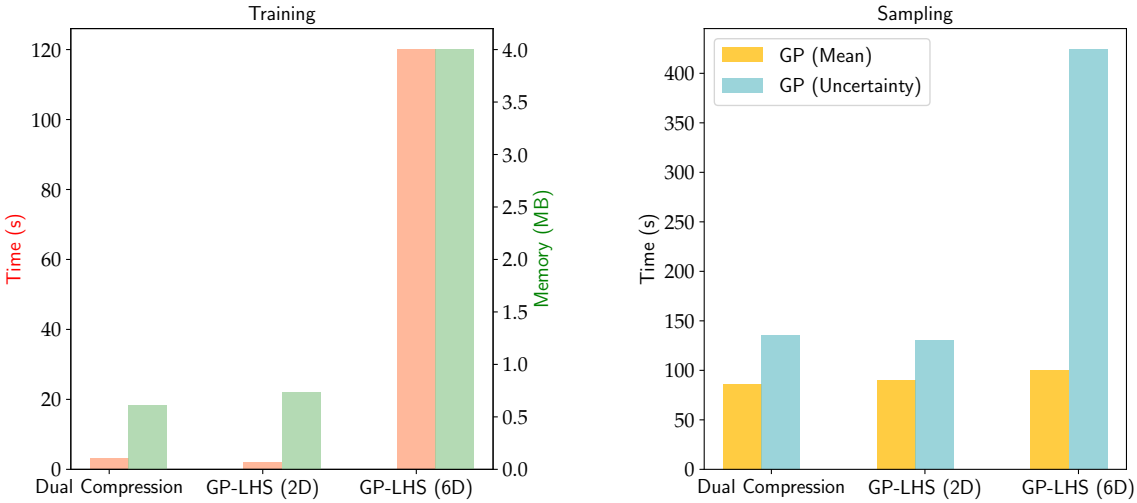


Figure 5.11 – The top panel shows the performance of each method during the training phase. In particular, the partitioning method is the most expensive part, not only in terms of time but also memory. The bottom panel shows the time in seconds to generate 120 000 MCMC samples. The fact that including the GP uncertainty in the inference step is not surprising because of an extra $\mathcal{O}(N^3)$ operation at each step for calculating the predictive variance.

Figure 5.11 shows the different performance of the different emulators. The top panel gives an indication of the time it takes to train the emulators and also the memory footprint. The lower panel indicates the time taken to sample the full posterior distribution when using the different emulators. It is expected that including the GP uncertainty, especially, when we have p separate Gaussian Processes will be expensive. However, using the mean of the emulator leads to very quick results, that is, it takes around 90 seconds to generate 120 000 MCMC samples compared to the full simulator which takes around one hour, if we were to use the uncompressed dataset.

5.7 Related Work and Discussion

In the same spirit of likelihood regressor inference, Alsing et al. (2018) introduced a novel method for performing parameter inference using Density-Estimation likelihood free Inference (DELFI) and has shown to outperform existing method such as Population Monte Carlo Approximate Bayesian Computation (PMC-ABC) when comparing the number of forward simulations required in each case. Moreover, the authors extended their work in Alsing & Wandelt (2019) by explicitly marginalising over the nuisance parameters and in Alsing et al. (2019), the posterior distribution of cosmological parameters were inferred in an active learning scenario using neural networks. These techniques are generally quite robust and require of the $\mathcal{O}(1000)$

forward simulations only for a typical ~ 10 dimensional problem.

On the other hand, Leclercq (2018) proposed Bayesian Optimisation for likelihood regressor Inference (BOLFI) to infer cosmological parameters using the JLA data. The technique presented in that work seeks to learn a smooth function for the likelihood in an active learning scenario using Bayesian Optimisation. 6000 forward simulations were used to infer the 2 cosmological parameters, with partial marginalisation of the nuisance parameters which yield smaller contours. Bayesian Optimisation technique also relies on using Gaussian Process to build a surrogate model of the likelihood and one can expect that the active learning procedure becomes increasingly and inherently more expensive as training points are added to improve the latent function. This is because, the posterior distribution of the Gaussian Process needs to be updated and this requires re-training, which involves an $O(N^3)$ each time the training set is augmented. In other words, the kernel hyperparameters need to be updated for every update in the training set.

Similar to the idea presented in §5.5.3, instead of emulating each MOPED output with a Gaussian Process, one could also emulate the likelihood directly. In this case, we simply use the expected MOPED numbers, $\langle \boldsymbol{y} \rangle$ in §5.5.3, evaluated at the 700 LH samples to compute the log-likelihood. The latter, together with the 700 LH samples, are then used to construct a surrogate model for the log-likelihood using Gaussian Processes.

This technique has various advantages and disadvantages. Emulating the likelihood directly, sidesteps the impediment of training many Gaussian Processes. This implies that we have to query a single surrogate model at each step in an MCMC and hence makes parameter inference significantly fast. The trained model also has less storage requirement compared to emulating many functions. Moreover, since Gaussian Process has this nice feature of being infinitely differentiable also suggests that the likelihood regressor could potentially be used as a proxy for doing optimisation. In the same spectrum, an important quantity for gradient-based Monte Carlo sampler, such as Hamiltonian Monte Carlo (HMC) is the gradient of the potential energy, the negative log-likelihood. One could therefore use Gaussian Process as a proxy to obtain an analytical expression for the gradient.

While the likelihood regressor method works well, it is not without problem. In particular, the uncertainty from the likelihood regressor will not be useful, unless one considers a scenario where training points are actively added (Leclercq, 2018). On the other hand, the likelihood is a function of all the parameters in the model. This results in a high dimensional space problem which is more likely to hinder the performance of not only the Gaussian Process, but many

other Machine Learning algorithms in general.

5.8 Summary

We briefly summarize our analysis in this section. We have shown how we can use a dual compression approach to infer the cosmological and nuisance parameters for the JLA dataset. We have also shown that by using the MOPED formalism, at the level of the most expensive part of the calculation, we can significantly accelerate the computation. If one chooses to emulate the function as a whole, this results in a 6-dimensional problem and requires more training point. Alternatively, instead of building multiple emulators, one can simply emulate the log-likelihood. In all scenarios, we are able to generate robust posterior distributions, reducing the computational time from roughly an hour to just a few minutes. Moreover, to quantify the discrepancy between two joint posterior distribution of parameters, we use MCEvidence and we see from Table 5.6.1 that the values of the evidence do not significantly differ from each other (at one decimal place), hence quantifying that the different emulators are robust.

Chapter 6

PARAMETER INFERENCE FOR WEAK LENSING USING GAUSSIAN PROCESSES AND MOPED

> Technology is just a tool. In terms of getting the kids working together and motivating them, the teacher is the most important
>
> *Bill Gates*

This chapter has been published as a work in Monthly Notices of the Royal Astronomical Society (MNRAS) peer-review journal and was also presented as a short work in the International Conference on Learning Representations (ICLR) conference. The work, code and video for the two works can be accessed through the following:

1. Parameter Inference for Weak Lensing using Gaussian Processes and MOPED
   A. Mootoovaloo, *A. Heavens, A. Jaffe and F. Leclercq*, MNRAS, 497, 2213-2226, 2020

2. Gaussian Processes and MOPED Compression for Weak Lensing
   A. Mootoovaloo, *A. Heavens, A. Jaffe and F. Leclercq*, ICLR Conference

The following content corresponds to a large extract from the MNRAS work. A. Mootoovaloo led the project and code development. A. Heavens, A. Jaffe and F. Leclercq constantly provided feedbacks, guides and ideas to ensure successful completion of the project. This chapter makes use of the KiDS-450 likelihood code*.

6.1 Overview

In this work, we propose a Gaussian Process (GP) emulator for the calculation both of tomographic weak lensing band-powers, and of coefficients of summary data massively compressed

*https://bitbucket.org/fkoehlin/kids450_qe_likelihood_public/

with the MOPED algorithm. In the former case cosmological parameter inference is accelerated by a factor of $\sim$ 10-30 compared with Boltzmann solver CLASS applied to KiDS-450 weak lensing data. Much larger gains of order 10^3 will come with future data, and MOPED with GPs will be fast enough to permit the Limber approximation to be dropped, with acceleration in this case of $\sim 10^5$. A potential advantage of GPs is that an error on the emulated function can be computed and this uncertainty incorporated into the likelihood. However, it is known that the GP error can be unreliable when applied to deterministic functions, and we find, using the Kullback-Leibler divergence between the emulator and CLASS likelihoods, and from the uncertainties on the parameters, that agreement is better when the GP uncertainty is not used. In future, weak lensing surveys such as Euclid, and the Legacy Survey of Space and Time (LSST), will have up to $\sim 10^4$ summary statistics, and inference will be correspondingly more challenging. However, since the speed of MOPED is determined not the number of summary data, but by the number of parameters, MOPED analysis scales almost perfectly, provided that a fast way to compute the theoretical MOPED coefficients is available. The GP provides such a fast mechanism. The data (band powers and covariance matrix) used in this work is described in much details in §4.1 in Chapter 4. Moreover, the Gaussian Process approach adopted in this Chapter is dicussed in Chapter 3.

6.2 Emulator

In this section, we use the formalism presented above to build the emulator. In brief, the latter involves 4 main stages, 1) generating a set of design points, 2) running the full forward simulator at these points, 3) training the emulator and 4) making predictions at test locations in the parameter space. Once this is done, the emulator is connected to an MCMC sampler to obtain the marginalised posterior distributions of the parameters in our model. A simple flow of the core idea is shown in Fig. 6.1. In the following, we touch briefly on the data we have used for our analysis before systematically going through the steps we have taken to build the emulator.

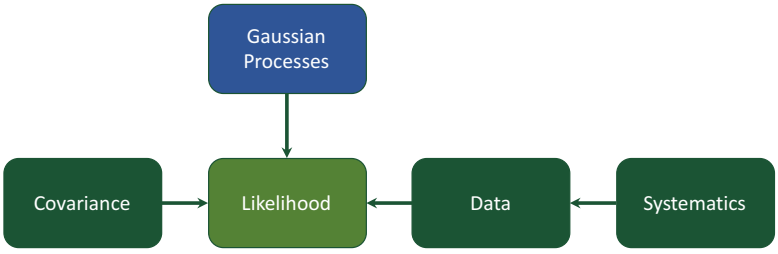


Figure 6.1 – A diagrammatic form of the core principle in this work. We substitute the most expensive part of the pipeline by surrogate models (Gaussian Processes) built at the level of the band powers. The other blocks in the inference procedure, for example, for the computations related to the nuisance parameters, are unaltered.

6.2.1 Data

We use the publicly-available weak lensing data from Köhlinger et al. (2017) to test the performance of our emulator. We use 3 tomographic redshift bins, namely, $0.10 < z < 0.30$, $0.30 < z < 0.60$ and $0.60 < z < 0.90$ and the convergence power spectrum is computed in the range $10 < \ell < 4000$. Moreover, we follow Köhlinger et al. (2017) and drop the first, second-to-last and last band powers in our analysis, that is, we use only the band powers corresponding to the following ℓ-ranges:$76 \leq \ell < 220$, $221 \leq \ell < 420$, $421 \leq \ell < 670$ and $671 \leq \ell < 1310$. For a 3-bin tomographic analysis, we have 6 auto- and cross- tomographic power spectra to calculate. The data and covariance matrix for this problem are shown in Fig. 4.1 and 4.2 respectively.

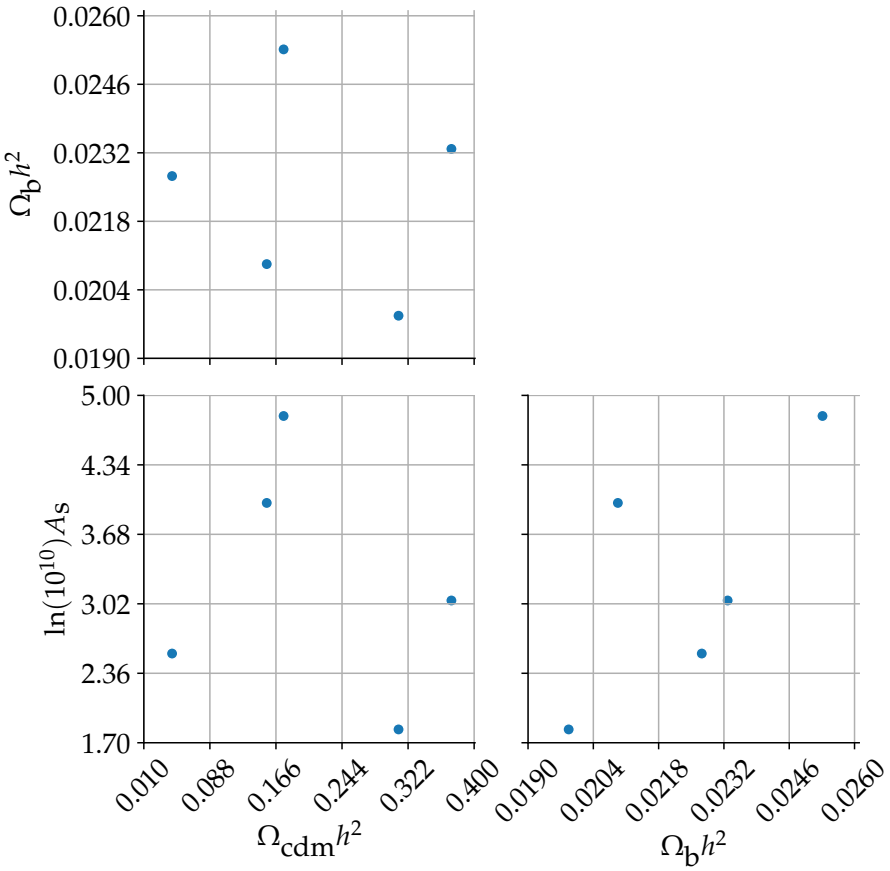


Figure 6.2 – Five Latin Hypercube samples (using the maximin method) projected in 2D. In particular, we generate five Latin Hypercube samples in 8D and we scale them according to our pre-defined priors. In the figure, we show the projection in 2D for 3 parameters and as expected, each point occupies it corresponding row and column.

The emulator can be built at the level of the power spectra or the band powers. Here we choose to build a GP for each band power, giving 24 GPs. Alternatively, for likelihood-free inference methods, one can also emulate the likelihood directly using the GPs (see Leclercq (2018) and Fendt & Wandelt (2007a)). For power spectrum reconstruction, one can use the PICO method or an alternative, but constrictive, stance is to adopt the approach taken by Habib et al.

(2007) to first learn a set of basis functions via Singular Value Decomposition (SVD) and model the resulting weights by a Gaussian Process. However, building an emulator for weak lensing analysis needs to account for systematic effects, but some of these can be included analytically without emulation, resulting in an 8-dimensional GP, rather than 12 (6 cosmological and 6 systematic parameters) if we were to emulate the likelihood.

6.2.2 Training Points

The generation of the training points is a key ingredient for the emulator to perform well. Accurate high-dimensional regression is not easy, mainly due to the curse of dimensionality. With the formalism presented in this work and depending on the complexity of the function, one can reconstruct the function precisely and accurately in low dimensions, hence leading to an accurate likelihood as would be the case if we were to use the full simulator, CLASS (Lesgourgues, 2011) in this case. As the dimensionality of the problem increases, we need an exponentially increasing number of training points to emulate the true function accurately.

In PICO, the training points were generated uniformly from a box whose sides were centred on the mean of a converged MCMC chain (consisting of $\sim$ 60000 cosmological models) and width 3σ along each direction. In the second release of PICO, they selected training points which lie within 25 log-likelihoods of the WMAP peak (Fendt & Wandelt, 2007a). On the other hand, Auld et al. (2007) first drew 2000 training points from the same box defined in PICO and also added an extra 5000 training points drawn from a Gaussian distribution, whose covariance was twice the expected covariance matrix, centred on the maximum likelihood. These techniques perform quite well for two reasons: 1) by restricting the prior volume of the training points to the high likelihood regions allows the sampler explicitly to explore this specific region in parameter space, 2) creating a data set with thousands of training points will also improve *any* regression method. A shortcoming of using these approaches is that the algorithm will not perform well in regions where there is no training point nearby (see Appendix A in Habib et al. (2007) for a comparison of their method with PICO). This is a typical manifestation of almost any Machine Learning algorithm. They are good at making reliable predictions within a pre-defined prior, provided they are trained with enough data points. Building Machine Learning algorithms in the small data regime is still in its infancy, hence an active area of research (Barz & Denzler, 2019).

Moreover, if the training points are naively generated randomly from our pre-defined priors, we might not obtain a suitable coverage of the parameter space. A possible solution to this,

is to use a grid but then the number of training points grows exponentially as the dimensionality of the problem increases. As an example, say, we have a 7D problem and we choose to have 10 points per parameter, then our training set will have 10 million points.

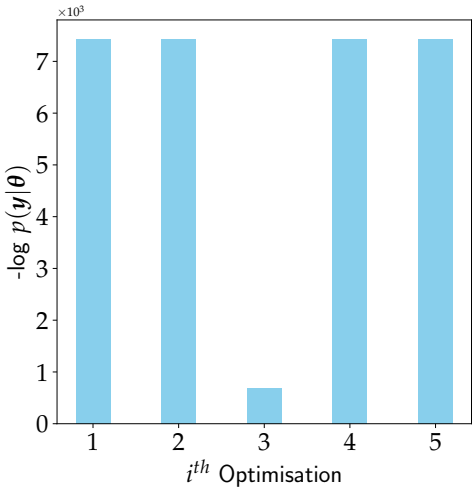


Figure 6.3 – The figure shows the marginal likelihood of the Gaussian Process (with 3000 training points) for the fourth band power matrix and $i = j = 2$. Note the local minimum for the 3$^{\text{rd}}$ run of the optimiser. The other bars have almost the same value, hence showing that $N_{\text{restart}} = 5$ is a good choice for training the GP.

Alternatively, we can use Latin Hypercube (LH) sampling (McKay et al., 1979) which is a method for generating random samples from a multidimensional dimensional distribution in a controlled (quasi-random) way. A point is assigned such that it uniquely occupies its row and column respectively. This procedure generalises to higher dimensional designs. In Fig. 6.2, we show the projection of 5 LH samples, which have been generated from a box in 8D and scaled by the pre-defined priors in §6.2.3, in 2D. In particular, we show the projection for 3 parameters only but the same applies for the other parameters, where each point uniquely occupies its corresponding row and its column. The LH method is now a ubiquitous tool for performing emulation in large simulation scenarios (Habib et al., 2007; Schneider et al., 2011; Schmit & Pritchard, 2018) and is seen to be quite efficient, not only in producing a fair interpolation, but also provides reasonable posterior densities.

In this work, we adopt the LH approach to generate our training set. The LH samples are generated using the `maximinLHS` function from the `lhs` R package (Carnell, 2012). This particular design relies on distance criterion (Johnson et al., 1990) and the final design is a result of maximising the minimum distance between points.

6.2.3 Priors

In our baseline emulator, we generate 1000 Latin Hypercube samples from a box, between 0 and 1. We first linearly transform these samples to the range of the pre-defined prior box for

the 6 cosmological and 2 systematics parameters,

$$\begin{aligned}
\Omega_{\rm cdm}h^2 &\sim \mathcal{U}[0.01, 0.40] \\
\Omega_b h^2 &\sim \mathcal{U}[0.019, 0.026] \\
\ln(10^{10}A_s) &\sim \mathcal{U}[1.70, 5.00] \\
n_s &\sim \mathcal{U}[0.70, 1.30] \\
h &\sim \mathcal{U}[0.64, 0.82] \\
A_{\rm bary} &\sim \mathcal{U}[0.0, 2.0] \\
A_{\rm IA} &\sim \mathcal{U}[-6.0, 6.0] \\
\Sigma m_\nu &\sim \mathcal{U}[0.06, 1.0]
\end{aligned}$$

followed by running the full simulator at these points to obtain the total band powers. $\mathcal{U}[a, b]$ denotes a uniform distribution with lower and upper limits a and b respectively. We apply a more restrictive prior than the original KiDS-450 prior [0.01,0.99] for $\Omega_{\rm cdm}h^2$ since otherwise a large fraction of the LH samples we generate lie outside the region of parameter space constrained by the current weak lensing analysis. Moreover, having a smaller volume of parameter space also improves the performance of the emulator. The prior for the $A_{\rm bary}$ is set to an upper limit of 2 (instead of 10 in Köhlinger et al. (2017)) because we found that, values of $A_{\rm bary} \gtrsim 3$ lead to negative b^2, which implies an unphysical negative power spectrum. In the same spirit, large values of $A_{\rm bary}$ lead to negative auto-correlated band powers and in some cases, the band power matrix (equation (6.2.1)) was not positive definite. We also found that large values of neutrino masses, $\Sigma m_\nu \gtrsim 1$eV result in almost half of the CLASS band powers in our training set to be `nan`. We therefore set an upper limit for Σm_ν to 1 eV.

6.2.4 Transformations

Training the Gaussian Processes with the LH samples from above might be suboptimal, the reason being that the volume occupied by a hypercube grows exponentially with increasing dimensions. On the other hand, a sphered training set (hypersphere) has a smaller volume compared to its corresponding hypercube but with the same scaling with dimension. This transformation step is analogous to the one used by Fendt & Wandelt (2007b). Schneider et al. (2011) assessed in detail the effect of various transformations prior to building an emulator for the CMB power spectrum. They found that de-correlating the input space leads to significant improvements compared to working with the original form of the input parameters. The inter-

polation can further be improved if one uses a known Fisher information matrix specific to the problem.

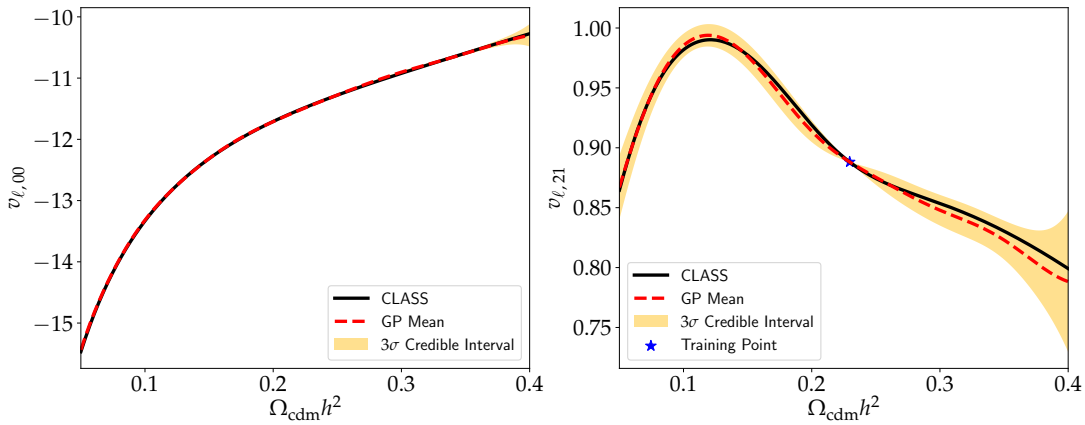


Figure 6.4 – The left plot shows the predicted band power across a slice in parameter space. In other words, we choose a point within the prior box and compute the GP mean, variance and the actual band power for $\Omega_{\rm cdm}h^2 \in [0.05, 0.40]$. The same procedure is repeated in the right plot, but we instead choose a point from a training set, to illustrate the fact that the predicted GP uncertainty tends to zero near the training point and the predictive variance increases towards the edge of the prior box.

The transformation matrix can be calculated as follows: we first compute the sample covariance, $\mathbf{C}_{\boldsymbol{\theta}}$ of the 1000 input parameters, $\boldsymbol{\theta}$ to the emulator (see Table 4.2.1), which we diagonalise, $\mathbf{C}_{\boldsymbol{\theta}} = \mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}$. $\mathbf{U}$ is a $d \times d$ orthonormal matrix and $\mathbf{D}$ is a diagonal $d \times d$ matrix consisting of the (necessarily positive) eigenvalues. The transformation matrix which whitens $\boldsymbol{\theta}$ is then $\mathbf{U}\mathbf{D}^{\frac{1}{2}}$, such that the transformed input covariates are $\mathbf{X} = \mathbf{U}\mathbf{D}^{\frac{1}{2}}\boldsymbol{\theta}$, and the covariance of $\mathbf{X}$ is the identity matrix. Also, having a pre-whitened basis also justifies the use of a diagonal kernel matrix such as the ARD kernel in equation (3.2.11), for which it is often blindly assumed (without transforming the inputs) that the correlation among the input parameters vanishes.

Next, we consider the transformation of the band powers. The distribution of the original band powers in our training set is left-skewed. For a fixed ℓ in our 3-bin tomographic analysis, the resulting 3×3 matrix,

$$\mathbf{B}_\ell = \begin{pmatrix} \mathrm{B}_{\ell,00} & \mathrm{B}_{\ell,01} & \mathrm{B}_{\ell,02} \\ \mathrm{B}_{\ell,10} & \mathrm{B}_{\ell,11} & \mathrm{B}_{\ell,12} \\ \mathrm{B}_{\ell,20} & \mathrm{B}_{\ell,21} & \mathrm{B}_{\ell,22} \end{pmatrix} \tag{6.2.1}$$

must be positive-definite and emulating the matrix elements individually will not guarantee this. To ensure that the 3×3 band power matrix remains positive-definite during the prediction phase when using the emulator, we instead build the latter on each element of the logarithm $\mathbf{B}_\ell$ (lower or upper triangular part, essentially all the unique elements),

$$\mathbf{V}_\ell = \mathbf{R}\tilde{\mathbf{\Lambda}}\mathbf{R}^{\mathrm{T}} = \log\left(\mathbf{B}_\ell\right), \tag{6.2.2}$$

where $\mathbf{B}_\ell = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{\mathrm{T}}$, $\tilde{\mathbf{\Lambda}}_{\nu\nu} = \log(\mathbf{\Lambda}_{\nu\nu})$ and $\mathbf{\Lambda}$ and $\tilde{\mathbf{\Lambda}}$ are diagonal. Moreover, since we normally assume a Gaussian Process with mean zero and kernel, $\mathbf{K}$, we do an additional linear scaling such that the mean of the band powers in our training set is zero and has a standard deviation of one, for example, for the i^{th} transformed band power,

$$\boldsymbol{v}_i' \to \frac{\boldsymbol{v}_i - \bar{v}_i}{\sigma_i} \tag{6.2.3}$$

and the predictive mean and variance are

$$\begin{aligned} \mathbb{E}[v_{i(*)}] &= \sigma_i\, \mathbb{E}[v'_{i(*)}] + \bar{v}_i \\ \mathrm{var}[v_{i(*)}] &= \sigma_i^2\, \mathrm{var}[v'_{i(*)}]. \end{aligned} \tag{6.2.4}$$

6.2.5 Training the Emulator

We now have our training set $\{\mathbf{X}, \mathbf{V}_{\ell,ij}\}$. Therefore we have a set of 24 Gaussian Processes due to each element of the transformed band powers. Prior to building the emulator, a crucial step is to choose a kernel function for the Gaussian Process. Here we use the ARD kernel, defined in equation (3.2.11).

To ensure a good performance, we have to find the set of hyper-parameters which maximises the marginal likelihood, as discussed in §3.2.1 in Chapter 3. An important ingredient is the analytical gradient of the marginal likelihood with respect to the kernel hyper-parameters to guarantee convergence to the global minimum. The gradients are

$$\begin{aligned} \frac{\partial k_{pq}}{\partial A} &= \frac{2}{A} k_{pq}^{\mathrm{ARD}} \\ \frac{\partial k_{pq}}{\partial \ell_i} &= k_{pq}^{\mathrm{ARD}} \frac{(\boldsymbol{\theta}_{p(i)} - \boldsymbol{\theta}_{q(i)})^2}{\ell_i^3}, \end{aligned} \tag{6.2.5}$$

where i indicates the i^{th} dimension of the problem. We use the Limited memory Broyden-Fletcher-Goldfarb-Shanno, L-BFGS-B algorithm (Zhu et al., 1997; Press et al., 2007) along with the gradients defined above to optimise for these hyper-parameters by minimising the negative log-marginal likelihood, in equation (3.2.12), via gradient descent. However, it is a known fact that training a Gaussian Process is not an easy task because the marginal likelihood has various local maxima (Rasmussen & Williams, 2006). We adopt the standard approach of restarting our optimiser at different positions and we find that $N_{\mathrm{restart}} = 5$ was sufficient in practice to

ensure that we find the set of hyper-parameters corresponding to the global optimum (see Fig. 6.3). Although this is not guaranteed, we also want to emphasise that the use of the gradients was required to find the global optimum. Once the Gaussian Process is trained, the kernel parameters are fixed at the optimised values of the hyper-parameters and then use equations (3.2.10) to make predictions.

6.2.6 The GP Uncertainty

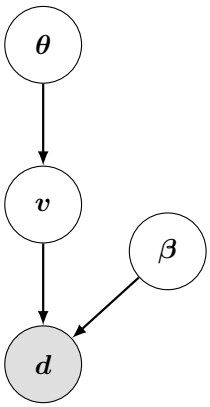


Figure 6.5 – The full forward model can be understood as follows: at each step in the inference procedure, a random set of samples of the cosmological, $\boldsymbol{\theta}$ and nuisance, $\boldsymbol{\beta}$ is drawn from the prior, followed by a random realisation of the probabilistic band powers, centred on its mean and variance before computing the likelihood. Note that the kernel hyper-parameters are fixed to their optimised values.

In this section, we look into propagating the GP uncertainty through the full forward model when we use the emulator. To be more specific, we seek the posterior distributions of the cosmological parameters and the two nuisance parameters $(A_{\rm IA},\, A_{\rm bary})$, that is,

$$\boldsymbol{\theta} = \left[\Omega_{\rm cdm}h^2,\, \Omega_{\rm b}h^2,\, \ln(10^{10}A_{\rm s}),\, n_s,\, h,\, A_{\rm bary},\, A_{\rm IA},\, \Sigma m_\nu\right]$$

and the other 4 nuisance parameters,

$$\boldsymbol{\beta} = [A_1,\, A_2,\, A_3,\, m]$$

marginalised over the probabilistic band powers. A_1, A_2, A_3 correspond to free parameters which determine excess noise in the autocorrelation power spectrum, while m is the shear multiplicative bias parameter (Köhlinger et al., 2017). Using equation (4.2.6) and defining $\boldsymbol{v}$ as the total band powers, we can write the joint posterior, $p(\boldsymbol{\theta},\, \boldsymbol{\beta}\,|\boldsymbol{d}\,)$ as

$$
\begin{aligned}
p\left(\boldsymbol{\theta}, \boldsymbol{\beta} \,|\boldsymbol{d}\right) &= \int p\left(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{v} \,|\boldsymbol{d}\right) \, \mathrm{d}\boldsymbol{v} \\
&= \int p\left(\boldsymbol{d} \,|\boldsymbol{v}, \boldsymbol{\beta}\right) p\left(\boldsymbol{v} \,|\boldsymbol{\theta}\right) \, \mathrm{d}\boldsymbol{v} \, p\left(\boldsymbol{\theta}\right) p\left(\boldsymbol{\beta}\right) .
\end{aligned}
\tag{6.2.6}
$$

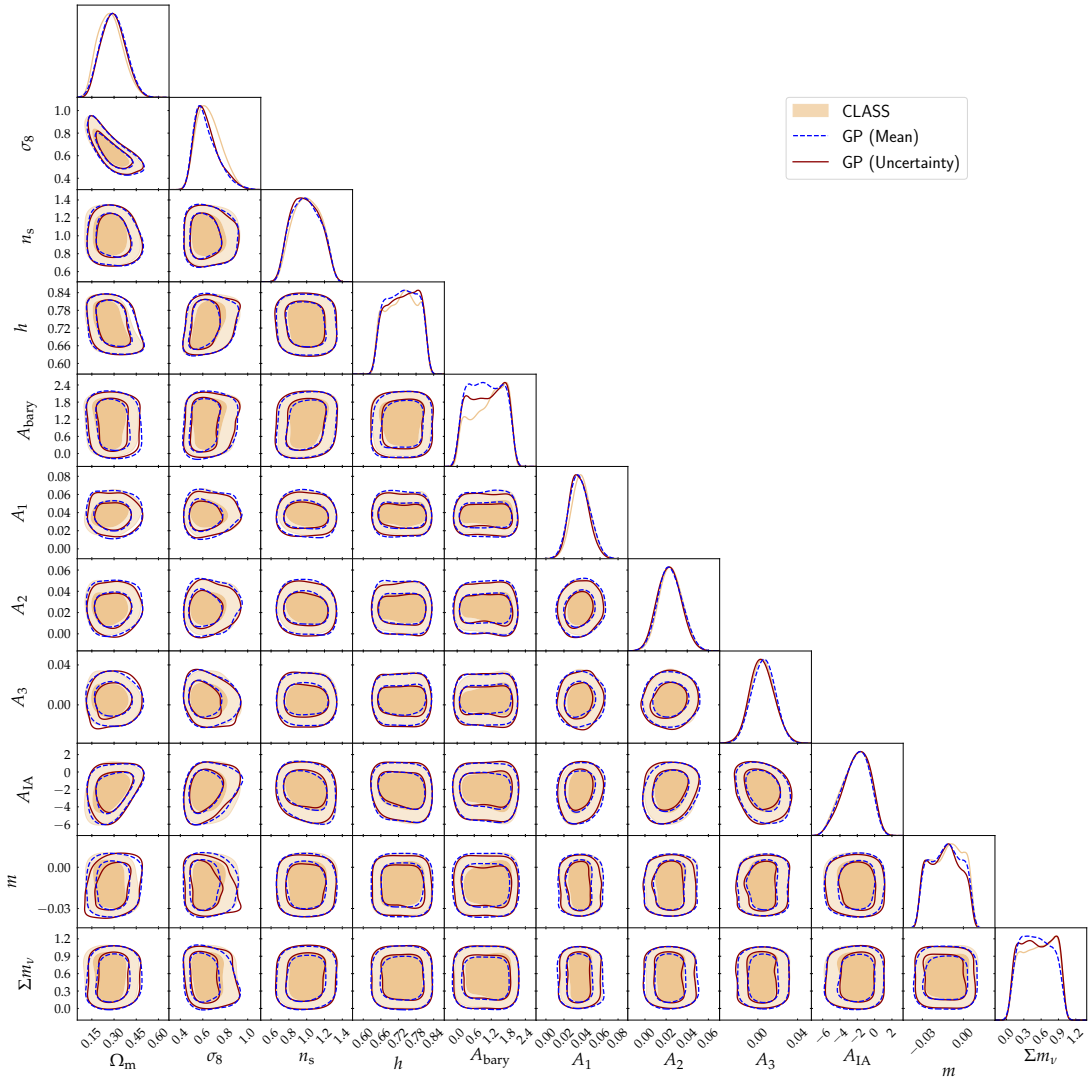


Figure 6.6 – The full 1D and 2D marginalised posterior distributions obtained using three different methods - The one in tan colour corresponds to posterior distributions with the full simulator (CLASS) while the solid brown one corresponds to the Gaussian Process emulator when random functions of the band powers are drawn, hence marginalising over the Gaussian Process uncertainty. The posterior in blue shows the distributions obtained when only the mean of the Gaussian Process was used in the inference routine. The contours denote the 68 % and 95 % credible interval respectively. Note that some parameters are dominated by their respective priors and are not constrained at all. A similar conclusion was drawn by (Köhlinger et al., 2017). However, the important point here is that the posterior from the GP is close to that obtained with CLASS.

If $p(\boldsymbol{\nu}|\boldsymbol{\theta})$ were a Gaussian distribution of the band power from the Gaussian Process, the above integration would be a convolution of two Gaussian distributions and the likelihood part would be Gaussian.

However, in our analysis, the predictive distribution is Gaussian in each element of the logarithm of the band power matrix. For example, in Fig. 6.4, we show the GP mean and

variance for two elements across a slice in parameter space. As previously discussed, if the GP predictions were Gaussian in the band powers, we could marginalise analytically over the theoretical uncertainty. Since they are Gaussian in each element of $\mathbf{V}_{\ell,ij}$, we marginalise by drawing samples of the cosmological and nuisance parameters (see Fig. 6.5) and perform a Monte Carlo integration, which is relatively fast and approximating the joint posterior as

$$p(\boldsymbol{\theta}, \boldsymbol{\beta} \,|\boldsymbol{d}\,) \propto \frac{p(\boldsymbol{\theta})p(\boldsymbol{\beta})}{N_s} \sum_{i=1}^{N_s} p\left(\boldsymbol{d}\,\middle|\mathbf{V}_{\ell,ij}\right). \tag{6.2.7}$$

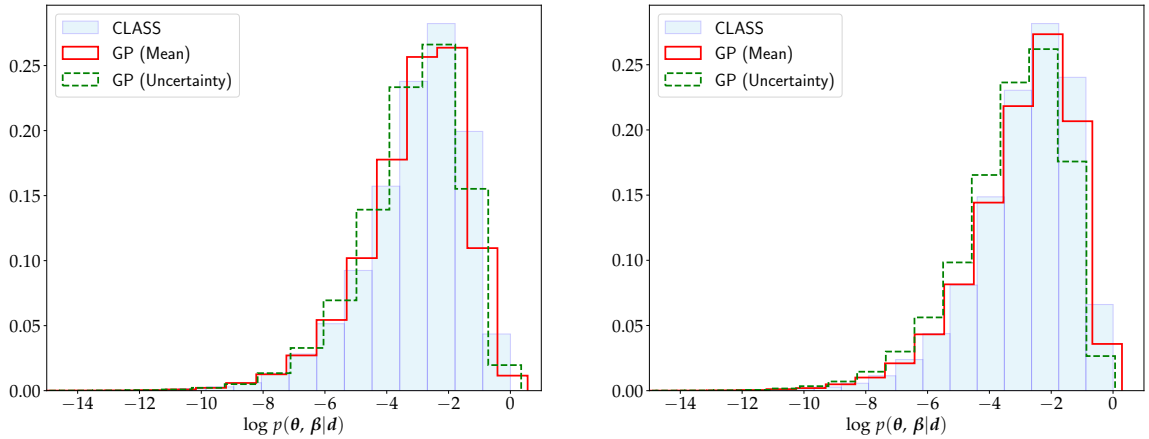
N_s is the number of random band powers drawn after computing the predictive mean and variance. We use $N_s = 20$ at every step in the MCMC to take into account the uncertainty from the Gaussian Process. Recall that each band power is being modelled independently by a GP and hence the Monte Carlo integral in equation (6.2.7) requires few draws of the probabilistic band powers.

6.3 Data Compression

The next era of weak lensing surveys such as Euclid and LSST will have $\sim$ 10 tomographic bins, and with multiple band powers or correlation functions per bin, the number of summary statistics will be large, $\sim 10^3 - 10^4$. As an example, Euclid Collaboration et al. (2019) considered 100 band powers per bin, and 10 tomographic bins, which gives a minimum of 1000 summaries, and 5500 if cross-band powers are included. The setup, in the previous section, is not a scalable approach for these future surveys. In particular, emulating each band power is not an entirely feasible approach because one will have to train and store thousands of separate Gaussian Processes and this process in itself can be quite expensive.

Hence, we use the MOPED data compression algorithm to compress all the band powers to a smaller set of MOPED coefficients. In the previous applications of the MOPED algorithm, it was assumed that the covariance matrix is fixed. In our case, Köhlinger et al. (2017) constructed a covariance matrix which depends on the m parameter, the multiplicative bias. In this work, we fix $\mathbf{C}$ at the average fiducial value provided[†] in the data. Data compression with parameter-dependent covariance matrix has been explored by Heavens et al. (2017b). For current weak lensing analysis, the gain is not significant (since we are working with only 24 band powers) but the method proposed in this work is expected to yield fast parameter inference in the regime of a large number of band powers, $N \sim 10^4$, with only $p \sim 10$ parameters of interest.

[†]Cosmological parameter inference depends mildly on the parameter m.

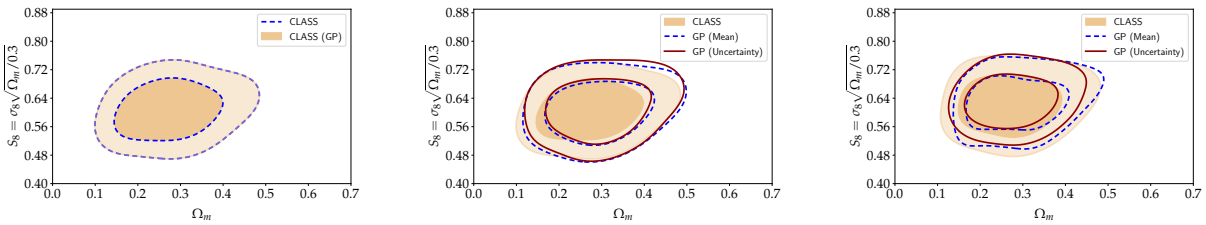


(a) log-Posterior without Compression

(b) log-Posterior with MOPED Compression

Figure 6.7 – Samples of the log-posterior obtained with the 3 methods investigated - In panel (a), the pale blue histogram refers to the log-posterior samples from CLASS while the red and green step histogram correspond to the mean and error of the GP respectively. A similar plot is shown in panel (b) but after applying the MOPED compression step.

By emulating the MOPED coefficients directly with separate Gaussian Processes, we have a very powerful tool. The GPs are still functions of just 8 parameters (6 cosmological and 2 systematics) and we now have only 11 separate GPs. Crucially, this setup is interesting because increasing the number of band powers (for example, in forthcoming lensing surveys) will not affect the MOPED timings at all.



(a) Emulator for computing σ_8

(b) Without MOPED Compression

(c) With MOPED Compression

Figure 6.8 – S_8 versus Ω_m plane for our analysis. The left panel shows that the Gaussian Process emulator, which is a function of our cosmological parameters, for computing σ_8 is accurate and precise enough compared to CLASS. The middle panel shows the constraints without MOPED compression while the right panel includes MOPED compression. The inner and outer contours correspond the 68% and 95% credible interval respectively.

6.4 Results

Fig. 6.4 shows 2 band powers, evaluated across the $\Omega_{\rm cdm}h^2$ slice in parameter space. In particular, the function in black corresponds to the accurate solver, CLASS while the broken red function corresponds to the GP mean, with the tan shading giving the 3σ credible interval of the GP. Note also that the right panel shows the GP prediction through a given training point

and as expected, the GP uncertainty tends to zero. As seen in Fig. 6.4, the GP is able to predict the band powers quite well.

Since the predictive function is a Gaussian distribution, we can build a simple emulator by just using the mean, or propagate the uncertainty from the Gaussian Process through the model. Either method gives reasonable posterior densities as shown in Fig. 6.6. On a high end desktop computer, the evaluation is quite fast. Computing one likelihood with the mean of the Gaussian Process takes 0.03 seconds compared to 0.09 seconds if we include the Gaussian Process uncertainty with 20 Monte Carlo samples to marginalise over the GP uncertainty. On the other hand, CLASS takes 0.65 seconds for a likelihood evaluation. If we use 1000 training points, this yields an overall speed-up by a factor of $\sim 12 - 30$ depending on whether we use the mean or the GP variance. In our case, we generate 360000 MCMC samples using EMCEE (Goodman & Weare, 2010; Foreman-Mackey et al., 2013) for which the full simulator takes about 44 hours while the Gaussian Process emulator, using the mean, takes about ~ 1.5 hours. On the other hand, when we emulate the MOPED coefficients using 1000 training points, each likelihood computation takes ~ 0.03 seconds with either the mean or the variance of the GPs. As an example, with the MOPED compression, CLASS takes ~ 44 hours to generate 330000 MCMC samples (note that there is no significant improvement in speed-up because we have just 24 band powers and each likelihood computation with or without the MOPED compression is almost the same). However, with the emulator, we obtain the same number of MCMC samples in ~ 1.5 hours with either the mean or variance of the GPs. All experiments with EMCEE were run on a single core. An interesting additional feature for the emulation scheme would be to exploit parallelization to speed-up inference further.

The distribution of the log-posterior (up to a normalisation constant) of the MCMC samples obtained by using CLASS (in pale blue) is shown in Fig. 6.7. In the same plot, the red and green histograms show the distribution of the log-posterior when using the mean and error from the GP respectively. In the same figure in panel (b), we show the log-posterior of the samples obtained after compressing the data using the MOPED formalism. Note that, the distribution of the log-posterior of the different MCMC samples gives an indication of how faithful the function reconstruction with the GP is. With a small number of training points, there is a small shift of the log-posterior distribution of the GP emulator (either with the mean or the uncertainty) relative to the CLASS distribution.

To compare the two distributions, we compute the Kullback-Leibler (KL) divergence between the CLASS distribution and the GP distribution, that is,

Table 6.4.1 – Computational cost comparison between CLASS and the GP emulator

N_{train}	**Training**	**MCMC (Mean)**	**MCMC (Error)**	D_{KL} (Mean)	D_{KL} (Error)
1000	20	84	216	0.84	1.00
1500	48	85	290	0.63	0.89
2000	92	86	396	0.60	0.81
2500	139	88	524	0.47	0.68
3000	209	90	692	0.09	0.65

Note: The training and sampling time (columns 2,3,4) are given in minutes and the KL divergence is computed in units of nats (scaled by a constant); the largest D_{KL} is with the 1000 training points when we include the GP uncertainty, in which case, $D_{\text{KL}} = 2.03 \times 10^{-12}$.

$$D_{\text{KL}}\left(p\,\|q\right) = \sum p\left(\boldsymbol{\theta},\,\boldsymbol{\beta}\,|\boldsymbol{d}\right)\,\log\left[\frac{p\left(\boldsymbol{\theta},\,\boldsymbol{\beta}\,|\boldsymbol{d}\right)}{q\left(\boldsymbol{\theta},\,\boldsymbol{\beta}\,|\boldsymbol{d}\right)}\right] \tag{6.4.1}$$

where $p\left(\boldsymbol{\theta},\,\boldsymbol{\beta}\,|\boldsymbol{d}\right)$ and $q\left(\boldsymbol{\theta},\,\boldsymbol{\beta}\,|\boldsymbol{d}\right)$ are the posterior probabilities computed using CLASS and the GP at the same points in parameter space. Since the posterior probability is cheap to compute with the GP, we use all the MCMC samples obtained using CLASS to compute $q(\boldsymbol{\theta},\,\boldsymbol{\beta})$. The KL-divergence in nats, as a function of the number of training points, is shown in Table 6.4.1. In general, the reconstruction of the band powers is almost perfect as the number of training points increases. This can also be deduced from the 5^{th} column in Table 6.4.1 where the KL divergence decreases with increasing training points. If one could afford additional simulations, one option would be to just use the mean of the GP to sample the posterior distribution since it is not only faster compared to the case where the GP uncertainty is included, but is also closer to the actual true posterior distribution.

To assess the convergence of our MCMC chains, we also compute the Gelman-Rubin statistics (Gelman & Rubin, 1992) for different scenarios. The latter is simply defined as $\hat{R} = {}^{V}/_{W}$, where V is the between-chain variance and W is the within chain variance. $\hat{R}$ is calculated for different cases, for example, for a fixed number of training points, we use the MCMC samples using the GP (mean) and the MCMC samples obtained using CLASS. This is repeated with the MCMC samples where the GP uncertainty is included. In all cases, we apply a threshold of 1.05 to ensure that the chains satisfy the ergodicity condition.

We are also interested in the $S_8 = \sigma_8\sqrt{\Omega_{\text{m}}/0.3}$ cosmological parameter constraint. Recall that the GPs for sampling the posterior are built using the 8 parameters (6 cosmological and 2 systematics) and they do not allow us to predict σ_8 directly. However, the latter is a function of

just the 6 cosmological parameters, since it involves an integration over the power spectrum. Therefore, as we compute the band powers with the 1000 training points, we also record the σ_8 values, as generated by CLASS. We then construct a training set with inputs

$$\left[\Omega_{\rm cdm}h^2,\, \Omega_{\rm b}h^2,\, \ln(10^{10}A_{\rm s}),\, n_{\rm s},\, h,\, \Sigma m_\nu\right]$$

which is then used to build an additional GP for σ_8. This then allows us to predict σ_8 at any point in the parameter space within the prior box. We find that it takes only 1 minute to predict σ_8 for 360 000 MCMC samples.

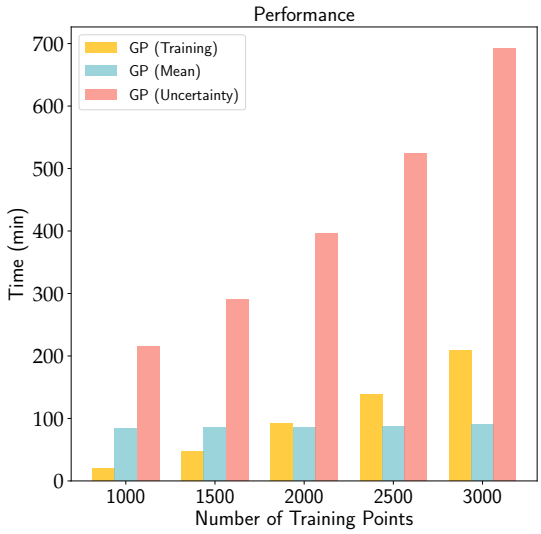


Figure 6.9 – Illustrating the performance of the emulator as a function of the number of training points. The expensive computations reside in training and predicting the GP uncertainty. Sampling the posterior with the GP mean is quick, even with the 3000 training points. The graphs do not perfectly follow the expected scaling with N because of various overheads.

In Fig. 6.8, we show the 2D marginalised posterior distribution of the derived parameters S_8 and $\Omega_{\rm m}$ using three different methods is shown. In particular, we compare the distribution obtained from CLASS with the mean and uncertainty of the GP and we conclude that we are able to recover comparable posterior densities for these two quantities, S_8 and $\Omega_{\rm m}$.

In high dimensions, the GP uncertainty inflates between any two points. It is expected that adding more training points will improve the performance of the emulator (either with the mean or GP uncertainty) since the reconstruction of the emulated function will converge to the original function. In general, with increasing number of training points, the GP uncertainty will also decrease. The effect of the number of training points is indicated by the values of the KL-divergence in the last two columns of Table 6.4.1. However, we empirically found that the KL-divergence when we use the mean of the emulator, is always better compared to the GP uncertainty.

One might expect the inclusion of the GP uncertainty to broaden the likelihoods, so the KL divergence would not be an appropriate measure of success. However, this does not appear to be the case: marginal errors are not noticeably increased. Our conclusion is that inclusion of the GP uncertainty does not improve results, but this might vary with application. The reason is probably that we are emulating a precise function, where the training points have zero error, and in this circumstance, the GP (which makes some assumptions that do not hold in detail) provides an error that is only approximately correct (Karvonen et al., 2020; Wang, 2020).

6.5 Conclusions

We have designed a principled and detailed Gaussian Process emulator for constraining not only weak lensing cosmological parameters but also the nuisance parameters. In summary, for this problem, 1) the (expensive) solver is queried a few thousand times only, to generate a training set (compared to a conventional MCMC routine where the solver is queried at every likelihood computation, 2) the emulator is $\sim$ 20 times faster compared to the full solver and this makes inference very quick and 3) by emulating the MOPED coefficients, the number of separate Gaussian Processes is equal to the number of parameters in the model and inference, irrespective of the number of data points, and is a very powerful technique for analysing large datasets. Moreover, the posterior distributions obtained from the emulator are quite robust compared to the full run of the simulator, with and without MOPED.

We have also demonstrated that the emulator can be used to emulate the MOPED coefficients directly. Both combined are expected to accelerate cosmological parameter inference. Emulating the MOPED compressed data has two major advantages. The first is a feature of MOPED itself, that the compressed data set does not grow at all as the original dataset increases in size, so scales exceptionally well to Euclid and LSST. The second is that MOPED is only fast if the theoretical values of the MOPED coefficients can be computed very quickly. The GP provides this functionality. This is the most important conclusion of this work.

In addition, we have used the KL-divergence as a metric to assess the performance of the emulator in obtaining reliable high dimensional posterior distributions. As evident from Table 6.4.1, the larger the number of training points, the better the reconstruction of the emulated function and hence the lower the KL-divergence between the accurate CLASS posterior distribution and the emulator posterior distribution.

We also recommend using the mean of the emulator for this application. In Table 6.4.1, the KL-divergence between the emulator posterior and the CLASS posterior shows that the mean

is always better than the emulator with the GP uncertainty. From a computational perspective, this has various other advantages, for example, inference is very fast since the GP mean prediction requires $\mathcal{O}(N)$ operations (recall that the GP mean is a linear predictor) and storage.

An exciting application of this emulator can be in the case where one requires non-Limber computation of the power spectra. This certainly applies to galaxy clustering statistics (Fang et al., 2020) and the weak lensing bi-spectrum (Deshpande & Kitching, 2020), even if for the weak lensing power spectrum it is a good approximation (Kitching & Heavens, 2017; Kilbinger et al., 2017). In general, the latter is computationally expensive to be calculated accurately, especially at large ℓ because of the rapid oscillations of the spherical Bessel functions (Lemos et al., 2017). For example, if the CLASS run were to be repeated without the Limber approximation, the emulator would have been $\sim 10^3$ times even faster. In future surveys, because the number of tomographic bins will be large, one would require more power spectra computations. For example, 10 tomographic bins lead to 55 auto- and cross- power spectra and the emulator would be $\sim 10^3$ and $\sim 10^5$ faster with and without the Limber approximation respectively.

Emulation has various other key advantages, apart from speeding up inference. As an example, one has to choose a good proposal distribution, which often requires tuning, to run an MCMC chain with the full simulator. The emulator can be used to explore the parameter space quickly and learn a suboptimal proposal distribution which can then be used with the full simulator.

The accuracy of the reconstructed function can be improved by adding more training points as we have demonstrated. However, scaling Gaussian Processes to large number of training points results in a major computational bottleneck, mainly due to $\mathcal{O}(N^3)$ operations in training and $\mathcal{O}(N^2)$ in predicting the uncertainty (see Fig. 6.9). Fortunately, here a few hundred training points suffices to give cosmological results with only a few percent degeneracy in error bars. Moreover, in this work, the training points have been placed according to the prior range itself. However, the interpolation scheme can be improved if we have more constrained parameters where we can use better prior information such as a Fisher matrix to intelligently place the training points. Alternatively, one can also do a quick optimisation to find the maximum likelihood estimator and the Hessian matrix, both of which can be used to construct an optimal design for the training points.

An alternative option to accelerate the computation of GP uncertainty is to intelligently partition the training set by using a clustering algorithm, for example, $k-$means clustering (Hastie et al., 2001). During the prediction step, one can then use a local expert, which has a

smaller kernel size, to compute the GP uncertainty swiftly.

A quantity which is often required in optimisation or Monte Carlo methods such as Hamiltonian Monte Carlo (HMC) is the gradient with respect to the negative log-likelihood (cost function). Conveniently, the gradient with respect to the mean of the Gaussian Process surrogate model is analytic and this opens a new avenue towards accelerating gradient computation as well.

Gaussian Processes should not only be interpreted as a method for just accelerating computations. Instead, it effectively allows us to compute the posterior distribution of a function by placing a prior over it. In this work, the EE band powers and MOPED coefficients are modelled independently as Gaussian Processes and we have shown that we can recover robust cosmological parameters, whilst still marginalising over the nuisance parameters.

6.6 Summary

In this chapter, we have improved upon the exploratory analysis we did in Chapter 4. In particular, we have shown that using the LH samples, along with the two transformations, namely the pre-whitening step at the input level and the matrix logarithm transformation for the band powers, only a few thousands of forward simulations are sufficient. Moreover, by emulating the MOPED coefficients directly, we show that we only have p regressors and this number does not grow as the size of the data set increases. The techniques developed in this chapter will significantly improve inference speed in future surveys.

Chapter 7

SEMI-PARAMETRIC GAUSSIAN PROCESSES

> You cannot hope to build a better world without improving the individuals. To that end, each of us must work for his own improvement and, at the same time, share a general responsibility for all humanity, our particular duty being to aid those to whom we think we can be most useful.
>
> *Marie Curie*

In this chapter, we propose a technique which takes into account the option of including prior information in the modelling a function. Generally, in most application a zero mean Gaussian Process is often assumed. Instead, one can also add an explicit mean function and marginalise over the residuals. If the mean function is not deterministic, we also have to marginalise over the regression coefficients. A full sketch of the technique is provided in §7.1.2

Specifying a mean function has multiple advantages. First, it allows us to model the function as much as we can by specifying a set of basis functions. Second, irrespective of the number of training points, the inferred function is expected to be better than a zero mean Gaussian Process. In the latter, the mean prior becomes irrelevant if there is a sufficient number of training points, but this is rarely the case.

Following our work in Chapter 6, in this chapter, we choose to emulate the MOPED coefficients and we take a slightly different approach where we focus only on the cosmological part of the analysis, whilst still marginalising over all the nuisance parameters. This work uses the publicly available likelihood code* for the KiDS-450 and extends it to incorporate our method, that is, the compression and emulation of the MOPED coefficients via the semi-parametric Gaussian Process. The code and analysis for this chapter is shared on Github and is available at:

https://github.com/Harry45/semi_gp

*https://bitbucket.org/fkoehlin/kids450_qe_likelihood_public/

This chapter is also a precursor to Chapter 8 where we will explore the semi-parametric Gaussian Process approach to build an emulator for the most expensive part of a weak lensing analysis pipeline, that is, the 3D matter power spectrum, $P_\delta(k, z)$.

7.1 The Emulating Scheme

In this section, we walk through the steps to build a model for the MOPED coefficients. We assume we have run the simulator (computing the forward model and compressing the data/-theory) at N design points, $\boldsymbol{\theta}$, such that we have a training set, $\{\boldsymbol{\theta}, \boldsymbol{g}_i\}$. The index i corresponds to the i^{th} compressed data. Note that in our application, the fact that the compressed data are uncorrelated, allows us to model each MOPED coefficient independently.

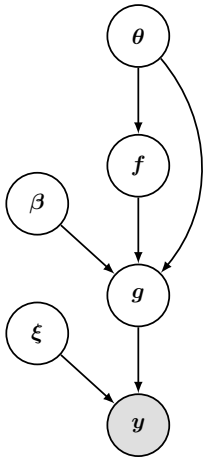


Figure 7.1 – Directed Acyclic Graph (DAG) with the different latent parameters. $\boldsymbol{\theta}$ is the vector of cosmological parameters and $\boldsymbol{\beta}$ is the set of regression coefficients in the model. $\boldsymbol{f}$ refers to the vector of residuals, modelled by a zero mean GP and $\boldsymbol{y}$ is the compressed MOPED data vector. $\boldsymbol{\xi}$ is a set of nuisance parameters which are also marginalised over in the inference mechanism.

Once we have our training set, our goal is to learn the functional relationship between the MOPED coefficients, $\boldsymbol{g}$ (we have dropped the index i but the same steps apply to all MOPED coefficients) and the inputs $\boldsymbol{\theta}$. In other words, we model the data, $\boldsymbol{g}$, as

$$\boldsymbol{g} = \boldsymbol{h}(\boldsymbol{\theta}) + \boldsymbol{\epsilon} \tag{7.1.1}$$

where $\boldsymbol{h}$ is underlying assumed model. For example, in the cosmological context, we often fit a ΛCDM model to the observed data.

7.1.1 Polynomial Regression

However, in our application, $\boldsymbol{h}$ might be a deterministic function but is completely unknown to us. A straightforward approach is to assume a polynomial approximation to the data, that is,

$$\boldsymbol{g} = \boldsymbol{\Phi}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{7.1.2}$$

where $\boldsymbol{\Phi}$ is a design matrix, whose columns contain the basis functions $[1, \boldsymbol{\theta}_1, \dots \boldsymbol{\theta}_p^n]$ and n is the order of the polynomial. $\boldsymbol{\beta}$ is a vector of regression coefficients (also referred to as weights) and $\boldsymbol{\epsilon}$ is the noise vector and $\text{cov}(\boldsymbol{\epsilon}) = \boldsymbol{\Sigma}$. Using Bayes' theorem, the full posterior distribution of the weights is

$$p(\boldsymbol{\beta}\,|\boldsymbol{g}) = \frac{p(\boldsymbol{g}\,|\boldsymbol{\beta})p(\boldsymbol{\beta})}{p(\boldsymbol{g})}. \tag{7.1.3}$$

$p(\boldsymbol{\beta}\,|\boldsymbol{g})$ is the posterior distribution of $\boldsymbol{\beta}$, $p(\boldsymbol{g}\,|\boldsymbol{\beta})$ is the likelihood of the data, $p(\boldsymbol{\beta})$ is the prior for $\boldsymbol{\beta}$ and $p(\boldsymbol{g})$ is the marginal likelihood (Bayesian evidence) which does not depend on $\boldsymbol{\beta}$. In what follows, the notation $\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \mathbf{C})$ denotes a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{C}$.

Assuming a Gaussian likelihood for the data, $\mathcal{N}(\boldsymbol{g} \mid \boldsymbol{\Phi}\boldsymbol{\beta}, \boldsymbol{\Sigma})$ and a Gaussian prior for the weights, $\mathcal{N}(\boldsymbol{\beta} \mid \boldsymbol{\mu}, \mathbf{C})$, the posterior distribution of $\boldsymbol{\beta}$ is another Gaussian distribution, $\mathcal{N}(\boldsymbol{\beta} \mid \bar{\boldsymbol{\beta}}, \boldsymbol{\Lambda})$ with mean and covariance given respectively by

$$\begin{aligned} \bar{\boldsymbol{\beta}} &= \boldsymbol{\Lambda}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{g} + \mathbf{C}^{-1}\boldsymbol{\mu}) \\ \boldsymbol{\Lambda} &= (\mathbf{C}^{-1} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi})^{-1}. \end{aligned} \tag{7.1.4}$$

In general, we are also interested in learning the (posterior) predictive distribution at a given test point $\boldsymbol{\theta}_*$, that is, $p(g_* \mid \boldsymbol{g}, \boldsymbol{\theta}_*)$ and this is another Gaussian distribution,

$$p(g_* \mid \boldsymbol{g}, \boldsymbol{\theta}_*) = \mathcal{N}(g_* \mid \boldsymbol{\Phi}_*\bar{\boldsymbol{\beta}}, \sigma^2 + \boldsymbol{\Phi}_*\boldsymbol{\Lambda}\boldsymbol{\Phi}_*^{\mathrm{T}}). \tag{7.1.5}$$

For noise-free regression, the noise variance, $\sigma^2 \approx 0$ (although a tiny jitter term is often used for numerical stability) and the predictive uncertainty is dominated by the term $\boldsymbol{\Phi}_*\boldsymbol{\Lambda}\boldsymbol{\Phi}_*^{\mathrm{T}}$. Moreover, in practice, the noise term at the test point is barely known and is hence approximated by $\boldsymbol{\Phi}_*\boldsymbol{\Lambda}\boldsymbol{\Phi}_*^{\mathrm{T}}$.

On the other hand, we are also interested in understanding the model, that is, the number of

basis functions we would need to fit the data. An important quantity is the marginal likelihood which penalises model complexity (Trotta, 2008). In this case, this quantity can be analytically derived and is given by

$$p(\boldsymbol{g}) = \mathcal{N}(\boldsymbol{g} \mid \boldsymbol{\Phi\mu}, \boldsymbol{\Sigma} + \boldsymbol{\Phi C \Phi}^{\mathrm{T}}). \tag{7.1.6}$$

Note that this quantity is independent of $\boldsymbol{\beta}$ and is an integral of the numerator with respect to *all* the latent variables (in our case $\boldsymbol{\beta}$), that is,

$$p(\boldsymbol{g}) = \int p(\boldsymbol{g}\,|\boldsymbol{\beta}) p(\boldsymbol{\beta})\, \mathrm{d}\boldsymbol{\beta}. \tag{7.1.7}$$

To this end, one can compute the Bayesian evidence for a series of (polynomial) models and choose the model which yields the maximum Bayesian evidence (Kunz et al., 2006).

7.1.2 Modelling the residuals

The above formalism works well in various cases but (1) polynomial model fitting is generally a *global* fitting approach, (2) there exists a large number of choice for the number of basis functions and (3) the functional relationship between the data and the model might be a very complicated function. In this section, we therefore propose a Bayesian technique which models the residuals, that is, the difference between our proposed polynomial approximation and the underlying model. We will re-write equation (7.1.2) as

$$\boldsymbol{g} = \boldsymbol{\Phi\beta} + \boldsymbol{f} + \boldsymbol{\epsilon}, \tag{7.1.8}$$

where $\boldsymbol{f} = \boldsymbol{h} - \boldsymbol{\Phi\beta}$ is the deterministic error component of the model (Blight & Ott, 1975; Rasmussen & Williams, 2006). Under the assumption that we have modelled $\boldsymbol{g}$ as much as we can with the polynomial model, it is fair to make an a priori assumption for the distribution of f, that is $f \sim \mathcal{N}(0, A^2)$, where A is a constant. Moreover, in function space, points which are close to each other will depict similar values for f and as we move further away from a given design point, it is expected that the degree of similarity will decrease. In other words, the correlation between $f(\boldsymbol{\theta}_i)$ and $f(\boldsymbol{\theta}_j)$ decreases monotonically as the distance between $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ increases. This prior knowledge can be encapsulated by using a covariance (kernel) function such as the Gaussian distribution, that is,

$$\text{cov}(f_i, f_j) = A^2 \exp\left[-\frac{1}{2}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)^{\mathrm{T}} \boldsymbol{\Omega}^{-1} (\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)\right], \tag{7.1.9}$$

where $\boldsymbol{\Omega} = \text{diag}(\omega_1^2 \dots \omega_d^2)$ and ω_i^2 is the characteristic length-scale for each dimension. $\boldsymbol{\nu} = \{A, \omega_1, \dots \omega_d\}$ is the set of hyper-parameters for this kernel. In the same spirit, the full prior distribution for $\boldsymbol{f}$ is a multivariate normal distribution, that is,

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f} \mid \mathbf{0}, \mathbf{K}) \tag{7.1.10}$$

where the kernel matrix has elements $k_{ij} \equiv \text{cov}(f_i, f_j)$. At this point, we will assume that the hyper-parameters are fixed but we will later consider learning them via optimisation.

7.1.2.1 Inference

Now that we have a model for the data (MOPED coefficients), we seek the full posterior distribution of the latent variables $\boldsymbol{\beta}$ and $\boldsymbol{f}$. We assume a Gaussian prior for $\boldsymbol{\beta}$, that is, $p(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta} \mid \boldsymbol{\mu}, \mathbf{C})$. Using Bayes' theorem, the posterior distribution of $\boldsymbol{\beta}$ and $\boldsymbol{f}$ is

$$p(\boldsymbol{\beta}, \boldsymbol{f} \mid \boldsymbol{g}) = \frac{p(\boldsymbol{g} \mid \boldsymbol{\beta}, \boldsymbol{f}) p(\boldsymbol{\beta}, \boldsymbol{f})}{p(\boldsymbol{g})} \tag{7.1.11}$$

To simplify the derivation, we will rewrite equation (7.1.8) as

$$\boldsymbol{g} = \mathbf{D}\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{7.1.12}$$

where $\mathbf{D} = [\boldsymbol{\Phi}, \mathbb{I}]$ is an augmented, new (block) design matrix, consisting of the existing design matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N \times m}$ and the identity matrix, $\mathbb{I}$ of size $N \times N$. $\boldsymbol{\alpha} = [\boldsymbol{\beta}, \boldsymbol{f}]^{\mathrm{T}}$ is now a vector of length $N + m$, consisting of both $\boldsymbol{\beta}$ and $\boldsymbol{f}$. The sampling distribution of $\boldsymbol{g}$ is a Gaussian distribution, $\mathcal{N}(\boldsymbol{g} \mid \mathbf{D}\boldsymbol{\alpha}, \boldsymbol{\Sigma})$. On the other hand, we can re-write the full prior distribution of both set of parameters, $\boldsymbol{\beta}$ and $\boldsymbol{f}$ as $\mathcal{N}(\boldsymbol{\alpha} \mid \boldsymbol{\gamma}, \mathbf{R})$, where

$$\boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\mu} \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{R} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{K} \end{bmatrix}$$

Using a similar approach as in the previous section, the posterior of $\boldsymbol{\alpha}$ is another Gaussian distribution, that is,

$$p(\boldsymbol{\alpha} \mid \boldsymbol{g}) = \mathcal{N}(\boldsymbol{\alpha} \mid \mathbf{A}^{-1}\boldsymbol{b}, \mathbf{A}^{-1}), \tag{7.1.13}$$

where $\mathbf{A} = \mathbf{D}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{D} + \mathbf{R}^{-1}$ and $\boldsymbol{b} = \mathbf{D}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{g} + \mathbf{R}^{-1}\boldsymbol{\gamma}$. Simplifying further and using the block

matrix inversion lemma, we can also derive the covariance for each block in the following:

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{V}_\beta & \mathbf{V}_{\beta f} \\ \mathbf{V}_{f\beta} & \mathbf{V}_f \end{bmatrix}.$$

Noting that $\mathbf{V}_{f\beta} = \mathbf{V}_{\beta f}^{\mathrm{T}}$, the expression for each block is:

$$\mathbf{V}_f = \left[\mathbf{K}^{-1} + (\mathbf{\Sigma} + \mathbf{\Phi}\mathbf{C}\mathbf{\Phi}^{\mathrm{T}})^{-1}\right]^{-1} \tag{7.1.14}$$

$$\mathbf{V}_\beta = \left[\mathbf{\Phi}^{\mathrm{T}}\left(\mathbf{K} + \mathbf{\Sigma}\right)^{-1}\mathbf{\Phi} + \mathbf{C}^{-1}\right]^{-1} \tag{7.1.15}$$

$$\mathbf{V}_{\beta f} = -\mathbf{V}_\beta \mathbf{\Phi}^{\mathrm{T}} \mathbf{\Sigma}^{-1} (\mathbf{\Sigma}^{-1} + \mathbf{K}^{-1})^{-1} \tag{7.1.16}$$

In the same spirit, the posterior mean for $\boldsymbol{\beta}$ and f can be derived and are given respectively by

$$\hat{\boldsymbol{\beta}} = \mathbf{V}_\beta \left[\mathbf{\Phi}^{\mathrm{T}}\left(\mathbf{K} + \mathbf{\Sigma}\right)^{-1} g + \mathbf{C}^{-1}\boldsymbol{\mu}\right] \tag{7.1.17}$$

and

$$\hat{f} = \mathbf{V}_f \mathbf{\Sigma}^{-1}\left[g - \mathbf{\Phi}\bar{\boldsymbol{\beta}}\right] \tag{7.1.18}$$

Recall that $\bar{\boldsymbol{\beta}}$ is the expression for the posterior distribution of $\boldsymbol{\beta}$ when we use the polynomial model only. There are also some useful remarks and sanity checks which we can make from Equations 7.1.14, 7.1.17 and 7.1.18. In Equation 7.1.14, for the covariance of $\boldsymbol{\beta}$, if we had ignored the other latent variables f, in other words, in the absence of the kernel matrix, $\mathbf{K}$, we recover the posterior covariance for $\boldsymbol{\beta}$ when we use a polynomial model only. A similar argument applies for Equation 7.1.17 in which case we also recover the posterior distribution of $\boldsymbol{\beta}$ in the polynomial model. Equation 7.1.18 has a nice interpretation. The posterior mean of f is a weighted sum of the residuals, $g - \mathbf{\Phi}\bar{\boldsymbol{\beta}}$.

7.1.2.2 Prediction

Now that we have the full posterior distribution of the latent variables, another key ingredient is learning the predictive distribution at a given test point, $\boldsymbol{\theta}_*$. The joint distribution of the data and the function at the test point can be written as

$$\begin{bmatrix} g \\ g_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\Phi}\boldsymbol{\beta} \\ \boldsymbol{\Phi}_*\boldsymbol{\beta} \end{bmatrix}, \begin{bmatrix} \mathbf{K}+\boldsymbol{\Sigma} & k_* \\ k_*^{\mathrm{T}} & k_{**}+\sigma_*^2 \end{bmatrix} \right) \tag{7.1.19}$$

and the conditional distribution of g_* is a Gaussian distribution

$$p(g_* \mid \boldsymbol{g}, \boldsymbol{\theta}_*) = \mathcal{N}(g_* \mid \bar{g}_*, \mathrm{var}(g_*)) \tag{7.1.20}$$

where $\bar{g}_*$ and $\mathrm{var}(g_*)$ are the mean and variance given respectively by

$$\begin{aligned} \bar{g}_* &= \mathbf{X}_*\hat{\boldsymbol{\beta}} + f_* \\ \mathrm{var}(g_*) &= \mathbf{X}_*\mathbf{V}_\beta\mathbf{X}_*^{\mathrm{T}} + k_{**} + \sigma_*^2 - k_*^{\mathrm{T}}\mathbf{K}_g^{-1}k_* \end{aligned} \tag{7.1.21}$$

and we have defined $\mathbf{K}_g = \mathbf{K} + \boldsymbol{\Sigma}$, $\mathbf{X}_* = \boldsymbol{\Phi}_* - k_*^{\mathrm{T}}\mathbf{K}_g^{-1}\boldsymbol{\Phi}$ and $f_* = k_*^{\mathrm{T}}\mathbf{K}_g^{-1}\boldsymbol{g}$. This is another interesting result because if we did not have the parametric polynomial model, then the prediction corresponds to that of a zero mean Gaussian Process (GP) (Rasmussen & Williams, 2006). Until now, we have assumed a fixed set of kernel hyper-parameters. In the next section, we will explain how we can learn them via optimisation.

Table 7.1.1 – Symbols and notations with corresponding meanings

Symbol	Meaning
N	Number of training points
m	Number of basis functions
$\boldsymbol{g}$	Response of size N
$\boldsymbol{\theta}$	Inputs to the emulator
$\boldsymbol{\xi}$	Additional nuisance parameters
$\boldsymbol{y}$	MOPED compressed data
$\boldsymbol{\beta}$	Regression coefficients of size m
$\boldsymbol{f}$	Deterministic error component of size N of the model
$\boldsymbol{\Phi}$	Design matrix of size $N \times m$
$\mathbf{K}$	Kernel matrix of size $N \times N$
$\mathbf{C}$	Prior covariance matrix of $\boldsymbol{\beta}$ of size $m \times m$
$\boldsymbol{\mu}$	Prior mean of $\boldsymbol{\beta}$ of size m
$\mathbf{D}$	$\mathbf{D} = [\boldsymbol{\Phi}, \mathbb{I}]$ is a new design matrix of size $N \times (m+N)$
$\boldsymbol{\alpha}$	$\boldsymbol{\alpha} = [\boldsymbol{\beta}, \boldsymbol{f}]^{\mathrm{T}}$ is a vector of size $m+N$
$\mathbf{R}$	Prior covariance matrix of size $(m+N) \times (m+N)$
γ	$\gamma = [\boldsymbol{\mu}, \mathbf{0}]^{\mathrm{T}}$ prior mean of size $m+N$
$\boldsymbol{\Sigma}$	Noise covariance matrix of size $N \times N$
d	Dimension of the problem
$\boldsymbol{\nu}$	Kernel hyper-parameters

7.1.2.3 Kernel Hyper-parameters

An important quantity in learning the kernel hyper-parameters is the marginal likelihood (Bayesian evidence), which is obtained by marginalising over all the latent variables $\boldsymbol{\alpha}$ and is given by

$$p(\boldsymbol{g}) = \int p(\boldsymbol{g} \mid \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) \, \mathrm{d}\boldsymbol{\alpha}. \tag{7.1.22}$$

Fortunately, the above integration is a convolution of two multivariate normal distributions, $\mathcal{N}(\boldsymbol{g} \,|\mathbf{D}\boldsymbol{\alpha},\, \boldsymbol{\Sigma})$ and $\mathcal{N}(\boldsymbol{\alpha} \,|\boldsymbol{\gamma},\, \mathbf{R})$ and hence can be calculated analytically, that is,

$$p(\boldsymbol{g}) = \mathcal{N}(\boldsymbol{g} \mid \boldsymbol{\Phi}\boldsymbol{\mu},\, \mathbf{K}_g + \boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^{\mathrm{T}}) \tag{7.1.23}$$

and the log-marginal likelihood is

$$\log p(\boldsymbol{g}) = -\frac{1}{2}(\boldsymbol{g} - \boldsymbol{\Phi}\boldsymbol{\mu})^{\mathrm{T}}(\mathbf{K}_g + \boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^{\mathrm{T}})^{-1}(\boldsymbol{g} - \boldsymbol{\Phi}\boldsymbol{\mu}) - \frac{1}{2}\log\left|\mathbf{K}_g + \boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^{\mathrm{T}}\right| + \text{constant}. \tag{7.1.24}$$

The first term in equation (7.1.24) encourages the fit to the data while the second term (the determinant term) controls the model complexity. Recall that the kernel matrix, $\mathbf{K}$ is a function of the hyper-parameters $\boldsymbol{\nu} = \{A, \omega_1, \ldots \omega_d\}$. We want to maximise the marginal likelihood with respect to the kernel hyper-parameters and this step is equivalent to minimising the cost, that is, the negative log-marginal likelihood. In other words,

$$\boldsymbol{\nu}_{\text{opt}} = \arg\min_{\boldsymbol{\nu}} J(\boldsymbol{\nu}) \tag{7.1.25}$$

where we have defined $J(\boldsymbol{\nu}) \equiv -2\log p(\boldsymbol{g})$. An important ingredient for the optimisation to perform well is the gradient of the cost with respect to the kernel hyper-parameters, which is given by

$$\frac{\partial J(\boldsymbol{\nu})}{\partial \boldsymbol{\nu}_i} = \operatorname{tr}\left[\left((\mathbf{K}_g + \boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^{\mathrm{T}})^{-1} - \boldsymbol{\eta}\boldsymbol{\eta}^{\mathrm{T}}\right)\frac{\partial \mathbf{K}}{\partial \boldsymbol{\nu}_i}\right], \tag{7.1.26}$$

where $\boldsymbol{\eta} = (\mathbf{K}_g + \boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^{\mathrm{T}})^{-1}\boldsymbol{g}$. There are a few computational aspects which we should consider when implementing this method. In particular, for a single predictive variance calculation (see equation (7.1.21)) an $\mathcal{O}(N^2)$ operation is required whereas training (that is, learning the kernel hyper-parameters) requires an $\mathcal{O}(N^3)$ operation. On the other hand, the mean is quick to compute.

7.1.2.4 Derivatives

An important by-product from the trained model is the gradient of the emulated function with respect to the input parameters. This can be of paramount importance if we are using a more sophisticated Monte Carlo sampling scheme such as Hamiltonian Monte Carlo, HMC. In essence, the gradient of the MOPED log-likelihood is simply the sum of gradients of individual, independent emulated function. In this section, we provide the derived analytical gradients of the mean function with respect to the inputs. In particular,

$$\frac{\partial \bar{g}_*}{\partial \boldsymbol{\theta}_*} = \frac{\partial \boldsymbol{\Phi}_*}{\partial \boldsymbol{\theta}_*}\hat{\boldsymbol{\beta}} + \left[k_* \odot \mathbf{Z}_* \boldsymbol{\Omega}^{-1}\right]^{\mathrm{T}} \mathbf{K}_g^{-1}(g - \boldsymbol{\Phi}\hat{\boldsymbol{\beta}}) \tag{7.1.27}$$

where $\odot$ refers to element-wise multiplication. $\mathbf{Z}_* \in \mathbb{R}^{N \times d}$ and is defined as:

$$\mathbf{Z}_* = [\boldsymbol{\theta}_1 - \boldsymbol{\theta}_*, \boldsymbol{\theta}_2 - \boldsymbol{\theta}_* \ldots \boldsymbol{\theta}_N - \boldsymbol{\theta}_*]^{\mathrm{T}}$$

.

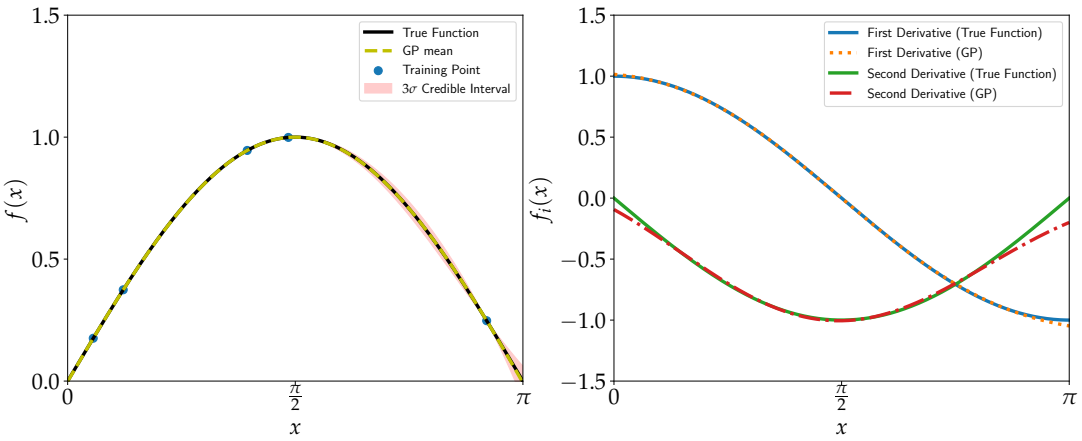


Figure 7.2 – Example of a GP regression in 1D - We have have five 'observed' data points and we use a GP with an ARD kernel to fit the data. The true function is $y = x\sin x$. The left panel shows the predicted mean and credible interval of the emulator while in the right panel, the first and second derivatives are shown. Note that $f'(x) = \cos x$ and $f''(x) = -\sin x$

.

Importantly, as seen from equation 7.1.27, the gradient is the sum of the gradients corresponding to the parametric part and the residual, which is modelled by a kernel. Moreover, higher order derivatives can also be calculated analytically. For example, the second order auto- and cross- derivatives are

$$\frac{\partial^2 \bar{g}_*}{\partial \boldsymbol{\theta}_*^2} = \frac{\partial^2 \boldsymbol{\Phi}_*}{\partial \boldsymbol{\theta}_*^2}\hat{\boldsymbol{\beta}} + \left[\Omega^{-1}\frac{\partial k_*}{\partial \boldsymbol{\theta}_*}\mathbf{Z}_* - \Omega^{-1} \odot k_*\right] \mathbf{K}_g^{-1}(g - \boldsymbol{\Phi}\hat{\boldsymbol{\beta}}). \tag{7.1.28}$$

As a result of this procedure, one can analytically calculate the first and second derivatives of

an emulated function using kernel methods. While the first derivatives are particularly useful in HMC sampling method, the second derivatives are more relevant in the calculation of, for example, the Fisher information matrix.

7.2 Emulating MOPED Coefficients

In this section, we briefly touch upon the data we use in this work before going through the steps we take towards building the emulator. We then elaborate on the theoretical cosmological model, followed by a discussion on the joint compression and emulating scheme.

7.2.1 Data

We use the publicly available band powers KiDS-450 data[†] (Köhlinger et al., 2017) which is a product after applying a quadratic estimator algorithm (Bond et al., 1998) to extract (tomographic) band powers for $76 \leq \ell \leq 1310$. Further details on the algorithm is detailed in Köhlinger et al. (2017). A different approach is to perform a shear correlation function analysis and this technique, applied to the KiDS-450 data is presented by (Hildebrandt et al., 2017). In this work, we resort to a tomographic band powers analysis.

Alternatively, one could also use a Bayesian Hierarchical Model, BHM, as presented by Alsing et al. (2016) to sample the shear field as well as the tomographic shear power spectra via Gibbs sampling. This technique has further been applied to the CFHTLens data not only for generating shear power spectra but also shear maps and cosmological parameter constraints (Alsing et al., 2017; Porqueres et al., 2021). Another emerging technique for generating summary statistics from lensing data is the Complete Orthogonal Sets of E/B-mode Integrals, COSEBIs statistics (Schneider et al., 2010; Asgari et al., 2019). The latter has been used for deriving cosmological constraints from the KiDS-450 and the DES-Y1 via a joint shear analysis (Asgari et al., 2020).

7.2.2 Cosmological Model

We follow the same theoretical model as explained in §4.2.1. The total lensing power spectrum is a weighted sum of the weak lensing power spectrum and the two intrinsic alignment power spectra, GI and II as follows:

$$C^{\rm tot}_{\ell,ij} = C^{\rm EE}_{\ell,ij} + A^2_{\rm IA} C^{\rm II}_{\ell,ij} - A_{\rm IA} C^{\rm GI}_{\ell,ij} \tag{7.2.1}$$

[†] http://kids.strw.leidenuniv.nl/sciencedata

and there is also a contribution due to noise in the data (see Equation 4.2.11 for further reference). Therefore, if we apply a two-step compression, that is, from power spectrum to band powers and band powers to MOPED coefficients, we have

$$g_{\rm tot} = g_{\rm EE} + A_{\rm IA}^2 g_{\rm II} - A_{\rm IA} g_{\rm GI}. \tag{7.2.2}$$

Note that we use the letter 'g' to denote this compression and the total MOPED coefficient is denoted by the letter 'y' (see Figure 7.1). Therefore, in this work we emulate $g_{\rm EE}$, $g_{\rm II}$ and $g_{\rm GI}$ and these are functions of the six cosmological parameters:

$$\left[\Omega_{\rm cdm}h^2,\ \Omega_{\rm b}h^2,\ \ln(10^{10}A_s),\ n_s,\ h,\ \Sigma m_\nu\right]$$

and one nuisance parameter, $A_{\rm bary}$ to account for baryon feedback (see §4.2.1 for further details).

7.2.3 Training Points

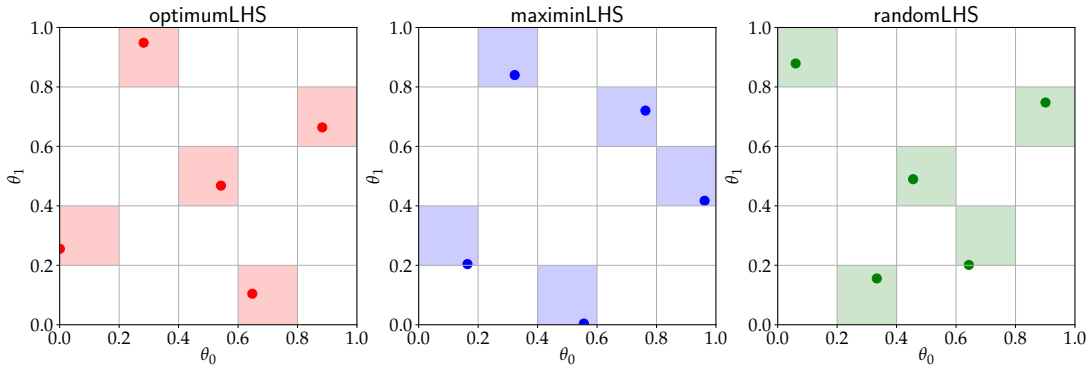


Figure 7.3 – Here we show three different options for generating the input training points. In the left, middle and right panel, we have `optimumLHS`, `maximinLHS` and `randomLHS` respectively. The `optimumLHS` attempts to separate the training points as far from each other, but is very expensive to compute due to the challenging optimisation procedure.

The training points (inputs to the emulator) are generated using Latin Hypercube (LH) Sampling (McKay et al., 1979). This is crucial since it has a better space filling property compared to standard sampling techniques. We use the publicly available R routine `lhs` to generate the LH samples. We have various options, for example, `maximinLHS`, `optimumLHS` and `randomLHS` to do so. In particular, from the 3 LH sampling technique, we have found that the `maximinLHS` procedure gives the best performance (see §7.2.5 for further details). Hence, throughout this paper, we use the `maximinLHS` procedure. Once the LH samples are generated, we compute and store the MOPED coefficients at these points in parameter space.

7.2.4 Inference Mechanism

The inference engine is built in two steps. The first step involves learning the kernel hyper-parameters while the second stage involves learning the posterior distribution of the cosmological and nuisance parameters.

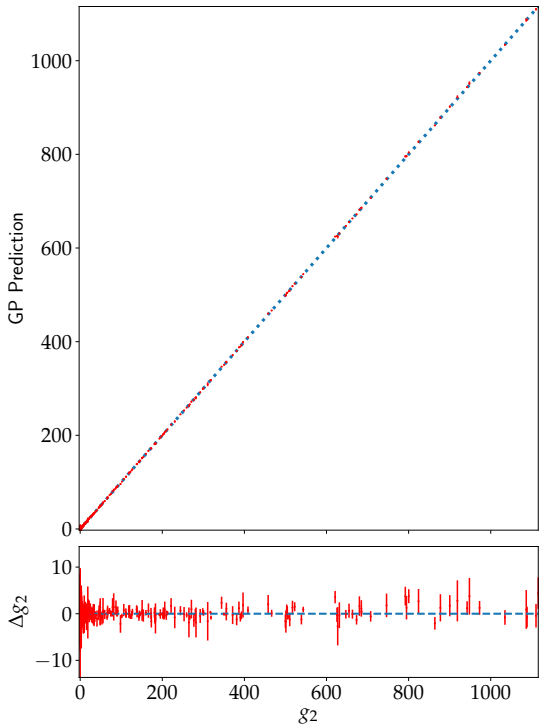


Figure 7.4 – In this figure, the $y-$axis corresponds to the predicted emulated function (here, the second MOPED coefficients) at 500 test points. These points do not form part of the training set. The $x-$axis shows the accurate MOPED coefficients as evaluated using CLASS. We do expect a one-to-one relationship between the predicted emulated function and the accurate g_i. The bottom panel shows the residuals, that is, $\Delta g_2 = \tilde{g}_2 - g_2$, where $\tilde{g}_2$ is the predicted function with the GP.

An important step prior to inferring the kernel hyper-parameters is to first pre-whiten the inputs such that the sample covariance matrix of the training set is just the identity matrix. We model each MOPED coefficient independently with a parametric function with a set of basis functions, $[1, \theta, \theta^2]$ and model the residual with a zero mean Gaussian Process model (see Equation 7.1.8). The marginal likelihood is then maximised with respect to the kernel hyper-parameters via optimisation.

Once the GP models are trained, they are stored. They can further be connected to an MCMC sampler to infer the full posterior distribution of the cosmological and nuisance parameters. Note that the process of calculating the different power spectra, band power and MOPED coefficients via the full solver is completely substituted with the GP models. Inference with the latter is not only around 30 times faster compared to the full solver but also faithfully generates robust credible intervals for the inferred parameters - see Figure 7.7.

7.2.5 Diagnostics for the Emulator

Various techniques have been proposed by Bastos & O'Hagan (2009) to assess the performance of an emulator. These diagnostics are generally based on the comparisons between the emulator and simulator runs for new test points in input parameter space. These test points should cover the input parameter space over which the training points were previously generated. In our approach, we randomly choose 500 independent test points from the prior range and evaluate the simulator and the emulator at these points.

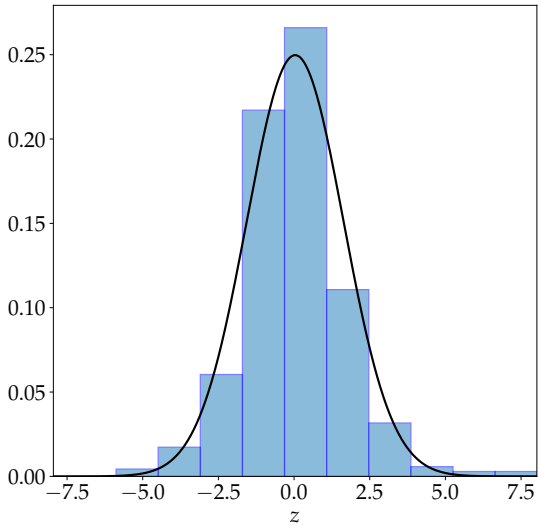


Figure 7.5 – In this figure, a histogram of the standardised residuals, $z = \tilde{g}_2 - g_2/\tilde{\sigma}_2$ is shown for the second MOPED coefficients. $\tilde{g}_2$ and $\tilde{\sigma}_2$ are the mean and standard deviations of the predicted MOPED coefficients with the GP model.

Despite a very simple diagnostic, one can adopt a graphical approach to inspect for any odd behaviour. For example, one can just plot the predictions from the emulator against the outputs from the simulator. It is expected that the points lie close to the 45° line through the origin. Moreover, the residual errors (see bottom panel of Figure 7.4 for an example) fluctuate around 0 and with no specific trends. Any odd behaviours in either plot (upper and lower panel of Figure 7.4) suggest different possible issues, for example, local fitting problem, non-stationary behaviour of the function or lack of training points in the parameter space. Sometimes, these issues are difficult to diagnose, especially in high dimensions or when we have multiple emulated functions as we do in our case.

A second diagnostic is the calculation of the standardised prediction error

$$z = \frac{\tilde{g}_i - g_i}{\tilde{\sigma}_i} \tag{7.2.3}$$

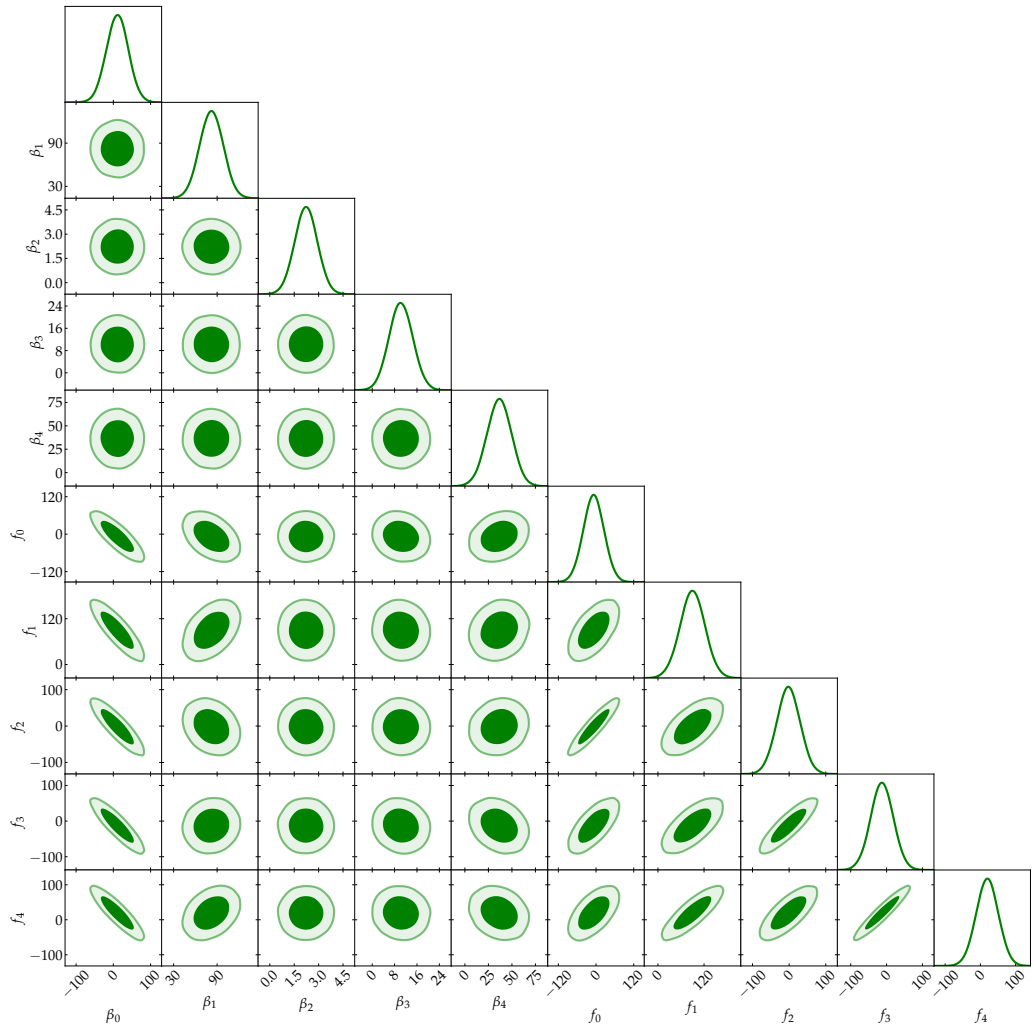


Figure 7.6 – Once the kernel hyper-parameters are learnt, the posterior distributions of $\boldsymbol{\beta}$ and $\boldsymbol{f}$ are inferred. This triangle plot shows the marginalised posterior of the first 5 regression coefficients from the $\boldsymbol{\beta}$ vector and the first 5 distributions of $\boldsymbol{f}$ vector. The inner and outer contours refer to the 68% and 95% credible intervals respectively.

where $\tilde{g}$ and $\tilde{\sigma}$ denotes the mean prediction and standard deviation of the i^{th} emulator. An emulator is deemed to faithfully represent the simulator if the standardised prediction error, z follow a Student-t distribution (Bastos & O'Hagan, 2009). Note that the shape of a Student-t distribution is analogous to a normal distribution. In the example shown in Figure 7.5, the mean and standard deviation are estimated to be 0.03 and 1.60 respectively and a normal distribution based on these two summary statistics is shown in black in Figure 7.5. If the number of training points is large enough, the distribution can be assumed to be standard normally distributed. Any large deviation from the bell-shaped distribution indicates some systematic problem, for example, misspecification of a prior mean function.

7.3 Results and Discussions

In the section, we describe the different results obtained as part of this exploratory analysis using semi-parametric Gaussian Process. Since we emulate $g_{\rm EE}$, $g_{\rm GI}$ and $g_{\rm II}$ and we have 11 parameters (6 cosmological and 5 nuisance), we have $3 \times 11 = 33$ MOPED coefficients (GP emulators). The systematic part is quick to compute and can just be added to the total MOPED coefficients, in other words, we have

$$y = g_{\rm tot} + g_{\rm sys}. \tag{7.3.1}$$

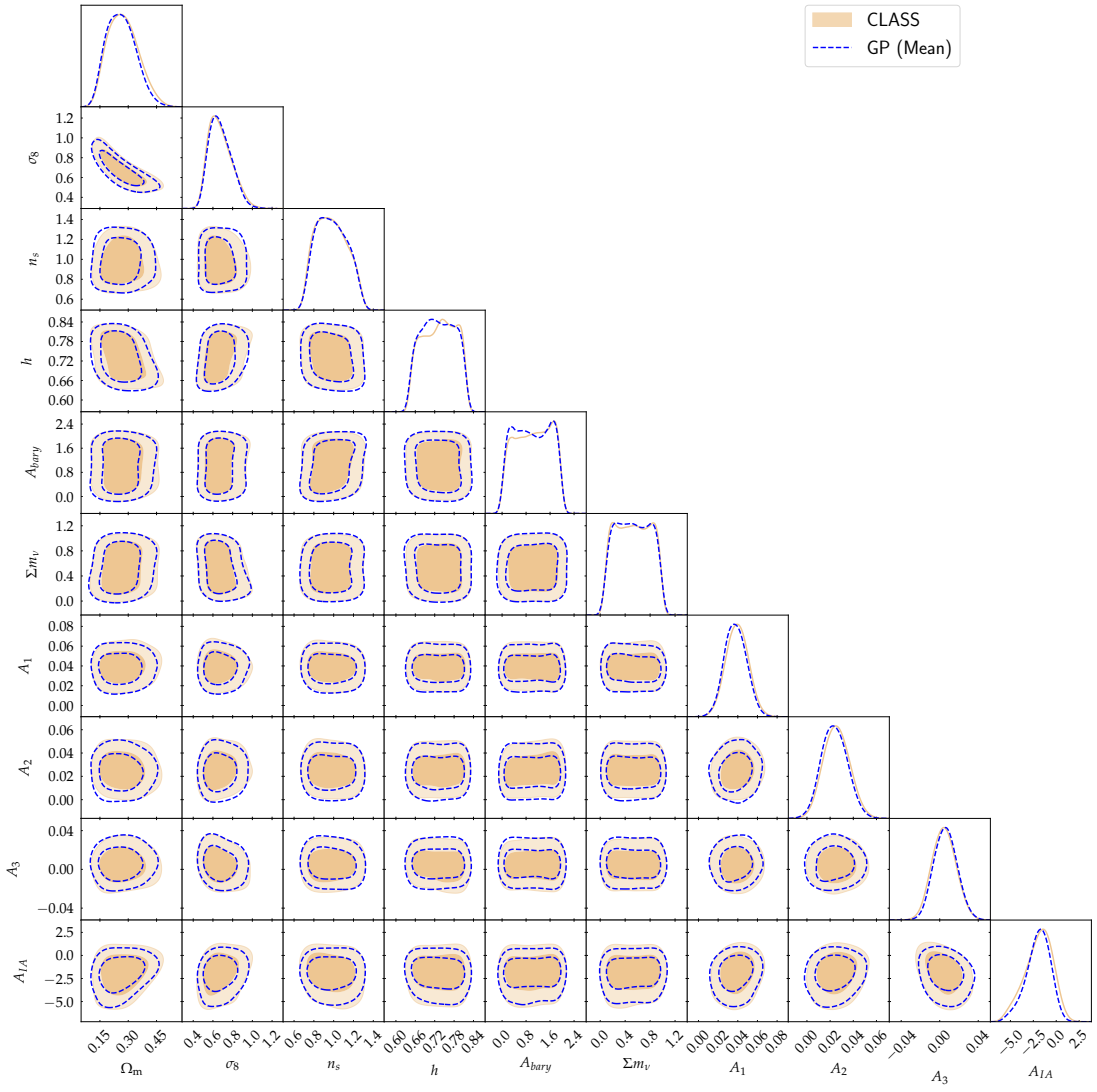


Figure 7.7 – The posterior distribution of the cosmological and nuisance parameters as obtained using the accurate solver CLASS and the kernel approach. The tan shading shows the posterior with CLASS while the credible intervals and marginalised posterior distributions using the emulator are shown in blue. The inner and outer contours correspond to 68% and 95% credible intervals respectively.

All the emulators show good performance, in terms of the prediction at test points. For example, in Figure 7.5, the standardised prediction error, that is, the difference between the

solver (CLASS) and the emulator, normalised by the GP uncertainty, is centred on zero and follows roughly a normal distribution. Moreover, if we look at the residuals in the lower panel in Figure 7.4, they are centred on the zero line.

Importantly, the semi-parametric approach allows one to recover a full posterior for the regression coefficients, $\boldsymbol{\beta}$ and the residuals, $\boldsymbol{f}$. In Figure 7.6, we show the full 1D and 2D posterior for the first five regressions coefficients $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4)$ and for the first five residuals as well, that is, $(f_1, f_2, f_3, f_4, f_5)$. Note that the residuals has the same shape as the number of training points. The plot also shows the full covariance between the regression coefficients and the residuals.

Once we have our emulator, we can sample the full posterior of the cosmological and nuisance parameters and the 1D and 2D projection of the inferred parameters are shown in Figure 7.7. Note that we plot a slightly different set of parameters to show the derived parameters, Ω_m and σ_8. We obtain remarkable posterior with the emulator compared to the full solver, CLASS with just 1000 training points.

7.4 Summary

In this chapter, we have investigated the semi-parametric Gaussian Process approach and we have shown that the method reproduces robust posterior densities for both cosmological and nuisance parameters. It is not only ~ 30 times faster than the full solver, but also allows us to embed an approximate function, by defining a set of basis functions, to emulate the different MOPED coefficients.

Now that we have a working methodology for performing emulation, an important ingredient of a weak lensing analysis is the 3D matter power spectrum, and it is in fact the most expensive part of a likelihood calculation. If we can have an emulator for $P_\delta(k, z)$, this will give us the flexibility of calculating all weak lensing and intrinsic alignment power spectra, since they all involve $P_\delta(k, z)$ in the integrations, which can be calculated swiftly with numerical techniques. Moreover, we can also specify our own inferred $n(z)$ distributions from galaxy surveys. Hence, in the next chapter, we will look into building an emulator for the 3D matter power spectrum.

Chapter 8

KERNEL-BASED EMULATOR FOR THE 3D MATTER POWER SPECTRUM FROM CLASS

> Creativity is just connecting things. When you ask creative people how they did something, they feel a little guilty because they didn't really do it, they just saw something. It seemed obvious to them after a while. That's because they were able to connect experiences they've had and synthesize new things.
>
> *Steve Jobs*

This chapter has been accepted for publication in Astronomy & Computing journal. The paper, code and documentation can be accessed through the following:

Kernel-Based Emulator for the 3D Matter Power Spectrum from CLASS (, ,)
A. Mootoovaloo, *A. Jaffe, A. Heavens and F. Leclercq*, arXiv:2105.02256, 2021

The content as put forward here is a large extract from the paper. A. Mootoovaloo led the project and code development. A. Heavens, A. Jaffe and F. Leclercq constantly provided feedbacks, guides and ideas to ensure successful completion of the project. From the cosmology perspective, A. Jaffe proposed the main idea of splitting the 3D matter power spectrum into three separate components.

8.1 Overview

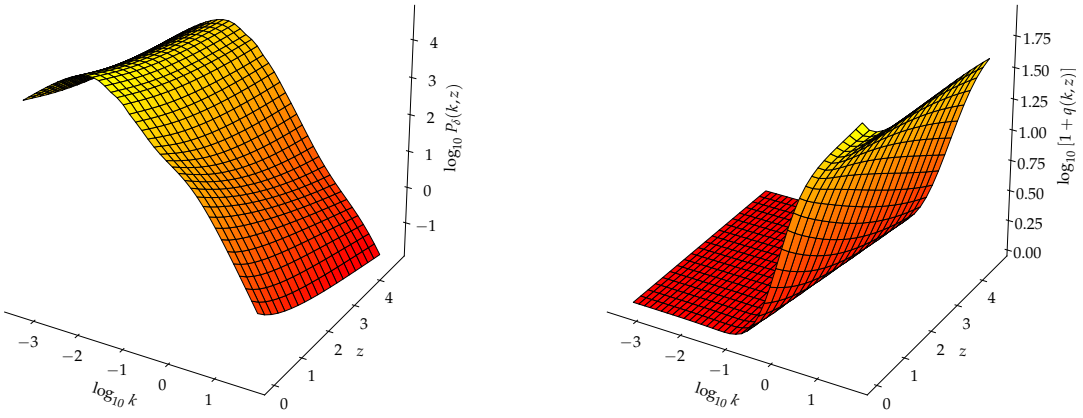
The 3D matter power spectrum, $P_\delta(k, z)$ is a fundamental quantity in the analysis of cosmological data such as large-scale structure, 21cm observations, and weak lensing. Existing computer models (Boltzmann codes) such as CLASS can provide it at the expense of immoderate

computational cost. In this work, we propose a fast Bayesian method to generate the 3D matter power spectrum, for a given set of wavenumbers, k and redshifts, z. Our code allows one to calculate the following quantities: the linear matter power spectrum at a given redshift (the default is set to 0); the non-linear 3D matter power spectrum with/without baryon feedback; the weak lensing power spectrum. The gradient of the 3D matter power spectrum with respect to the input cosmological parameters is also returned and this is useful for Hamiltonian Monte Carlo samplers. The derivatives are also useful for Fisher matrix calculations. In our application, the emulator is accurate when evaluated at a set of cosmological parameters, drawn from the prior, with the fractional uncertainty, $\Delta P_\delta / P_\delta$ centred on 0. It is also ~ 300 times faster compared to CLASS, hence making the emulator amenable to sampling cosmological and nuisance parameters in a Monte Carlo routine. In addition, once the 3D matter power spectrum is calculated, it can be used with a specific redshift distribution, $n(z)$ to calculate the weak lensing and intrinsic alignment power spectra, which can then be used to derive constraints on cosmological parameters in a weak lensing data analysis problem. The software (`emuPK`) can be trained with any set of points and is distributed on Github, and comes with with a pre-trained set of Gaussian Process (GP) models, based on 1000 Latin Hypercube (LH) samples, which follow roughly the current priors for current weak lensing analyses.

8.2 Introduction

The 3D matter power spectrum, $P_\delta(k, z)$ is a key quantity which underpins most cosmological data analysis, such as galaxy clustering, weak lensing, 21 cm cosmology and various others. Crucially, the calculation of other (derived) power spectra can be fast if $P_\delta(k, z)$ is precomputed. In practice, the latter is the most expensive component and can be calculated either using Boltzmann solvers such as CLASS or CAMB, or via simulations, which can be computationally expensive depending on the resolution of the experiments.

For the past 30 decades or so, with the advent of better computational facilities, various techniques have been progressively devised and applied to deal with inference in cosmology. In brief, some of these techniques include Monte Carlo (MC) sampling, variational inference, Laplace approximation and recently we are witnessing other new approaches such as density estimation (Alsing et al., 2018, 2019; Alsing & Wandelt, 2019) which makes use of tools like Expectation-Maximisation (EM) algorithm and neural networks (NN). Recently, Charnock et al. (2018) designed the information maximizing neural networks (IMNNs) to learn non-linear functional of data that maximize Fisher information. In this work, we explore another branch



(a) 3D Matter Power Spectrum, $P_\delta(k,z)$ **(b)** The function $1+q(k,z)$

Figure 8.1 – The left panel shows the 3D matter power spectrum at a fixed input cosmology to CLASS for $k \in [5 \times 10^{-4}, 50]$ and $z \in [0.0, 4.66]$. The grid shows the region where we choose to model the function, that is, 40 wavenumbers, equally spaced in logarithm scale and 20 redshifts, equally spaced in linear scale.

of Machine Learning (ML) which deals with kernel techniques.

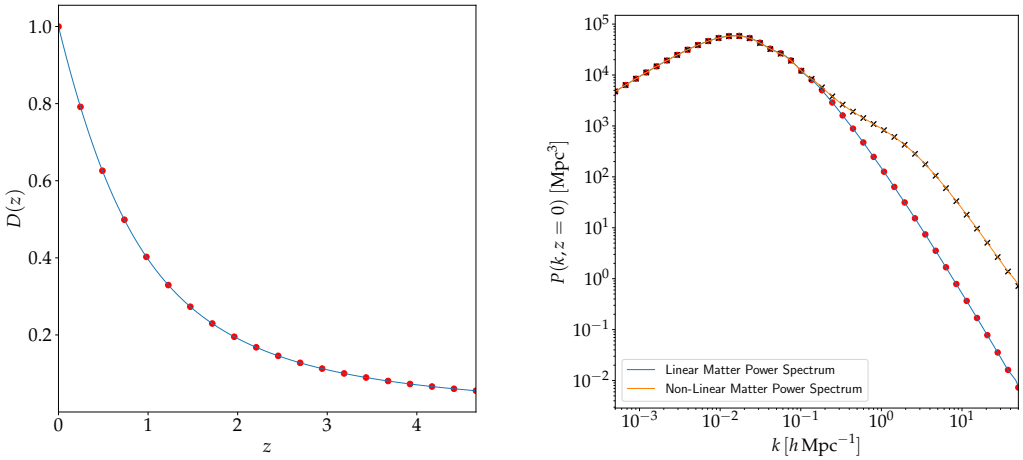
The ML techniques discussed previously will slowly pave their way in various weak lensing (WL) analysis. Indeed, in the analysis of the cosmic microwave background (CMB), Fendt & Wandelt (2007b) designed the Parameters for the Impatient Cosmologist (PICO) algorithm for interpolating CMB power spectra at test points in the parameter space. In the same spirit, Auld et al. (2007) built a neural network algorithm, which they refer to as CosmoNet, for interpolating CMB power spectra. Neural networks have been used in other applications as well, for example, in simulations. Agarwal et al. (2012, 2014) used neural networks for interpolating non-linear matter power spectrum based on 6 cosmological parameters while Schmit & Pritchard (2018) used neural networks for emulating the 21cm power spectrum in the context of epoch of reionisation. In the context of weak lensing analysis, Manrique-Yus & Sellentin (2020) used neural networks for accelerating cosmological parameter inference by combining cosmic shear, galaxy clustering, and tangential shear. While we were finishing this work, the work of Aricò et al. (2021) and Ho et al. (2021), both related to emulating the matter power spectrum, appeared on arXiv. In particular, Ho et al. (2021) used GPs to build an emulator for the matter power spectrum at fixed redshifts using N-body simulations while Aricò et al. (2021) used neural networks and a combination of LH points (an 8D input parameter space with 156 000 training points), which they refer to as the standard and cosmological space to emulate the linear matter power spectrum as well as other cross-spectra of linear fields. Spurio Mancini et al. (2021) also used neural networks to emulate the 3D non-linear matter power spectrum,

with at least 10^5 training points depending on their applications and the redshift is also treated as an input to the neural network.

On the other end, Gaussian Processes have been used in the Coyote Universe collaboration (Habib et al., 2007; Heitmann et al., 2009, 2010, 2014; Lawrence et al., 2010) for emulating the matter power spectrum for large-scale simulations. Recently, Leclercq (2018) used Gaussian Processes in the context of likelihood-free inference, where the data (training points) is augmented in an iterative fashion via Bayesian Optimisation, hence the procedure being referred to as Bayesian Optimisation for Likelihood-Free Inference, BOLFI (Gutmann & Corander, 2016). Each emulating scheme has its own pros and cons (we defer to §8.7 for a short discussion on the advantages and possible limitations of Gaussian Processes).

Different emulating schemes have been designed for the matter power spectrum and most of them are based on combining Singular Value Decomposition (SVD) and Gaussian Processes. The emulator from Habib et al. (2007) is among the first in the context of large simulations. Emulating $P_\delta(k, z)$ is not a trivial task because it is strictly a function of 3 inputs, k, the wavenumber, z, the redshift and $\boldsymbol{\theta}$, the cosmological parameters. Neural networks seem to be the obvious choice because they can deal with multiple outputs but they generally require a large number of training points.

Our contributions in this work are three fold. First, it addresses the point that we do not always need to assume a zero mean Gaussian Process model for performing emulation, in other words, one can also include some additional basis functions prior to defining the kernel matrix. This can be useful if we already have an approximate model of our function. Moreover, if we know how a particular function behaves, one can adopt a stringent prior on the regression coefficients for the parametric model, hence allowing us to encode our degree of belief about that specific parametric model. Second, since we are using a Radial Basis Function (RBF) kernel and the fact that it is infinitely differentiable enables us to estimate the first and second derivatives of the 3D matter power spectrum. The derived expressions for the derivatives also indicate that there is only element-wise matrix multiplication and no matrix inverse to compute. This makes the gradient calculations very fast. Finally, with the approach that we adopt, we show that the emulator can output various key power spectra, namely, the linear matter power spectrum at a reference redshift z_0 and the non-linear 3D matter power spectrum with/without an analytic baryon feedback model. Moreover, using the emulated 3D power spectrum and the tomographic redshift distributions, we also show that the weak lensing power spectrum and the intrinsic alignment (II and GI) can be generated in a very fast way using existing numerical



(a) The growth factor, $D(z)$

(b) The linear and non-linear matter power spectrum

Figure 8.2 – The left panel shows the growth factor, as function of redshift. In this case, to generate the training set, the growth factor is calculated at 20 redshifts, equally spaced in linear scale (shown by the red scatter points) and the linear matter power spectrum, $P_{\rm lin}(k, z_0)$ is calculated at 40 different wavenumbers, k, equally spaced in logarithm space (red scatter points). We also show the non-linear matter power spectrum in (b). These functions are evaluated at different cosmological parameters to build a training set.

techniques.

In Mootoovaloo et al. (2020), we found that using the mean of the GP and ignoring the error always results in better posterior densities. This is a known feature when GPs emulate a deterministic function (Bastos & O'Hagan, 2009). As a result, in this work, we work only with the mean of the GP in all experiments. Importantly, we use 1000 training points and once the emulator is trained and stored, it takes about 0.1 seconds to generate the non-linear 3D matter power spectrum, compared to CLASS which takes about 30 seconds to generate an accurate and smooth power spectrum. Hence, the method presented in this work also opens a new avenue towards building emulators for large-scale simulations where a single high-resolution forward simulation might take minutes to compute.

The chapter is organised as follows: in §8.3, we describe the 3D power spectrum, which can be decomposed in different components, and the analytic baryon feedback model, which can be used in conjunction with $P_\delta(k, z)$. In §8.4 and §8.5, we highlight briefly the steps behind building the emulator. In §8.6, using a pair of toy $n(z)$ tomographic redshift distributions, we show how the emulator can be used to generate different weak lensing power spectra and in §8.7, we describe briefly the different functionalities that the code supports and we highlight the main results in §8.8. Finally, we conclude in §8.9.

8.3 Model

In this section, we describe the model which we want to emulate. Central to the calculation is the 3D matter power spectrum, $P_\delta(k, z; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ refers to a vector of cosmological parameters. In what follows, we will drop the $\boldsymbol{\theta}$ vector notation for clarity. The matter power spectrum is generally the most expensive part to calculate, especially if one chooses to use large-scale simulation to generate the 3D matter power spectrum. In the simple case, one can just emulate $P_\delta(k, z)$ but we consider a different approach, which enables us to include baryon feedback, to calculate the linear matter power spectrum at a reference redshift and to calculate the non-linear 3D matter power spectrum itself.

Baryon feedback is one of the astrophysical systematics which is included in a weak lensing analysis. This process is not very well understood but is deemed to modify the matter distribution at small scales, hence resulting in the suppression of the matter power spectrum at large multipoles. In general, to model these effects, large hydrodynamical simulations provide a proxy to model baryon feedback. In particular, it is quantified via a bias function, $b^2(k, z)$ such that the resulting modified 3D matter power spectrum can be written as

$$P_\delta^{\text{bary}}(k, z) = b^2(k, z) P_\delta(k, z), \tag{8.3.1}$$

where $P_\delta^{\text{bary}}(k, z)$ and $P_\delta(k, z)$ are the 3D matter power spectra, including and excluding baryon feedback respectively. The bias function is modelled by the fitting formula

$$b^2(k, z) = 1 - A_{\text{bary}} \left[A_z e^{(B_z x - C_z)^3} - D_z x e^{E_z x} \right] , \tag{8.3.2}$$

where A_{bary} is a flexible nuisance parameter and we allow it to vary over the range $A_{\text{bary}} \in [0.0, 2.0]$. The quantity $x = \log_{10}(k\ [\text{Mpc}^{-1}])$, and A_z, B_z, C_z, D_z and E_z depend on the redshift and other constants. See Harnois-Déraps et al. (2015) for details and functional forms. Note that setting $A_{\text{bary}} = 0$ implies no baryon feedback. Moreover, since we have a functional form for the baryon feedback model, which is not expensive to compute, we will apply it as a bolt-on function on top of the emulated non-linear 3D matter power spectrum.

Next, we consider the non-linear 3D matter power spectrum without baryon feedback. It can be decomposed into three components as follows:

$$P_\delta(k, z) = D(z)[1 + q(k, z)] P_{\text{lin}}(k, z_0) \tag{8.3.3}$$

where $D(z)$ is the growth factor, $q(k,z)$ is a 2D function (in terms of k and z) representing the scale-dependence of the growth factor, plus the non-linear contributions and $P_{\rm lin}(k,z_0)$ is the linear matter power spectrum at fixed redshift z_0. See Figures 8.1 and 8.2 for an illustration of the decomposition of the 3D matter power spectrum at fixed cosmological parameters. Emulating the three different components separately has the advantage of calculating the linear matter power spectrum at the reference redshift for any given input cosmology.

Following current weak lensing analysis, we define some bounds on the redshifts, z and wavenumbers, k. For example, the maximum redshift in the tomographic weak lensing analysis performed by Köhlinger et al. (2017) is ~ 5 and the maximum wavenumber is set to 50. With these numbers in mind, we choose $z \in [0.0, 5]$ and $k \in [5 \times 10^{-4}, 50]$. We will elaborate more on these settings in the sections which follow. On the other hand, for the cosmological parameters, we assume the following range to generate the training set:

Table 8.3.1 – Default parameter prior range inputs to the emulator

Description	**Range**
CDM density, $\Omega_{\rm cdm}h^2$	[0.06, 0.40]
Baryon density, $\Omega_{\rm b}h^2$	[0.019, 0.026]
Scalar spectrum amplitude, $\ln(10^{10}A_s)$	[1.70, 5.0]
Scalar spectral index, n_s	[0.7, 1.3]
Hubble parameter, h	[0.64, 0.82]

Current weak lensing analyses also assume a fixed sum of neutrino mass, Σm_ν. Hence, in all experiments, $\Sigma m_\nu = 0.06\,\text{eV}$. This quantity can be fixed by the user prior to running all experiments with the pipeline we have developed. However, we can also treat it as a varying parameter before building the emulator.

8.4 Procedures

In the existing likelihood code from Köhlinger et al. (2017), the accurate solver, CLASS, is queried at 39 wavenumbers k and 72 redshifts z, corresponding to the centres of each tophat in the $n(z)$ distribution and a standard spline interpolation is carried out along the k axis. Following a similar approach, we choose to have a model of the $P_\delta(k,z)$ at 40 values of k, equally spaced on a logarithmic grid and 20 values of redshift, equally spaced in linear scale from 0 to 4.66 (the maximum redshift in the KiDS-450 analysis) and we can perform a standard 2D interpolation, such as spline interpolation, along k and z. See Figures 8.1 and 8.2 for an illustration.

In this section, we will walk through the steps to build a model for the 3D matter power

spectrum. It is organised as follows: in §8.4.1 we discuss how the input training points are generated and this is crucial for the emulator to work with a reasonable number of training points. We will use the concepts derived in §7.1.2 in Chapter 7 to model the different output from CLASS.

We denote the response (or target), that is the function we want to model as y. In this particular case, we have three different components, namely the growth factor, $D(z)$, the $q(k, z)$ function and the linear matter power spectrum $P_{\rm lin}(k, z_0)$. We assume we have run the simulator, CLASS, at N design points, $\boldsymbol{\theta}$, such that we have a training set, $\{\boldsymbol{\theta}, \boldsymbol{y}_i\}$. Throughout this work, we are using the fitting function, *halofit* (Takahashi et al., 2012) implemented in CLASS to generate the training set. The index i corresponds to the $i^{\rm th}$ response. Note that in our application, we model each function independently with the emulating scheme proposed below.

8.4.1 Training Points

An important ingredient in designing a robust emulator lies in generating the input training points. Points which are drawn randomly and uniformly from the pre-defined range (see Table 8.3.1) do not show a space-filling property. As the dimensionality of the problem increases, the volume increases and hence there are large spatial fluctuations in the density of points. Hence, the emulator will lack information of its neighbourhood and the prediction can be very poor in these regions. Moreover, one would need a large number of training points to accurately model the power spectrum. For example, a recent work by Spurio Mancini et al. (2021) shows that one would need $\sim 10^5$ training points to build an emulator with deep neural networks with uniform random sampling.

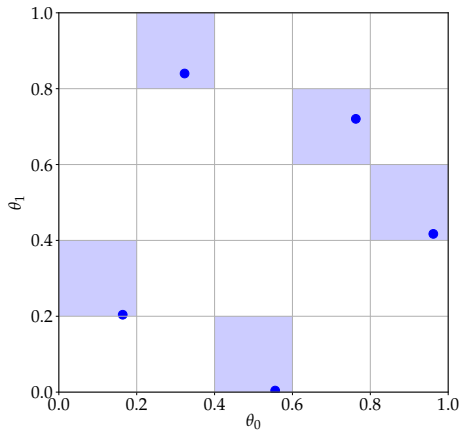


Figure 8.3 – An example of a Latin Hypercube (LH) design in two dimensions. 5 LH points are drawn randomly using the `maximin` procedure and each point will occupy a single cell, that is, if a point occupies cell (i, j), then there is not a point occupying cell (j, i). This procedure remains exactly the same when we generate LH samples from a hypercube.

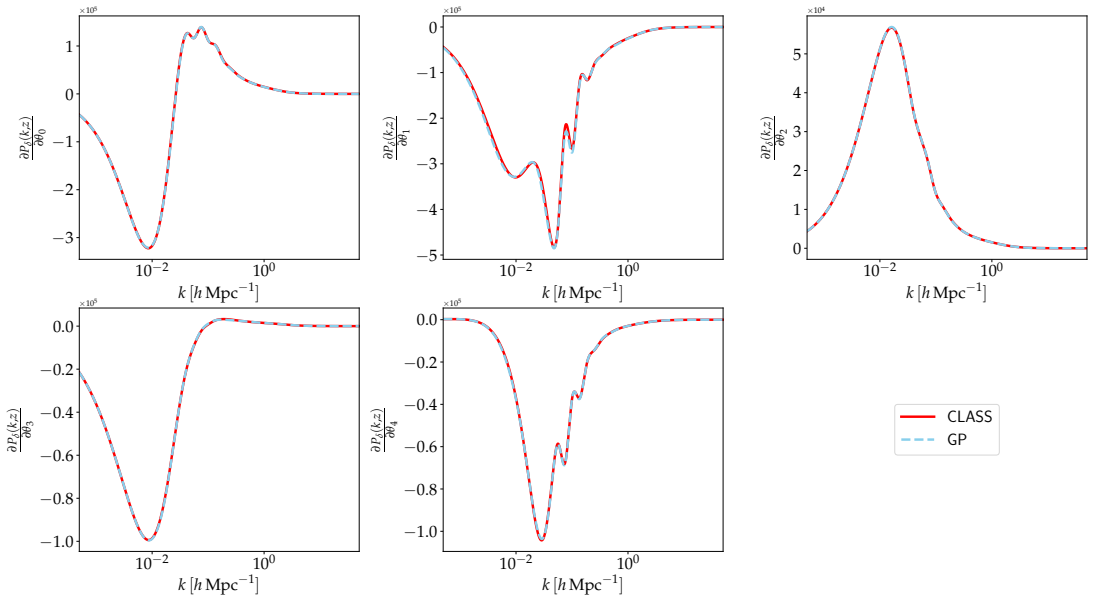


Figure 8.4 – Gradients with respect to the input cosmologies. $\boldsymbol{\theta}$ corresponds to the following cosmological parameters: $\boldsymbol{\theta} = (\Omega_{\mathrm{cdm}}h^2,\ \Omega_{\mathrm{b}}h^2,\ \ln(10^{10}A_s),\ n_s,\ h)$. Note that since we are emulating the 3D power spectrum, the gradient is also a 3D quantity. In this figure, we are showing the predicted function with the GP model in broken blue and the accurate gradient function calculated with CLASS in solid red, at a fixed redshift.

To circumvent these issues, the natural choice is to instead generate Latin Hypercube (LH) samples, which demonstrate a space-filling property as shown in Figure 8.3. The idea behind a LH design is that a point will always occupy a single cell. For example, if we consider the design shown in Figure 8.3, each column and row contains precisely one training point (in 2D). Similarly, in a 3D case, each row, column and layer will have one training point and this extends to higher dimensions. Intuitively, for a 2D design, this is analogous to the problem of positioning n rooks on an $n \times n$ chessboard such that they do not attack each other. This ensures that the LH points generated cover the parameter space as much as possible, hence enabling the emulator to predict the targets at test points. `emuPK` can also be trained on a different set of training points, for example, one which has been generated using a different LH sampling scheme.

8.5 Gradients

An important by-product from the trained model is the gradient of the emulated function with respect to the input parameters. This can be of paramount importance if we are using a sophisticated Monte Carlo sampling scheme such as Hamiltonian Monte Carlo (HMC) to infer cosmological parameters in a Bayesian analysis. The gradients of the log-likelihood with respect to the cosmological parameters are important in such a sampling scheme. Hence, with some linear algebra and using the gradient of the power spectra, generated with the emulator, the

desired gradients can be derived. The analytical gradient of the mean function with respect to the inputs, at a fixed redshift and wavenumber is

$$\frac{\partial \bar{y}_*}{\partial \boldsymbol{\theta}_*} = \frac{\partial \boldsymbol{\Phi}_*}{\partial \boldsymbol{\theta}_*}\hat{\boldsymbol{\beta}} + \left[k_* \odot \mathbf{Z}_* \boldsymbol{\Omega}^{-1}\right]^{\mathrm{T}} \mathbf{K}_y^{-1}(\boldsymbol{y} - \boldsymbol{\Phi}\hat{\boldsymbol{\beta}}) \tag{8.5.1}$$

where $\odot$ refers to element-wise multiplication (Hadamard product). $\mathbf{Z}_* \in \mathbb{R}^{N \times d}$ corresponds to the pairwise difference between the test point, $\boldsymbol{\theta}_*$ and the training points, that is, $\mathbf{Z}_* = [\boldsymbol{\theta}_1 - \boldsymbol{\theta}_*, \boldsymbol{\theta}_2 - \boldsymbol{\theta}_* \ldots \boldsymbol{\theta}_N - \boldsymbol{\theta}_*]^{\mathrm{T}}$. Importantly, as seen from equation 8.5.1, the gradient is the sum of the gradients corresponding to the parametric part and the residual, which is modelled by a kernel. Moreover, higher order derivatives can also be calculated analytically. For example, the second order auto- and cross- derivatives are

$$\frac{\partial^2 \bar{y}_*}{\partial \boldsymbol{\theta}_*^2} = \frac{\partial^2 \boldsymbol{\Phi}_*}{\partial \boldsymbol{\theta}_*^2}\hat{\boldsymbol{\beta}} + \left[\Omega^{-1}\frac{\partial \boldsymbol{k}_*}{\partial \boldsymbol{\theta}_*}\mathbf{Z}_* - \Omega^{-1} \odot \boldsymbol{k}_*\right] \mathbf{K}_y^{-1}(\boldsymbol{y} - \boldsymbol{\Phi}\hat{\boldsymbol{\beta}}). \tag{8.5.2}$$

As a result of this procedure, one can analytically calculate the first and second derivatives of an emulated function using kernel methods. While the first derivatives are particularly useful in HMC sampling method, the second derivatives are more relevant in the calculation of, for example, the Fisher information matrix.

Once the gradients with respect to each component of the non-linear 3D matter power spectrum are derived, the first and second derivatives with respect to the non-linear matter spectrum can be derived via chain rule and are given by:

$$\frac{\partial P_\delta}{\partial \boldsymbol{\theta}} = \frac{\partial D}{\partial \boldsymbol{\theta}}(1+q)P_{\mathrm{lin}} + D\frac{\partial q}{\partial \boldsymbol{\theta}}P_{\mathrm{lin}} + D(1+q)\frac{P_{\mathrm{lin}}}{\partial \boldsymbol{\theta}} \tag{8.5.3}$$

and

$$\begin{aligned}\frac{\partial^2 P_\delta}{\partial \boldsymbol{\theta}^2} = &\frac{\partial^2 D}{\partial \boldsymbol{\theta}^2}(1+q)P_{\mathrm{lin}} + D\frac{\partial^2 q}{\partial \boldsymbol{\theta}^2}P_{\mathrm{lin}} + D(1+q)\frac{\partial^2 P_{\mathrm{lin}}}{\partial \boldsymbol{\theta}^2} \\ &+ 2\frac{\partial D}{\partial \boldsymbol{\theta}}\frac{\partial q}{\partial \boldsymbol{\theta}}P_{\mathrm{lin}} + 2\frac{\partial D}{\partial \boldsymbol{\theta}}(1+q)\frac{\partial P_{\mathrm{lin}}}{\partial \boldsymbol{\theta}} + 2D\frac{\partial q}{\partial \boldsymbol{\theta}}\frac{\partial P_{\mathrm{lin}}}{\partial \boldsymbol{\theta}}.\end{aligned} \tag{8.5.4}$$

Once $\mathbf{K}_y^{-1}(\boldsymbol{y} - \boldsymbol{\Phi}\hat{\boldsymbol{\beta}})$ is pre-computed (after learning the kernel hyperparameters, $\boldsymbol{\nu}$) and stored, the first and second derivatives can be computed very quickly. In the case of finite difference methods, if a poor finite step size is specified, numerical derivatives can become unstable. This is not the case in this framework. In Figure 8.4, we show the first derivatives with respect to the input cosmological parameters, $\boldsymbol{\theta} = (\Omega_{\mathrm{cdm}}h^2, \Omega_{\mathrm{b}}h^2, \ln(10^{10}A_s), n_s, h)$. The first derivatives with CLASS (in red) are calculated using finite central difference method.

8.6 Weak Lensing Power Spectra

A crucial application of the 3D matter power spectrum is in a weak lensing analysis, where the calculation of the different power spectra types is required. In the absence of systematics, most of the cosmological information lies in the curl-free (E-) component of the shear field. The Limber approximation (Limber, 1953; Loverde & Afshordi, 2008) is typically assumed and under the assumption of no systematics, the E-mode lensing power spectrum is equal to the convergence power spectrum and is given by:

$$C^{\mathrm{EE}}_{\ell,ij} = \int_0^{\chi_H} \mathrm{d}\chi \, \frac{w_i(\chi) w_j(\chi)}{\chi^2} \, P_\delta^{\mathrm{bary}}(k,\chi). \tag{8.6.1}$$

and

$$w_i(\chi) = A\chi(1+z) \int_\chi^{\chi_{\mathrm{H}}} \mathrm{d}\chi' \, n_i(\chi) \left(\frac{\chi' - \chi}{\chi'} \right) \tag{8.6.2}$$

where $A = 3H_0^2\Omega_{\mathrm{m}}/(2c^2)$. χ is the comoving radial distance, χ_{H} is the comoving distance to the horizon, H_0 is the present day Hubble constant and Ω_{m} is the matter density parameter. w_i is the weight function which depends on the lensing kernel. The weight function is a measure of the lensing efficiency for tomographic bin i. Moreover, the redshift distribution, $n_i(z)$, as a function of the redshift, is related to the comoving distance via a Jacobian term, that is, $n(z)\,\mathrm{d}z = n(\chi)\,\mathrm{d}\chi$ and it is also normalised as a probability distribution, that is, $\int n(z)\,\mathrm{d}z = 1$.

8.6.1 Intrinsic Alignment Power Spectra

An important theoretical astrophysical challenge for weak lensing is intrinsic alignment (IA). It gives rise to preferential and coherent orientation of galaxy shapes, not because of lensing alone but due to other physical effects. Although not very well understood, it is believed to arise by two main mechanisms, namely the interference (GI) and intrinsic alignment (II) effects, such that the total signal is in fact a biased tracer of the true underlying signal, $C^{\mathrm{EE}}_{\ell,ij}$, that is,

$$C^{\mathrm{tot}}_{\ell,ij} = C^{\mathrm{EE}}_{\ell,ij} + A^2_{\mathrm{IA}} C^{\mathrm{II}}_{\ell,ij} - A_{\mathrm{IA}} C^{\mathrm{GI}}_{\ell,ij} \tag{8.6.3}$$

where A_{IA} is a free amplitude parameter, which allows for the flexibility of varying the strength of the power, arising due to the intrinsic alignment effect. In particular, the II term arises as a result of alignment of a galaxy in its local environment whereas the GI term is due to the correlation between the ellipticities of the foreground galaxies and the shear of the background

galaxies. Note that, the II term contributes positively towards the total lensing signal whereas the GI subtracts from the signal. The II power spectrum is given by

$$C^{\rm II}_{\ell,ij} = \int_0^{\chi_{\rm H}} {\rm d}\chi \, \frac{n_i(\chi)\, n_j(\chi)}{\chi^2} \, P_\delta^{\rm bary}(k,\chi) \, F^2(\chi) \tag{8.6.4}$$

and the GI power spectrum is

$$C^{\rm GI}_{\ell,ij} = \int_0^{\chi_{\rm H}} {\rm d}\chi \, \frac{w_i(\chi) n_j(\chi) + w_j(\chi) n_i(\chi)}{\chi^2} \, P_\delta^{\rm bary}(k,\chi) \, F(\chi), \tag{8.6.5}$$

where $F(\chi) = C_1 \rho_{\rm crit} \Omega_{\rm m} / D(\chi)$. $D(\chi)$ is the linear growth factor normalised to unity today, $C_1 = 5 \times 10^{-14} \ h^{-2} {\rm M}_\odot^{-1} {\rm Mpc}^3$ and $\rho_{\rm crit}$ is the critical density of the Universe today. As seen from Equations 8.6.1, 8.6.4 and 8.6.5, they all involve an integration of the form

$$C_\ell = \int_0^{\chi_{\rm H}} g(\chi) \, P_\delta^{\rm bary}(k,\chi) \, {\rm d}\chi. \tag{8.6.6}$$

Hence, an emulator for $P_\delta(k,z)$ will enable us to numerically compute all the weak lensing power spectra in a fast way. This will be useful in future weak lensing surveys where we will require many power spectra calculation as a result of the large number of auto- and cross-tomographic bins. For example, in the recent KiDS-1000 analysis (Asgari et al., 2021), five tomographic bins were employed, resulting in 15 (multiplied by 3 if we are including intrinsic alignment power spectra) power spectra calculations. In future surveys, it is expected that the number of redshift bins will be of the order 10, thus requiring at least 55 power spectra calculations for each power spectrum type (EE, GI and II).

8.6.2 Redshift Distribution

An important quantity for calculating the weak lensing power spectra is the redshift distribution. For an in-depth cosmological data analysis such as the Kilo Degree Survey (KiDS), it is crucial to calibrate the photometric redshift to obtain robust model predictions. For more advanced techniques for estimating the $n(z)$ from photometric redshifts, we refer the reader to techniques such as weighted direct calibration, DIR (Lima et al., 2008; Bonnett et al., 2016), calibration with cross-correlation, CC (Newman, 2008) and recalibration of photometric $P(z)$, BOR by Bordoloi et al. (2010). Recently Leistedt et al. (2016) developed a hierarchical Bayesian inference method to infer redshift distributions from photometric redshifts.

In this work, we use a toy Gaussian distribution to illustrate how we can use the 3D matter power spectrum, $P_\delta(k,z)$ in conjunction with the $n(z)$ distribution to calculate the different

weak lensing power spectra. Note that one can just replace this toy $n(z)$ distribution example by any redshift distribution as calculated by any one of the techniques mentioned above. Different $n(z)$ distributions are available as part of the software. The first 2 distributions are:

$$n(z) = B\, z^2 \exp\left(-\frac{z}{z_0}\right) \tag{8.6.7}$$

and

$$n(z) = B\, \alpha z \exp\left[-\left(\frac{z}{z_0}\right)^{\beta}\right]. \tag{8.6.8}$$

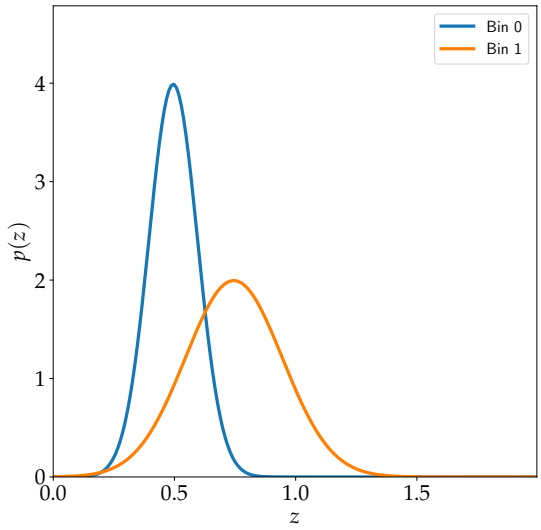


Figure 8.5 – To illustrate the calculation of the weak lensing power spectra, we use two analytic redshift distributions centred at redshift 0.50 and 0.75 respectively. The $n(z)$ distribution assumed here is a normal distribution and is given by Equation 8.6.9. The standard deviations for each normal distribution are set to 0.1 and 0.2 respectively.

For a *Euclid*-like survey, $z_0 \sim 0.7$, $\alpha = 2$ and $\beta = 1.5$ (Leonard et al., 2015). The third distribution implemented is just a Gaussian distribution with mean z_0 and standard deviation, σ

$$n(z) = B \exp\left[-\frac{1}{2}\left(\frac{z - z_0}{\sigma}\right)^2\right] \tag{8.6.9}$$

where B is a normalisation factor such that $\int n(z)\, \mathrm{d}z = 1$ in all cases above. As shown in Figure 8.5, we employ two redshift distributions, where the mean and standard deviation for the first distribution is 0.50 and 0.10 respectively and for the second distribution (in orange), the mean and standard deviation are set to 0.75 and 0.20 respectively.

8.7 Software

In this section, we briefly elaborate on how the code is set up and the different functionalities one can exploit. Note that any default values mentioned below can be adjusted according to the user's preferences. The default values of the minimum and maximum redshifts are set to 0 and 4.66 respectively and as discussed in §7.2, we also assume 20 redshifts spaced equally in the linear scale. For the wavenumbers in units of $h\,\mathrm{Mpc}^{-1}$, the minimum is set to 5×10^{-4} and the maximum to 50, with 40 wavenumbers equally spaced in logarithmic scale. A fixed neutrino mass of 0.06 eV is assumed but this can also be fixed at some other value or it can also be included as part of the emulation strategy. The code supports either choice.

The next step involves generating the training points. We generate 1000 LH design points using the `maximinLHS` function and we calculate and record the three quantities, the growth factor, $D(z)$, the non-linear function, $q(k,z)$ and the linear matter power spectrum, $P_{\rm lin}(k,z_0)$. At a very small value of k, which we refer to as $k_{\rm min}$, $q = 0$. The non-linear matter power spectrum is only relevant for some range of k_{nl} and $k_{nl} > k_{\rm min}$. Hence, the growth factor is just

$$D(z) = \frac{P_{\rm lin}(k_{\rm min}, z)}{P_{\rm lin}(k_{\rm min}, z_0)} \tag{8.7.1}$$

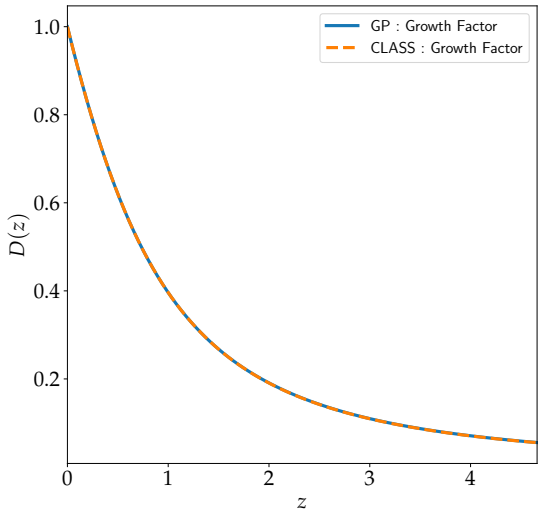


Figure 8.6 – The growth factor, $D(z)$ as predicted by the surrogate model (in blue) at a test point in parameter space. The accurate function is also calculated using CLASS and is shown in orange. Recall that the emulator is constructed for $z \in [0.0, 4.66]$, aligned with current weak lensing surveys.

Throughout our analysis, we use $z_0 = 0$. In some regions of the parameter space, we also found that the $q(k,z)$ were noisy and this can be alleviated by increasing the parameter `P_k_max_h/Mpc` when running CLASS. If a small value is assumed, the interpolation in the high-dimensional

space will not be robust. We set this value to 5000 to ensure the $q(k, z)$ function remains smooth as a function of the inputs. However, this procedure leads to CLASS being slower. It takes $\sim$ 30 seconds on average to do 1 forward simulation. For example, in our application, it took 520 minutes to generate the targets $(D,\ q,\ P_{\rm lin})$ for 1000 input cosmologies. We have also found that CLASS occasionally fails to compute the power spectrum and this is resolved as follows. We allocate a time frame (60 seconds in this work) for CLASS to attempt to calculate the power spectrum and if it fails, a small perturbation is added to the input training point parameters and we re-run CLASS, until the power spectrum is successfully calculated. In the failing cases, the maximum number of attempts is only 3. Moreover, the code currently supports polynomial functions of order 1 and 2, that is, the set of basis functions for an order 2 polynomial is $[1,\ \boldsymbol{\theta},\ \boldsymbol{\theta}^2]$. For example, Schneider et al. (2011) implemented a first and second order polynomial function to design an emulator for the CMB while Fendt & Wandelt (2007b) used a fourth order polynomial function. In this case, recall that we are also marginalising over the residuals analytically by using the kernel function. Training the emulator, that is, learning the kernel hyperparameters, for the different targets, took around 340 minutes. All experiments were conducted on an Intel Core i7-9700 CPU desktop computer.

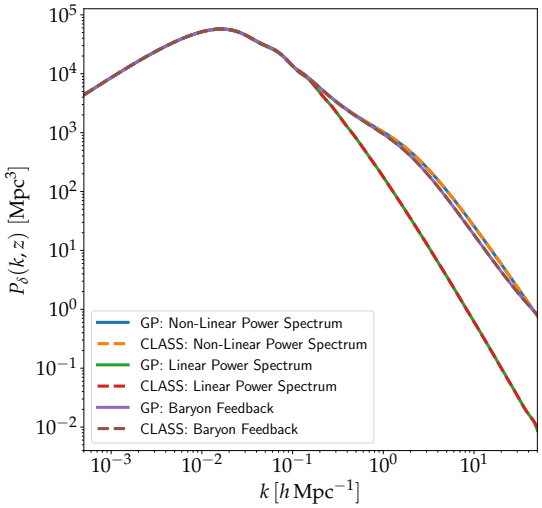


Figure 8.7 – The linear power spectrum at a fixed redshift, z_0, the 3D non-linear matter power spectrum, $P_\delta(k, z)$ and the 3D non-linear matter power spectrum with baryon feedback, $P_\delta^{\rm bary}(k, z)$ can be calculated with our emulating scheme. The solid curves correspond to predictions from the model while the broken curves show the accurate functions as calculated with CLASS.

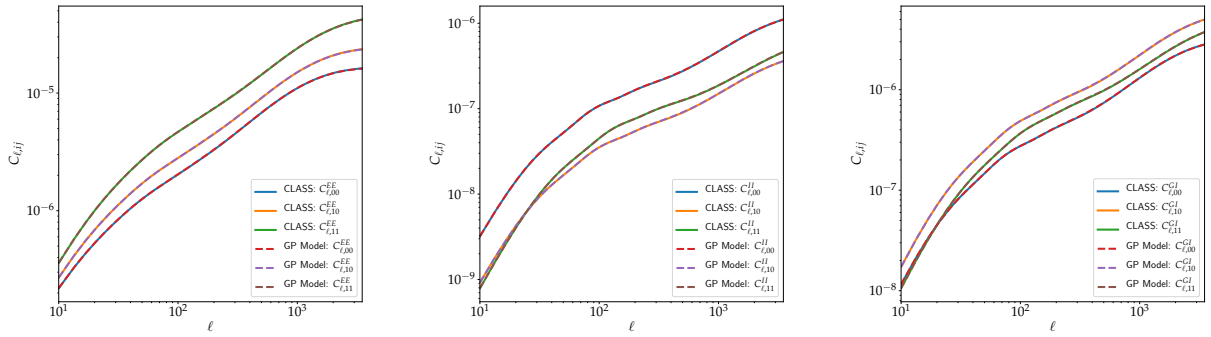
Note that we do not compute the emulator uncertainty for various reasons. As argued by Bastos & O'Hagan (2009), simulators such as CLASS are deterministic input-output models, that is, running the simulator again at the same input values will give the same outputs and the error returned by the GP is unreliable (Mootoovaloo et al., 2020). Moreover, the emulator uncertainty changes as a function of the number of training points and so do the accuracy and

precision of the predicted mean function from the emulator. However, in a small data regime, for example, band powers for current weak lensing surveys, the emulator uncertainty might have significant undesirable effects on the inference of the cosmological parameters. On a more technical note, storing and calculating the emulator uncertainty is a demanding process, both with $\mathcal{O}(N^2)$ computational cost respectively, where N is the number of training points.

Once all these processes (generating the training points and training the emulators) are completed, the emulator is very fast when we compute the 3D matter power spectrum. It takes around 0.1 seconds to do so compared to the average value of 30 seconds by CLASS. Note that the gradient calculation with the emulator is even more efficient compared to finite difference methods, where CLASS would need to be called 10 times for a 5D problem (assuming a central difference method). For an in-depth documentation on the code structure and technical details, we refer the reader to the beginning of this chapter, where we provide the links to the code and documentation.

8.8 Results

In this section, we highlight the main results, starting from the calculation of the 3D matter power spectrum to the calculation of the different weak lensing power spectra.



(a) The EE power spectrum, $C^{\mathrm{EE}}_{\ell,ij}$

(b) The II power spectrum, $C^{\mathrm{II}}_{\ell,ij}$

(c) The GI power spectrum, $C^{\mathrm{GI}}_{\ell,ij}$

Figure 8.8 – The left, centre and right panels show the different weak lensing power spectra as calculated by the emulator (broken curves) and the accurate model, CLASS, shown by the solid curves. The different power spectra within each panel correspond to the auto- and cross- power spectra, due to the 2 tomographic redshift distribution in Figure 8.5, hence leading to 00, 10, and 11 power spectra. These power spectra are then added, via the intrinsic alignment parameter, A_{IA} to construct a final model, $C^{\mathrm{tot}}_{\ell,ij}$ in a weak lensing analysis. See Equation 8.6.3.

In Figure 8.4, we show the gradient along at a fixed cosmological parameter (test point) and fixed redshift, $z = 0$. The red curve corresponds to the gradients as calculated by CLASS using central difference method and the blue curves show the gradients output from the emulator. In

particular, this gradient is strictly a 3D quantity, as a function of the wavenumber k, redshift, z and the cosmological parameters $\boldsymbol{\theta}$. In other words, the gradient calculation from the emulator will be a tensor of size (N_k, N_z, N_p), where N_k is the number of wavenumbers for $k \in [5 \times 10^{-4}, 50]$, N_z is the number of redshifts for $z \in [0.0, 4.66]$ and N_p is the number of parameters considered. In this case, $N_p = 5$ and the default values for a finer grid in k and z are $N_k = 1000$ and $N_z = 100$.

In Figure 8.6, we show the growth factor, $D(z)$ calculated using CLASS (in orange) and the emulator (in blue), while in Figure 8.7, we show three important quantities. First, since we are emulating the 3 different components of the non-linear matter power spectrum, we are able to compute the linear matter power spectrum at a test point, at the reference redshift, $z_0 = 0$. Note that the one calculated by CLASS and the one by the emulator agree quite well. Similarly, we can also calculate the 3D non-linear matter power spectrum and in Figure 8.7, in orange and blue, we have the power spectrum at a fixed redshift, excluding baryon feedback, calculated using CLASS and the emulator respectively. The same is repeated for the curves in purple and brown, but in this case including baryon feedback. As discussed in §8.3, we can also see the effect of baryon feedback which alters the power spectrum at large k.

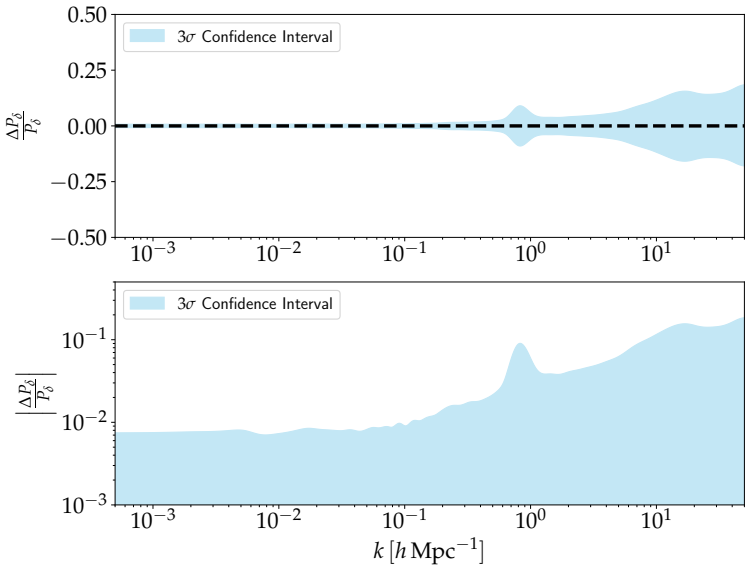


Figure 8.9 – To investigate the performance of the emulator, we draw an independent set of cosmological parameters, randomly from the prior and we calculate the fractional error between the predicted ones with the GP model and CLASS. The mean of $\Delta P_\delta / P_\delta$ is shown by the broken horizontal line and the 3σ confidence interval, derived from the standard deviations of all experiments, is shown in pale blue. For an accurate emulator, it is expected that the mean is centred on 0 and this demonstrates the robustness of this method. Note that in this procedure, one can also specify the number of desired power spectra for $z \in [0.0, 4.66]$. For example, for p cosmological parameters and n redshifts, we have np power spectra outputs. In the bottom panel, we are showing the error in logarithmic scale.

Various techniques have been proposed by Bastos & O'Hagan (2009) to assess the performance of an emulator. These diagnostics are generally based on the comparisons between the

emulator and simulator runs for new test points in input parameter space. These test points should cover the input parameter space over which the training points were previously generated. In this application, we randomly choose 100 independent test points from the prior range and evaluate the simulator and the emulator at these points. Since, we are emulating the 3D matter power spectrum, we can also generate it on a finer grid, unlike the previous setup where we used 40 wavenumbers and 20 redshifts. Hence, we generate all the power spectra for 1000 wavenumbers, equally spaced in logarithmic scale, $k \in [5 \times 10^{-4}, 50]$ and 100 redshifts, $z \in [0.0, 4.66]$, equally spaced in linear scale. For the 100 test points, this gives us a set of 10^4 power spectra.

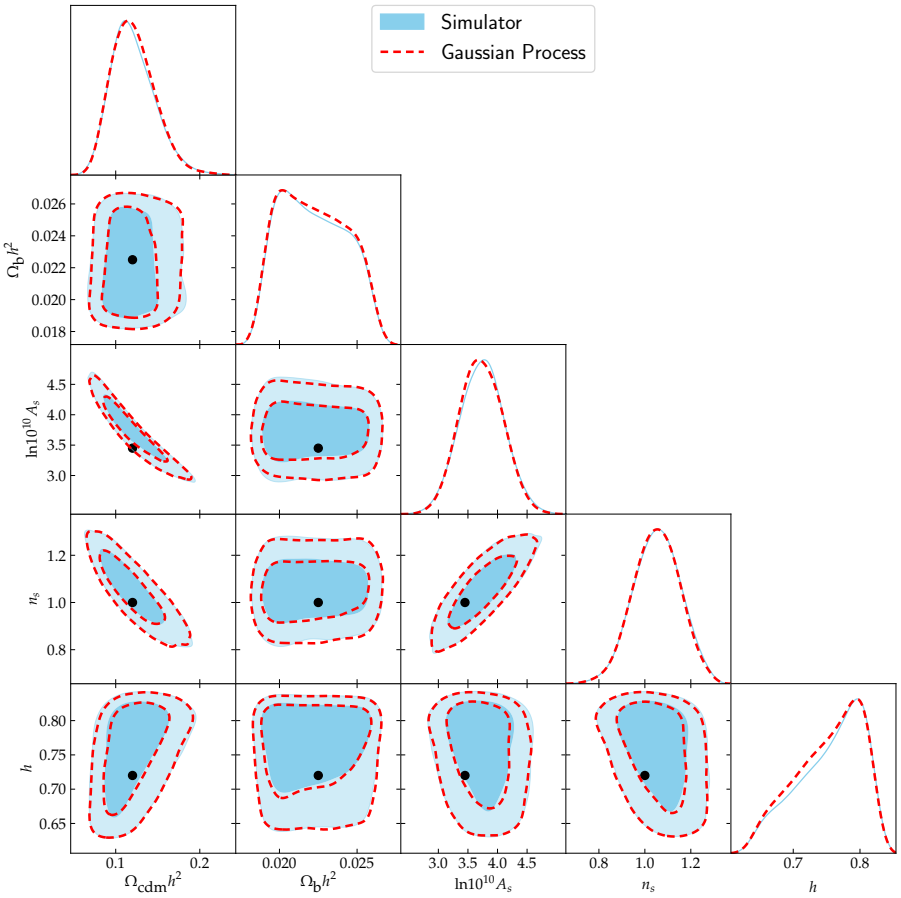


Figure 8.10 – In this figure, the blue colour refers to the posterior distribution of the parameters as inferred using the solver, CLASS and the broken red contours refer to the posterior distribution when using the emulator developed in this work. The black dots correspond to the fiducial point in parameter space where the data has been generated.

We define the fractional uncertainty as

$$\frac{\Delta P_\delta}{P_\delta} = \frac{P_\delta^{\rm emu} - P_\delta}{P_\delta} \tag{8.8.1}$$

and given the set of power spectra we have generated, we compute the mean and variance of $\Delta P_\delta / P_\delta$. For a robust emulator, the mean should be centred on zero and indeed, as seen, from

Figure 8.9, the mean is centred on 0. The variance, depicted by the 3σ confidence interval in pale blue, is also quite small.

In Figure 8.8, we show the different types of weak lensing power spectra calculated using CLASS and the emulator. The left, middle and right panel show the auto- and cross- EE, II and II power spectra due to the two tomographic bins, shown in Figure 8.5. In the three panels, the blue, orange and green curves correspond to the auto- and cross- power spectra, $C_{\ell,00}$, $C_{\ell,10}$ and $C_{\ell,11}$ as computed by CLASS. Similarly, the red, purple and brown broken curves are the power spectra generated by the emulator. The power spectra are in agreement when comparing CLASS and the emulator. Note that, in a typical weak lensing analysis, the three different types of power spectra (EE, GI and II) are combined together via the intrinsic alignment parameter, $A_{\rm IA}$ (see Equation 8.6.3).

We also test the emulator on simulated weak-lensing bandpowers. We assume measurements over $10 \leq \ell \leq 1500$ and 5 tomographic slices with Gaussian $n(z)$ as in Equation 8.6.9, centred on redshifts [0.5, 1.0, 1.5, 2.0, 2.5] and each having a standard deviation of 0.075. Ten bandpowers, equally spaced in logarithmic scale, are used and this gives us a set of 150 data points. Moreover, we simulate and then assume in the likelihood independent Gaussian errors with, for simplicity, $\sigma = 0.5\hat{\mathcal{B}}_\ell$, where $\hat{\mathcal{B}}_\ell$ is the bandpower evaluated at the fiducial set of cosmological parameters. For this particular case, we have set $A_{\rm IA} = 0$ but one can trivially include this factor and marginalise over it in the sampling process. The fiducial point $\boldsymbol{\theta}_{\rm fid} = [0.12, 0.0225, 3.45, 1.0, 0.72]$ is used to generate the data and is shown by the black dots in Figure 8.10. We use a Gaussian likelihood and uniform priors on all cosmological parameters, similar to the range of the inputs of the emulator. Figure 8.10 shows the results obtained when sampling the cosmological parameters on this toy data set. The red contours correspond to the result using the emulator while the pale blue colour refers to the posterior distributions using CLASS. We run three separate MCMC chains, each with 150 000 MCMC samples, two with the emulator and one with CLASS. On each of the three resulting pairs of runs, we compute the Gelman-Rubin convergence parameter, $\hat{R}$ (Gelman & Rubin, 1992). The worst $\hat{R}$ value is 1.002, consistent with all three chains being drawn from the same distribution, and corroborating the agreement shown in Figure 8.10. The emulator developed in this work is thus able to robustly recover the posterior distributions of all the cosmological parameters, compared to the accurate solver, CLASS.

8.9 Conclusions

In this work, we have proposed an emulator for the 3D matter power spectrum as calculated by CLASS across a wide range of cosmological parameters (see Table 8.3.1). This detailed methodology presented in this work entails a multifaceted view of the 3D power spectrum, which is an essential quantity in a weak lensing analysis. In particular, we have successfully demonstrated that as part of this routine, we can compute the linear matter power spectrum at a reference redshift z_0, the non-linear 3D matter power spectrum with and without the baryon feedback model described in §8.3, gradients of the 3D matter power spectrum with respect to the input parameters and the different auto- and cross- weak lensing power spectra (EE, GI and II) derived from $P_\delta^{\text{bary}}(k,z)$ and the given tomographic redshift distributions, $n_i(z)$. Note that the gradients of the weak lensing power spectra are also straightforward to calculate using the distributive property of gradients (see Equation 8.6.6 for a general form for the different weak lensing power spectra). Note that only $P_\delta^{\text{bary}}(k,z)$ is a function of the cosmological parameters.

The default emulator is built using 1000 training points only and because the mean of the surrogate model is just a linear predictor, the mean function is very quick to compute. In the same spirit, the first and second derivatives involve only element-wise matrix multiplication, and are therefore quick to compute. In the test cases, a full 3D matter power spectrum calculation takes 0.1 seconds compared to an average value of 30 seconds when CLASS is used. While the goal remains to have an emulating method which is faster than the computer model, it is also worth pointing out that it also quite accurate, following the diagnostics we have performed in this work, see Figure 8.9 as an example. The emulator can be made more accurate and precise as we add more and more training points, but this comes at an expense of $\mathcal{O}(N^3)$ cost at each optimisation step during the training phase. Fortunately, in this work, 1000 training points suffice to yield promising and robust power spectra.

Building an emulator for the 3D power spectrum is deemed to be a challenging task (Kobayashi et al., 2020), the main difficulties arising due to the fact that GP models cannot easily handle large datasets ($\sim 10^4$ training points) and it is not trivial to work with vector-valued functions, for example, $P_\delta(k,z;\boldsymbol{\theta})$ as in this work. Also, techniques such as multi-outputs GP result in large matrices, hence a major computational challenge. Fortunately, the method presented in this work, along with the projection method explained in §7.1.2, provides a simple and straightforward path towards building emulators.

Moreover, current weak lensing data do not constrain the cosmological parameters to a

high precision, hence motivating us to distribute 1000 training points across a large parameter space, according to the current prior distributions (hypercube) used in the literature. In future weak lensing surveys, with improved precision on the parameters, one can choose to use, for example, a multi-dimensional Gaussian prior (hypersphere) which will certainly have a much smaller volume compared to the hypercube used in this work. If we stick with 1000 training points, this will lead to very precise power spectra, or we can also opt to distribute fewer than 1000 training point across the parameter space. Fewer training points also imply that training the emulator will be faster.

The different aspects of the emulation scheme proposed in this work can easily pave their way into different cosmological data analysis problems. A nice example is an analysis combining the MOPED data compression algorithm (Heavens et al., 2000), the emulated 3D matter power spectrum and the $n(z)$ uncertainty in a weak lensing analysis. Moreover, if we want to use a more sophisticated sampler such as Hamiltonian Monte Carlo (HMC), one can leverage the gradients from the emulator to derive an expression for the gradient of the negative log-likelihood (the potential energy function in an HMC scheme) with respect to the input cosmological parameters, under the assumption that such an analytic derivation is possible. Furthermore, the second derivatives can be used in a Fisher Matrix analysis, or the first and second derivatives can be be used together in an approximate inference scheme based on Taylor expansion techniques, see for example, the recent work by Leclercq et al. (2019). In addition, similar concepts behind this work can be extended to build emulators for $P_\delta(k, z)$ from N-body simulations.

8.10 Summary

In this work, we have shown that we can emulate the 3D matter power spectrum, $P_\delta(k, z)$ by first decomposing it into three different parts, that is, $P_\delta(k, z) = D(z)[1 + q(k, z)]P_{\rm lin}(k, z_0)$. We also use the semi-parametric Gaussian Process approach developed and tested in Chapter 7 to emulate the three different components. In doing so, we can successfully calculate the 3D non-linear matter power spectrum, as well as the linear matter power spectrum at a fixed redshift, z_0. The gradient of $P_\delta(k, z)$ can also be calculated and one can take a step further and calculate the weak lensing and intrinsic alignment power spectra after specifying the redshift distributions, $n_i(z)$.

Chapter 9

Mathematical Methods for Weak Lensing Data Analysis

> We will always have STEM with us. Some things will drop out of the public eye and go away, but there will always be science, engineering, and technology. And there will always, always be mathematics.
>
> *Katherine Johnson*

In all weak lensing cosmological data analyis, the summary statistics and the tomographic redshift distribution play an important role to constrain cosmological and nuisance parameters. In this short chapter, a precursor for Chapter 10, in §9.1, we discuss briefly the different summary statistics, namely band powers, binned correlation functions and the Complete Orthogonal Sets of E/B-Integrals COSEBIs which are currently used in weak lensing data analysis. In addition, in §9.2, we provide an alternative method, which we will often be referred to as the 'double sum approach' to computing weak lensing and intrinsic alignment power spectra, when the redshift distributions are supplied as narrow-bin histograms. Finally, in §9.3, we propose marginalising over the $n(z)$ samples using numerical Monte Carlo methods.

9.1 Weak Lensing Statistics in Current Analyses

In this section, we will briefly cover some of these statistics, as employed for the KV-450 and KiDS-1000 analyses. The statistics band powers, 2PCFs and COSEBIs all involve some linear transformation of the weak lensing (shear) power spectrum, that is,

$$S^x = \int_0^\infty \ell \, W_\ell^x \, C_\ell^{\mathrm{EE}} \, \mathrm{d}\ell \tag{9.1.1}$$

where S^x is one of the 3 statistics and W_ℓ^x is a weight function, which itself is function of the Fourier scale, ℓ. $C_\ell^{\rm EE}$ is the E-mode angular power spectrum, usually a biased tracer of the gravitational lensing, hence often substituted with

$$C_\ell^{\rm tot} = C_\ell^{\rm EE} + C_\ell^{\rm GI} + C_\ell^{\rm II} \tag{9.1.2}$$

which includes the effect of intrinsic alignment. In most cosmic shear analysis, significant B-modes, another systematics, are not expected. In some cases, these are measured and included in the analysis for robust analysis. The 2PCFs was the main choice of the KV-450 analysis (see §10.1) and is strictly a linear combination of the E- and B-mode power spectra, that is,

$$\xi_\pm(\theta) = \int_0^\infty \frac{\ell}{2\pi} J_{0/4}(\ell\theta)[C_\ell^{\rm EE} \pm C_\ell^{\rm BB}] \tag{9.1.3}$$

and in this case, the Bessel functions of the first kind, $J_{0/4}$ are the weight functions. In the KiDS-1000 analysis, the 2PCFs are binned in the angular separation, θ and referred to as the $\theta-$bins. The full angular range employed was $\theta \in [0'.5, 300']$. On the other hand, the expressions for the COSEBIs statistics are

$$\begin{aligned} E_n &= \frac{1}{2\pi}\int_0^\infty \ell C_\ell^{\rm EE} W_\ell^n \, {\rm d}\ell \\ B_n &= \frac{1}{2\pi}\int_0^\infty \ell C_\ell^{\rm BB} W_\ell^n \, {\rm d}\ell \end{aligned} \tag{9.1.4}$$

and the expressions for W_ℓ^n are

$$\begin{aligned} W_\ell^n &= \int_{\theta_{\rm min}}^{\theta_{\rm max}} \theta T_{+n}(\theta) J_0(\ell\theta) \\ &= \int_{\theta_{\rm min}}^{\theta_{\rm max}} \theta T_{-n}(\theta) J_4(\ell\theta). \end{aligned} \tag{9.1.5}$$

Therefore, for COSEBIs, the weight functions, W_ℓ^n are Hankel transforms of the filter functions, $T_\pm(\theta)$. The latter is bounded for certain angular range, that is, $\theta \in [\theta_{\rm min}, \theta_{\rm max}]$. Moreover, band powers are binned angular power spectra and are given by

$$\begin{aligned} \mathcal{B}_\ell^{\rm EE} &= \frac{1}{2\mathcal{N}_\ell}\int_0^\infty \ell \left[W_\ell^{\rm EE} C_\ell^{\rm EE} + W_\ell^{\rm EB} C_\ell^{\rm BB}\right] {\rm d}\ell, \\ \mathcal{B}_\ell^{\rm BB} &= \frac{1}{2\mathcal{N}_\ell}\int_0^\infty \ell \left[W_\ell^{\rm BE} C_\ell^{\rm EE} + W_\ell^{\rm BB} C_\ell^{\rm BB}\right] {\rm d}\ell \end{aligned} \tag{9.1.6}$$

where $\mathcal{N}_\ell = \ln(\ell_{{\rm up},\ell}) - \ln(\ell_{{\rm low},\ell})$ and is simply a normalisation term which traces $\ell^2 C_\ell$. The

weight functions, W_ℓ are

$$\begin{aligned} W_\ell^{\mathrm{EE}} &= W_\ell^{\mathrm{BB}} \\ &= \int_0^\infty \theta T(\theta) \left[J_0(\ell\theta) g_+^\ell(\theta) + J_4(\ell\theta) g_-^\ell(\theta) \right] \end{aligned} \tag{9.1.7}$$

and

$$\begin{aligned} W_\ell^{\mathrm{EB}} &= W_\ell^{\mathrm{BE}} \\ &= \int_0^\infty \theta T(\theta) \left[J_0(\ell\theta) g_+^\ell(\theta) - J_4(\ell\theta) g_-^\ell(\theta) \right]. \end{aligned} \tag{9.1.8}$$

In this case, $T(\theta)$ is a selection and is unrelated to the term $T_{\pm n}(\theta)$ for the COSEBIs statistics and $g_\pm^\ell(\theta)$ are filter functions. We refer the reader to Asgari et al. (2021) and Joachimi et al. (2021b) for a more comprehensive explanation on the theoretical modelling of the different statistics, which have been explicitly tested on the KiDS-1000 data. Asgari et al. (2021) argued that out of the three statistics, COSEBIs and band powers provide a cleaner approach towards the separation of the cosmology-driven E- and systematics-driven B-modes.

9.2 Weak Lensing Power Spectra as Double Sums

In this section, we will re-write the weak lensing and intrinsic alignment power spectra as a double sum. The $n(z)$ distribution can be regarded as a series of tophat functions which constitute a histogram. This approach sidesteps multiple steps, for example, integrations, to approximate the power spectra numerically. Existing approach in the KiDS-450, KiDS+VIKING-450 (KV-450) and KiDS-1000 likelihoods use trapezoidal rule to perform numerical integrations. In the following, the indices i and j correspond to the auto and cross tomographic bins while the indices α and β correspond to the auto and cross elements of the finer bins within one redshift distribution.

EE Power Spectrum

The EE power spectrum is given by

$$C_{\ell,ij}^{\mathrm{EE}} = \int_0^{\chi_{\mathrm{H}}} \mathrm{d}\chi \, \frac{w_i(\chi)\, w_j(\chi)}{\chi^2} \, P_\delta(k;\chi) \tag{9.2.1}$$

where the weight function $w(\chi)$ is

$$w_i(\chi) = A\chi(1+z)\int_{\chi}^{\chi_{\rm H}} \mathrm{d}\chi'\, n_i(\chi)\left(\frac{\chi'-\chi}{\chi'}\right) \tag{9.2.2}$$

and $A = {}^{3H_0^2\Omega_m}/_{2c^2}$. The idea behind this work is to assume that the redshift bins within one tomographic bin are very small such that we can approximate them as a tophat (boxcar) function. If the bins are very small, then we can assume that the tomographic bin is a sum of δ functions, which has the fundamental property that

$$\int_{-\infty}^{\infty} f(x)\delta(x-a)\,\mathrm{d}x = f(a) \tag{9.2.3}$$

and in fact,

$$\int_{a-\epsilon}^{a+\epsilon} f(x)\delta(x-a)\,\mathrm{d}x = f(a) \tag{9.2.4}$$

for $\epsilon > 0$. We will first define a tophat (boxcar) function as

$$T_\alpha = \begin{cases} {}^{1}/_{\Delta z_\alpha} & z_\alpha - {}^{\Delta z_\alpha}/_2 < z < z_\alpha + {}^{\Delta z_\alpha}/_2 \\ 0 & \text{otherwise} \end{cases} \tag{9.2.5}$$

and the tomographic redshift $n_i(z)$ can be approximated as a sum over fine bins in redshift

$$n_i(z) \approx \sum_\alpha h_{i\alpha} T_\alpha \tag{9.2.6}$$

The integration term in the weight function can be re-written in terms of the newly defined $n(z)$ distribution, that is,

$$I_i = \sum_\alpha \Delta z_\alpha h_{i\alpha}\left(\frac{\chi_\alpha - \chi}{\chi_\alpha}\right) \tag{9.2.7}$$

and with some linear algebra, the EE power spectrum can be written as a double sum as follows

$$C_{\ell,ij}^{\rm EE} \approx \sum_\alpha\sum_\beta h_{i\alpha}h_{j\beta}\Delta z_\alpha \Delta z_\beta \int_0^{\min(\chi_\alpha,\chi_\beta)} \mathrm{d}\chi \left(\frac{\chi_\alpha-\chi}{\chi_\alpha}\right)\left(\frac{\chi_\beta-\chi}{\chi_\beta}\right) A^2(1+z)^2 P_\delta(k;\chi). \tag{9.2.8}$$

The correlations between the two fields are caused only by matter which is in the foreground of **both** narrow tomographic bins. The matter in between α and β will affect the lensing of the further bin but this additional fluctuation will not be correlated with the convergence in the near bin, so contributes nothing to the cross-correlation. At some **tiny** level, fluctuations

just beyond the nearer bin will be correlated with fluctuations just closer than the nearer bin, but this is tiny, and is heavily suppressed since the lensing kernel for the nearer bin is almost zero there. In the Limber approximation, it exactly vanishes. Equation 9.2.8 can further be simplified. If we define

$$F_\ell^{\mathrm{EE}}(k;\chi) = A^2(1+z)^2 P_\delta(k;\chi), \tag{9.2.9}$$

we have

$$C_{\ell,ij}^{\mathrm{EE}} \approx \sum_\alpha \sum_\beta h_{i\alpha} h_{i\beta} \Delta z_\alpha \Delta z_\beta \left[Q_{\ell,\alpha\beta}^{\mathrm{EE},0} - \left(\frac{\chi_\alpha + \chi_\beta}{\chi_\alpha \chi_\beta} \right) Q_{\ell,\alpha\beta}^{\mathrm{EE},1} + \frac{1}{\chi_\alpha \chi_\beta} Q_{\ell,\alpha\beta}^{\mathrm{EE},2} \right] \tag{9.2.10}$$

where

$$Q_{\ell,\alpha\beta}^{\mathrm{EE},r} = \int_0^{\min(\chi_\alpha,\chi_\beta)} \mathrm{d}\chi \, \chi^r F_\ell^{\mathrm{EE}}(k;\chi), \tag{9.2.11}$$

Once the three cosmology-dependent terms, $Q_{\ell,\alpha\beta}^{\mathrm{EE},r}$ for $r \in [0,1,2]$ are computed, the auto and cross EE power spectra can be calculated in a very fast way. Moreover, from an algorithmic perspective, we can easily input a specific set of redshift distribution for a weak lensing analysis.

Intrinsic Alignment Power Spectra

Using a similar approach as elaborated in the previous section for the EE power spectrum, the II and GI intrinsic alignment power spectra can also be expressed as a weighted sum in terms of the heights on the $n(z)$ redshift distributions. The II power spectrum (Hirata & Seljak, 2004) is modelled as

$$C_{\ell,ij}^{\mathrm{II}} = \int_0^{\chi_{\mathrm{H}}} \mathrm{d}\chi \, \frac{n_i(\chi)\, n_j(\chi)}{\chi^2} \, P_\delta(k;\chi) \, F^2(\chi) \tag{9.2.12}$$

where $F(\chi) = C_1 \rho_{\mathrm{crit}} \Omega_m / D_+(\chi)$. C_1 is a constant given by $5 \times 10^{-14}\ h^{-2} \mathrm{M}_\odot^{-1} \mathrm{Mpc}^3$, $D_+(\chi)$ is the linear growth factor normalised to unity today and ρ_{crit} is the critical density of the Universe today. We will first define

$$F_\ell^{\mathrm{II}}(k;\chi) = \frac{P_\delta(k;\chi)\, F^2(\chi)}{\chi^2}. \tag{9.2.13}$$

The II power spectrum can thus be written in terms of the heights of the $n_i(z)$ and $n_j(z)$ redshift

distribution as

$$C^{\rm II}_{\ell,ij} \approx \sum_{\alpha} h_{i\alpha} h_{j\alpha} \Delta z_{\alpha} F^{\rm II}_{\ell}(\chi_{\alpha}) \tag{9.2.14}$$

In this derivation, we have the product of two tophat functions, $T_\alpha T_\beta$ and hence the contribution to the total II power spectrum arises from $\alpha = \beta$ only and we have a single sum only compared to the EE power spectrum. Similarly, we can express the GI power spectrum as a double sum. The GI power spectrum (Hirata & Seljak, 2004) is modelled as

$$C^{\rm GI}_{\ell,ij} = \int_0^{\chi_{\rm H}} {\rm d}\chi \, \frac{w_i(\chi) n_j(\chi) + w_j(\chi) n_i(\chi)}{\chi^2} \, P_\delta(k;\chi) \, F(k;\chi) \tag{9.2.15}$$

and we define

$$F^{\rm GI}_{\ell}(k;\chi) = \frac{A(1+z)}{\chi} P_\delta(k;\chi) \, F(\chi). \tag{9.2.16}$$

The final approximate GI power spectrum is given by

$$C^{\rm GI}_{\ell,ij} \approx \sum_{\alpha} \sum_{\beta<\alpha} \left(h_{i\alpha} h_{j\beta} + h_{i\beta} h_{j\alpha} \right) \Delta z_\alpha \Delta z_\beta \left(1 - \frac{\chi_\beta}{\chi_\alpha} \right) F^{\rm GI}_{\ell}(k;\chi_\beta) \tag{9.2.17}$$

We now have a method for computing the three types of power spectra, often used in a weak lensing analysis. The advantage of this formalism is that we no longer have to compute integrations numerically using trapezoidal rule for each auto and cross power spectrum. Only $Q^{{\rm EE},r}_{\ell,\alpha\beta}$ term involves an integration. Instead, once all cosmology-dependent terms are precomputed, the final power spectra are given by a (double) sum involving the heights of any pairs of $n(z)$ distribution.

9.3 Marginalisation of the $n(z)$

An important aspect is the marginalisation of the $n(z)$ uncertainty since the uncertainty in the redshift distributions needs to be propagated to the final results. In the analysis by Hildebrandt et al. (2020), at each step in the MCMC, a random sample of the $n(z)$ is drawn and the log-likelihood is calculated. Denoting ρ as the additional set of nuisance parameters for the $n(z)$ distributions, we propose marginalising over ρ as follows

$$
\begin{aligned}
p(\boldsymbol{\theta}, \boldsymbol{\beta}, \rho|\boldsymbol{d}) &\propto p(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{\beta}, \rho)\, \pi(\boldsymbol{\theta}, \boldsymbol{\beta}, \rho) \\
p(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{d}) &= \int p(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{\beta}, \rho)\pi(\rho)\, \mathrm{d}\rho\ \pi(\boldsymbol{\theta}, \boldsymbol{\beta}) \\
&\approx \frac{1}{N}\sum_{i=1}^{N} p(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{\beta}, \rho_i)\ \pi(\boldsymbol{\theta}, \boldsymbol{\beta})
\end{aligned}
\tag{9.3.1}
$$

and we are assuming independent (separable) priors. The last step is a Monte Carlo estimate of the likelihood marginalised over ρ. In other words, we may draw more than one sample of $n(z)$ distributions at a fixed set of cosmological parameters, $\boldsymbol{\theta}$ and the other nuisance parameters, $\boldsymbol{\beta}$ and we estimate the log-likelihood. This is valid since the sampled redshift distributions are cosmology-independent.

9.4 Summary

In this short chapter, we have covered three topics, namely the different summary statistics currently being employed in different weak lensing surveys, the weak lensing and intrinsic alignment power spectra expressed as a (double) sum and finally, we discussed how we can marginalise over the $n(z)$ redshift distributions by drawing more than one sample of $n(z)$ for a fixed set of cosmological and nuisance parameters. This results in a Monte-Carlo estimate of the likelihood. In the next chapter, we will use these three mathematical methods to perform an analysis on the KV-450 data and the goal is also to extend the analysis and apply the tools to the latest KiDS-1000 data.

Chapter 10

WEAK LENSING DATA ANALYSIS OF DIFFERENT SURVEYS

> No one undertakes research in physics with the intention of winning a prize. It is the joy of discovering something no one knew before.
>
> *Stephen Hawking*

The determination of the redshift distributions, $n(z)$ will play a crucial role in future weak lensing data analyses since the cosmic signal is sensitive to the choice of $n(z)$. Indeed, the forward model, that is, the calculation of the different weak lensing and intrinsic alignment power spectra, relies on the $n(z)$ distributions. Hence, any interpretation of the result from a weak lensing data analysis is dependent on accurate $n(z)$ distributions. For example, Hildebrandt et al. (2017) argued that in their analysis, a 1σ uncertainty mis-specification of one of their tomographic redshift distributions can deteriorate cosmological parameters by $\sim 25\%$. This clearly demonstrates that a careful and meticulous analysis is required to deal with the systematics in weak lensing cosmology.

Existing methods for estimating the $n(z)$ distributions include the DIR method (Lima et al., 2008) which is a weighted direct calibration approach, the CC method, which is an angular cross-correlation-based calibration developed by Newman (2008) and the BOR method (Bordoloi et al., 2010), which essentially involves recalibrating the Bayesian probability estimate of of the redshift, BPZ (Benítez, 2000) in probability space. We refer the reader to Hildebrandt et al. (2017) who performed a detailed analysis by investigating the different redshift distributions, namely DIR, CC and BOR on the KiDS-450 dataset.

One option to estimate the photometric redshift distributions is to use the DIR method which makes use of available spectroscopic redshift distribution for a sample of object of in-

terest. The drawback is that in practice, the spectroscopic catalogues are never complete and hence are representative of only a subset of the full shear catalogue. In addition to this, a further complication is that deep spectroscopic redshift surveys cover a smaller area compared to photometric redshifts, hence sample variance being the main issue. Lima et al. (2008) proposed a $k-$nearest-neighbour (kNN) approach to estimate the volume density of objects in a multi-dimensional magnitude space and this is performed for both catalogues, that is, spectroscopic and photometric. This estimate is then used to up-weight and down-weight the spectroscopic redshift objects in magnitude space where they are under-represented and overrepresented respectively.

On the other hand, Newman (2008) used cross-correlation (CC) functions between photometric and spectroscopic objects for finding the photometric redshift distributions. In particular, the main advantage of this method is that as long as the full redshift range between spectroscopic and photometric is the same, the method is insensitive to spectroscopic selection function in terms of galaxy type and magnitude. However, in order to develop the method, a good knowledge of the angular selection function, which can be estimated from masks, is required. Hildebrandt et al. (2017) implemented variants of this algorithms in the analysis of the KiDS-450 data, where the cosmological data analysis was performed using correlation functions as summary statistics.

Another approach is the BOR method proposed by Bordoloi et al. (2010). This uses the posterior probability distribution of the redshift of an object determined using BPZ. A representative set of spectroscopic sample of objects is used to investigate the properties of the corresponding photometric redshift likelihoods. A main limitation is that the spectroscopic training sample should be completely representative of the photometric sample, but this is not the case in practice. For example, in the KiDS-450 analysis, a recalibration method along with a re-weighting procedure in magnitude space had to be employed to estimate the $n(z)$ redshift distributions.

Our contribution in this chapter is three-fold. First, we use the method established by Leistedt et al. (2016) to develop a Bayesian Hierarchical method for inferring the $n(z)$ redshift distributions*, with the aim of performing cosmological data analyses of recent releases of weak lensing data sets, in particular, the KV-450 and the KiDS-1000 data. Second, the fact that the emulator developed in Chapter 8 produce robust and reliable results, this is then used to substitute CLASS to compute the 3D matter power spectrum. Third, we express all the weak lensing

*The samples of $n(z)$ were computed by George Kyriacou.

and intrinsic alignment power spectra as a (double) sum of the product of the some quantities, which are pre-computed and they depend on the cosmological parameters, and the heights of the fine bins of the $n(z)$ distributions. This approach offers a simple and quick way to compute power spectra.

The chapter is organised as follows. In §10.1, we elaborate briefly on the KiDS+VIKING, henceforth KV-450, survey before explaining the data and covariance in §10.2. Next, our approach is different compared to the original analysis and hence, this results in a different set of parameters which we discuss in §10.3. The novel method for finding the $n(z)$ redshift distribution and this is covered briefly in §10.4 and we present the results from our analysis in §10.5. We also touch briefly upon the latest KiDS-1000 survey, data and performed a brief analysis in §10.6 as part of our future work.

10.1 The KV-450 Survey

The KV-450 is a weak lensing analysis which combines data from the Kilo-Degree Survey (KiDS) and the VISTA Kilo-Degree Infrared Galaxy Survey (VIKING), spanning a wavelength of $320\,\mathrm{nm} \lesssim \lambda \lesssim 2350\,\mathrm{nm}$. Interestingly, KIDS overlaps with VIKING and the latter is well-suited for calculating more accurate photometric redshift (photo-z), an important requirement for a cosmic shear analysis (Hildebrandt et al., 2020). The infrared data from VIKING becomes important in the high redshift regime where the performance of the photo-z can be better. Hence, this factor can be exploited and this information can be added to a weak lensing analysis. As a result, the cosmic shear results do not only improve in terms of robustness but also in precision.

In the original KV-450 analysis, the photo-z are calibrated using spectroscopic surveys. Different surveys such as zCOSMOS (Lilly et al., 2009), DEEP2 Redshift Survey (Newman et al., 2013) and among others were used for the KV-450 photo-z calibration. As in any cosmic shear analysis, the galaxies are binned in tomographic redshift bins and for KV-450, 5 tomographic bins were employed for the redshift range $0.1 < z < 1.2$, with a bin width of $\Delta z = 0.2$, except for the last one, which has a bin width of $\Delta z = 0.3$.

Moreover, the DIR method was used to find the redshift distributions. This was first developed by Lima et al. (2008) and is based on the k^{th} nearest neighbour (kNN) algorithm. To account for the variance on the redshift distributions, Hildebrandt et al. (2020) adopted a spatial bootstrapping approach. Moreover, they also define a set of nuisance parameters, δz_i for $i \in [1,5]$ to linearly shift each tomographic bin, that is, $n_i(z) \rightarrow n_i(z_i + \delta z_i)$ when calculating

the shear correlation function. This accounts for shifts in the median redshift, but is clearly limited in applicability.

10.2 Data

The summary statistics employed in the KV-450 are the 2-point shear correlation functions, ξ_+ and ξ_- (see Chapter 9 for a discussion on summary statistics). Nine logarithmically spaced bins are defined in the interval $[0'.5, 300']$ and the first seven data points and the last six bins are used for the ξ_+ and ξ_- respectively. In particular, these are chosen such that baryon feedback will have less than $\sim 20\%$ effect on the overall signal. See Hildebrandt et al. (2020) for a detailed discussion. Since five tomographic redshift distributions are used in the analysis, the data vector of KV-450 consists of $(6+7) \times 15 = 195$ elements and the plots for both ξ_+ and ξ_- are shown in Figure 10.1 in the lower and upper panel respectively.

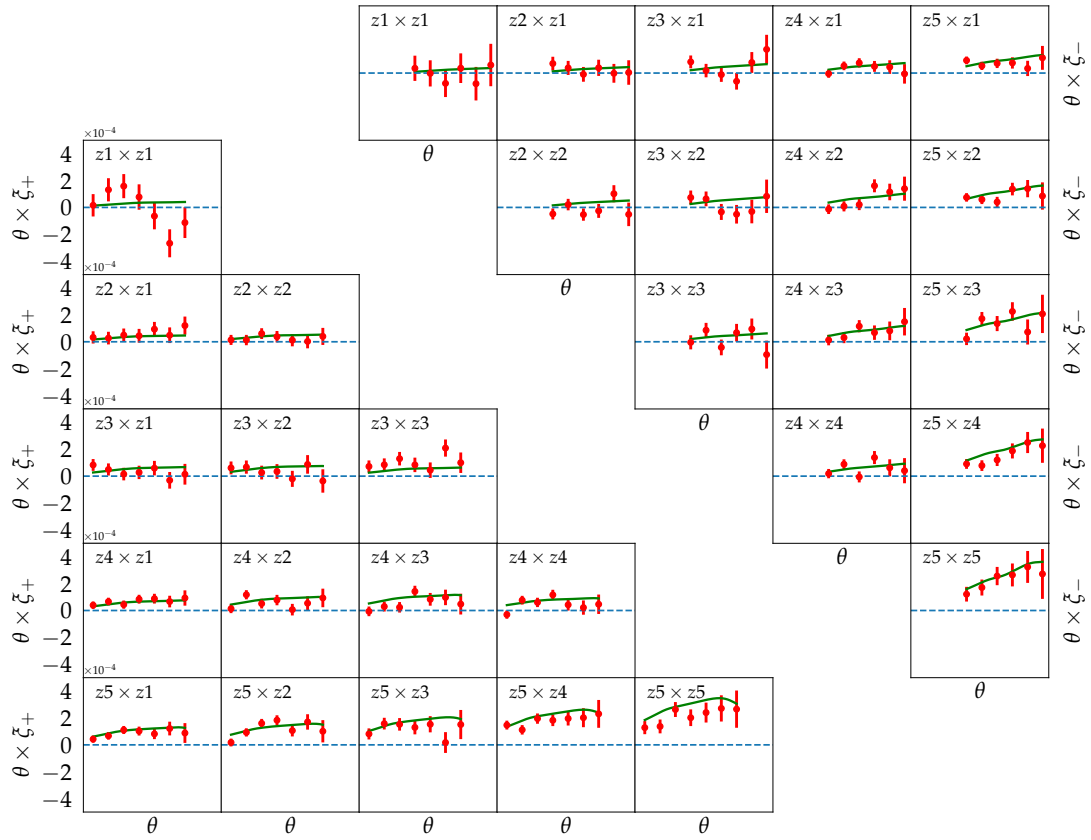


Figure 10.1 – The upper and lower plots show the 2-point shear correlation functions, ξ_- and ξ_+ respectively for the the 5 tomographic redshift bins. This results in a total of 15 auto and cross shear correlation functions. There are 7 data points for the ξ_+ and 6 for the ξ_-, resulting in a data vector of length, $(6+7) \times 15 = 195$. The error bars are given by the square root of the diagonal of the covariance matrix. The green solid curves show the theoretical model computed using the emulator and this includes all the systematics, that is, the intrinsic alignment model, baryon feedback and the observational biases similar to Hildebrandt et al. (2020).

Moreover, the data covariance matrix, of size 195×195 is shown in Figure 10.2. The covariance matrix is determined via an analytical recipe which is discussed in further details by

Hildebrandt et al. (2017) when the first analysis of the KiDS-450 data was performed using correlation functions.

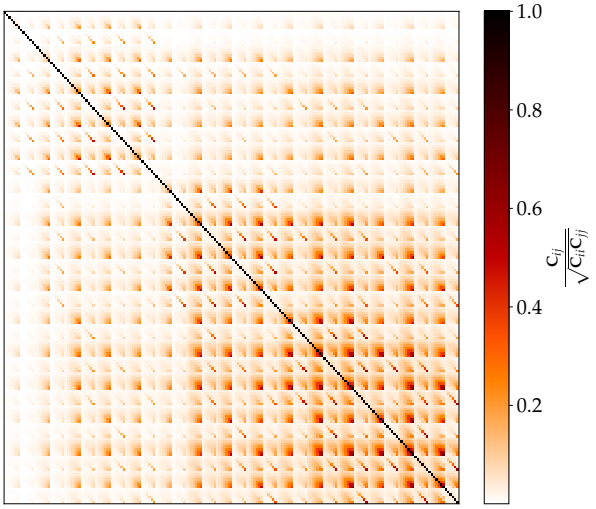


Figure 10.2 – The data covariance matrix (scaled such that the maximum is 1) for the KV-450 analysis. This matrix is of size 195.

10.3 Parameters

In this section, we briefly elaborate on the cosmological and nuisance parameters used in the KV-450 analysis by Hildebrandt et al. (2020). Our approach, as discussed in the next section, will be slightly different. The shear measurements are generally biased and this bias is commonly parametrised in terms of the multiplicative bias term, m and the additive bias c, such that the observed shear is

$$g \simeq (1+m)g_{\text{true}} + c. \tag{10.3.1}$$

Note that g, m, c are all complex numbers. In the work by Hildebrandt et al. (2020), m is very small and is expected to have small effect on the overall cosmological parameter constraints. They introduce a δc parameter for the c-term offset and a Gaussian prior, $c \sim \mathcal{N}(0, 2 \times 10^{-4})$ is assumed. Moreover, an additional parameter, A_c is introduced to account for position-dependent additive bias and a Gaussian prior is assumed, that is, $A_c \sim \mathcal{N}(1.01, 0.13)$. See Hildebrandt et al. (2020) for further details.

Unlike the analysis in Chapter 6 where a set of band powers were used to constrain cosmology, for KV-450, a 2-point shear correlation function analysis is performed. The correlation function is

$$\xi_{\pm,ij}(\theta) = \frac{1}{2\pi}\int_0^\infty \ell\, C^{\kappa\kappa}_{\ell,ij}\, J_{0,4}(\ell\theta)\, \mathrm{d}\theta, \tag{10.3.2}$$

where $J_{0,4}(\ell\theta)$ are Bessel functions of the first kind and $C^{\kappa\kappa}_{\ell,ij}$ is the convergence power spectrum, in the Born approximation, given by:

$$C^{\kappa\kappa}_{\ell,ij} = \int_0^{\chi_\mathrm{H}} \mathrm{d}\chi\, \frac{w_i(\chi)\, w_j(\chi)}{\chi^2}\, P_\delta(k;\chi). \tag{10.3.3}$$

w_i is the lensing efficiency and is elaborated at the beginning of this Chapter (see Equation 9.2.2). In addition, we also have to account for intrinsic alignment effects and hence the total shear correlation function is a linear combination of the original correlation function (Equation 10.3.2) and two additional terms, GI and II,

$$\xi^{\mathrm{tot}}_{\pm,ij} = \xi_{\pm,ij} + \xi^{\mathrm{GI}}_{\pm,ij} + \xi^{\mathrm{II}}_{\pm,ij} \tag{10.3.4}$$

We refer the reader to §9.2 for further details on the expressions for the intrinsic alignments. In practice, all power spectra (convergence, GI and II) are first computed and then the final transform, essentially, the Hankel transform in Equation 10.3.2 is applied to calculate the 2-point shear correlation functions. An additional nuisance parameter, A_{IA} is used to model the amplitude of the intrinsic alignment effects and typically, A_{IA} is marginalised over in the sampling procedure.

Next, an important component is to model other systematics in the analysis. Hildebrandt et al. (2020) used HMcode (Mead et al., 2015) to model baryon feedback. In HMcode, B, the amplitude of the halo mass-concentration and $\eta_0 = 0.98 - 0.12B$, the halo bloating parameter can be varied. Hildebrandt et al. (2020) applied a flat prior on $B \sim \mathcal{U}[2.00, 3.13]$ and marginalised over it when sampling the posterior distribution of the cosmological and nuisance parameters. As a result, in the Hildebrandt et al. (2020) analysis, they have 5 cosmological parameters, which we denote by $\boldsymbol{\theta}$ and 9 nuisance parameters, which we denote by $\boldsymbol{\beta}$

$$\boldsymbol{\theta} = \left[\Omega_{\mathrm{cdm}}h^2,\, \ln(10^{10}A_\mathrm{s}),\, \Omega_\mathrm{b}h^2,\, n_\mathrm{s},\, h\right]$$

and

$$\boldsymbol{\beta} = \left[A_{\mathrm{IA}},\, B,\, \delta c,\, A_c,\, \delta z_1,\, \delta z_2,\, \delta z_3,\, \delta z_4,\, \delta z_5\right].$$

10.4 Bayesian Hierarchical Model for $n(z)$ Distributions

An important ingredient in cosmic shear analysis is the inference of the redshift distribution, $n_i(z)$ from galaxy observations. The most common choice for finding photometric redshifts is via template-fitting method. However, in cosmic shear analysis, we are not strictly interested in the redshift estimate but rather in the distribution of the redshifts, which is crucial for inferring cosmological parameters from two-point statistics such as correlation functions, band powers or COSEBIs. In general, once the photometric redshifts are estimated, these are stacked to generate the $n(z)$. This does not offer a clean approach since there are redshift uncertainties are not propagated in a likelihood analysis, ad the stacked likelihood is not in general a good estimate of $n(z)$ (Malz, 2021).

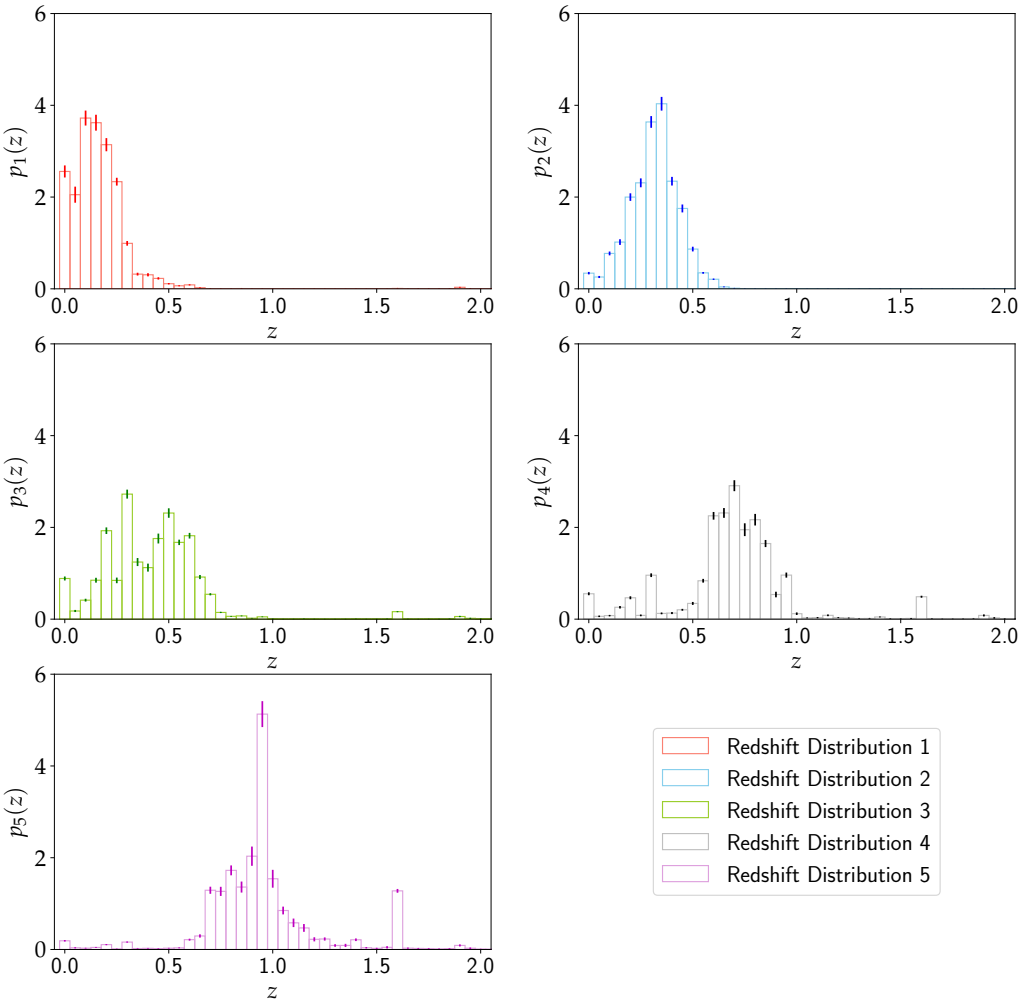


Figure 10.3 – The figure shows a set of redshift distributions generated using the BHM approach. We have 5 $n(z)$ distributions in the redshift range, $z \in [0, 3]$. The distribution lies mostly in the redshift range, $z \in [0, 2]$. These can then be used to calculate the weak lensing and intrinsic alignment power spectra in a weak lensing cosmological analysis.

Common methods for finding redshift estimates include template fitting, Machine Learn-

ing and clustering. On the other hand, Leistedt et al. (2016) proposed a Bayesian Hierarchical Model (BHM) to infer the redshift distributions, as well as the individual redshifts from catalogues. Here, we provide a short summary of the idea behind this work. Each galaxy has some intrinsic properties such as types, t, redshifts, z and apparent magnitudes, m (or fluxes) and these are assumed to be drawn from a joint distribution, $p(t, z, m|\text{survey, galaxy})$. This distribution can be regarded as a combination of a series of a piecewise constant, parametrised by $\{f_{ijk}\}$, that is,

$$p(t,z,m|\{f_{ijk}\}) = \sum_{ijk} \frac{f_{ijk}}{(z_{i,\max} - z_{j,\min})(m_{k,\max} - m_{k,\min})} \times \delta_{t,t_i}\Theta(z - z_{j,\min})\Theta(z_{j,\max} - z)\Theta(m - m_{k,\min})\Theta(m_{k,\max} - m) \tag{10.4.1}$$

where δ is the Kronecker delta function and Θ is the step-Heaviside function. Marginalising over the redshift and magnitude essentially gives the probability, f_{ijk} of finding an object in that voxel ijk, that is,

$$f_{ijk} \equiv \int_{z_{j,\min}}^{z_{j,\max}} \int_{m_{k,\min}}^{m_{k,\max}} p(t_i, z, m)\, \mathrm{d}z\mathrm{d}m. \tag{10.4.2}$$

We refer the reader to Leistedt et al. (2016) for a full mathematical description of their work. The posterior distribution of f_{ijk} turns out to be a Dirichlet distribution and a Gaussian likelihood is assumed for the photometric fluxes of each given galaxy. Leistedt et al. (2016) then used a two-step Gibbs sampler to sample the full posterior distribution to obtain samples of the desired quantities. This framework has been adapted by George Kyriacou for his thesis work, including selection effects appropriate for KV-450.

10.5 Analysis and Results

In our analysis, we follow Hildebrandt et al. (2020) and focus on the 2PCFs. We take a slightly different approach for the $n(z)$, which is determined using the method explained in §10.4. Moreoer, we do not require the shifts, δz_i for the $n(z)$ distributions (see Figure 10.3) generated using the BHM approach. We also use the analytical baryon feedback model (see Chapter 4, Equation 4.2.5) to account for baryon feedback. This model has an amplitude parameter, A_{bary} which we marginalise over in the sampling procedure. Importantly, we can easily couple the emulator we developed in Chapter 8 to the KV-450 likelihood code. Since both the emulator and the KV-450 have the same set of input cosmological parameters, this step was quite

straightforward. The number of nuisance parameters is reduced from 9 to 4 and $\boldsymbol{\beta}$ is now

$$\boldsymbol{\beta} = \left[A_{\rm IA},\, A_{\rm bary},\, \delta c,\, A_c\right].$$

Having established that the emulator for the 3D matter power spectrum, $P_\delta(k,z)$ (see for example, Figure 8.10 in Chapter 8) and CLASS give essentially the same cosmological results, for fixed $n(z)$, we use the faster emulator to undertake the more expensive task of marginalising over the $n(z)$ uncertainty. The emulator is therefore used to calculate $P_\delta(k,z)$ at each step in the MCMC sampling scheme.

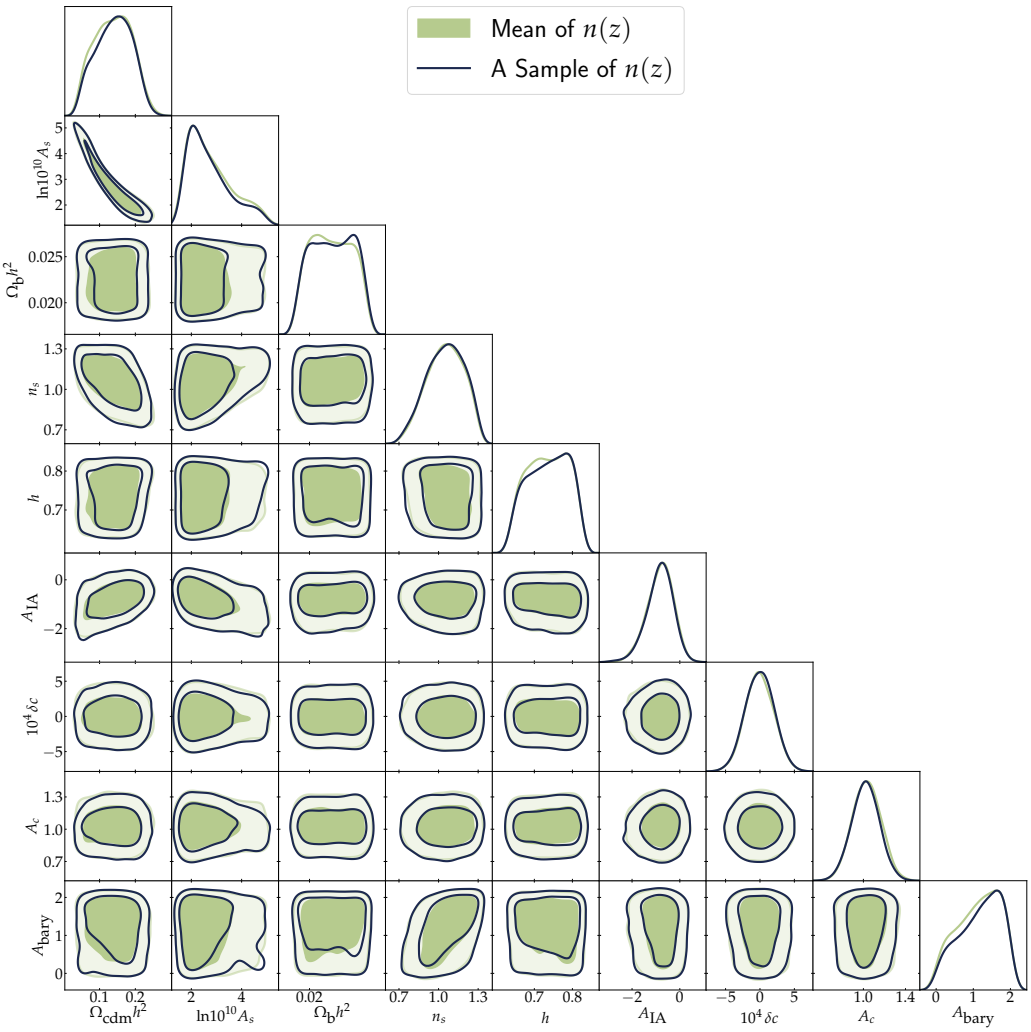
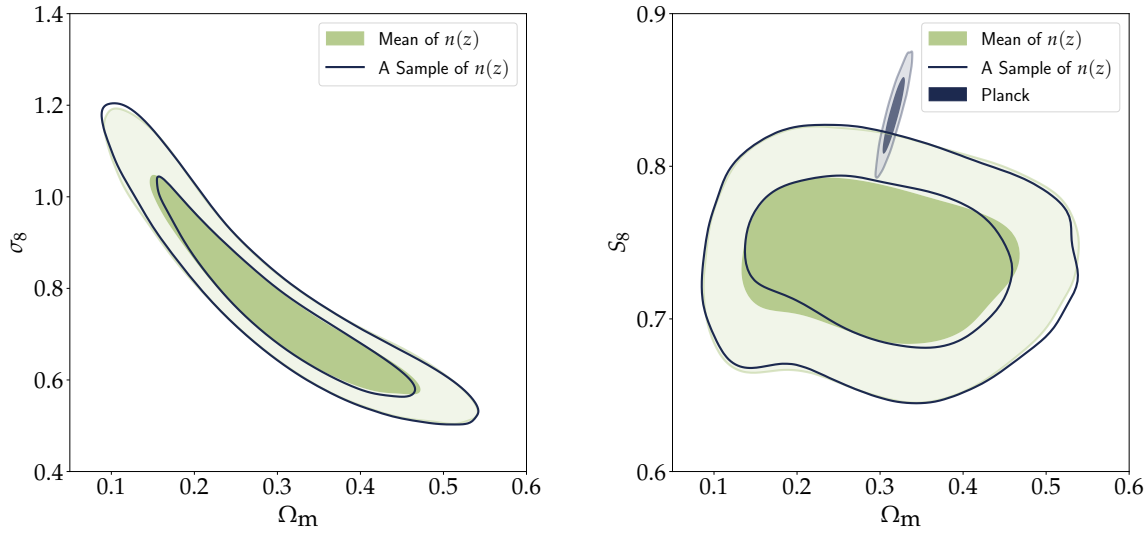


Figure 10.4 – The figure shows the posterior distribution of all cosmological and nuisance parameters. The contours contain 68% and 95% of the posterior. The olive green and solid dark green colours show the posterior obtained when sampling the posterior using the mean and a single draw of the $n(z)$ distributions respectively.

The next step involves calculating the convergence power spectrum and the two intrinsic alignment power spectra (GI and II). We use the double sum approach (see §9.2 in Chapter 9) for further details) in this step. In addition, we can perform three separate analyses using the $n(z)$ distributions as determined using BHM approach. We can either use the

- the mean, $\bar{n}(z)$,
- one random sample of $n(z)$ for each $\boldsymbol{\theta}$ or
- N samples of $n(z)$ for each $\boldsymbol{\theta}$, the Monte-Carlo marginalisation (see §9.3 in Chapter 9).

Hildebrandt et al. (2020) used the mean of the SOM method of the $n(z)$ distributions and in our case, one has the option to choose either the mean or a random sample of $n(z)$. Performing the Monte-Carlo integration to estimate the log-likelihood, that is, drawing N samples of the $n(z)$ for each set of cosmological and nuisance parameters is a computationally expensive step, which we do not apply in this data analysis problem. The reason it is expensive is because after the computation of the weak lensing and intrinsic alignment power spectra using the emulator, performing the Hankel transform (see §9.1 in Chapter 9) to compute the 2PCFs for each realisation of the $n(z)$ distribution is usually expensive. Therefore, the computation time grows linearly depending on the choice of N.



(a) σ_8 versus Ω_m plane **(b)** S_8 versus Ω_m plane

Figure 10.5 – The left panel shows the marginalised joint posterior distribution of σ_8 and Ω_m while the right panel shows S_8 and Ω_m for the KV-450 data . S_8 is re-parametrised in terms of σ_8 and Ω_m such that $S_8 := \sigma_8\sqrt{\Omega_m/0.3}$. The blue contours shows the joint posterior of the S_8 parameter against Ω_m for the latest Planck 2018 result.

Figure 10.4 shows the marginalised 1D and 2D distributions of all cosmological and nuisance parameters for the KV-450 data using the mean of the $n(z)$ distributions (in olive green) and the results corresponding to samples of $n(z)$ are shown in black green. The left panel of Figure 10.5 shows the $\sigma_8 - \Omega_m$ plane while, in the right panel, we compare the constraints on (S_8, Ω_m) to the Planck 2018 results (Planck Collaboration et al., 2020).

The analysis performed in this chapter is the first which applies an emulator for the 3D

matter power spectrum and a novel calibration of the $n(z)$ redshift distributions. It also makes use a new technique for re-writing the weak lensing power and intrinsic alignment power spectra as a (double) sum. Once the emulator is trained and stored, sampling the joint posterior distribution of the cosmological and nuisance parameters took around 30 hours on a Desktop computer using the EMCEE sampler (Foreman-Mackey et al., 2013).

Using the mean of the $n(z)$ distributions, the parameter S_8 is estimated to be $S_8 = 0.738^{+0.035}_{-0.036}$ and a similar estimate is obtained if a single sample of $n(z)$ is used at every step in the MCMC. These constraints are not very different when compared to the fiducial analysis of Hildebrandt et al. (2020), where the constraint is $S_8 = 0.737^{+0.040}_{-0.036}$. We believe that the sampling the posterior distribution will be faster with the emulator and the double sum approach combined, in the case where band powers are used as summary statistics. The main reason is that the additional step of computing 2PCFs using the Hankel transform is not required. See §10.6 where a future work is to apply the tools developed in this thesis to KiDS-1000 band power data. The fact that the emulator is also able to yield robust constraints when applied to an actual data from the KV-450 weak lensing survey demonstrates its robustness and therefore it can be used for future weak lensing surveys. The emulator and the Bayesian Hierarchical method developed for inferring the $n(z)$ can also be used in map-cosmology sampling scheme (Alsing et al., 2017) to constrain cosmological parameters from data directly, without the need for computing two-point summary statistics.

10.6 Future Work

Following the analysis on the KV-450 data above, we also intend to use the latest KiDS-1000 data to infer cosmological and nuisance parameter. In particular, the KiDS-1000 is a nine-band optical and near-infrared photometry survey and contains 1006 deg^2 of images. It is the fourth data release from KiDS, the Kilo-Degree Survey. Asgari et al. (2021) performed an in-depth cosmological analysis using the KiDS-1000 with three different sets of statistics namely, band powers, Complete Orthogonal Sets of E/B-Integrals (COSEBIs) and shear two-point correlation functions (2PCFs).

Cosmic shear, as elaborated in this thesis, has the capability of not only testing the cosmological model, but also constrains the amplitude of the matter density fluctuations, typically measured using the $S_8 = \sigma_8\sqrt{\Omega_\mathrm{m}/0.3}$ parameter. Ω_m is the matter density parameter and σ_8 yields the standard deviation of matter density in spheres of $8\,h^{-1}$Mpc. Despite the recent rapid development of cosmic shear analyses, not only through KiDS but also DES and HSC, they de-

mand for critical revision and comprehensive improvements. For example, observational and astrophysical systematics remain a major challenge and require some careful analysis. However, KiDS has some unique properties for it allows accurate measurement of the gravitational shear, as well as the accurate determination of the $n(z)$ redshift distributions.

An important extension of the work done for KV-450 would be to use the latest KiDS-1000 data to infer the redshift distributions[†] and couple the emulator with the likelihood code. In particular, since the emulator is built at the level of the 3D matter power spectrum, this trivially allows for the use of different summary statistics in the KiDS-1000 analysis.

10.7 Summary

In this chapter, we are adapting the KV-450 and KiDS-1000 likelihood codes so that they can support the different tools we have developed in this thesis. In particular, we are focusing on the weak lensing power spectra which we have re-written as a (double) sum in terms of the heights on the $n(z)$ distributions, the emulator which we have built in Chapter 8 and the $n(z)$ distributions generated using Bayesian Hierarchical Modelling (Leistedt et al., 2016).

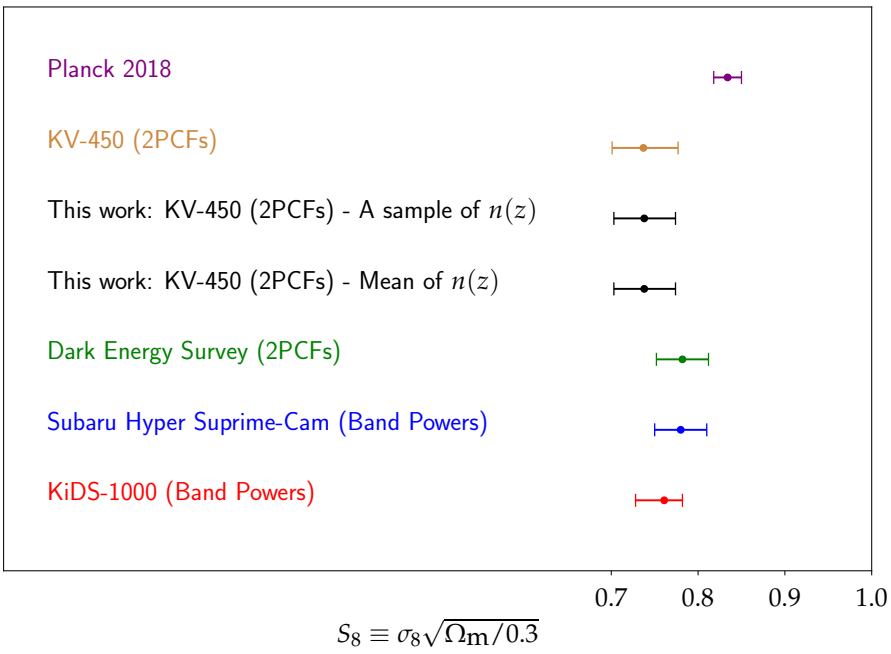


Figure 10.6 – The inferred values of S_8 in this work, compared to other weak lensing surveys. Note the tension with Planck's inferred S_8 value.The 68% credible interval is indicated by the horizontal bar for each experiment.

Early results obtained using the KV-450 data are robust when compared to the previous work based on cosmic shear, for example, the Dark Energy Survey (DES) (Troxel et al., 2018)

[†]Using the BHM developed by George Kyriacou

and Subaru Hyper Suprime-Cam (HSC) (Hikage et al., 2019). This opens a new avenue towards incorporating the emulator, the BHM method for the $n(z)$ distribution and the mathematical framework for calculating the weak lensing and intrinsic alignment power spectra, as part of a more complex, but principled Bayesian Hierarchical inference engine. The latter can possibly relax the assumption of a Gaussian assumption for weak lensing analysis. In other words, Bayesian Hierarchical inference of non-Gaussian shear field can be possible with efficient samplers such as Hamiltonian Monte Carlo. This fits perfectly with the emulator developed in this thesis, since the emulator also outputs derivatives with respect to the input cosmological parameters.

In the near future, we would like to extend the analysis in this chapter and apply the tools developed in this thesis to the KiDS-1000 data (Asgari et al., 2021). At this stage, we have successfully adapted the KiDS-1000 likelihood code so we can use the EMCEE sampler and we can now trivially couple the emulator of the 3D matter power spectrum, $P_\delta(k, z)$ with the likelihood.

Throughout this work, we have assumed a flat ΛCDM cosmological model to infer cosmological parameters. The results obtained in this work are close to other cosmic shear analysis such as the DES and HSC results. Since the inferred value of S_8 is lower compared to the latest *Planck* results (see Figure 10.6 for a comparison), there can be two possible explanations. One possibility lies in refining the cosmological model or it could also be that there are systematics in the data which are not fully accounted. The former raises various questions whether a wCDM model would instead be preferred. Troxel et al. (2018) performed an analysis based upon the wCDM model and reports $w = -0.95^{+0.33}_{-0.39}$ using the Dark Energy Survey (DES) Year 1 data. By computing the log-Bayes factor between a ΛCDM and wCDM model, they did not find significant preference for a model which allows for $w \neq -1$. Another source for the discrepancy can be in the handling of the systematics, for example, baryon feedback, intrinsic alignment modelling and neutrino, and the estimation of the redshift distributions. One possible test would be a combined analysis of the three surveys (KiDS, HSC and DES), including a robust and principled method for not only finding the redshifts of sources but also the estimation of the $n(z)$ distributions. A bias due to the small number and perhaps miscalculated spectroscopic redshifts might also be a possible source of systematics. With upcoming larger surveys, much diligent effort would be needed to determine the source of tension/discrepancy in the inferred value of S_8.

Chapter 11

CONCLUSIONS

> Research is what I'm doing when I don't know what I'm doing.
>
> *Wernher von Braun*

In this thesis, we have performed an in-depth analysis of the different emulating methods that one can adopt, not only for future weak lensing surveys, but for *any* cosmological data analysis pipeline. The principles behind remain the same, depending on the quantity (power spectrum, band power, MOPED coefficient) we want to emulate. In what follows, we will briefly summarise the motivation and purpose of each chapter in this thesis before elaborating on the possible use-cases of these techniques as future applications, not only in cosmology but also in the machine learning community. In some chapters, for example, Chapters 4, 5 and 7, we first test our methods and perform exploratory analyses before developing robust methodologies for publications. Moreover, Chapter 10 is still work in progress and the expectation is to have the three ingredients, namely the double sum approach for computing the weak lensing and intrinsic alignment power spectra, the Bayesian hierarchical method for estimating the $n(z)$ distributions and the emulator in a weak lensing analysis.

11.1 Summary

In this section, we describe briefly what we have covered in each chapter and how the ideas from each chapter follow from one another. This provides a summary of the wealth of topics and information we have explored, organised and applied in different weak lensing analyses.

In Chapter 1, we systematically go through the different theoretical concepts of weak lensing cosmology. This chapter serves as a cornerstone to the whole field of weak lensing and is crucial in the development and application of theoretical model(s) in any weak lensing data analysis problem.

In Chapter 2, we dive deep into the Bayesian methodology and motivate its use in the cosmology community. Arguably, it remains the favoured method in a cosmological data analysis problem, albeit the cost of evaluating the likelihood in a sampling procedure. Novel techniques based on (Bayesian/Statistical) machine learning are currently being developed and can pave their way in a likelihood analysis. One such example is a Bayesian emulator which is central in this thesis.

In Chapter 3, we extend the parametric Bayesian approach covered in Chapter 2 to non-parametric Bayesian techniques and one such approch is Gaussian Processes (GP). Instead of working in weight (parameter) space, the principle is to work in data space, hence allowing us to model any function given a sufficient number of training points. Crucially, the kernel function is the fundamental concept behind non-parametric Bayesian analysis.

In Chapter 4, we cover different techniques, namely polynomial models, Gaussian Processes and neural networks to illustrate **scalable** emulating methods, applied to the KiDS-450 band powers. In particular, we test the PICO algorithm (Fendt & Wandelt, 2007b) and develop an analogous technique for Gaussian Processes, called Product-of-Experts in the machine learning community. All three methods produce consistent results and open a new avenue to performing scalable emulation for weak lensing.

In Chapter 5, we use the JLA supernova data as a test case to show that we can emulate the MOPED coefficients using Gaussian Processes. Importantly, unlike the method explored in Chapter 4 where we had to emulate many functions, the MOPED formalism allows us to emulate only p functions and p is the number of parameters in our model. We also show that one can focus on the most expensive part of the pipeline, which is usually a function of a small number of parameters. Hence, the number of forward simulations can be reduced to a large extent.

In Chapter 6, we extend the concepts developed in the previous chapter and apply them to an actual likelihood analysis. In particular, we use the KiDS-450 likelihood code to couple an emulator for the band powers and the MOPED coefficients. This is a challenging problem because we have only 24 band powers and the data is not very informative about the parameters. Despite this hurdle, we are able to recover reliable posterior distributions for the cosmological and nuisance parameters using the emulator.

In Chapter 7, we introduce for the first time, in the cosmology community, an existing approach in the statistics community, which deals with semi-parametric Gaussian Processes. While the zero-mean Gaussian Process applied in the various test cases work well, one can

also embed existing prior information about the function we want to learn using the semi-parametric Gaussian Process approach. We test this method on the MOPED coefficients and we are able to generate robust posterior densities for all cosmological and nuisance parameters.

In Chapter 8, we use the semi-parametric Gaussian Process model developed in the previous chapter to build an emulator for the 3D matter power spectrum. The motivation behind is to emulate the most expensive part of a weak lensing analysis pipeline. The other power spectrum calculations, such as the convergence power spectrum and the intrinsic alignment power spectra can be done easily. Crucially, we show that the first and second derivatives of $P_\delta(k, z)$ with respect to the input cosmological parameters can also be computed analytically.

In Chapter 9, we highlight briefly different types of summary statistics which are used in current weak lensing data analyses. In addition, we derive expressions for calculating the weak lensing and intrinsic alignment power spectra, which involves summing over the product of the heights of the $n(z)$ redshift distributions and other pre-computed quantities which depend on the cosmological parameters. We also provide a simple method for marginalising over the $n(z)$ uncertainties.

In Chapter 10, we apply the emulator for the 3D matter power spectrum developed in the previous chapter, along with a Bayesian hierarchical method for determining the $n(z)$ distribution of sources, to infer cosmological parameters for the KiDS+VIKING-450 survey. In addition, we re-write the weak lensing and intrinsic alignment power spectra as a double sum. We discuss how these techniques can be extended and applied in future weak lensing surveys.

11.2 Future Applications

In this section, we highlight briefly different and feasible projects that one can consider. In particular, we discuss them from a cosmology and machine learning perspectives separately but these two fields are inextricably linked and the different ideas proposed can be used together in a single project.

Cosmology

An important and ubiquitous ingredient in most Bayesian analysis is the choice of the sampler. Unlike the traditional Metropolis-Hastings algorithm, gradient-based sampling techniques have been shown to be superior in many ways. As shown in Chapter 8, the fact that we can derive the gradient of the power spectrum with respect to the input cosmological parameters is a huge advantage if we choose to use a Hamiltonian Monte Carlo sampler. For the latter, we require the gradient of the negative log-likelihood with respect to the parameters we want to sample.

A different application of our work in Chapter 8 is to derive cosmological parameter constraints (and nuisance parameters) from an approximate inference perspective. For example, techniques such as Simulator Expansion for Likelihood-Free Inference (SELFI) (Leclercq et al., 2019) depend on expanding a black-box function around an expansion point, $\boldsymbol{\theta}_*$ and infer parameters analytically, that is, the resulting expression for the posterior distribution is a multivariate normal distribution. Suppose, $\boldsymbol{S}(\boldsymbol{\theta})$ is the simulator. The simulator can be approximated as

$$\boldsymbol{S} \approx \boldsymbol{S}_* + \nabla_{\boldsymbol{\theta}} \boldsymbol{S}(\boldsymbol{\theta} - \boldsymbol{\theta}_*)$$

at first order and leads to analytical solutions for the posterior. However, if we choose to extend this approach and include second order terms, then,

$$\boldsymbol{S} \approx \boldsymbol{S}_* + \nabla_{\boldsymbol{\theta}} \boldsymbol{S}(\boldsymbol{\theta} - \boldsymbol{\theta}_*) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_*)^{\mathrm{T}} \mathbf{H}_{\boldsymbol{\theta}} (\boldsymbol{\theta} - \boldsymbol{\theta}_*)$$

where $\nabla_{\boldsymbol{\theta}} \boldsymbol{S}(\boldsymbol{\theta})$ is the first derivative of the simulator and $\mathbf{H}_{\boldsymbol{\theta}}^{ij} \equiv \frac{\partial^2 S}{\partial \boldsymbol{\theta}_i \partial \boldsymbol{\theta}_j}$ corresponds to the Hessian matrix, that is, a matrix consisting of the second-order auto- and cross- derivatives of the simulator. Note that these are evaluated at the expansion point in the above expressions. If we choose to use the first derivative, then one requires only p forward simulations and at least $\frac{1}{2}p(p+1)$ forward simulations if we choose to include the second order terms. Note that we are assuming forward finite differencing. If the simulator is expensive, then one can use the emulator as a proxy to obtain the first and second derivatives and the inference follows naturally.

An important common tool is the Fisher information matrix calculation in cosmology. In a Fisher Matrix analysis, we are dealing with the negative log-likelihood, $\mathcal{L} \equiv -\ln p(\boldsymbol{x}|\boldsymbol{\theta})$. The Fisher matrix, $\mathbf{F}$ is simply the expectation value of the inverse of the Hessian matrix, $\mathbf{H}$ at $\boldsymbol{\theta} = \boldsymbol{\theta}_*$, that is, $\mathbf{F} = \langle \mathbf{H} \rangle$ and $\mathbf{F} \equiv \langle \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta}_i \partial \boldsymbol{\theta}_j} \rangle$. The fact that we have access to the first and second derivatives of the 3D matter power spectrum through the GP emulator will enable us to use them in a Fisher matrix analysis, albeit the extra steps required to explicitly calculate the derivatives of $\mathcal{L}$.

Machine Learning

From a machine learning perspective, we can further improve upon the methods developed in Chapter 6. Instead of placing initially the training points across the whole prior range, one can instead attempt to augment the training points in an active learning scenario, see for ex-

ample, Figure 11.1. This can be achieved using Bayesian optimisation which attempts to find the maximum of a function by leveraging exploitation (regions of high mean) and exploration (regions of high uncertainty). Common applications augment the training set in a serial fashion, but with MOPED, one can do this in parallel, that is, we can have a GP regressor for each MOPED coefficient. For each regressor, we can find the next best training point to be added to the training set. Note that, the functions (emulator for the MOPED coefficients) are different and the utility function in Bayesian optimisation generally has multiple local minima. Hence, each emulator will likely optimise a different set of parameters at each step of the optimisation procedure.

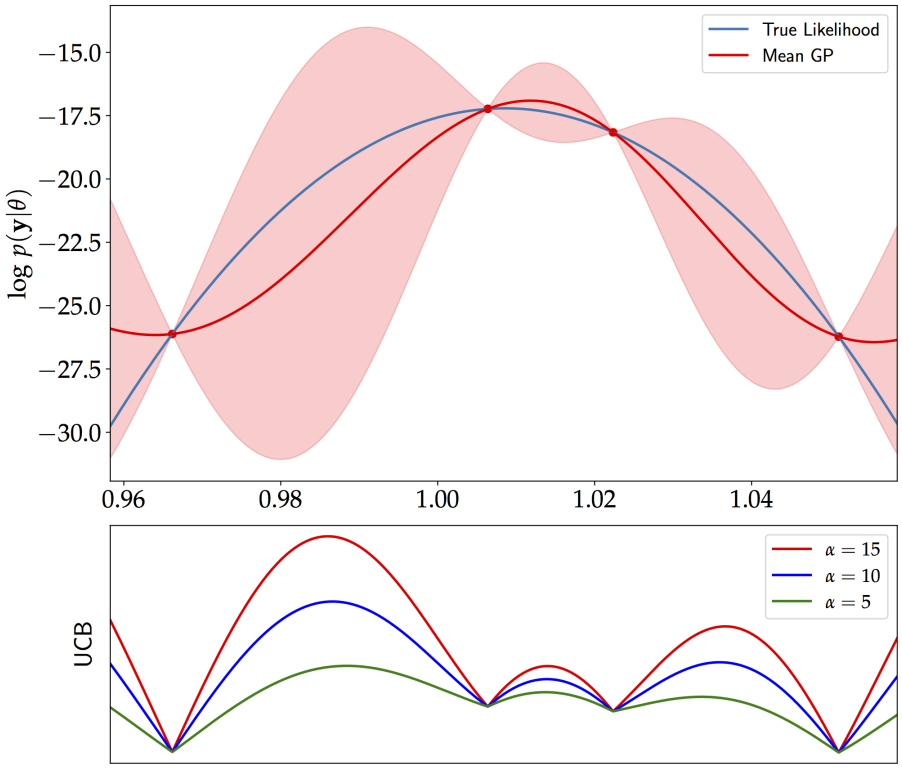


Figure 11.1 – An example of Bayesian optimisation where a log-likelihood function is emulated (initially with just 4 points) using a Gaussian Process. The utility function used here is referred to as the Upper Confidence Bound (UCB), $u\boldsymbol{\theta} = \mu + \alpha\sigma$ and as seen in the bottom panel, different choices of α lead to different shape of the utility function. Crucially, it also has different optima.

Another extension of the research carried out in this thesis is to adopt or perhaps devise techniques for scalable Gaussian Process. We have presented one such method in Chapter 4 based on Product-of-Experts. Another example includes the use of inducing point method (Quiñonero-Candela & Rasmussen, 2005). This is a promising approach since the number of training points involved in training/prediction is reduced from N to m, $m < N$, hence the computational complexity is reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(Nm^2)$. An application of such method can be in the case where we want to build a regressor for the log-likelihood. For example, doing

a full likelihood analysis for the Planck data is deemed to be very expensive. One can either run a short MCMC or evaluate the likelihood at a large set of LH samples, build and store an emulated model for the likelihood. Alternatively, one can use existing chains of MCMC samples, perhaps from previous experiments, to build the training set.

An emerging trend in the machine learning community is to make use of automatic differentiation (autodiff) for gradient computation. While we were finishing this work, a useful Python package called `KeOps` (Kernel Operations on the Graphic Processing Unit, GPU, with autodiff) was released and has the advantage of not only leveraging automatic differentiation but also GPU computing, importantly without resulting in memory overflows. Recall that when working with kernel methods, for example, for a Gaussian Process, we have to deal with an $N \times N$ matrix and this can lead to memory overflow if N is too big. Developing technologies based on the ideas developed in this thesis, along with new code development such as `KeOps` will certainly bring about new statistical machine learning methods in weak lensing cosmology in the near future.

The different aspects of the work in this thesis can be applied in future weak lensing surveys, such as *Euclid* and the Vera Rubin Observatory. In particular, with the emergence of large data sets, the combination of the MOPED and Gaussian Process formalism will play an important role in finding constraints on cosmological parameters in a very fast way. On the other hand, the emulator for the 3D matter power spectrum is not only very fast to compute but also outputs analytical derivatives with respect to input cosmological parameters. This opens a new avenue for working with efficient samplers such as the Hamiltonian Monte Carlo method, which requires derivatives of the log-likelihood with respect to the inputs. These techniques are aligned with the long term scientific goals of *Euclid* and the Vera Rubin Observatory, for which there is a demand to develop fast and robust tools, whilst being able to marginalise of hundreds of nuisance parameters to deal with the systematics.

BIBLIOGRAPHY

Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, software available from tensorflow.org

Abbott, T., Abdalla, F. B., Allam, S., et al., Cosmology from cosmic shear with Dark Energy Survey Science Verification data. 2016, *Phys. Rev. D*, 94, 022001

Abbott, T. M. C., Abdalla, F. B., Alarcon, A., et al., Dark Energy Survey year 1 results: Cosmological constraints from galaxy clustering and weak lensing. 2018, *Phys. Rev. D*, 98, 043526

Abbott, T. M. C., Allam, S., Andersen, P., et al., First Cosmology Results using Type Ia Supernovae from the Dark Energy Survey: Constraints on Cosmological Parameters. 2019, *ApJ*, 872, L30

Agarwal, S., Abdalla, F. B., Feldman, H. A., Lahav, O., & Thomas, S. A., PkANN - I. Non-linear matter power spectrum interpolation through artificial neural networks. 2012, *MNRAS*, 424, 1409

Agarwal, S., Abdalla, F. B., Feldman, H. A., Lahav, O., & Thomas, S. A., PkANN - II. A non-linear matter power spectrum interpolator developed using artificial neural networks. 2014, *MNRAS*, 439, 2102

Aihara, H., Arimoto, N., Armstrong, R., et al., The Hyper Suprime-Cam SSP Survey: Overview and survey design. 2018, *PASJ*, 70, S4

Alsing, J., Charnock, T., Feeney, S., & Wandelt, B., Fast likelihood-free cosmology with neural density estimators and active learning. 2019, *MNRAS*, 488, 4440

Alsing, J., Heavens, A., & Jaffe, A. H., Cosmological parameters, shear maps and power spectra from CFHTLenS using Bayesian hierarchical inference. 2017, *MNRAS*, 466, 3272

Alsing, J., Heavens, A., Jaffe, A. H., et al., Hierarchical cosmic shear power spectrum inference. 2016, *MNRAS*, 455, 4452

Alsing, J. & Wandelt, B., Generalized massive optimal data compression. 2018, *MNRAS*, 476, L60

Alsing, J. & Wandelt, B., Nuisance hardened data compression for fast likelihood-free inference. 2019, *MNRAS*, 488, 5093

Alsing, J., Wandelt, B., & Feeney, S., Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology. 2018, *MNRAS*, 477, 2874

Amon, A., Gruen, D., Troxel, M. A., et al., Dark Energy Survey Year 3 Results: Cosmology from Cosmic Shear and Robustness to Data Calibration. 2021, arXiv e-prints, arXiv:2105.13543

Aricò, G., Angulo, R. E., & Zennaro, M., Accelerating Large-Scale-Structure data analyses by emulating Boltzmann solvers and Lagrangian Perturbation Theory. 2021, arXiv e-prints, arXiv:2104.14568

Asgari, M., Heymans, C., Hildebrandt, H., et al., Consistent cosmic shear in the face of systematics: a B-mode analysis of KiDS-450, DES-SV and CFHTLenS. 2019, *A&A*, 624, A134

Asgari, M., Lin, C.-A., Joachimi, B., et al., KiDS-1000 cosmology: Cosmic shear constraints and comparison between two point statistics. 2021, *A&A*, 645, A104

Asgari, M., Tröster, T., Heymans, C., et al., KiDS+VIKING-450 and DES-Y1 combined: Mitigating baryon feedback uncertainty with COSEBIs. 2020, *A&A*, 634, A127

Auld, T., Bridges, M., Hobson, M. P., & Gull, S. F., Fast cosmological parameter estimation using neural networks. 2007, *MNRAS*, 376, L11

Barz, B. & Denzler, J., Deep Learning on Small Datasets without Pre-Training using Cosine Loss. 2019, arXiv e-prints, arXiv:1901.09054

Bastos, L. S. & O'Hagan, A., Diagnostics for Gaussian Process Emulators. 2009, *Technometrics*, 51, 51

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M., Automatic differentiation in machine learning: a survey. 2017, The Journal of Machine Learning Research, 18, 18

Benítez, N., Bayesian Photometric Redshift Estimation. 2000, *ApJ*, 536, 571

Betancourt, M., A Conceptual Introduction to Hamiltonian Monte Carlo. 2017, arXiv e-prints, arXiv:1701.02434

Betoule, M., Kessler, R., Guy, J., et al., Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples. 2014, *A&A*, 568, A22

Bingham, E., Chen, J. P., Jankowiak, M., et al., Pyro: Deep Universal Probabilistic Programming. 2019, J. Mach. Learn. Res., 20, 20

Bishop, C. M. 2006, Pattern recognition and machine learning (springer)

Blight, B. J. N. & Ott, L., A Bayesian approach to model inadequacy for polynomial regression. 1975, *Biometrika*, 62, 62

Bond, J. R., Jaffe, A. H., & Knox, L., Estimating the power spectrum of the cosmic microwave background. 1998, *Phys. Rev. D*, 57, 2117

Bonnett, C., Troxel, M. A., Hartley, W., et al., Redshift distributions of galaxies in the Dark Energy Survey Science Verification shear catalogue and implications for weak lensing. 2016, *Phys. Rev. D*, 94, 042005

Bordoloi, R., Lilly, S. J., & Amara, A., Photo-z performance for precision cosmology. 2010, *MNRAS*, 406, 881

Buchner, J., A statistical test for Nested Sampling algorithms. 2014, arXiv e-prints, arXiv:1407.5459

Buchner, J., Collaborative Nested Sampling: Big Data versus Complex Physical Models. 2019, *PASP*, 131, 108005

Cao, Y. & Fleet, D. J., Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions. 2014, arXiv e-prints, arXiv:1410.7827

Carnell, R., lhs: Latin hypercube samples. 2012, R package version 0.10

Carnell, R., Package 'lhs'. 2019, 780, 780

Castro, P. G., Heavens, A. F., & Kitching, T. D., Weak lensing analysis in three dimensions. 2005, *Phys. Rev. D*, 72, 023516

Charnock, T., Lavaux, G., & Wandelt, B. D., Automatic physical inference with information maximizing neural networks. 2018, *Phys. Rev. D*, 97, 083004

Collister, A. A. & Lahav, O., ANNz: Estimating Photometric Redshifts Using Artificial Neural Networks. 2004, *PASP*, 116, 345

de Jong, J. T. A., Verdoes Kleijn, G. A., Kuijken, K. H., & Valentijn, E. A., The Kilo-Degree Survey. 2013, *Experimental Astronomy*, 35, 25

Deisenroth, M. P. & Ng, J. W., Distributed Gaussian Processes. 2015, arXiv e-prints, arXiv:1502.02843

Desautels, T., Krause, A., & Burdick, J., Parallelizing Exploration-Exploitation Tradeoffs with Gaussian Process Bandit Optimization. 2012, arXiv e-prints, arXiv:1206.6402

Deshpande, A. C. & Kitching, T. D., Post-Limber weak lensing bispectrum, reduced shear correction, and magnification bias correction. 2020, *Phys. Rev. D*, 101, 103531

Dickey, J. M., The Weighted Likelihood Ratio, Linear Hypotheses on Normal Location Parameters. 1971, *The Annals of Mathematical Statistics*, 42, 42

Dodelson, S. 2003, Modern Cosmology (Amsterdam: Academic Press)

Dodelson, S. 2017, Gravitational Lensing (Cambridge University Press)

Duane, S., Kennedy, A., Pendleton, B. J., & Roweth, D., Hybrid Monte Carlo. 1987, *Physics Letters B*, 195, 195

Eifler, T., Miyatake, H., Krause, E., et al., Cosmology with the Roman Space Telescope - multi-probe strategies. 2021, *MNRAS*, 507, 1746

Elsner, F. & Wandelt, B. D., Efficient Wiener filtering without preconditioning. 2013, *A&A*, 549, A111

Erben, T., Hildebrandt, H., Miller, L., et al., CFHTLenS: the Canada-France-Hawaii Telescope Lensing Survey - imaging data and catalogue products. 2013, *MNRAS*, 433, 2545

Euclid Collaboration, Blanchard, A., Camera, S., et al., Euclid preparation: VII. Forecast validation for Euclid cosmological probes. 2019, arXiv e-prints, arXiv:1910.09273

Fang, X., Krause, E., Eifler, T., & MacCrann, N., Beyond Limber: efficient computation of angular power spectra for galaxy clustering and weak lensing. 2020, *J. Cosmology Astropart. Phys.*, 2020, 010

Fendt, W. A. & Wandelt, B. D., Computing High Accuracy Power Spectra with Pico. 2007a, arXiv e-prints, arXiv:0712.0194

Fendt, W. A. & Wandelt, B. D., Pico: Parameters for the Impatient Cosmologist. 2007b, *ApJ*, 654, 2

Feroz, F., Hobson, M. P., & Bridges, M., MULTINEST: an efficient and robust Bayesian inference tool for cosmology and particle physics. 2009, *MNRAS*, 398, 1601

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J., emcee: The MCMC Hammer. 2013, *PASP*, 125, 306

Freedman, W. L., Madore, B. F., Hoyt, T., et al., Calibration of the Tip of the Red Giant Branch. 2020, *ApJ*, 891, 57

Gelman, A. & Rubin, D. B., Inference from Iterative Simulation Using Multiple Sequences. 1992, *Statistical Science*, 7, 457

Geman, S. & Geman, D., Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. 1984, IEEE Transactions on Pattern Recognition, 6, 6

Ghahramani, Z., Bayesian non-parametrics and the probabilistic approach to modelling. 2013, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 371, 371

Ghahramani, Z., Probabilistic machine learning and artificial intelligence. 2015, Nature, 521, 521

Goldberg, J. N., Macfarlane, A. J., Newman, E. T., Rohrlich, F., & Sudarshan, E. C. G., Spin-s Spherical Harmonics and ð. 1967, *Journal of Mathematical Physics*, 8, 2155

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., et al., Generative Adversarial Networks. 2014, arXiv e-prints, arXiv:1406.2661

Goodman, J. & Weare, J., Ensemble samplers with affine invariance. 2010, *Communications in Applied Mathematics and Computational Science*, 5, 65

Gutmann, M. U. & Corander, J., Bayesian Optimization for Likelihood-Free Inference of Simulator-Based Statistical Models. 2016, Journal of Machine Learning Research, 17, 17

Habib, S., Heitmann, K., Higdon, D., Nakhleh, C., & Williams, B., Cosmic calibration: Constraints from the matter power spectrum and the cosmic microwave background. 2007, *Phys. Rev. D*, 76, 083503

Hajian, A., Efficient cosmological parameter estimation with Hamiltonian MonteCarlo technique. 2007, *Phys. Rev. D*, 75, 083525

Handley, W. J., Hobson, M. P., & Lasenby, A. N., polychord: nested sampling for cosmology. 2015a, *MNRAS*, 450, L61

Handley, W. J., Hobson, M. P., & Lasenby, A. N., POLYCHORD: next-generation nested sampling. 2015b, *MNRAS*, 453, 4384

Harnois-Déraps, J., van Waerbeke, L., Viola, M., & Heymans, C., Baryons, neutrinos, feedback and weak gravitational lensing. 2015, *MNRAS*, 450, 1212

Hastie, T., Tibshirani, R., & Friedman, J. 2001, The Elements of Statistical Learning (Springer series in statistics New York)

Hastings, W. K., Monte Carlo Sampling Methods Using Markov Chains and Their Applications. 1970, Biometrika, 57, 57

Heavens, A., 3D weak lensing. 2003, *MNRAS*, 343, 1327

Heavens, A., Alsing, J., & Jaffe, A. H., Combining size and shape in weak lensing. 2013, *MNRAS*, 433, L6

Heavens, A., Fantaye, Y., Mootoovaloo, A., et al., Marginal Likelihoods from Monte Carlo Markov Chains. 2017a, arXiv e-prints, arXiv:1704.03472

Heavens, A. F., Jimenez, R., & Lahav, O., Massive lossless data compression and multiple parameter estimation from galaxy spectra. 2000, *MNRAS*, 317, 965

Heavens, A. F., Kitching, T. D., & Taylor, A. N., Measuring dark energy properties with 3D cosmic shear. 2006, *MNRAS*, 373, 105

Heavens, A. F., Sellentin, E., de Mijolla, D., & Vianello, A., Massive data compression for parameter-dependent covariance matrices. 2017b, *MNRAS*, 472, 4244

Heitmann, K., Higdon, D., White, M., et al., The Coyote Universe. II. Cosmological Models and Precision Emulation of the Nonlinear Matter Power Spectrum. 2009, *ApJ*, 705, 156

Heitmann, K., Lawrence, E., Kwan, J., Habib, S., & Higdon, D., The Coyote Universe Extended: Precision Emulation of the Matter Power Spectrum. 2014, *ApJ*, 780, 111

Heitmann, K., White, M., Wagner, C., Habib, S., & Higdon, D., The Coyote Universe. I. Precision Determination of the Nonlinear Matter Power Spectrum. 2010, *ApJ*, 715, 104

Hensman, J., Fusi, N., & Lawrence, N. D., Gaussian Processes for Big Data. 2013, arXiv e-prints, arXiv:1309.6835

Heymans, C., Grocutt, E., Heavens, A., et al., CFHTLenS tomographic weak lensing cosmological parameter constraints: Mitigating the impact of intrinsic galaxy alignments. 2013, *MNRAS*, 432, 2433

Higson, E., Handley, W., Hobson, M., & Lasenby, A., Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation. 2019, *Statistics and Computing*, 29, 891

Hikage, C., Oguri, M., Hamana, T., et al., Cosmology from cosmic shear power spectra with Subaru Hyper Suprime-Cam first-year data. 2019, *PASJ*, 71, 43

Hildebrandt, H., Köhlinger, F., van den Busch, J. L., et al., KiDS+VIKING-450: Cosmic shear tomography with optical and infrared data. 2020, *A&A*, 633, A69

Hildebrandt, H., Viola, M., Heymans, C., et al., KiDS-450: cosmological parameter constraints from tomographic weak gravitational lensing. 2017, *MNRAS*, 465, 1454

Hinshaw, G., Nolta, M. R., Bennett, C. L., et al., Three-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Temperature Analysis. 2007, *ApJS*, 170, 288

Hirata, C. M. & Seljak, U., Intrinsic alignment-lensing interference as a contaminant of cosmic shear. 2004, *Phys. Rev. D*, 70, 063526

Ho, M.-F., Bird, S., & Shelton, C. R., A Multi-Fidelity Emulator for the Matter Power Spectrum using Gaussian Processes. 2021, arXiv e-prints, arXiv:2105.01081

Hoekstra, H., The effect of imperfect models of point spread function anisotropy on cosmic shear measurements. 2004, *MNRAS*, 347, 1337

Hu, W., Weak lensing of the CMB: A harmonic approach. 2000, *Phys. Rev. D*, 62, 043007

Hu, W. & White, M., Power Spectra Estimation for Weak Lensing. 2001, *ApJ*, 554, 67

Huterer, D., Takada, M., Bernstein, G., & Jain, B., Systematic errors in future weak-lensing surveys: requirements and prospects for self-calibration. 2006, *MNRAS*, 366, 101

Jasche, J. & Lavaux, G., Matrix-free large-scale Bayesian inference in cosmology. 2015, *MNRAS*, 447, 1204

Jeffreys, H., An Invariant Form for the Prior Probability in Estimation Problems. 1946, *Proceedings of the Royal Society of London Series A*, 186, 453

Joachimi, B. & Bridle, S. L., Simultaneous measurement of cosmology and intrinsic alignments using joint cosmic shear and galaxy number density correlations. 2010, *A&A*, 523, A1

Joachimi, B., Köhlinger, F., Handley, W., & Lemos, P., When tension is just a fluctuation. How noisy data affect model comparison. 2021a, *A&A*, 647, L5

Joachimi, B., Lin, C. A., Asgari, M., et al., KiDS-1000 methodology: Modelling and inference for joint weak gravitational lensing and spectroscopic galaxy clustering analysis. 2021b, *A&A*, 646, A129

Johnson, M., Moore, L., & Ylvisaker, D., Minimax and maximin distance designs. 1990, *Journal of Statistical Planning and Inference*, 26, 26

Jones, D. M. & Heavens, A. F., Bayesian photometric redshifts of blended sources. 2019a, *MNRAS*, 483, 2487

Jones, D. M. & Heavens, A. F., Gaussian mixture models for blended photometric redshifts. 2019b, *MNRAS*, 490, 3966

Kaiser, N., Squires, G., & Broadhurst, T., A Method for Weak Lensing Observations. 1995, *ApJ*, 449, 460

Karvonen, T., Wynne, G., Tronarp, F., Oates, C. J., & Särkkä, S., Maximum likelihood estimation and uncertainty quantification for Gaussian process approximation of deterministic functions. 2020, arXiv e-prints, arXiv:2001.10965

Kilbinger, M., Cosmology with cosmic shear observations: a review. 2015, *Reports on Progress in Physics*, 78, 086901

Kilbinger, M., Heymans, C., Asgari, M., et al., Precision calculations of the cosmic shear power spectrum projection. 2017, *MNRAS*, 472, 2126

Kingma, D. P. & Ba, J., Adam: A Method for Stochastic Optimization. 2014, arXiv e-prints, arXiv:1412.6980

Kitching, T. D. & Heavens, A. F., Unequal-time correlators for cosmology. 2017, *Phys. Rev. D*, 95, 063522

Kitching, T. D., Miller, L., Heymans, C. E., van Waerbeke, L., & Heavens, A. F., Bayesian galaxy shape measurement for weak lensing surveys - II. Application to simulations. 2008, *MNRAS*, 390, 149

Kobayashi, Y., Nishimichi, T., Takada, M., Takahashi, R., & Osato, K., Accurate emulator for the redshift-space power spectrum of dark matter halos and its application to galaxy power spectrum. 2020, *Phys. Rev. D*, 102, 063504

Köhlinger, F., Viola, M., Joachimi, B., et al., KiDS-450: the tomographic weak lensing power spectrum and constraints on cosmological parameters. 2017, *MNRAS*, 471, 4412

Kunz, M., Trotta, R., & Parkinson, D. R., Measuring the effective complexity of cosmological models. 2006, *Phys. Rev. D*, 74, 023503

Laureijs, R., Amiaux, J., Arduini, S., et al., Euclid Definition Study Report. 2011, arXiv e-prints, arXiv:1110.3193

Lawrence, E., Heitmann, K., White, M., et al., The Coyote Universe. III. Simulation Suite and Precision Emulator for the Nonlinear Matter Power Spectrum. 2010, *ApJ*, 713, 1322

Leclercq, F., Bayesian optimization for likelihood-free cosmological inference. 2018, *Phys. Rev. D*, 98, 063511

Leclercq, F., Enzi, W., Jasche, J., & Heavens, A., Primordial power spectrum and cosmology from black-box galaxy surveys. 2019, *MNRAS*, 490, 4237

Leistedt, B., Mortlock, D. J., & Peiris, H. V., Hierarchical Bayesian inference of galaxy redshift distributions from photometric surveys. 2016, *MNRAS*, 460, 4258

Lemos, P., Challinor, A., & Efstathiou, G., The effect of Limber and flat-sky approximations on galaxy weak lensing. 2017, *J. Cosmology Astropart. Phys.*, 2017, 014

Leonard, A., Lanusse, F., & Starck, J.-L., Weak lensing reconstructions in 2D and 3D: implications for cluster studies. 2015, *MNRAS*, 449, 1146

Lesgourgues, J., The Cosmic Linear Anisotropy Solving System (CLASS) I: Overview. 2011, arXiv e-prints, arXiv:1104.2932

Liddle, A. R. 1998, An introduction to modern cosmology

Lilly, S. J., Le Brun, V., Maier, C., et al., The zCOSMOS 10k-Bright Spectroscopic Sample. 2009, *ApJS*, 184, 218

Lima, M., Cunha, C. E., Oyaizu, H., et al., Estimating the Redshift Distribution of Faint Galaxy Samples. 2008, *Mon. Not. Roy. Astron. Soc.*, 390, 390

Lima, M., Cunha, C. E., Oyaizu, H., et al., Estimating the redshift distribution of photometric galaxy samples. 2008, *MNRAS*, 390, 118

Limber, D. N., The Analysis of Counts of the Extragalactic Nebulae in Terms of a Fluctuating Density Field. 1953, *ApJ*, 117, 134

Loverde, M. & Afshordi, N., Extended Limber approximation. 2008, *Phys. Rev. D*, 78, 123506

MacKay, D. J. 2003, Information theory, inference and learning algorithms (Cambridge university press)

Malz, A. I., How not to obtain the redshift distribution from probabilistic redshift estimates: Under what conditions is it not inappropriate to estimate the redshift distribution N (z) by stacking photo-z PDFs? 2021, *Phys. Rev. D*, 103, 083502

Mandelbaum, R., Weak Lensing for Precision Cosmology. 2018, *ARA&A*, 56, 393

Manrique-Yus, A. & Sellentin, E., Euclid-era cosmology for everyone: neural net assisted MCMC sampling for the joint 3×2 likelihood. 2020, *MNRAS*, 491, 2655

McKay, M. D., Beckman, R. J., & Conover, W. J., Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. 1979, Technometrics, 21, 21

Mead, A. J., Peacock, J. A., Heymans, C., Joudaki, S., & Heavens, A. F., An accurate halo model for fitting non-linear cosmological power spectra and baryonic feedback models. 2015, *MNRAS*, 454, 1958

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E., Equation of State Calculations by Fast Computing Machines. 1953, *J. Chem. Phys.*, 21, 1087

Miller, L., Heymans, C., Kitching, T. D., et al., Bayesian galaxy shape measurement for weak lensing surveys - III. Application to the Canada-France-Hawaii Telescope Lensing Survey. 2013, *MNRAS*, 429, 2858

Miller, L., Kitching, T. D., Heymans, C., Heavens, A. F., & van Waerbeke, L., Bayesian galaxy shape measurement for weak lensing surveys - I. Methodology and a fast-fitting algorithm. 2007, *MNRAS*, 382, 315

Mootoovaloo, A., Heavens, A. F., Jaffe, A. H., & Leclercq, F., Parameter inference for weak lensing using Gaussian Processes and MOPED. 2020, *MNRAS*, 497, 2213

Mukhanov, V. 2005, Physical Foundations of Cosmology (Cambridge University Press)

Mustafa, M., Bard, D., Bhimji, W., et al., CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks. 2019, *Computational Astrophysics and Cosmology*, 6, 1

Nau, R. F., De Finetti was right: probability does not exist. 2001, Theory and Decision, 51, 51

Neal, R. 2011, MCMC Using Hamiltonian Dynamics, 113–162

Newman, E. T. & Penrose, R., Note on the Bondi-Metzner-Sachs Group. 1966, *Journal of Mathematical Physics*, 7, 863

Newman, J. A., Calibrating Redshift Distributions beyond Spectroscopic Limits with Cross-Correlations. 2008, *ApJ*, 684, 88

Newman, J. A., Calibrating Redshift Distributions Beyond Spectroscopic Limits with Cross-Correlations. 2008, *Astrophys. J.*, 684, 684

Newman, J. A., Cooper, M. C., Davis, M., et al., The DEEP2 Galaxy Redshift Survey: Design, Observations, Data Reduction, and Redshifts. 2013, *ApJS*, 208, 5

Nightingale, J. W., Massey, R. J., Harvey, D. R., et al., Galaxy structure with strong gravitational lensing: decomposing the internal mass distribution of massive elliptical galaxies. 2019, *MNRAS*, 489, 2049

Perlmutter, S., Aldering, G., Goldhaber, G., et al., Measurements of Ω and Λ from 42 High-Redshift Supernovae. 1999, *ApJ*, 517, 565

Planck Collaboration, Aghanim, N., Akrami, Y., et al., Planck 2018 results. VI. Cosmological parameters. 2020, *A&A*, 641, A6

Porqueres, N., Heavens, A., Mortlock, D., & Lavaux, G., Lifting weak lensing degeneracies with a field-based likelihood. 2021, arXiv e-prints, arXiv:2108.04825

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 2007, Numerical recipes 3rd edition: The art of scientific computing (Cambridge university press)

Quiñonero-Candela, J. & Rasmussen, C. E., A Unifying View of Sparse Approximate Gaussian Process Regression. 2005, Journal of Machine Learning Research, 6, 6

Rasmussen, C. E. & Williams, C. K. I. 2006, Gaussian Processes for Machine Learning

Riddell, A., Hartikainen, A., & Carter, M. 2021, pystan (3.0.0), PyPI

Riess, A. G., Casertano, S., Yuan, W., Macri, L. M., & Scolnic, D., Large Magellanic Cloud Cepheid Standards Provide a 1% Foundation for the Determination of the Hubble Constant and Stronger Evidence for Physics beyond ΛCDM. 2019, *ApJ*, 876, 85

Riess, A. G., Filippenko, A. V., Challis, P., et al., Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant. 1998, *AJ*, 116, 1009

Rodríguez, A. C., Kacprzak, T., Lucchi, A., et al., Fast cosmic web simulations with generative adversarial networks. 2018, *Computational Astrophysics and Cosmology*, 5, 4

Salvatier, J., Wiecki, T. V., & Fonnesbeck, C., Probabilistic programming in Python using PyMC3. 2016, *PeerJ Computer Science*, 2, 2

Santos, M. G., Heavens, A., Balbi, A., et al., Multiple methods for estimating the bispectrum of the cosmic microwave background with application to the MAXIMA data. 2003, *MNRAS*, 341, 623

Schmit, C. J. & Pritchard, J. R., Emulation of reionization simulations for Bayesian inference of astrophysics parameters using neural networks. 2018, *MNRAS*, 475, 1213

Schneider, M. D., Holm, Ó., & Knox, L., Intelligent Design: On the Emulation of Cosmological Simulations. 2011, *ApJ*, 728, 137

Schneider, P. 2006, Extragalactic Astronomy and Cosmology

Schneider, P., Eifler, T., & Krause, E., COSEBIs: Extracting the full E-/B-mode information from cosmic shear correlation functions. 2010, *A&A*, 520, A116

Schneider, P. & Seitz, C., Steps towards nonlinear cluster inversion through gravitational distortions. I. Basic considerations and circular clusters. 1995, A&A, 294, 411

Seitz, C. & Schneider, P., Steps towards nonlinear cluster inversion through gravitational distortions. III. Including a redshift distribution of the sources. 1997, A&A, 318, 687

Semboloni, E., Hoekstra, H., Schaye, J., van Daalen, M. P., & McCarthy, I. G., Quantifying the effect of baryon physics on weak lensing tomography. 2011, *MNRAS*, 417, 2020

Singh, S. & Mandelbaum, R., Intrinsic alignments of BOSS LOWZ galaxies - II. Impact of shape measurement methods. 2016, *MNRAS*, 457, 2301

Skilling, J. 2004, in American Institute of Physics Conference Series, Vol. 735, Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 24th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, ed. R. Fischer, R. Preuss, & U. V. Toussaint, 395–405

Skilling, J., Nested sampling for general Bayesian computation. 2006, *Bayesian Analysis*, 1, 1

Slivkins, A., Introduction to Multi-Armed Bandits. 2019, arXiv e-prints, arXiv:1904.07272

Snelson, E. & Ghahramani, Z., Sparse Gaussian processes using pseudo-inputs. 2005, Advances in neural information processing systems, 18, 18

Spurio Mancini, A., Piras, D., Alsing, J., Joachimi, B., & Hobson, M. P., *CosmoPower* : emulating cosmological power spectra for accelerated Bayesian inference from next-generation surveys. 2021, arXiv e-prints, arXiv:2106.03846

Takahashi, R., Sato, M., Nishimichi, T., Taruya, A., & Oguri, M., Revising the Halofit Model for the Nonlinear Matter Power Spectrum. 2012, *ApJ*, 761, 152

Tanaka, M., Coupon, J., Hsieh, B.-C., et al., Photometric redshifts for Hyper Suprime-Cam Subaru Strategic Program Data Release 1. 2018, *PASJ*, 70, S9

Tegmark, M., Taylor, A. N., & Heavens, A. F., Karhunen-Loève Eigenvalue Problems in Cosmology: How Should We Tackle Large Data Sets? 1997, *ApJ*, 480, 22

Treu, T., Strong Lensing by Galaxies. 2010, *ARA&A*, 48, 87

Trotta, R., Bayes in the sky: Bayesian inference and model selection in cosmology. 2008, *Contemporary Physics*, 49, 71

Trotta, R., Bayes in the sky: Bayesian inference and model selection in cosmology. 2008, *Contemporary Physics*, 49, 49

Troxel, M. A., MacCrann, N., Zuntz, J., et al., Dark Energy Survey Year 1 results: Cosmological constraints from cosmic shear. 2018, *Phys. Rev. D*, 98, 043528

van Daalen, M. P., Schaye, J., Booth, C. M., & Dalla Vecchia, C., The effects of galaxy formation on the matter power spectrum: a challenge for precision cosmology. 2011, *MNRAS*, 415, 3649

Van Waerbeke, L. & Mellier, Y., Gravitational Lensing by Large Scale Structures: A Review. 2003, arXiv e-prints, astro

Vijayakumar, S., D'souza, A., & Schaal, S., Incremental Online Learning in High Dimensions. 2005, *Neural Comput.*, 17, 17

Wang, W., On the Inference of Applying Gaussian Process Modeling to a Deterministic Function. 2020, arXiv e-prints, arXiv:2002.01381

Weinberg, D. H., Mortonson, M. J., Eisenstein, D. J., et al., Observational probes of cosmic acceleration. 2013, *Phys. Rep.*, 530, 87

Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J., Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization. 1997, *ACM Trans. Math. Softw.*, 23, 23