UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede amministrativa: Università degli Studi di Padova
Dipartimento di Fisica e Astronomia "Galileo Galilei"

SCUOLA DI DOTTORATO DI RICERCA IN: ASTRONOMIA
CICLO XXVII

MACHINE LEARNING AND ADVANCED STATISTICS IN ASTRONOMY: TWO APPLICATIONS

Direttore della Scuola: Ch.mo Prof. Giampaolo Piotto

Supervisore: Dott. Andrea Baruffolo

Dottorando: Marco De Pascale

Sommario

Nel campo della spettroscopia e della fotometria, la mole di dati prodotta dalle survey sta aumentando molto velocemente, e continuerà a farlo sempre più nei prossimi anni. Un'analisi che estragga informazioni in tempi utili può essere affidata a metodi automatici sviluppati utilizzando tecniche statistiche e della scienza computazionale. Questo lavoro presenta lo sviluppo e l'applicazione di due metodi automatici. La tesi è divisa in due parti.

La prima parte riporta l'utilizzo dell'algoritmo MATISSE, sviluppato all'Observatoire de la Côte d'Azur, e della pipeline AMBRE per la parametrizzazione di $\sim 126\,000$ spettri prodotti dallo spettrografo ESO:HARPS. I parametri estratti da MATISSE sono temperatura effettiva, gravità, metallicità e abbondanza di elementi α, completi di errori. Il sottoinsieme di parametri che ha superato i criteri di qualità definiti per il campione, è stato confrontato con i risultati di lavori indipendenti mostrando un ottimo accordo. Inoltre, i risultati identificano la grande maggioranza delle stelle come di tipo spettrale G e K, in accordo con il tipo di oggetti osservato da HARPS. Questo conferma MATISSE come un ottimo algoritmo di parametrizzazione.

La seconda parte è dedicata all'analisi di grandi quantità di dati fotometrici. Qui è descritto lo sviluppo di un classificatore di supernovae e la sua applicazione a curve di luce simulate. Il metodo è sviluppato seguendo un approccio detto "data-driven", in cui si cerca di estrarre dai dati tutta l'informazione necessaria a risolvere il problema, affidandosi al minor numero possibile di assunzioni. A questo scopo, il metodo fa affidamento a tecniche del "machine learning", in grado di far apprendere a un computer la regola che trasforma l'input nell'output usando campioni di esempio. Nello specifico vengono utilizzati i processi gaussiani per l'interpolazione delle curve di luce, le "diffusion maps" per la parametrizzazione e le "random forest" per costruire il classificatore vero e proprio. Lo scopo è quello di replicare la classificazione spettroscopica nei tre tipi Ia, Ib/c e II usando solo curve di luce. In questo il metodo fallisce, non riuscendo a classificare le Ib/c in maniera soddisfacente. La causa maggiore è da ricercarsi nell'insieme di esempi disponibili, non rappresentativo della popolazione di supernovae osservata. Invece, confrontato con risultati indipendenti, il metodo presentato risulta competitivo nell'identificazione delle supernovae Ia.

Abstract

In spectroscopy and photometry domains, the amount of data produced by surveys is rapidly increasing, and this trend will continue thanks to next future surveys. To extract information from these data in a useful time scale, the analysis can be done by means of techniques from statistics and computer science. This work presents the development and application of two automatic methods. The thesis is in two parts.

The first part describes the use of MATISSE, a parameterisation algorithm for stellar spectra developed at the Observatoire de la Côte d'Azur, and part of the AMBRE project. It has been applied to $\sim$ 126 000 spectra observed by the ESO:HARPS spectrograph. The parameters extracted by MATISSE are effective temperature, gravity, metallicity and α elements abundance and comes with relative errors. Quality selection criteria have been defined. The accepted subsample of parameters, has been compared with results from independent works, showing very good agreement. Additionally, these parameters identify the great majority of stars as of spectral type G and K, in agreement with the type of targets observed by HARPS. This confirms MATISSE as an excellent parameterisation algorithm.

The second part is concerned with the analysis of large amounts of photometric observations. It describes the development of a supernova classifier and its application to a set of simulated light curves. The method is developed using a *data-driven* approach. The aim is to extract from the data all the information necessary to solve the problem, using the as less assumption as possible. For this purpose, techniques from the machine learning domain are exploited. These techniques are able to make a computer learn the rule transforming input into output using example observations. The machine learning algorithms used are Gaussian processes to perform light curve interpolation, diffusion maps to extract parameters, and random forest to build the classification model. The goal is to reproduce the spectroscopy-based classification in the three classes of type Ia, Ib/c and II, using only light curves. In this respect the method fails, since it is not reliable in classifying type Ib/c. The main cause of this failure is to be found in the set of example light curves, not representative of the observed population of supernovae. On the other hand, when compared with independent results, the method developed results competitive in the identification of supernovae Ia.

Contents

List of Figures

Introduction

In the last decade, the advances in technology have permitted the development of highly automated surveys in many fields of astronomy. One of the most ambitious is the ESA mission Gaia. Mainly devoted to astrometric measurements in the Galaxy, Gaia will provide also spectroscopic and photometric data. All this information will amount to thousands of terabytes.

The same goes for surveys designed to observe transient events: the ongoing Panoramic Survey Telescope and Rapid Response System (PanSTARSS) and Dark Energy Survey (DES) (Bernstein et al. 2009), and the planned Large Synoptic Survey Telescope (LSST) (Ivezic et al. 2008), will produce a huge amount of data.[1]

The data production is thus quickly increasing, and is most likely to increase more with the future surveys. Astronomy is facing an era of data-flooding, where there will be much more data then we are able to analyse with classical methods. The way to deal with this flood, the way in which we can extract scientific information in a short time scale, is using techniques developed in the field of statistics and computer science.

In this framework, in this work are presented two applications, one using spectroscopic data and the other photometric data. The first is the use of an automatic method called MATISSE to determine atmospheric parameters from stellar spectra. The second is the development of a data driven classifier for supernovae using photometric information alone.

Spectroscopic survey of the Milky Way

Many ground based spectroscopic surveys target stars in the Milky Way: RAVE (RAdial Velocity Experiment, Steinmetz et al. (2006)) and SEGUE (Sloan Extension for Galactic Understanding and Exploration, Yanny et al. (2009)) at low resolution, and APOGEE (Apache Point Observatory for Galactic Evolution Experiment, Majewski et al. (2007)) at high resolution. A deep, high resolution survey is GES (Gaia-ESO Survey, Gilmore et al. (2012)), employing the ESO VLT FLAMES instrument. It is designed to complement in resolution and magnitude the information provided by the Radial Velocity Spectrometer aboard Gaia ($R \simeq 7000 - 11500$ down to a magnitude 20 in g band). The data obtained from these surveys will help to trace with great detail the kinematic and chemical history of the Galaxy. This is the field of galactic archaeology: to study the evolution of the Milky Way using information retrieved from stars as if they were fossils. As an example, the chemical abundance ratio of α-elements over Iron, $[\alpha/\mathrm{Fe}]$, provides information on the star formation time-scale.

To extract homogeneous information from spectroscopic data produced by different surveys and different instrument, automatic spectral analysis algorithm have been developed,

[1] `http://pan-starrs.ifa.hawaii.edu`

one of the tools to analyse these spectra is MATISSE. I describe the use of this tool in Part I.

Photometric classification of supernovae

Historically, the classification scheme of supernovae is based on spectroscopy. However, spectroscopy is expensive in term of telescope time, and not all candidates can be classified. This is true in particular for large survey, like SDSS, PanSTARRS, DES and LSST.

On the other hand, planned and ongoing big supernova surveys, produces photometric output in the form of time series of flux measurements, light curves that can provide useful information.

The need for an automatic classifier here is twofold: the number of light curves collected is too high for them to be classified "by hand", and this number is far beyond the spectroscopic facilities capabilities to classify them. There exist several software dedicated to classification of supernovae using photometry. Most of them rely on the use of light curve templates built from previous observations of other objects. One problem in using templates for classification, is that they rule out the possibility of identifying new classes. A big and growing sample gives access to a large variety where what is usually flagged as outlier can actually be identified as a whole new class of objects.

For this reason the approach to the problem has been to use few assumptions, to let data describe themselves. *Machine learning* techniques are special algorithms, built on statistic, and designed to make a computer able to learn from data: as such, they are optimal candidates to a *data-driven* approach.

Using as few assumptions as possible is a way to say there is no prior knowledge on the event producing the light curve. Flux measurements in a light curve are taken on an irregular time grid (or at different time steps). Given two light curves, a classification method has to answer the question: "Are they similar?" or, equivalently, "Are they different?". The first step is to make the light curves comparable one to the other. To do so, the time grid of the two light curves have to be the same. The method has to fit each light curve with some function. To satisfy the condition of having no prior, it has been used a method from *non-parametric statistics*. The technique is called *Gaussian processes* (also known as *kriging* in 2D modelling), and is described in Chapter 7.

The second step is to find a parameter space in which similar light curves form a group (or *cluster*), distinguished from other groups containing light curves of different shapes (represented by different parameters). One possible way to achieve this result is by using *diffusion maps*. Diffusion maps are a way to reduce the dimensions of an arbitrary parameter space, retaining only those that best describe differences between elements in the data set. They are described in Chapter 8.

To build the classification model, a learning algorithm has to be trained on light curves for which the classification is already known by means of spectroscopy (the *training set*). These light curves will act like flags, giving names to clusters identified in the previous step. The learning phase is a delicate issue. The training set should ideally contain an even number of examples for each of the different classes. If this is not the case, the model will learn to identify a particular class better then others. This will increase the possibility to mis-classify an object. In Section 4.4 will be explained why, in practice, when classifying supernovae it is not possible to have such an even training set. The machine learning technique used is called *random forest*. It has been developed by Breiman (2001) and will be explained in Chapter 9.

The final step is to use the learned model to classify all the light curves not used during training. This is a test to assess the goodness of the technique.

The implementation of this classification method is publicly available on `https://github.com/mdepasca/miniature-adventure/`.

Part I

Parameterisation of stellar spectra

Chapter 1

Automatic parameterisation of stellar spectra

1.1 Introduction

In the last decade, astronomy has entered an era of very large data surveys with the scientific goal to expand our understanding of the formation and evolution of the Universe. In particular, several spectroscopic surveys (such as RAVE, APOGEE, SEGUE, Gaia-ESO survey) are dedicated to the study of the Milky Way to comprehend its kinematic and chemical history in detail. The ESA Gaia mission is the pinnacle of all of these spectroscopic surveys during which its Radial Velocity Spectrometer (RVS) will observe tens of millions of stars, for which the radial velocity, atmospheric parameters, and chemical abundances will be determined.

The analysis of such a large quantity of data using 'by-hand' methods is not feasible on a short time scale. This has pushed the astronomical community to develop complex algorithms able to automatically determine the stellar parameters for large spectral datasets efficiently and reliably.

The literature presents several methods to derive stellar parameters from spectra. The common aim is to find stellar parameters (mainly effective temperature, surface gravity, metallicity and individual chemical abundances) defining a synthetic spectrum that is a optimal fit to an observed spectrum. This process of parameter estimation cannot be done analytically, the reason being the complex physics included in models describing stellar atmosphere and in theory of line formation. As a consequence, the parameter estimate is performed using grids built with either synthetic spectra (Kordopatis et al. 2011) or already parameterised observed spectra (such as Prugniel et al. 2007), looking for the synthetic spectrum minimising the distance function, defined by:

$$D = \sum_{j=1}^{J} |O(j) - S(j)|^2 \,, \tag{1.1}$$

where $O(j)$ and $S(j)$ are the observed and synthetic spectra, defined using J variables (such as the spectrum pixels). The common problem that all automated methods can encounter is the non-convexity of the distance function; when secondary minima exist, the method could fail to converge to the absolute minimum (the best solution), thus producing the wrong estimates for the stellar parameters.

As presented by Recio-Blanco (2014), the automatic parameter estimate from stellar spectra can be performed with three different mathematical approaches: optimisation, classification and projection.

The simplest optimisation approach is to explore the whole grid of synthetic spectra to find the nearest neighbour to the observed spectrum. Such approach calculate the distance in Equation (1.1) between the observed spectrum and each of the spectra in the synthetic grid. Although very time consuming, the identification of the absolute minimum is ensured, with a precision limited by the parameter step in the grid. To reduce the computation time, the Nelder-Mead algorithm, as in Allende Prieto et al. (2006), or the Gaussian-Newton algorithm have been used. This comes at the cost of having no guarantee of identifying the absolute minimum. Spectroscopy Made Easy algorithm by Valenti & Piskunov (1996) is a further example of optimisation based methods, since it works by optimising the χ^2.

Methods using the classification approach, handle the parameter estimation problem as a pattern recognition problem. From this point of view, the grid of synthetic spectra is treated as a known set of *patterns*; the aim is to identify, among them, the observed spectra. DEGAS (Kordopatis et al. 2011), is one such method.

Projection methods, to estimate stellar parameters, use a set of so called projection vectors, calculated before application. Each of these vector contains the most important signatures of the flux allowing the derivation of a given physical parameter. The value of each parameter is determined by projecting the observed spectrum on the relative vector. With respect to optimisation methods, this approach leads to a much lower computation time; nevertheless, the method can be trapped in secondary minima. MATISSE (MATrix Inversion for Spectral SynthEsis) is one such algorithm (Recio-Blanco et al. 2006) that has been developed at the Observatoire de la Côte d'Azur (OCA) and is part of the automated pipeline that will analyse and parameterise the spectra from Gaia-RVS.

In this context, the AMBRE project (Archéologie avec Matisse Basée sur les aRchive de l'ESO, de Laverny et al. 2013), is a collaboration between ESO and OCA. It was established to convert the archived spectra collected with the four ESO high-resolution spectrographs FEROS, HARPS, UVES and GIRAFFE, into a comprehensive spectral library of homogeneously determined stellar parameters: effective temperature ($T_{\rm eff}$), surface gravity ($\log g$), metallicity ([M/H]), and the abundance of α-elements versus iron ([α/Fe]). These quantities will be made publicly available to the international scientific community, as advanced data products via the ESO archive. The AMBRE Project has two other main objectives: first, to rigorously test MATISSE on large spectral datasets over a range of wavelengths and resolutions that include those of the Gaia-RVS and, second, to produce a chemo-kinematical map of the Galaxy using the combined ESO archive samples to unravel galactic formation and evolution.

The first part of the AMBRE project consisted of the analysis of the $\sim$ 6 500 FEROS archived spectra. This has been presented in Worley et al. (2012), and the parameters are now publicly available. What follows appeared in De Pascale et al. (2014) which reports the work I carried out on the $\sim$ 126 000 HARPS archived spectra provided by ESO. The analysis of $\sim$ 52 000 UVES spectra has also been performed and submitted by Worley et al. (2015), whereas the GIRAFFE spectra analysis is in progress.

My contribution to the presented work has been in the adaptation of the already existing AMBRE:FEROS analysis pipeline (see Worley et al. 2012) to the HARPS spectra. This consisted both in a speed up of the computation and in data analysis. I achieved the speed up by exploiting the radial velocity information provided by ESO within the header of the

majority of HARP spectra (see Section 1.3). I also implemented an embarrassingly parallel code in the `bash` language to distribute the pipeline's computation over a number of nodes in the OCA computing cluster SIGAMM. Such a kind of parallelisation does not require any communication between two processes, and it has been possible since the parameterisation of one spectrum is a completely independent process from the parameterisation of a second spectrum.

My contribution on the data analysis has been in identifying the wavelength regions suitable for MATISSE analysis, and the calculation of the FWHM to be used to smooth both synthetic and observed spectra (Section 1.4.1). Also, I defined the S/N thresholds for spectra rejection after parameterisation (Section 1.4.3). Finally I carried out the external error calculation, comparing AMBRE:HARPS results with two independent parameterisations (Section 2.2).

The chapter is organised as follows: in Section 1.2, I describe the data set and its properties, with the adaptation of the AMBRE:FEROS analysis pipeline to the AMBRE:HARPS sample and the derivation of the radial velocities. In Section 2.1 I explain how I used a sample of stars with repeated observations to determine the internal error. Section 2.2 describes the external errors estimates by the comparison of key samples with literature parameter values. Finally, Section 3 presents the parameterisation of the accepted AMBRE:HARPS spectra with their delivery to the ESO archive, and I conclude in Section 3.3 with a summary.

1.2 The AMBRE analysis of the HARPS spectra

The High Accuracy Radial velocity Planet Searcher (HARPS Mayor et al. 2003) is a fiber-fed, cross-dispersed echelle spectrograph that was built by a consortium of four institutes: the *Observatoire de Genève*, the *Observatoire de Haute Provence*, the *Universität Bern*, and the Service d'Aéronomie of CNRS in collaboration with ESO. It was installed and commissioned on the ESO 3.6m Telescope at La Silla, Chile, in 2003[1]. The HARPS high resolution ($R \simeq 120\,000$) spectra and the long term instrument stability ensure a radial velocity accuracy of about 1 m s^{-1} (Lo Curto 2011), making HARPS the prime facility for exoplanet hunting.

The AMBRE analysis of the HARPS spectra comprises of the spectra observed from October 2003 to October 2010. They have been homogeneously reduced by ESO with the HARPS pipeline and made publicly available through the ESO archives. This sample was delivered to OCA from the ESO archive department and it includes calibration and science spectra. The ESO:HARPS pipeline produces different types of science spectra: the extracted 2-Dimensional spectra, where each line contains the extracted flux from one spectral order; the extracted 1-Dimensional spectra, which contains the re-binned and merged spectral orders; and the radial velocity cross correlation function (CCF), which is computed between each order and a template mask. For the AMBRE analysis, we used the 126 688 extracted 1-Dimensional science spectra, of which about 85% of these spectra have a corresponding CCF spectrum, and thus a radial velocity ($V_{\rm rad}$) estimate. For the remaining 15%, we calculated the $V_{\rm rad}$ using the AMBRE automatic program (see Worley et al. 2012, Section 4.2 and Section 1.3 of the present paper).

This AMBRE:HARPS sample of 126 688 spectra is composed of several thousands of distinct stars with some of them being observed a few tens of times. We have found that the

[1] `http://www.eso.org/sci/facilities/lasilla/instruments/harps/index.html`

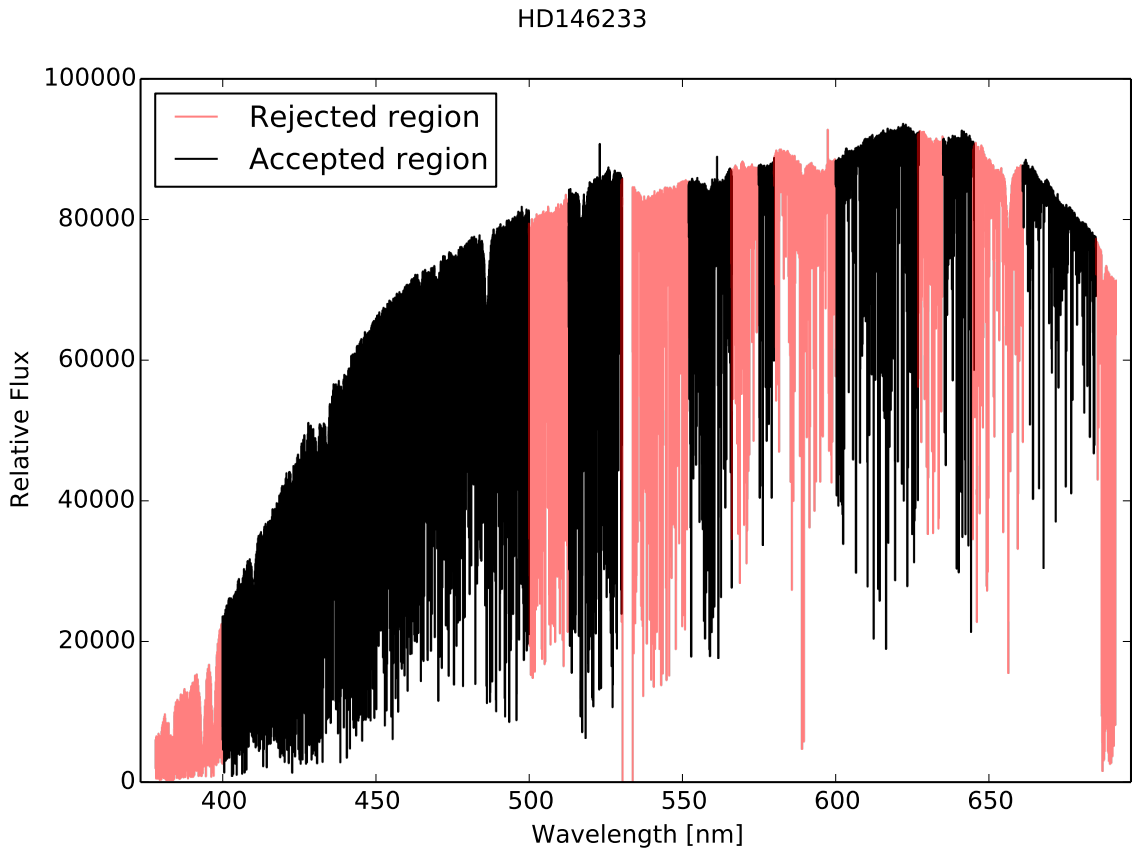

Figure 1.1: Raw 1-D spectrum of HD146233 (18Sco) observed by HARPS in 2004. At 396.8 nm and 393.4 nm is the Ca H&K doublet. Hα and Hβ lines are clearly visible, at 656.3 nm and 486.1 nm. The drop in flux around 530 nm is due to the gap between the two CCDs detectors of the instrument. The spectrum will be shifted for radial velocity and normalised so to determine the parameters using MATISSE. The meaning of the color coding is described in Section 1.4.1.

object ID available in the file header was not always a reliable indicator; thus, we have determined the total number of distinct stars included in this sample by performing a coordinate matching analysis whereby a maximum distance radius was imposed. Within a radius of $r \simeq 5''$, we obtain 17 218 distinct stars. The counts of the number of science spectra and, thus, distinct objects that have been observed by HARPS in the given period are reported in Figure 1.2 with the number of spectra with $V_{\rm rad}$ calculated by the ESO:HARPS pipeline.

Figure 1.3 shows the distribution of the repeated observations, which shows the number of different spectra available for the same star within the AMBRE:HARPS sample. About $\sim$ 40% of the stars were observed only once, and $\sim$ 95% of the sample less than 20 times. Within the search radius of $\sim$5 $''$, we have found that only 182 targets have been observed more then 110 times with a maximum of 1 211 repeated observations for Procyon AB binary system.

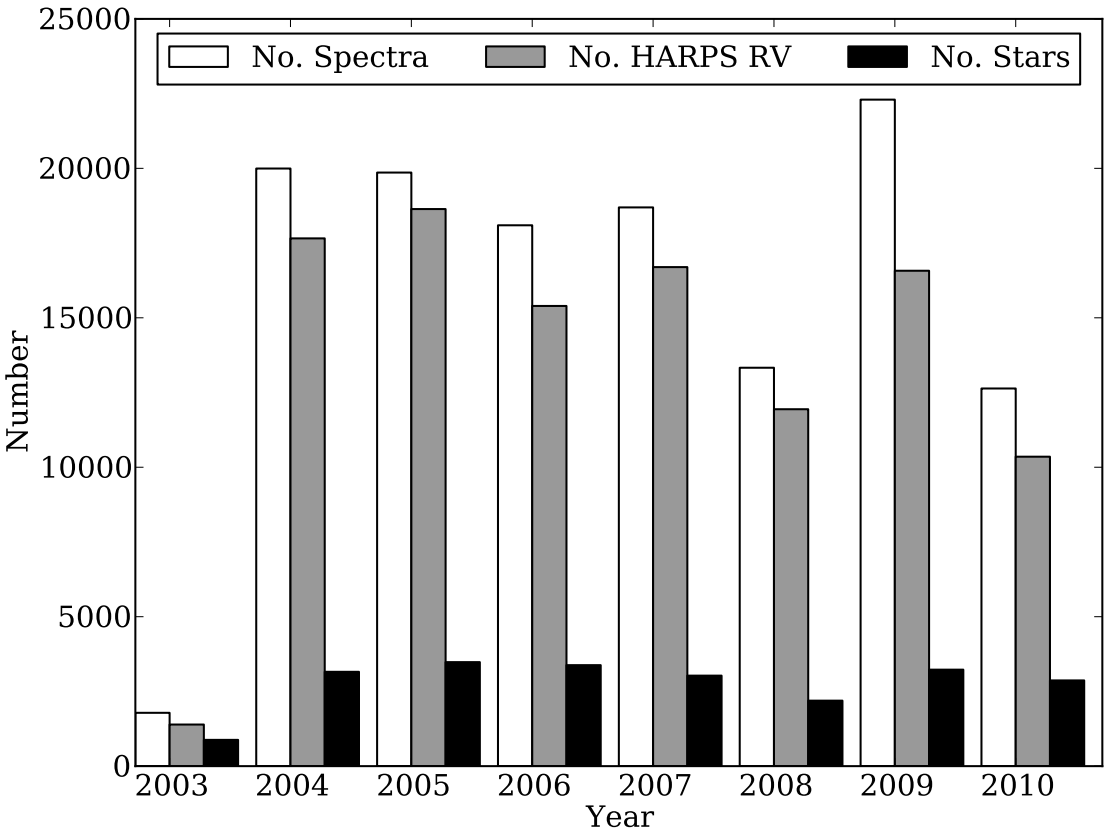

Figure 1.2: Number of HARPS spectra per year analysed by the AMBRE Project (in white). The number of spectra for which a $V_{\rm rad}$ was calculated by the ESO:HARPS pipeline are shown in gray. The number of different stars observed per year are shown in black.

1.3 Radial velocity

To analyse the observed spectra with MATISSE, it is necessary to correct them by the radial velocity of the star. Since the main scientific goal of HARPS is the search for exoplanets by measuring variations in the radial velocity of the host star, the data reduction pipeline (DRS) of HARPS determines the radial velocities with extremely high accuracy ($\sim$1 m s^{-1}). Radial velocity error estimates are also provided. Given the large number of spectra to be analysed and for the purpose of homogeneity with the ESO archives, we chose to adopt the radial velocity and the associated error provided in the header of the delivered reduced spectra when available.

However, as previously mentioned, among the sample of spectra delivered to OCA, approximately 15% do not have a HARPS:DRS radial velocity. Almost all of this subsample of spectra have a radial velocity set to a default value from the DRS, while a tiny fraction ($\sim$ 2%) have calculated values that are larger than 500 km s^{-1} in modulus. This indicates that the DRS radial velocity routine had most likely not converged. We therefore calculated the $V_{\rm rad}$ for this subsample of spectra using the AMBRE pipeline (see Worley et al. 2012, Section 4.2). Briefly, the radial velocity routine performs a cross-correlation between each spectrum and a set of 56 synthetic masks specifically computed for the AMBRE:HARPS sample. For each

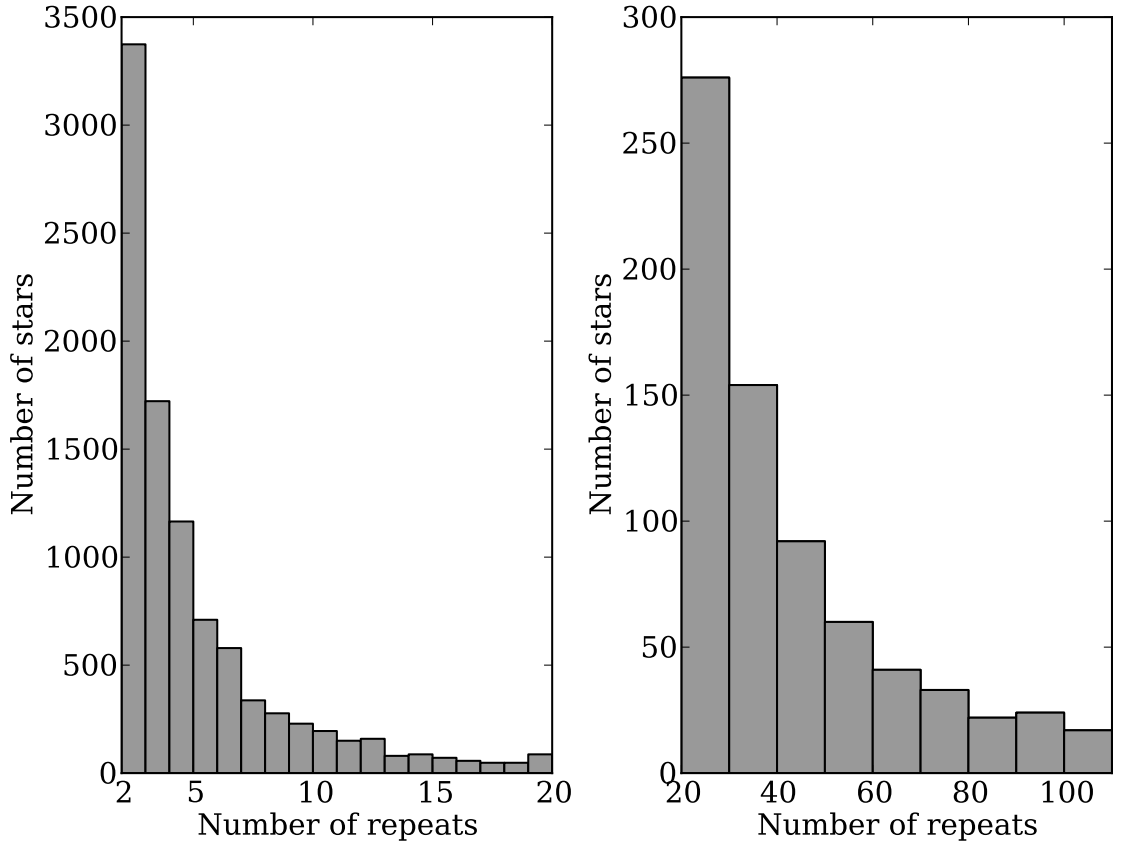

Figure 1.3: Approximate number of stars in the sample as a function of the number of observations. The total number of observed stars is 17 218. *Left panel*: Distribution of the repeats for stars observed between 2 and 20 times. *Right panel*: Similarly for stars observed more than 20 times (see text for more details).

spectrum, the output is therefore a set of 56 radial velocity determinations with associated errors; the radial velocity with the lowest error has been adopted. We point out that a final check on the validity of the adopted $V_{\rm rad}$ has been performed once the analysed spectra have been parametrized. We indeed always found that the atmospheric parameters estimated for a HARPS spectrum and those of the adopted mask agree with each other.

The AMBRE $V_{\rm rad}$ procedure also gives an estimate of the uncertainty associated with the derived radial velocity. To check the consistency of these results with the HARPS $V_{\rm rad}$, we also determined the radial velocity for a sample of spectra having a $V_{\rm rad}$ computed by HARPS:DRS. For that purpose, we considered a sample of ∼17 000 HARPS spectra, which were observed in 2004.

This comparison between the HARPS:DRS $V_{\rm rad}$ and the $V_{\rm rad}$ derived by AMBRE is shown in Figure 1.5. The distribution of the residuals between the two estimates in the bottom panel of Figure 1.5, shows a Gaussian distribution centered close to zero ($\langle V_{\rm rad}^{\rm HARPS} - V_{\rm rad}^{\rm AMBRE} \rangle = -0.13$ km s^{-1}). Additionally, 89% of the spectra have an absolute value of the residual, which is smaller than 1 km s^{-1}, whereas 96% of them have a difference smaller than 2 km s^{-1}. This test confirms the high enough consistency between the HARPS $V_{\rm rad}$ and the AMBRE $V_{\rm rad}$, such that the use of the AMBRE $V_{\rm rad}$ for those spectra without a HARPS $V_{\rm rad}$ still provides

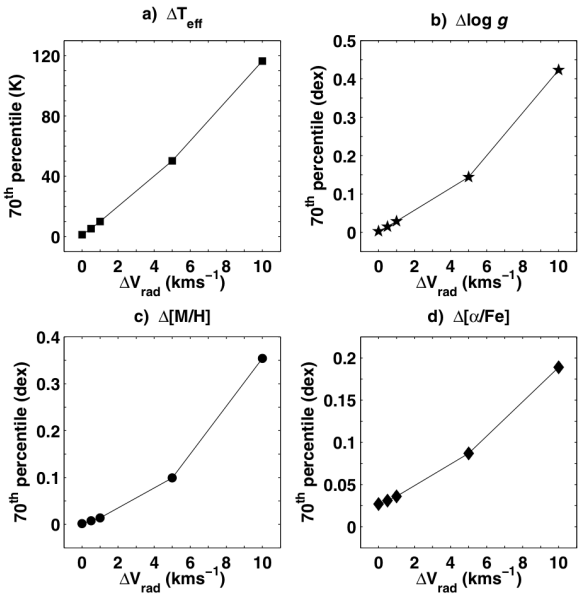

Figure 1.4: Internal error for each parameter as function of uncertainty on the $V_{\rm rad}$. Panel a reports the $\Delta T_{\rm eff}$ that the 70% of the synthetic sample were less than or equal to when calculating the difference between the nominal $T_{\rm eff}$ and the $T_{\rm eff}$ determined for the respective $V_{\rm rad}$ uncertainty. Panel b as for a but for $\log g$. Panel c as for a but for [M/H]. Panel d as for a but for [α/Fe]. Figure 11 from Worley et al. (2012).

a homogeneous parameterisation analysis across the entire sample. Moreover, with reference to (Worley et al. 2012, Figure 11), we note that an error of 2 km s^{-1} on the radial velocity has very little effect on the determination of the atmospheric parameters.

In summary, we therefore adopted the radial velocity and associated error provided by HARPS:DRS for the AMBRE analysis whenever possible. When these quantities were not available, we calculated them using the AMBRE procedure that has been shown to be consistent with the HARPS:DRS results.

1.4 The AMBRE:HARPS parameterisation pipeline

For the analysis of the HARPS spectra, we started from the pipeline that was developed for the AMBRE:FEROS analysis, which is described in detail in Worley et al. (2012). Due to the inherent differences between the two instruments in configuration (principally resolution and spectral range) and in the ESO reduction pipeline products (the radial velocity is provided for each HARPS spectra, for instance), the AMBRE:FEROS pipeline was adapted to obtain an optimal analysis for the HARPS spectra. These modifications are highlighted in the following subsections.

1.4.1 Adaptation of the HARPS spectra for the AMBRE analysis

The parameterisation of the stellar spectra in AMBRE is performed by a kind of comparison of the observed spectra with a library of synthetic spectra using the MATISSE algorithm (Recio-Blanco et al. 2006). We remind that MATISSE is a local multi-linear regression method. It acts as a projection method in the sense that the input observed spectra are

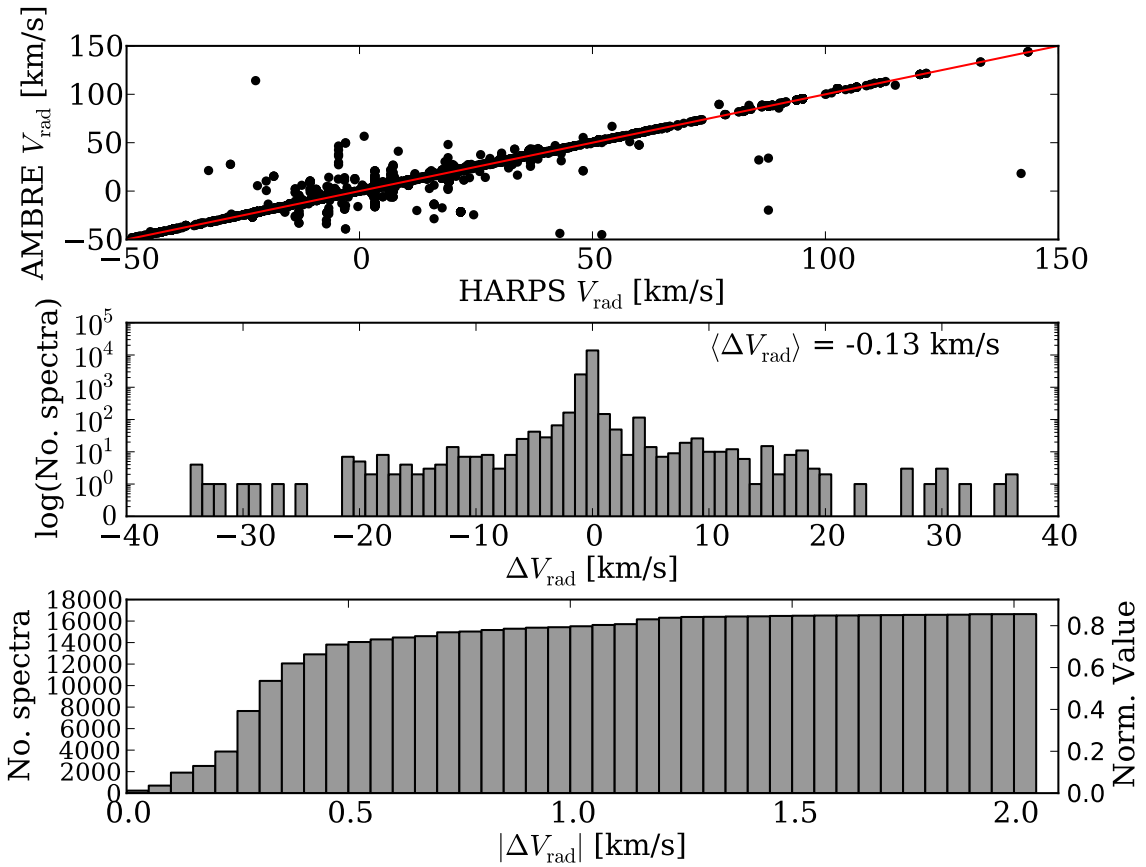

Figure 1.5: *Top panel*: Comparison between the radial velocity as calculated from the AMBRE radial velocity program and from the HARPS pipeline for the 2004 sample ($\sim$ 17 000 spectra). The distribution is Gaussian. *Middle panel*: Distribution of $\Delta V_{\rm rad}$ between HARPS and AMBRE radial velocities using a logarithmic scale. *Bottom panel*: Cumulative distribution of the $|\Delta V_{\rm rad}|$ in bins of 0.05 km s^{-1}. Almost 70% of the spectra have a $|\Delta V_{\rm rad}|$ smaller than 0.6 km s^{-1}. Moreover, almost 90% of the spectra are found between $\Delta V_{\rm rad} = \pm 1$ km s^{-1}.

projected onto a set of vectors derived during a learning phase of MATISSE. These vectors are a linear combination of reference spectra (i.e. the synthetic spectra described below) and could be roughly viewed as the derivatives of these spectra with respect to the different stellar parameters. We point out that we adopted exactly the same version of MATISSE for the analysis of the FEROS and HARPS spectra (see comment in Section 3.1).

Moreover, the same grid of synthetic spectra, the AMBRE grid, has also been adopted. Shortly, this grid of $\sim$17 000 high-resolution synthetic spectra has been computed from MARCS model atmospheres (Gustafsson et al. 2008), taking into account the most complete atomic and molecular linelists. The spectra cover the whole optical domain for cool to very cool stars of any luminosity (from dwarfs to supergiants) with metallicities varying from 10^{-5} to 10 times the Solar value. Large variations in the chemical composition of the α-elements with respect to iron have also been considered. As in the MARCS models, a constant microturbulent velocity has been adopted for dwarfs (1 km/s) and giants (2 km/s). More details on the properties of the AMBRE grid and how it has been computed can be found in de Laverny et al. (2012).

Table 1.1: Selected HARPS wavelength domains for the AMBRE analysis.

Region	λ Min (nm)	λ Max (nm)
1	400.0	500.0
2	513.0	530.0
3	552.0	566.0
4	575.0	580.0
5	600.0	627.0
6	635.0	645.0
7	661.0	685.0

Notes. These listed wavelength intervals are not (or are weakly) polluted by absorption and telluric features and do not contain the gap present between the two CCDs, the lowest S/N regions, nor the regions of the Ca II H and K lines.

The wavelength coverage of the AMBRE synthetic spectra grid goes from 300 nm to 1 200 nm, which is the whole optical wavelength domain. We, thus, were able to select only those wavelengths corresponding to the HARPS wavelength domain that were useful for the analysis. HARPS disperses light on 68 orders covering the spectral range between 378 nm and 691 nm with a gap from 530 nm to 533 nm due to the two CCDs that form the detector system of the instrument. Since we analysed the extracted 1-Dimensional spectra, we first discarded the blue and red edges of the CCDs where the signal-to-noise (S/N) can be significantly lower with respect to the remainder of the spectrum.

Then, from the two wavelength domains defined by the two CCDs, we rejected sections that contained sky absorption and telluric features. In addition, we rejected the very broad Ca II H & K lines, since they can be poorly synthesised for some parameter combinations and they are difficult to normalise automatically. The accepted wavelength regions for the AMBRE:HARPS analysis are listed in Table 1.1 and displayed in red in Figure 1.1.

A further refined selection of the wavelength ranges was then performed by comparing the observed normalised spectra of the Sun and Arcturus line by line (the two stars are representatives of standard dwarf and giant stars and taken from (Wallace et al. 1998; Hinkle et al. 2000) with the corresponding synthetic spectra in the AMBRE grid. Lines at matching rest wavelengths were rejected when the percentage difference between their two fluxes was larger then a fixed threshold. After testing, we selected a threshold of 0.05% for the Sun and 0.15% for Arcturus. Both thresholds had to be met for the spectral line to be included. These thresholds removed obvious mismatches between the observed and synthetic spectra. A higher threshold was set for Arcturus since the underlying physics of giants is less understood than it is for the Sun and hence the spectra of giants are not as well synthesised. This threshold is also allowed for greater inaccuracies in normalisation between the Arcturus and its corresponding synthetic spectrum.

This procedure allowed us to define the final list of selected wavelengths for the AMBRE:HARPS parameterisation. It consists of $\sim$ 500 intervals sampling the previously selected HARPS domains reported in Table 1.1 and spanning a total range of about 147 nm. Using this final list of wavelength intervals, we extracted the corresponding ranges from the grid of synthetic spectra resulting in the AMBRE:HARPS synthetic spectra grid.

As the resolution and pixel sampling of HARPS are very high, the computing time required

to analyse the original spectra is also correspondingly high. As for the AMBRE:FEROS analysis, the resolution and pixel sampling can be lowered to optimise computing time but without sacrifying the key spectral informations with the goal being to keep an as good as possible accuracy on the derived stellar parameters. This has been explored in the previous AMBRE:FEROS analysis and was have found that a resolution $R \sim 15\,000$ was sufficient to achieve the required accuracy.

Convolution of spectra by FWHM

The HARPS spectra have a constant resolution ($R = \lambda/\Delta\lambda$) and hence a varying $\Delta\lambda$ with λ (where $\Delta\lambda$ is the full-width-at-half-maximum (FWHM) of the spectral feature at λ). This is contrary to the synthetic spectra, which have been computed without any instrumental nor macroturbolence profiles, and a constant wavelength sampling of 0.001 nm.

By degrading (or convolving) both the synthetic and observed spectra to a lower resolution and mapping the observed FWHM profile onto the synthetic FWHM profile, computation time can be decreased and the comparison between the two sets of spectra is then consistent. The convolution was performed by "smoothing" the spectra with a Gaussian for which the FWHM was greater (and therefore of lower resolution) than that of HARPS.

The synthetic grid was convolved with a Gaussian of FWHM=0.02218 nm to produce a synthetic grid with resolution less than that of HARPS.

For each observed spectrum, the measured variation of $\Delta\lambda$ as a function of λ was interpolated to a linear function providing a uniformly increasing FWHM profile for each spectrum (where possible). Similarly, for a subsample of synthetic spectrum in the grid, their FWHM profile was also measured by the same procedure, confirming the constant FWHM profile. The mean value from the combination of all of these synthetic FWHM profiles was found to be $\mathrm{FWHM}_{\mathrm{mean}} = 0.022$ nm, as expected from the grid convolution, confirming that our procedure is valid. This was taken to be the nominal synthetic grid FWHM.

To map the convolution of the observed onto the synthetic for each bin in λ, the FWHM of the smoothing Gaussian was calculated using $\mathrm{FWHM}_{\mathrm{mean}}$ and the observed was linearly interpolated FWHM for that bin. This resulted in the convolved observed spectra having a constant FWHM profile of the same resolution as the convolved synthetic grid.

The final wavelength sampling was chosen to fulfill the Shannon criterion resulting in a sampling of 0.0085 nm/pixel. The same sampling was also applied to the synthetic spectra of the AMBRE:HARPS grid from which we generated the learning functions of MATISSE to parameterise the observed spectra. We remind that the broadening caused by the micro-turbulent velocity is dominated by the adopted one for the analysis. In consequence, this parameter is not estimated by the pipeline that relies on the adopted constant micro-turbulent velocity of the synthetic grid. We also do not derive the projected rotational velocity and reject every spectra having a too large line-broadening that could affect our parameter estimates (see the discussion in Worley et al. 2012).

1.4.2 Spectral processing A, B and C

In Figure 1.6, a description of the several steps that are part of the AMBRE:HARPS pipeline with a graphical representation of the process is reported. In this Section, the three main stages of the AMBRE pipeline are reviewed. The reader is referred to the AMBRE:FEROS analysis for more detail.

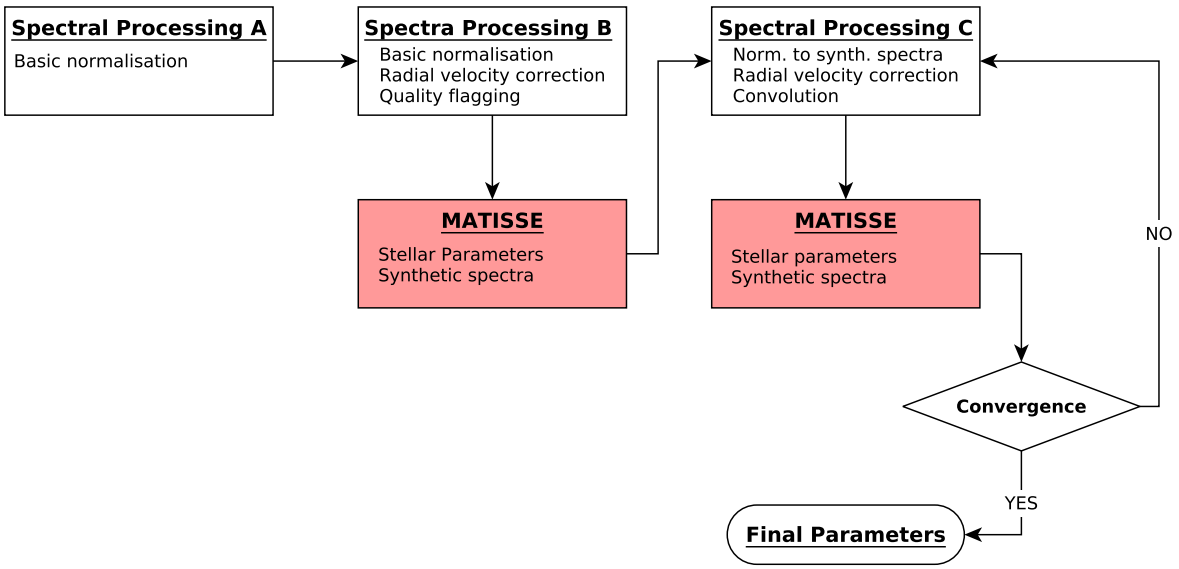

Figure 1.6: The AMBRE:HARPS analysis pipeline. The key stages are displayed in order of analysis. Coloured boxes highlight parts of the pipeline involving the use of MATISSE algorithm.

First, it is worth to remind that the HARPS spectra delivered for analysis by ESO are the products of the standard data reduction pipeline. As a consequence, some degradations could still be present in some spectra as, for instance, some possible contaminations by the wavelength calibration lamp for the low S/N ones (about 7% of the whole AMBRE:HARPS sample has S/N < 20). To keep the analysis of the whole sample as homogeneous as possible, such effects will be ignored in the following.

Spectral Processig A

In spectral processing A (SPA), the observed spectra are prepared for radial velocity computation, if this is needed (see discussion in Section 1.3), and spectral FWHM measurements, and a first quality check is performed identifying noisy or problematic spectra. Each spectrum is sliced in the wavelength regions defined in Table1.1 and roughly normalised to unity. Previously in Worley et al. (2012), the spectral FWHM was measured during the next stage. However, as this was heavy in computing time, it has been developed as a standalone routine for AMBRE:HARPS and AMBRE:UVES that can be run in parallel to the radial velocity routine on the normalised spectra of SPA. The radial velocity program was used only on the spectra for which the HARPS radial velocity was not available, and the radial velocity testing sample (see Section 1.3). During SPA, the S/N of the spectra was calculated (since it was not reported in the header of the reduced spectra), as done in the AMBRE:FEROS analysis.

Spectral Processig B

Spectral processing B (SPB) is the stage at which the first estimate of the stellar parameters is made using MATISSE. The observed spectra are, thus, normalised and convolved to be consistent with the AMBRE:HARPS synthetic grid (see discussion in Section 1.4.1). At this stage, there is still no determination of the atmospheric parameters; thus, the spectra are normalized to unity. A key point is also the flagging and rejection of problematic spectra;

spectra flagged as "bad normalized", "noisy" or "missing wavelength" are rejected before the subsequent stage. We point out that, contrary to the SPB of Worley et al. (2012), no iteration were performed for the HARPS spectra in SPB since the SPB estimate of the stellar parameters was always close to the final solution.

Spectral Processig C

Finally, rather than normalisation to unity, the normalisation of each spectrum in Spectral Processing C (SPC), is performed on a synthetic spectrum to better represent the continuum placement of the star. In the first instance, the normalisation is performed on the synthetic spectrum generated at the stellar parameters of the solution found in SPB. The resulting normalised spectrum is then analysed in MATISSE, which provides yet another solution and corresponding synthetic spectrum. Thus, SPC consists in iterating between normalisation and parameterisation, ultimately converging on the final stellar parameters, the final normalised spectrum and the final synthetic spectrum. A quality flag is produced, based on a χ^2 fit between the observed spectrum and synthetic spectrum at the determined stellar parameters.

1.4.3 Rejection criteria

The final parameters delivered to the ESO archive are a subsample of the entire set of parameters estimated by AMBRE:HARPS pipeline due to quality selection based on several rejection criteria. As in Worley et al. (2012, Section 7), the first set of rejection criteria that were applied were based on the radial velocity error ($\sigma_{V_{\rm rad}}$), the FWHM of the $V_{\rm rad}$ CCF, the S/N, and the quality of the fit of the normalised spectra to the synthetic spectra (χ^2):

- Following the rejection procedure of the AMBRE:FEROS data, all HARPS spectra with $\sigma_{V_{\rm rad}} > 10$ km s^{-1} were rejected as corresponding to large uncertainties in the parameter determination, see Worley et al. (2012, Section 3 & 5.2) and Figure 1.4. Specifically, for such large errors on the radial velocity, the uncertainties would be greater than ~ 120 K in $T_{\rm eff}$, ~ 0.4 dex in $\log g$, ~ 0.35 dex in [M/H] and ~ 0.17 dex in [α/Fe]. This step resulted in the rejection of 14 105 spectra (11% of the total sample).

- With reference to Worley et al. (2012, Section 7.2.3), all spectra with a FWHM of the CCF larger than 20 km s^{-1} (hot/fast rotating stars) were rejected. Such a threshold value was also chosen on the basis of the results obtained by Gazzano et al. (2010). It excluded another 5 199 spectra from the final results (4% of the total).

- Also excluded was every spectra having a S/N smaller than 10, since the parameterisation of such spectra are associated with rather large internal errors (see Section 2.1).

- Finally, to apply a rejection criterion based on the S/N of the HARPS spectra with the quality of their parameterisation, I first divided the remaining set of spectra into three different temperature domains:

  - hot stars ($T_{\rm eff} > 6\,500$ K),
  - warm stars ($5\,000$ K $< T_{\rm eff} \leq 6\,500$ K),
  - cool stars ($4\,000$ K $\leq T_{\rm eff} \leq 5\,000$ K).

For each of the three temperature domains, the S/N threshold used to reject spectra was determined by fitting a second degree polynomial to the distribution of χ^2 as a function of the S/N. This χ^2 corresponds to the sum of the squared differences between the synthetic and observed fluxes performed at every pixel:

$$\chi^2 \sim \sum_{\mathrm{px}} \left(F_{\mathrm{syn}} - F_{\mathrm{obs}}\right)^2 . \tag{1.2}$$

I investigated the effect of applying different thresholds in the three temperature domains by looking at the distribution of the rejected spectra in the HR diagram. The optimal selection for cool stars was obtained by retaining all the spectra below 0.5 times the standard deviation above the fit for cool stars. Similarly, we kept the warm star spectra by having a χ^2 smaller than three times the standard deviation above the fit (see Figure 1.7 and Table 3.2) and finally, every hot star spectra located below the fit in the S/N-χ^2 space. This last stage resulted in the rejection of 6 340 spectra (5% of the total).

As last rejection criterion, I applied the restrictions imposed by the synthetic grid boundaries, particularly excluding stars cooler then 4 000 K, since the determination of gravity was uncertain in the temperature range between 3 000 K and 4 000 K. The adopted grid limits are as follows:

- 4 000 K $\leq$ T_{eff} $\leq$ 7 625 K;
- 1 dex $\leq \log g \leq$ 5 dex;
- -3.5 dex $\leq$ [M/H] $\leq$ 1 dex;
- -0.4 dex $\leq$ [α/Fe] $\leq$ 0.4 dex if [M/H] $\geq$ 0.0 dex;
- -0.4 dex $\leq$ [α/Fe] $\leq$ 0.8 dex if -1.0 dex $<$ [M/H] $<$ 0.0 dex;
- 0.0 dex $\leq$ [α/Fe] $\leq$ 0.8 dex if [M/H] $\leq$ -1.0 dex.

After the application of these rejection criteria, stellar parameters for 93 116 spectra were accepted (i.e. $\simeq$73% of the initial sample). The stellar parameters derived for this AMBRE:HARPS sample of spectra are described in Chapter 3.

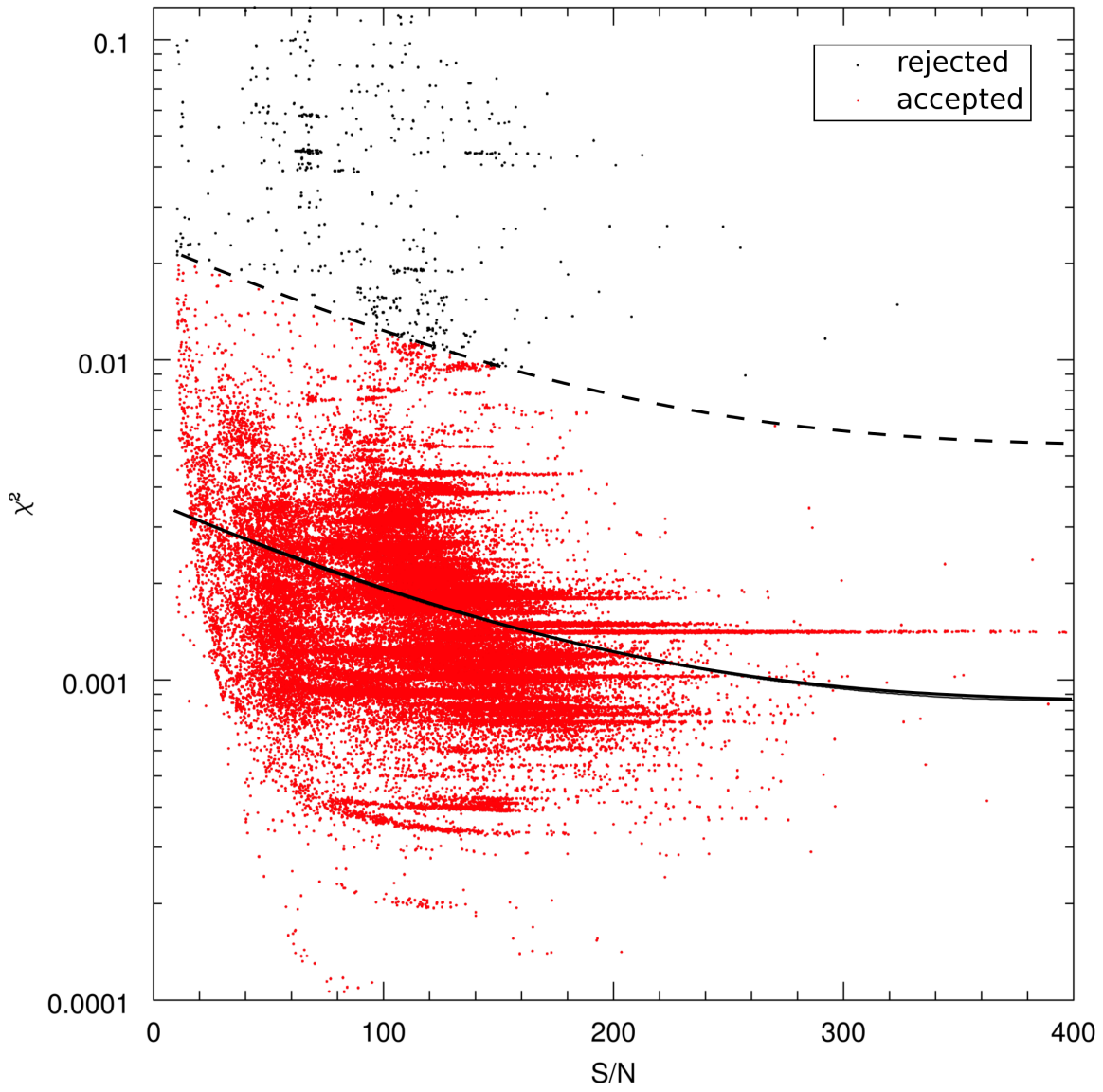

Figure 1.7: Spectra selection on the basis of the χ^2 quality criterion as function of S/N for stars with $5\,000\ \mathrm{K} < \mathrm{T_{eff}} \leq 6\,500\ \mathrm{K}$. The solid line traces the second degree polynomial fit to those spectra. The red points are the selected spectra, starting from a threshold of three times the standard deviation above the fit.

Chapter 2

Error analysis

The AMBRE:HARPS pipeline has been adapted from the AMBRE:FEROS pipeline as described in Chapter 1. The adaptation are mainly due to the higher resolution and wider wavelength range of HARPS. These two features, with respect to AMBRE:FEROS, provided enough information to allow a faster Spectral Processing B phase. Also, most of the FITS files containing the spectra provided the value of the radial velocity of the star; with this information much of the computation time devoted to radial velocity calculation has been avoided.

The stellar parameters produced as by AMBRE:HARPS pipeline have to come with associated errors. In this respect, it is possible to distinguish between internal and external error. The internal error is calculated by checking how uniform are the values of each parameter when calculated from different spectra of the same star.

The values for the external errors are defined through comparison of AMBRE:HARPS parameters width independent results obtained by different methods on a common sample of stars.

This chapter explains in detail how the values of these two kind of errors are calculated.

2.1 Internal error analysis

An estimate of some of the contributions to the internal errors associated with the analysis can be provided by injecting a large sample of interpolated (at random stellar parameter values) noised synthetic spectra with different uncertainties in radial velocity (see Worley et al. 2012, Section 5) into the pipeline. We point out that we only test the performances of the MATISSE method itself here by checking its ability to retrieve the atmospheric parameters in a very ideal case, since only a Gaussian white noise is assumed, and any possible mismatch between the synthetic and real spectra are assumed to be negligible. Since synthetic spectra that are almost the same spectral resolution and spectral coverage were used for AMBRE:HARPS as for AMBRE:FEROS, the error analysis of AMBRE:FEROS can be considered as valid for the analysis of the AMBRE:HARPS spectra. As shown in Worley et al. (2012):

- the behaviour of the 70^{th} percentile of the internal error of each atmospheric parameter as a function of the S/N show that for S/N > 10 the internal errors are negligible and, thus, have almost no effect on the determined parameters, see Figure 2.1. Since only the HARPS spectra with S/N > 10 have been retained, this argument is valid for the present parameterisation.

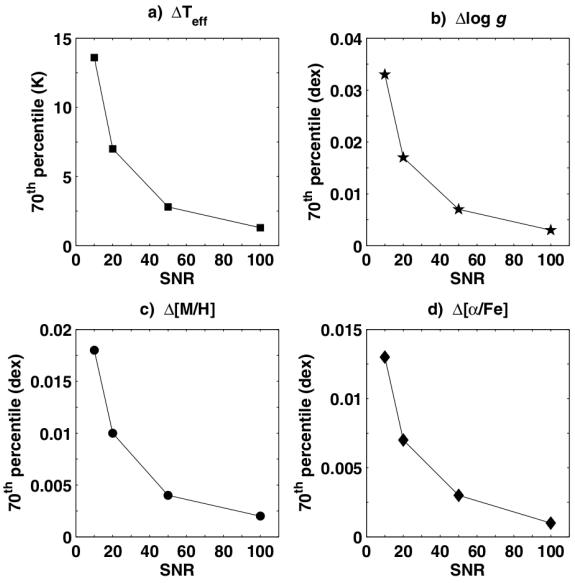

Figure 2.1: Internal error for each parameter with changes in S/N. Figure 10 from Worley et al. (2012).

- The error on radial velocity for the accepted sample has a small effect on the determination of the stellar parameters. This conclusion has been reached on the basis of Figure 1.4 and Figure 2.2. In Figure 1.4, the behaviour of the 70$^{\text{th}}$ percentile of the variation in each photospheric parameter calculated from the same spectrum as a function of artificial variations in the value of V_{rad} is reported. There, it is shown that the 70$^{\text{th}}$ percentile value for each parameter is relatively small for $\Delta V_{\text{rad}} < 10$ km s^{-1} (even smaller if ΔV_{rad} is limited to 6 km s^{-1}). Figure 2.2 shows that almost all the spectra ($\simeq 99\%$) that passed the rejection criteria described in Section 1.4.3 have a $\Delta V_{\text{rad}} < 6$ km s^{-1}. Putting together this result with what reported above, the conclusion is to neglect hereafter the contributions of V_{rad} uncertainties and low-quality spectra to the internal errors, since every spectra with $S/N < 10$ and error in V_{rad} greater than 10 km s^{-1} has been rejected.

The other possible sources of internal errors (and particularly any possible mismatch among observed spectra, real spectra and the effects of the real noise of the spectra) have been investigated by considering a second independent method. It consisted in estimating the internal error of the AMBRE:HARPS analysis by comparing the parameterisation of the repeated observations of the same star, which are characterized by different S/N and uncertainty on the radial velocity.

For that purpose, the sample of 93 116 parameterised spectra were investigated to identify those stars, which had been observed by HARPS at least 50 times, adopting a coordinate search radius of 5″. Using the web tool SIMBAD, I identified and then excluded from this repeat sample the observations of multiple stellar systems and variable stars, since the stellar parameters of these objects could vary with time.

This resulted in a sample of 61 313 spectra that corresponded to 6 094 distinct stars observed 50 times or more as per the radius search. For each set of repeated observations, the mean value of each stellar parameter and the deviations from the mean values for each

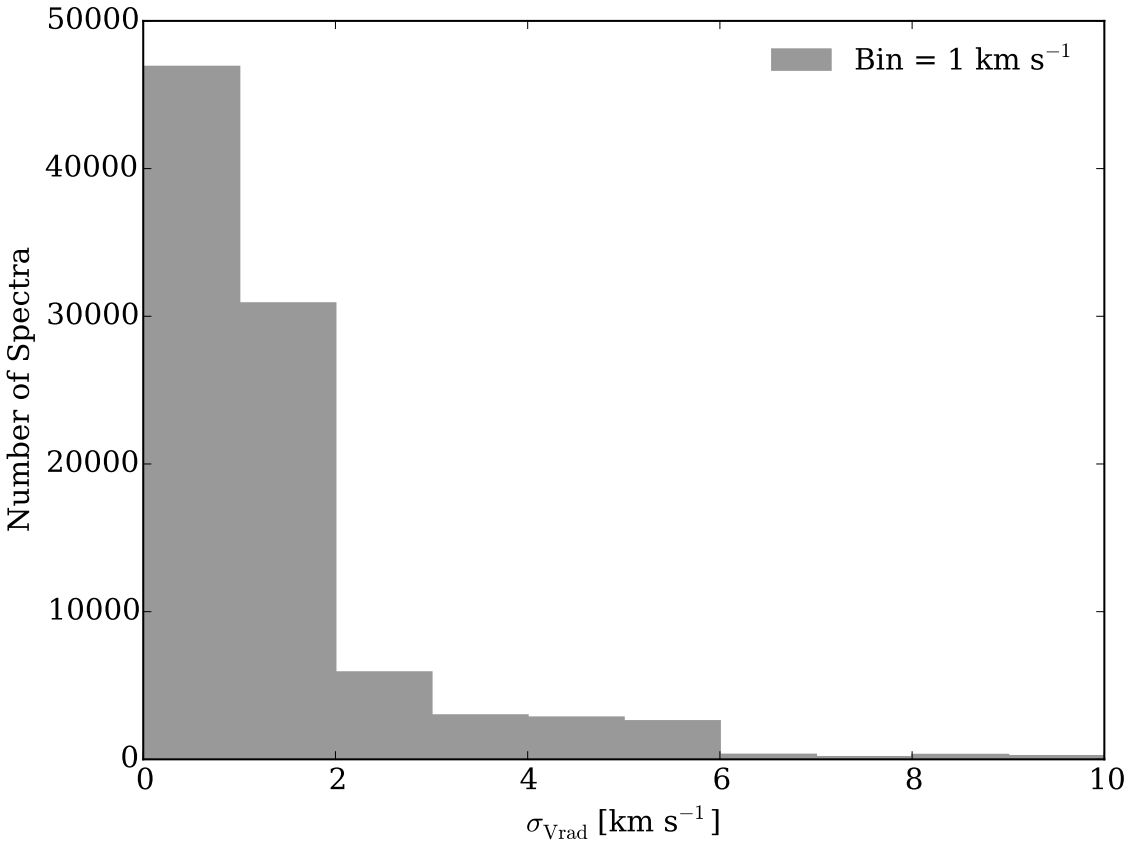

Figure 2.2: Histogram of measured V_{rad} uncertainty for each of the HARPS spectra that satisfied the rejection criteria. The errors come from the HARPS:DRS pipeline or the AMBRE radial velocity determination routine, as appropriate.

spectrum was calculated. In this way, I ended up with deviations from mean values as functions of S/N. The sets of repeat spectra were then sorted by their S/N to compute the 0.7 quantile of the parameter deviations for bins of ΔS/N = 20.

The results reported in Figure 2.3 show that the deviations in each stellar parameter decrease with increasing S/N, as expected. These trends were used to define the internal errors as follows. For a given spectrum with an associated S/N, the 0.7 quantile-value in the corresponding S/N-bin was adopted as the estimate of the internal error of the AMBRE:HARPS parameterisation.

For spectra with S/N > 160, a lower limit for the internal errors (set by the MATISSE internal error) was adopted as follows:

- $\sigma_{\mathrm{int}}(T_{\mathrm{eff}}) = 10$ K;
- $\sigma_{\mathrm{int}}(\log g) = 0.02$ dex;
- $\sigma_{\mathrm{int}}([\mathrm{M/H}]) = 0.01$ dex;
- $\sigma_{\mathrm{int}}([\alpha/\mathrm{Fe}]) = 0.005$ dex.

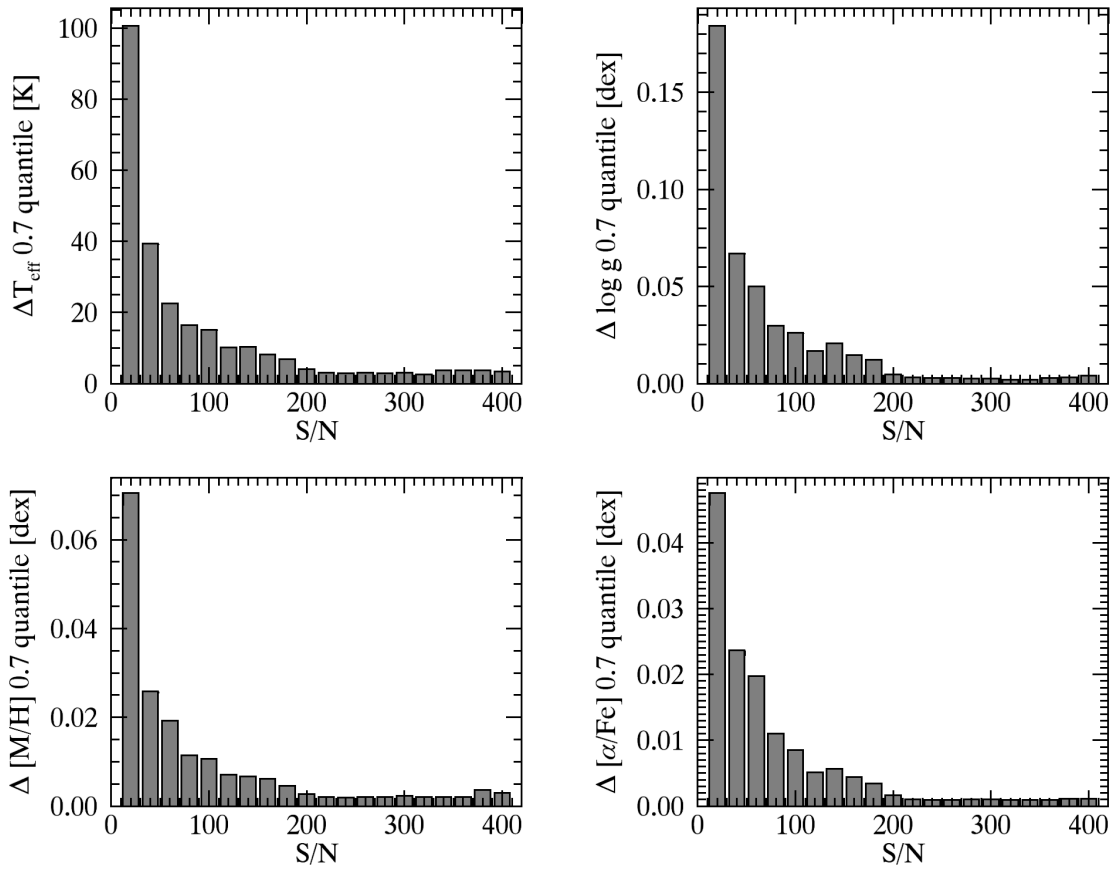

Figure 2.3: Histograms of the changes for each atmospheric parameter as a function of the S/N for the repeated spectra. For each bin, the 0.7 quantile of the deviation from the mean value is shown. These 0.7 quantiles define the internal errors associated with each parameter.

2.2 External error analysis

To quantify the external errors associated with the AMBRE:HARPS stellar parameters, they were compared with literature values for key samples within the dataset.

2.2.1 Benchmark stars

As a first estimate of the quality of this analysis, the results of the AMBRE:HARPS pipeline were investigated for a sample of well-studied reference stars. This sample is the FGK benchmark star sample defined for the Gaia mission and the Gaia ESO Survey (GES), for which the parameters are based on the homogeneous analysis of high quality data. For these benchmarks, the effective temperatures and surface gravities of Heiter et al. (2014, in prep.) were adopted, with the metallicities from Jofr, P. et al. (2014). Currently, there are no reference abundances of the α-elements available for these stars. The comparison between the AMBRE:HARPS and reference values for the stellar parameters of the ten FGK benchmark stars, which have a HARPS spectrum with S/N > 60 within the AMBRE:HARPS dataset in the three panels of is shown in Figure 2.4. The agreement is very good between both set of parameters with low biases and standard deviations, validating the results of the AMBRE:HARPS analysis for cool dwarfs with metallicities higher than -1.0 dex (i.e. the greater majority of the AMBRE:HARPS sample, see below). It is important, however, to point out that the agreement is slightly less good for the derived surface gravity of HD 22879 and the overall metallicity of the hot star Procyon, whose spectrum exhibits less metallic lines than the bulk of the AMBRE:HARPS sample.

2.2.2 Porto sample

An independent group previously published a large dataset of stellar atmospheric parameters estimated from HARPS spectra (Porto sample, hereafter) (see Sousa et al. 2008, 2011b,a; Adibekyan et al. 2012; Tsantaki et al. 2013). This sample consists in 1 111 stars for which stellar parameters have been estimated using a completely different method than that employed here, since the Porto method (ARES, Sousa et al. 2007) is based on an equivalent widths analysis of several selected lines.

The results of the Porto analysis provide a unique opportunity to perform a comparison between the parameterisation performed by two independent methods (with different line lists, model atmospheres, spectral analysis techniques, etc.) that analysed the same significantly large quantity of high quality spectra produced by the same instrument. Furthermore, this allows us to conduct a robust estimation of the external errors of the AMBRE:HARPS pipeline. Before describing the comparison between the two samples, I remind here that the AMBRE pipeline derives a mean [M/H] (i.e. taking into account all the metals), while the Porto group derives the [Fe/H] metallicity by considering only iron lines.

A cross match between the coordinates of the Porto sample and AMBRE:HARPS using a radius of $\sim 5''$ revealed 713 stars have been analysed by both methods. These 713 stars correspond to 3 991 spectra in the AMBRE:HARPS dataset with S/N ranging from $\sim$20 to $\sim$221 (see Figure 2.5). For the purpose of this comparison, the mean value of the derived AMBRE:HARPS stellar parameters (and the associated standard deviations) were calculated to represent the star when several spectra were available for the same star. The AMBRE:HARPS parameters were then compared to the Porto values in Figure 2.6.

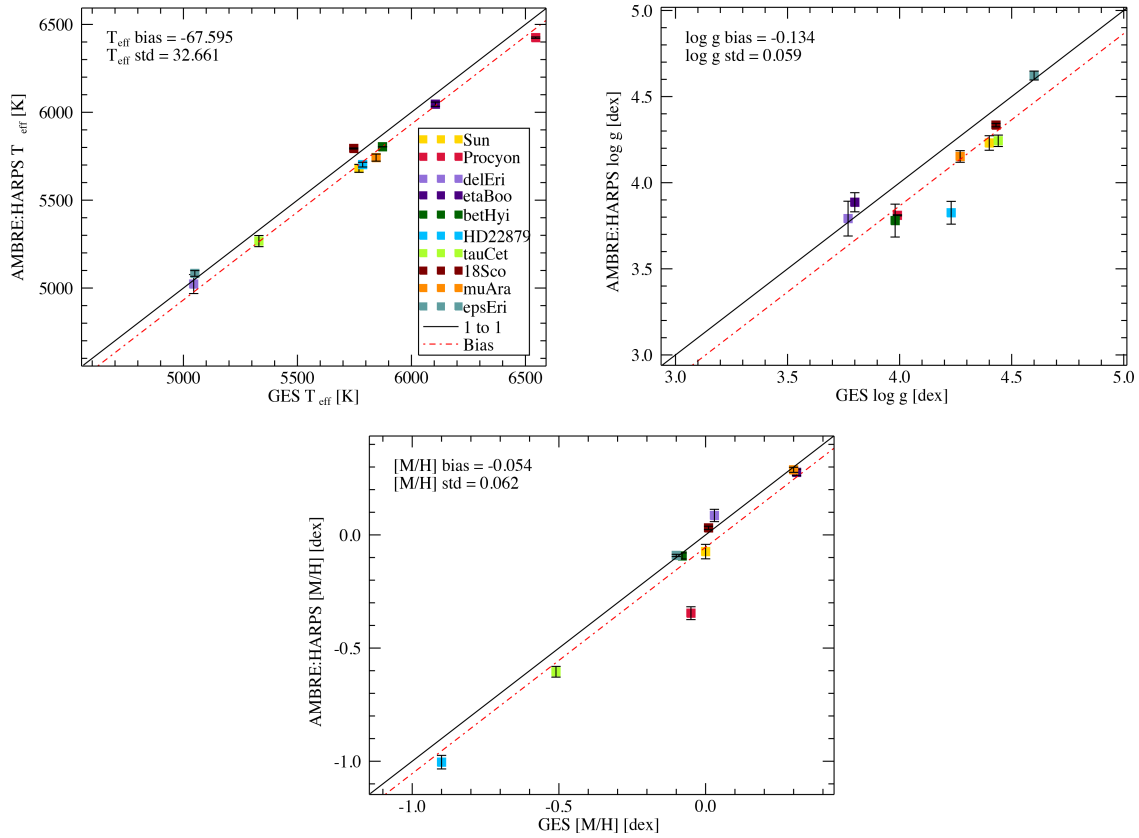

Figure 2.4: Comparison between the star atmospheric parameters for AMBRE:HARPS and FGK benchmarks from Jofr, P. et al. (2014) and Heiter et al. (2014, in prep.). The comparison is performed for $T_{\rm eff}$ (left panel), $\log g$ (right panel), and [M/H] (lower panel). Each point in the different panels corresponds to the mean value of the corresponding parameter when several AMBRE:HARPS spectra are available. For each parameter, the vertical error bars represent the standard deviation from the mean AMBRE:HARPS value.

The agreement between the Porto and AMBRE:HARPS stellar parameters are also illustrated in Figure 2.7 where the distribution of the residuals from the median value for each parameters are shown. The main characteristic of these distributions is that they are not perfectly centered on zero, but rather small biases exist that have been estimated as the mean value of the differences between Porto and AMBRE:HARPS.

Agreement in $T_{\rm eff}$

Figure 2.6-top left shows the comparison between the two set of effective temperatures. Despite the difference in the two methods, there is very good agreement between both results, the bias being around -59 K, and the dispersion around $\sim$87 K. It can be noted that the updated Porto effective temperatures of Tsantaki et al. (2013) for the cooler stars (below 5 000 K) are based on a new line list specifically built for these cool stars, which considerably improved the agreement in that temperature range. Indeed, their previous effective temperatures were slightly higher, leading to an absolute value of the bias between Porto and AMBRE:HARPS of 86 K (with standard deviation of 128 K) for the effective temperatures

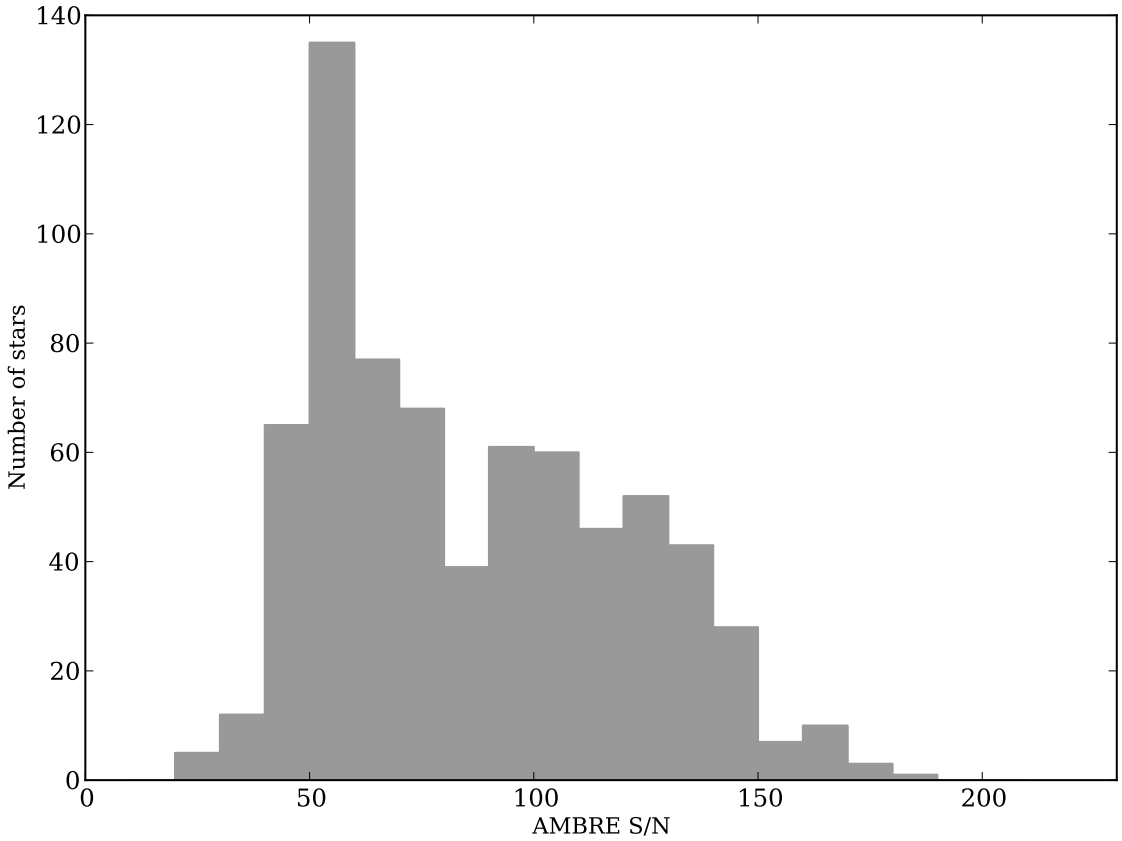

Figure 2.5: Histogram of S/N for stars in the AMBRE:HARPS dataset matching the Porto sample.

between 4 000 and 5 000 K. These bias and standard deviations decreased to an absolute value of 26 K and 117 K, respectively, when adopting the Porto revised values. In Figure 2.6 we also point out that, the small tail at large negative values is produced by stars hotter than 6 000 K, which are more difficult to parametrize than cooler stars.

Agreement in $\log g$

The agreement is not as good but still very reasonable for the stellar gravity comparison (see Figure 2.6), as it is probably the most difficult stellar atmospheric parameter to derive. The bias in gravity is -0.14 dex, and the standard deviation is 0.19 dex. In Figure 2.6, it can be seen that the possible cause of this bias and standard deviation could be that the Porto stellar gravities seem to span a smaller range of values than those obtained with the AMBRE:HARPS pipeline. Most of their sample data are indeed found between 4.3 and 4.6 dex, while the AMBRE:HARPS are mostly found between 4.2 and 4.7 dex. It can be seen in this figure that the large majority of the discrepant stars have an effective temperature larger than 6 000 K and/or a metallicity smaller than -0.4 dex, i.e. the cases for which spectra exhibit rather few lines to perform the parameterisation.

Agreement in [M/H]

On the other hand, Figure 2.6 reveals that the agreement between the stellar metallicities is very good (the bias is almost null and the dispersion is smaller than 0.1 dex). We note that we derive a slightly lower value than Porto, for stars with a mean metallicity smaller than -0.4 dex, and the small discrepancy seems to increase for lower metallicities. Indeed, when considering only stars, which have [M/H] < -0.4 dex, we calculate a bias and a standard deviation of -0.091 dex and 0.076 dex, respectively. Moreover, we point out that these stars are α-enriched (see Figure 2.6). In any case, the agreement is much better for more metal-rich stars, which consists of the bulk of the total sample.

Agreement in [α/Fe]

Finally, it was also possible to compare the abundance ratios of the α-elements with respect to iron. Adibekyan et al. (2012) published the individual abundances of the following α-species: Mg I, Si I, Ca I, Ti I, and Ti II. The mean of these five abundances was taken to compare to the AMBRE:HARPS [α/Fe] ratios (see Figure 2.6-lower right panel). Once again, the agreement is very satisfactory with a quasi-null bias and a standard deviation of only 0.03 dex. It can be, however, noted that, a small departure for this [α/Fe] ratio comparison, from the one-to-one line is present for the highest values of [α/Fe], or the most metal-poor stars (the Porto sample having slightly smaller [α/Fe] ratios when [α/Fe] > 0.2 dex). To understand this behaviour, we studied the values of the mean α abundances produced by our pipeline as a function of the abundances of the individual α-species provided by Adibekyan et al. (2012). Figure 2.8 reveals that the disagreement is probably mostly caused by the behaviour of the Ca and Si abundances that depart more than the other α-species.

Summary of comparison with Porto

It can be concluded that the agreement between the Porto and AMBRE:HARPS stellar parameters are very satisfactory. This comparison sample consists mainly of cool dwarf (solar-type) stars, and the agreement, therefore, validates the AMBRE:HARPS results for more than 90% of the spectra found in the AMBRE:HARPS sample. It has, however, to be noted that a lack of a good comparison sample did not allow us to estimate the systematic differences for non solar-type stars (although some estimates are given in Figure 2.6 for hot and metal-poor dwarfs, respectively). Finally, we point out that the Porto sample is also probably affected by trends and systematics and that some of the differences with AMBRE:HARPS could also partly originate in the Porto catalogue.

2.2.3 External error quantification

From the above comparisons, it can be concluded that only small biases are found between the AMBRE:HARPS stellar parameters and those of the Gaia Benchmark and Porto samples. Therefore no bias correction was performed, contrary to what has been done for the AMBRE:FEROS parameters. In this way, I also avoid needing to correct the stellar parameters of the remainder of AMBRE:HARPS spectra sample that are not found within the reference and Porto samples.

As for the external errors associated with the AMBRE:HARPS parameters, they have been defined using the Porto sample, which is a statistically significant comparison sample. It must be noted that the Porto sample has its own sources of error, and therefore, the adopted

external errors are probably overestimated. These external errors are estimated from the 0.7 quantiles reported in Figure 2.7 and are

- $\sigma_{T_{\rm eff},\rm ext} = 93$ K;
- $\sigma_{\log g,\rm ext} = 0.26$ dex;
- $\sigma_{\rm [M/H],ext} = 0.08$ dex;
- $\sigma_{\rm [\alpha/Fe],ext} = 0.04$ dex.

It is important here to point out that these external errors are based on dwarf star comparisons only. However, these values were also adopted for the few giants of the AMBRE:HARPS sample since (i) there was a lack of a good reference sample for giants in AMBRE:HARPS and (ii) as shown in Worley et al. (2012), except perhaps for the surface gravity, the external errors can be considered as constant for most of the stellar types.

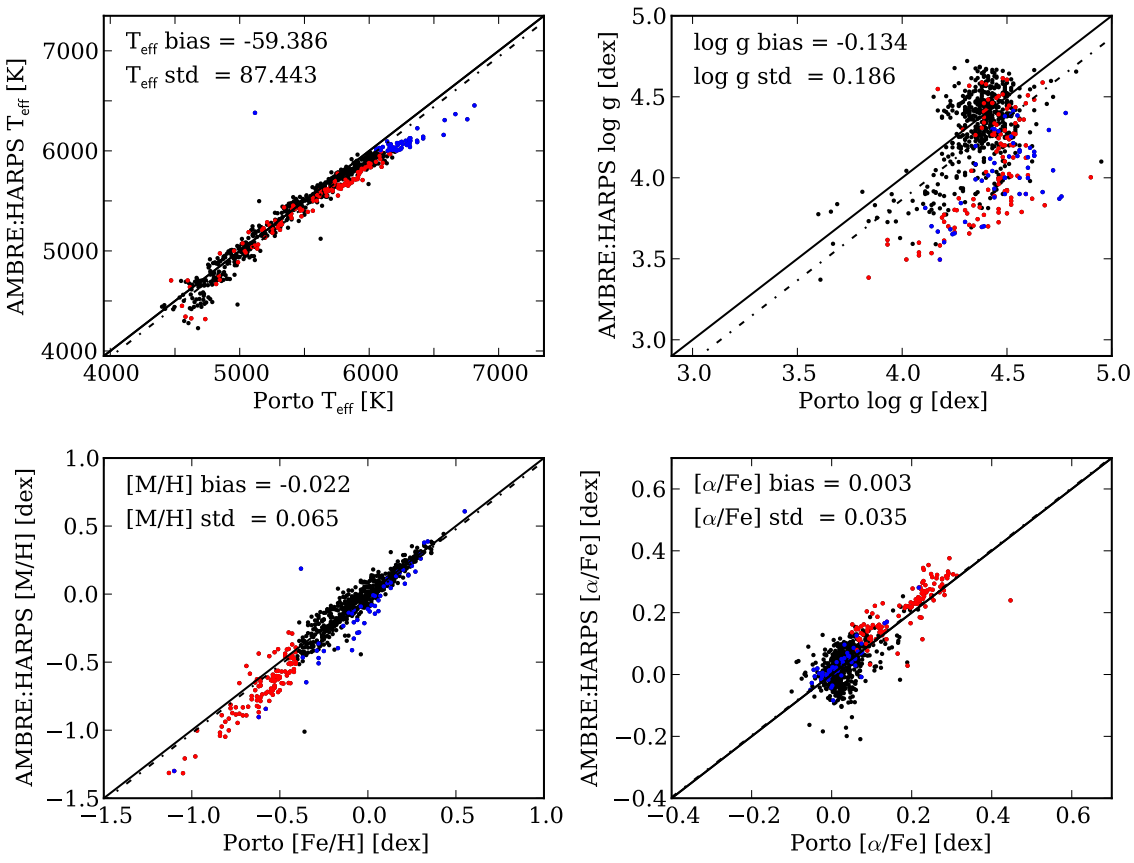

Figure 2.6: Comparison between the stellar atmospheric parameters derived by the AMBRE:HARPS pipeline and the reference sample from Porto. The solid line in each panel traces the one-to-one relation, while the dot-dashed line shows the location of the bias between the samples. Stars with $T_{\rm eff} > 6\,000$ K are plotted in the panels with blue dots; stars with [Fe/H] < -0.4 dex are the red dots. The biases in $T_{\rm eff}$, $\log g$, [M/H] and [α/Fe] for stars marked with red dots are -90.1 K, -0.349 dex, -0.098 dex, and 0.030 dex, respectively. For stars marked with blue dots the biases are -161.3 K, -0.451 dex, -0.093 dex, and 0.011 dex, respectively.

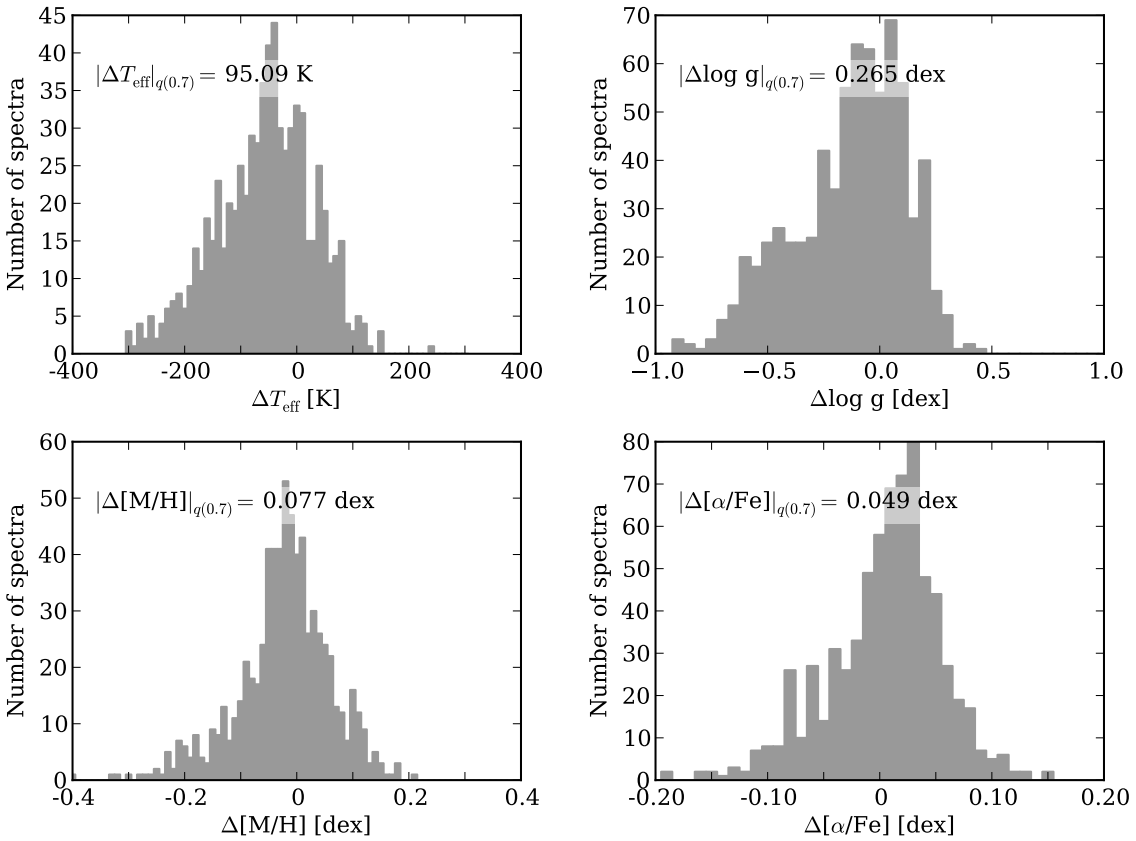

Figure 2.7: Distribution of the residuals $\theta_i^{\rm AMBRE} - \theta_i^{\rm Porto}$ for the stellar parameters $\theta_i = (T_{\rm eff}, \log g$, [M/H] and [$\alpha$/Fe]). For the AMBRE:HARPS stars with repeated observations, the mean of the AMBRE:HARPS derived parameters are shown. The asymmetries in the distributions reflect the asymmetries to the 1 to 1 lines that can be seen in Figure 2.6. In each panel the 0.7 quantile of the absolute value of the residuals corrected for the bias is reported.

As the final error associated to each parameter will be the quadratic sum of the internal and external errors, the consistency of this value with the deviations of the AMBRE:HAPRS parameters from the reference was verified. Referring to the Porto sample, I confirmed that the error associated to AMBRE:HARPS parameters is a good approximation to the deviations from the literature at all S/N.

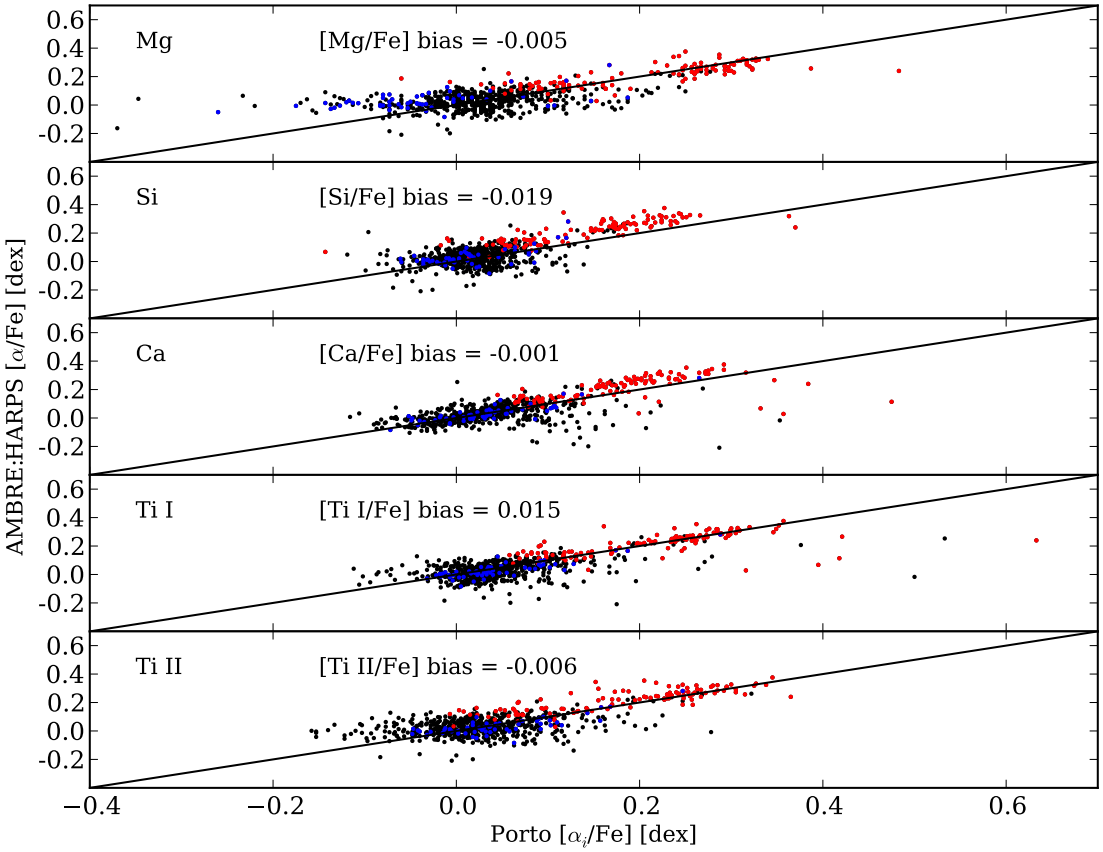

Figure 2.8: Comparison between the mean α-abundances derived by the AMBRE pipeline and the abundances of the individual α-elements determined by Porto. The largest departures from the 1-to-1 line are found for the Si and Ca abundances and could explain the departure observed in Figure 2.6 for the most metal-poor stars. As in Figure 2.7, stars with $T_{\rm eff} > 6\,000$ K are plotted in the panels with blue dots; stars with $[\mathrm{Fe}/\mathrm{H}] < -0.4$ dex are the red dots.

Chapter 3

Results

3.1 Final parameters of AMBRE:HARPS analysis

The final accepted stellar parameters of the 93 116 AMBRE:HARPS spectra are shown in Figure 3.1 (HR diagram in the plane $T_{\rm eff}$ - $\log g$), Figure 3.2 - 3.4 (histograms of the distribution of the three main stellar parameters), and Figure 3.5 (different combinations of the AMBRE:HARPS stellar parameters).

The main characteristic of Figure 3.1 is that the great majority of the spectra are populating the main sequence, confirming that more than 90% of the entire HARPS:AMBRE sample constitutes of cool dwarf stars of G and K spectral types (see also Figure 3.2, 3.3, and Figure 3.5). Actually, a large fraction ($\simeq$ 42%) of the spectra correspond to stars with an effective temperature close to the solar. Furthermore, most of these dwarfs ($\sim$86%) have a solar metallicity, which is larger than -0.5 dex (see Figure 3.4). The red giant branch is clearly seen, although it is much less populated. From Figure 3.3, it can be seen that about 4% of the total sample is composed of giant stars (defined as $\log g < 3.5$ dex). We refer to a forthcoming paper for a deeper analysis of the AMBRE:HARPS sample characteristics.

We finally point out that MATISSE relies on a learning phase based on the discrete grid of synthetic spectra. The products are vectors on which the observed spectra are projected to retrieve their parameters. For the AMBRE analysis, we have decided to adopt a version of these projection vectors computed from a direct inversion of the correlation matrix of the synthetic spectra (see Kordopatis et al. 2011). This assumption, giving better results for high quality spectra, can lead in some cases to pixelization effects due to an overfitting of the data (as it can be noticed in Figure 3.5).

3.2 ESO table description

The derived stellar parameters and [α/Fe] abundances of the 90 174 AMBRE:HARPS spectra are being ingested into the ESO archives (see `http://archive.eso.org/cms/eso-archive-news/first-data-release-from-the-matisse-oca-eso-project-ambre.html`). Table 3.1 describes the exact data that is being delivered to ESO.

To the values of the parameters calculated by the AMBRE:HARPS pipeline, the internal and external errors for each of the four parameters as defined in Section 2.1 and Section 2.2, respectively, have been added. The null value adopted for each parameter is also reported in this table. Moreover, each spectra has been flagged as a function of the χ^2, which is

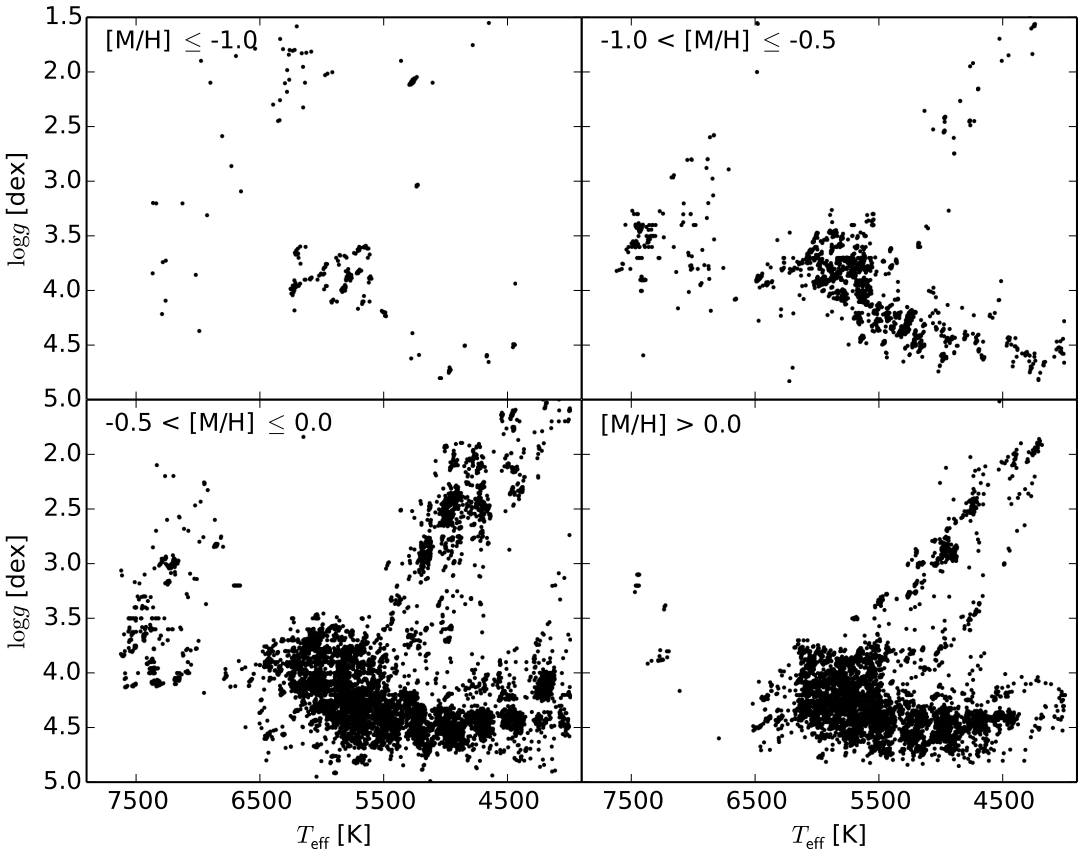

Figure 3.1: HR diagram of the AMBRE:HARPS stellar atmospheric parameters. The different panels correspond to different [M/H] bins, as indicated in the figure. Almost 14% of the retained spectra have a metallicity [M/H]< -0.5 dex.

derived from the agreement between the synthetic and the observed spectrum. As reported in Table 3.1, three values have been adopted for this flag: *very good*, *good*, and *acceptable*.

The limits applied by which to attribute these flags are defined differently for each temperature domain using as reference the interpolation curves that describe the χ^2 as a function of the S/N (see Section 1.4.3). The percentage of spectra associated with each flag are reported in Table 3.2 with the limiting values on the χ^2 defining the flag itself. The great majority of spectra that have a *good* or *very good* χ^2 are observations from cold or warm stars ($4\,000\text{K} \leq T_{\text{eff}} \leq 6\,500\text{K}$). On the other hand, 60% of the spectra of the hot stars ($T_{\text{eff}} > 6500\text{K}$) show a worse χ^2. These results reflect the ability of any spectral analysis method (including MATISSE) to better determine reliable parameters for stars with effective temperatures between $\sim$4 000 K and $\sim$7 000 K (namely F, G and K-type stars). For hotter stars (or metal-poor ones), fewer spectral signatures are available leading to more uncertainty in the derived stellar parameters.

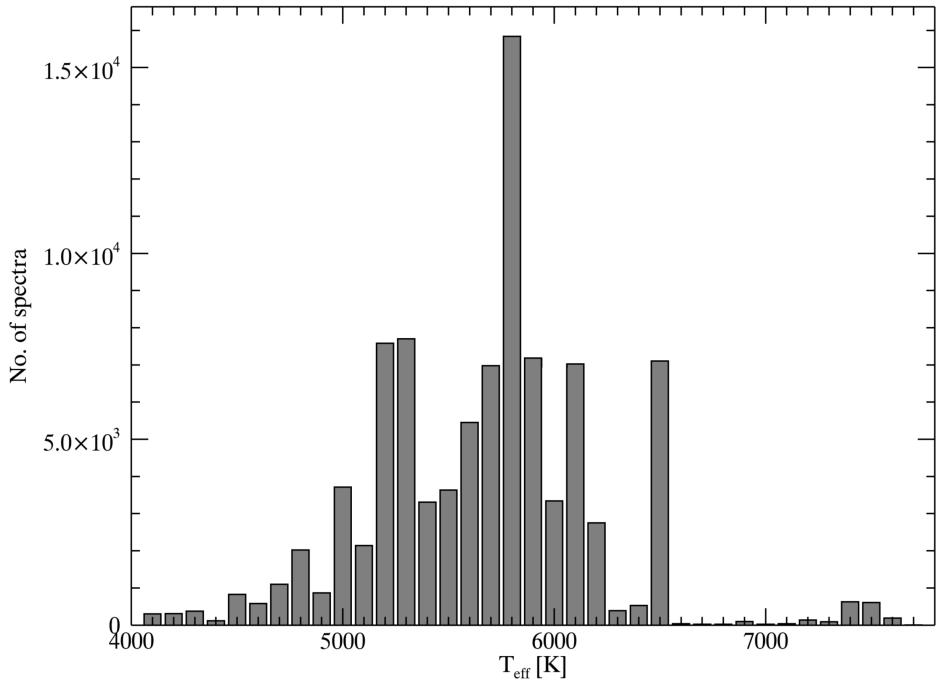

Figure 3.2: Distribution of the derived AMBRE:HARPS $T_{\rm eff}$ values. The distribution has a main peak around 5 700 K and a second one between 5200 K and 5300 K.

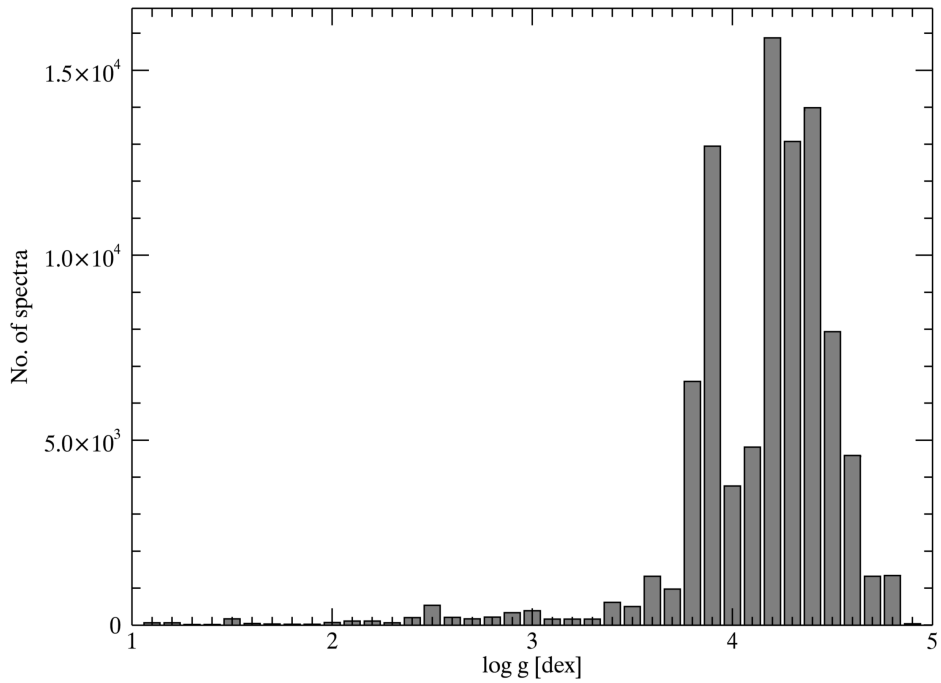

Figure 3.3: Distribution of the derived AMBRE:HARPS $\log g$ values. This distribution peaks around the solar gravity.

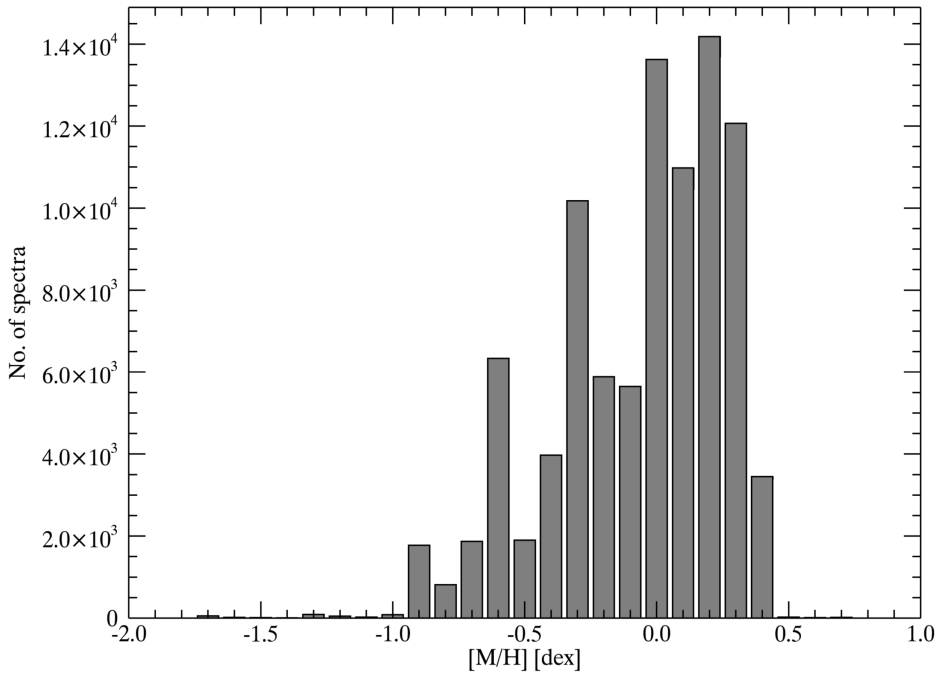

Figure 3.4: Distribution of the derived AMBRE:HARPS metallicities.

3.3 Summary and discussion

Automatic stellar parameterisation algorithms are needed to analyse the large number of spectra produced by surveys of the Galaxy, both ongoing and planned. MATISSE is one such algorithm, it uses the projection approach and is part of the pipeline analysing spectra produced by RVS mounted on the Gaia spacecraft. MATISSE is used also by the AMBRE project, to estimate parameters of stars observed by ESO facilities.

This part presented the automatic stellar parameters determination of more than 90 000 HARPS spectra collected between 2003 and 2010 and archived at ESO. These spectra correspond to more than 10 000 different stars that have been observed between one and several hundreds of times.

Stellar parameters have been determined for more than 70% of the total sample of HARPS spectra delivered by ESO. The main rejection criteria are possibly from low S/N, broad line (e.g. due to fast-rotation) spectra, poor synthetic fit of the observed spectra, and parameters found outside the synthetic spectra grid boundaries. The stellar parameters obtained for the HARPS spectra, which were derived using the AMBRE pipeline based on the MATISSE algorithm, are being delivered to ESO for ingestion in their archives.

The stellar parameters are the effective temperature, the surface gravity, the mean metallicity ([M/H]), the abundance of the α-elements with respect to iron ([α/Fe]), and their associated internal and external errors. We also provide the radial velocity (with the associated error and the FWHM of the corresponding CCF) and a quality flag for the parameters.

The AMBRE:HARPS sample is an extensive dataset of mainly solar-like stars for which the stellar parameters have been homogeneously determined. Given the richness of the HARPS archival data, the present dataset constitutes an invaluable tool to pursue a range of projects for both galactic archaeology and exoplanet host star analyses. Parameters obtained

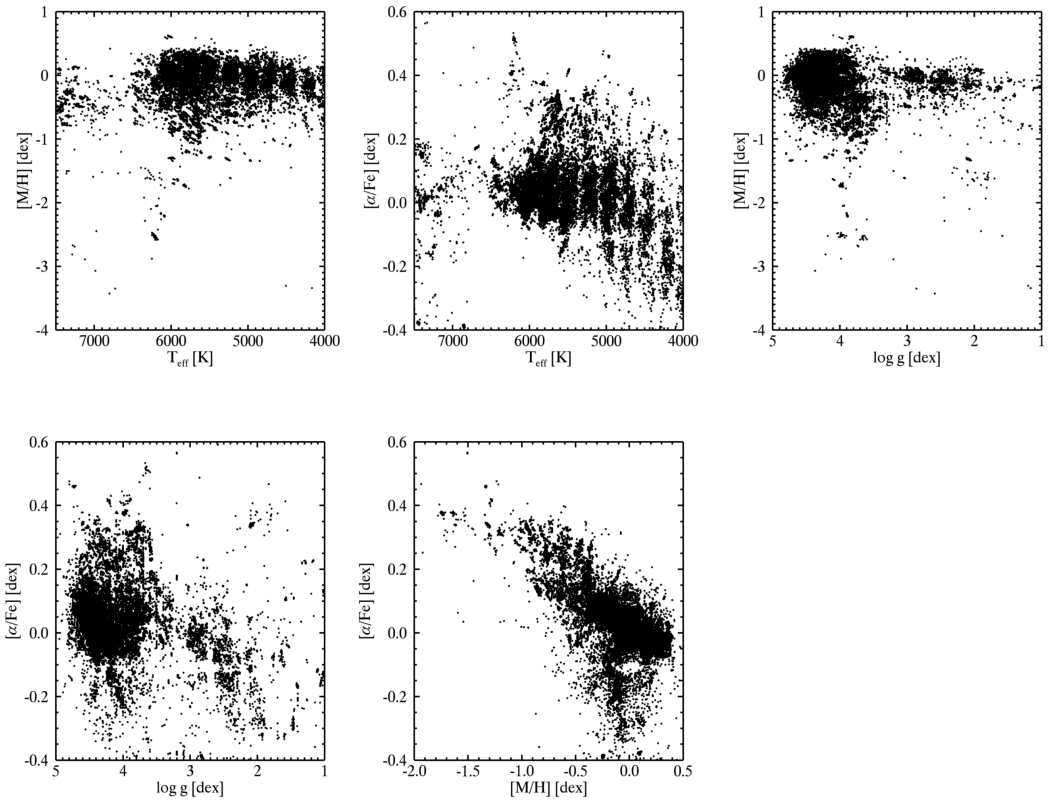

Figure 3.5: The final AMBRE:HARPS stellar atmospheric parameters. Different combinations of the four derived parameters. Top-left: [M/H] vs. $T_{\rm eff}$. Top-center: [α/Fe] vs. $T_{\rm eff}$. Top-right: [M/H] vs. $\log g$. Lower-left: [α/Fe] vs. $\log g$ and Lower-center: [M/H] vs. [α/Fe].

through the AMBRE pipeline are being used to map the heavy element contents (Mikolaitis et. al, 2015, to be submitted) and Li abundances of stars in the Solar vicinity. Also, they are employed to study the separation of the stellar populations forming the thin and thick disk of the Galaxy, the first younger than the second (de Laverny et al., 2015, in preparation). A similar study has been published by the GES collaboration in Recio-Blanco et al. (2014), employing parameters estimated from the Gaia ESO Survey ESO:GIRAFFE spectra, using also MATISSE as algorithm. Exploiting the AMBRE:HARPS to study thin and thick disk sample is possible, since [M/H] and [α/Fe] distribute in the related plane (lower-center panel of Figure 3.5) with the characteristic dichotomy of a population sampling these two galactic structures, the α-rich being the thick disc (see Reddy et al. 2006; Prieto 2010).

The AMBRE database is also used by GES and Gaia survey as calibration data (de Laverny et al. 2013).

Finally, this sample is the second phase of the AMBRE project and hes been followed by the parameterisation of the UVES archive, of which (Worley et al. 2014) reports preliminary results.

This concludes Part I of the manuscript. Part II will treat the automatic analysis of photometric data, in particular the attempt to build a classification model for supernovae using only their light curves.

Table 3.1: Description of the columns in the table of HARPS stellar parameters delivered to ESO.

Keyword	Definition	Value range	Null value	Determination
DP_ID	ESO dataset identifier	—	—	FITS file header.
OBJECT	Object designation as read in ORIGFILE	—	—	FITS file header.
TARG_NAME	Target designation as read in ORIGFILE	—	—	FITS file header.
RAJ2000	Telescope pointing (right ascension, J2000)	—	—	Units = deg. FITS file header.
DEJ2000	Telescope pointing (declination, J2000)	—	—	Units = deg. FITS file header.
MJD_OBS	Start of observation date	—	—	Units = Julian day. FITS file header.
EXPTIME	Total integration time	—	—	Units = sec. FITS file header.
S/N	Signal-to-noise ratio as estimated by the pipeline	$(10, \infty)$	`NaN`	Internal routine.
VRAD	Stellar radial velocity	$[-500, 500]$	`NaN`	Units = km s^{-1}.
ERR_VRAD	Error on the radial velocity	$(0, \infty)$	`NaN`	Units = km s^{-1}. FITS file header/AMBRE radial velocity calculation routine.
VRAD_CCF_FWHM	FWHM of the CCF between the spectrum and the binary mask	$(0, \infty)$	`NaN`	FITS file header/AMBRE radial velocity calculation routine.
VRAD_FLAG	Quality flag on the radial velocity analysis	0, 1, 2, 3, 4, 5	`NaN`	0 = Excellet determination ... 5 = Poor determination, when $V_{\rm rad}$ determined by AMBRE pipeline. `NaN` when $V_{\rm rad}$ determined by HARPS pipeline.
TEFF	Stellar effective temperature ($T_{\rm eff}$) as estimated by the pipeline	[4000, 7625]	`NaN`	Units = K. Null value used if $T_{\rm eff}$ outside accepted parameter limits.
ERR_INT_TEFF	Effective temperature internal error	[10, 100]	`NaN`	Units = K. Defined as function of S/N, it is equal to the 0.7 quantile to the bin to which the S/N value belongs to; see Figure 2.3 and Section 2.1.
ERR_EXT_TEFF	Effective temperature external error	93	`NaN`	Units = K. Defined using the Porto sample; see Section 2.2.3.
LOG_G	Stellar surface gravity ($\log g$) as estimated by the pipeline	[0, 5.0]	`NaN`	Units = dex. Null value used if $\log g$ outside accepted parameter limits.
ERR_INT_LOG_G	Surface gravity internal error	[0.02, 0.184]	`NaN`	Units = dex. See ERR_INT_TEFF for definition.
ERR_EXT_LOG_G	Surface gravity external error	0.26	`NaN`	Units = dex. See ERR_EXT_TEFF for definition.
M_H	Mean metallicity [M/H] as estimated by the pipeline	$[-3.5, 1]$	`NaN`	Units = dex. Null value used if [M/H] outside accepted parameter limits.
ERR_INT_M_H	Mean metallicity internal error	[0.01, 0.07]	`NaN`	Units = dex. See ERR_INT_TEFF for definition.
ERR_EXT_M_H	Mean metallicity external error	0.08	`NaN`	Units = dex. See ERR_EXT_TEFF for definition.
ALPHA	α-elements over iron enrichment ([α/Fe]) as estimated by the pipeline	$[-0.4, 0.8]$	`NaN`	Units = dex. Null value used if [α/Fe] outside accepted parameter limits.
ERR_INT_ALPHA	α-elements over iron enrichment internal error	[0.005, 0.048]	`NaN`	Units = dex. See ERR_INT_TEFF for definition.
ERR_EXT_ALPHA	α-elements over iron enrichment external error	0.04	`NaN`	Units = dex. See ERR_EXT_TEFF for definition.
CHI2	$\log(\chi^2)$ of the fit between the observed and reconstructed synthetic spectrum at the MATISSE parameters	$[-5, \infty)$	`NaN`	Goodness of fit between final normalised and final reconstructed spectra.
CHI2_FLAG	Quality flag on the fit between the observed and reconstructed synthetic spectrum at the MATISSE parameters	0, 1, 2	`NaN`	0 = Very good fit, 1 = Good fit, 2 = Poor fit
ORIGFILE	ESO file name of the original spectrum being analysed	—	—	

Table 3.2: Definition of the quality flags of the fit between the observed and the reconstructed synthetic spectra.

$T_{\rm eff}$ domain [K]	Flag	spectra %	χ^2 limits
$4000 \leq T_{\rm eff} \leq 5000$	0	10	$\chi^2 \leq \chi^2_{\rm cold,fit}(1-\sigma_{\rm cold})$
	1	59	$\chi^2_{\rm cold,fit}(1-\sigma_{\rm cold}) < \chi^2 \leq \chi^2_{\rm cold,fit}$
	2	31	$\chi^2_{\rm cold,fit} < \chi^2 \leq \chi^2_{\rm cold,fit}(1+0.5\sigma_{\rm cold})$
$5000 < T_{\rm eff} \leq 6500$	0	44	$\chi^2 \leq \chi^2_{\rm warm,fit}$
	1	51	$\chi^2_{\rm warm,fit} < \chi^2 \leq \chi^2_{\rm warm,fit}(1+1.5\sigma_{\rm warm})$
	2	4	$\chi^2_{\rm warm,fit}(1+1.5\sigma_{\rm warm}) < \chi^2 \leq \chi^2_{\rm warm,fit}(1+3\sigma_{\rm warm})$
$T_{\rm eff} > 6500$	0	26	$\chi^2 \leq \chi^2_{\rm hot,fit}(1-\sigma_{\rm hot})$
	1	14	$\chi^2_{\rm hot,fit}(1-\sigma_{\rm hot}) < \chi^2 \leq \chi^2_{\rm hot,fit}(1-0.5\sigma_{\rm hot})$
	2	60	$\chi^2_{\rm hot,fit}(1-0.5\sigma_{\rm hot}) < \chi^2 \leq \chi^2_{\rm hot,fit}$

Notes. In the second column, the flags are defined as: 0=*Very Good*, 1=*Good*, and 2=*Acceptable*. The third column refers to the percentage of spectra associated with the corresponding χ^2 flag. The last column defines the selection criteria for the flags. $\chi^2_{\rm cold,fit}$ and $\sigma_{\rm cold}$; $\chi^2_{\rm warm,fit}$ and $\sigma_{\rm warm}$; and $\chi^2_{\rm hot,fit}$ and $\sigma_{\rm hot}$ refer to 11 %, 83 %, and 2 % of the delivered spectra, respectively. The percentages of the different temperature domains does not sum to 100 % because we have excluded spectra with $T_{\rm eff} < 4000$ K.

Part II

Photometric classification of supernovae

Chapter 4

Automatic classification of supernovae

4.1 Introduction

Some star ends its life with a catastrophic explosion. Such explosions are very powerful in that they can release up to about 10^{44} J of mechanical energy. The brightness of these events can even outshine the host galaxy. When the explosion leaves a remnant, this is a compact object: a neutron star or a black hole.

Supernovae play an important role in galaxy evolution. After explosion, the interstellar medium (ISM) is enriched by heavy elements synthesised during both the star's life (via nuclear fusion) and the explosion itself. A side effect of the explosion is sweeping the ISM around the progenitor, this has two opposite effects. The region being swept by the ISM, will see a star formation quenching, the *negative supernova feedback*. Outside that region, the stellar wind will condense the surrounding gas exciting the star formation, the *positive supernova feedback* (or trigger).

Exploding stars called supernova Ia are the best distance indicators on cosmological scale. Their luminosity distance is thus used in observational cosmology to determine the values of the two parameters governing the expansion of the Universe in Friedmann-Robertson-Walker models. One is the Hubble constant H_0 (see Branch & Tammann 1992 and Leibundgut 2007), which is the current expansion rate of the Universe. The second is the deceleration parameter q_0, bound to dark energy (Carroll et al. 1992), and whose sign indicates if the expansion is accelerating ($q_0 < 0$) or not. Using light curves of supernovae Ia at high redshift, Riess et al. (1998) and Perlmutter et al. (1999) showed evidence for an accelerating Universe.

Finally, *supernova rates* reflect the evolution of the star formation rate (SFR), and can be used to trace both the evolution, with redshift, of the SFR and the nature of the SN progenitors.

Central to all these application is the classification of the supernova. In the following section I will explain which characteristic of the supernova are used to determine its type.

4.2 Supernova types

Historically, supernova types are defined on the basis of features in the supernova spectrum at maximum light. The absence or presence of hydrogen lines defines the two main classes

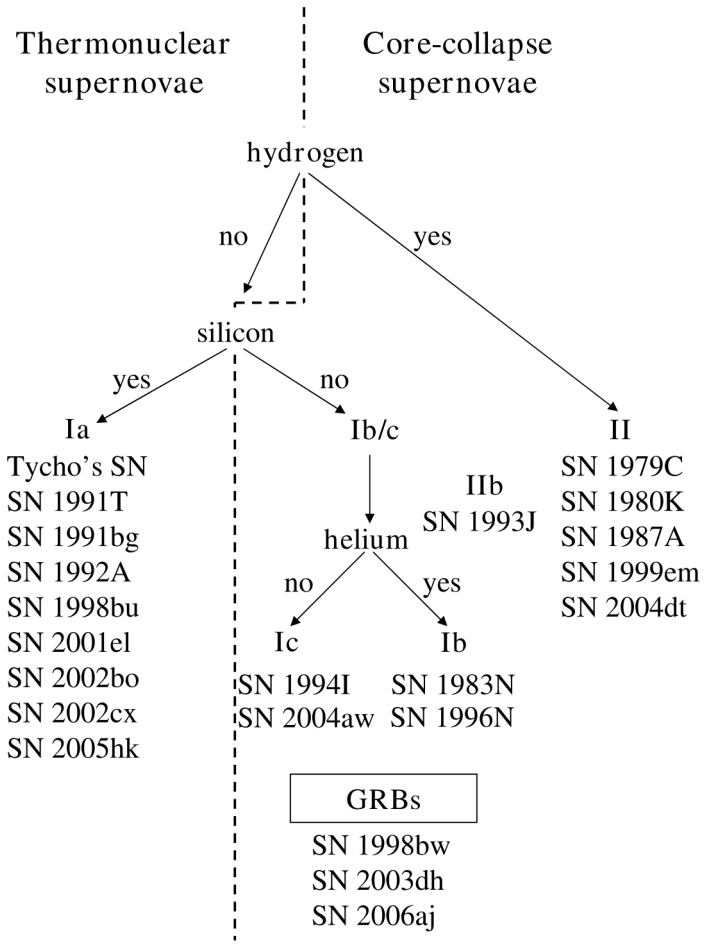

Figure 4.1: Classification scheme for supernovae. It is based on the presence or absence of specific spectral features at maximum light. In each class are reported IAU names of some observed events. From Leibundgut (2007).

of type I and type II. Spectra of supernovae of type I are further distinguished between type Ia and type Ib/c if they display or lack the absorption line from Si II. Spectra from supernovae type Ib are different from the one produced by Ic in being dominated by helium lines. Type II supernovae are also divided into sub-types, on the basis of either light curve shape or additional features in their spectra. Supernovae of type II-P (plateau) show light curves with constant brightness within ~ 1 day from the maximum, for an extended period (~ 100 days). On the other end, events showing a light curve with linear decline are collected as type II-L (linear). Supernovae with strong narrow lines in their spectra are classified as II-n (narrow), and finally, what is thought to be a transitional type, the IIb, showing an evolving helium line and a light curve similar to the type Ib/c. A view of this classification scheme is reported in Figure 4.1; there is also shown the distinction made on the basis of the explosion mechanism: thermonuclear or via core-collapse. The only supernovae produced by a thermonuclear explosion of the progenitor (a CO white dwarf) are the type Ia. All the other are the result of the collapse of the core of a massive star.

Despite the identification of the supernova type in principle needs spectroscopy, we are in a situation where this approach is often not applicable. The number of candidates identified

by photometric surveys is increasing at a pace spectroscopic facilities can not keep up to. Thus, the need for an automatic tool able to process the light curves and classify them, possibly reproducing the classification scheme described in this section. Section 4.3 outlines the classification method, developed using `Python 2.7`[1] and `R` (R Core Team 2015) languages. The project's core idea is to use a data-driven approach.

4.3 The method

The development of this project has been driven by the idea of relying as much as possible on the data, introducing no assumptions. When this is not be viable, the assumption has to be as weak as possible. Therefore, the data have to "describe themselves" with no use of strong prior hypothesis. Such an approach is called *data-driven*, and can be put into place provided a large statistical sample.

In classification, a data-driven approach offers the possibility to identify a new class of astrophysical events. Such events, using a more traditional template-based classification method, at best would be recognised as outliers, because dissimilar to any template. The use of templates, or the introduction of strong assumptions on how a light curve is shaped, prevents from readily identify new classes.

Techniques from statistics theory can be combined together to achieve a data-driven result. Examples of such techniques are machine learning, a branch of artificial intelligence (see Chapter 5), and non-parametric statistics (see Chapter 7).

The introduction to which steps the developed method needs to take, it is best made by reasoning backward, from what has the final result to be, to what kind of pre-processing is necessary.

The goal of classification is to identify clusters in some parameter space. Clusters are defined on the basis of similarities between observations (or events): similar objects are expected to be near in a parameter space, which is the reason of the clustering. Considering the light curves, they are formed by flux measurements taken at different time steps on an irregular time-grid. As a consequence, to define parameters and assess similarities, they have to be reported on the same regular time-grid. This is done by interpolating the data points. Finally, before interpolation, astrophysical effects have to be taken into account and the light curves corrected for them.

The classification process developed with this work can than be outlined in four main steps:

1. *Light curves pre-processing.* That is correction for the astrophysical effects of galactic absorption and time dilation, and K correction. Both time dilation and K correction require an estimate of the supernova redshift; this is obtained either photometrically from the host galaxy, or from the supernova spectrum. The data set I used, described in Section 4.4, provided the host galaxy redshift for each light curve, while the spectroscopic redshift only for a small subset. When available, time dilation correction and K correction were calculated using the spectroscopic redshift. This method does not account for missing redshift information, see Section 9.3 for a discussion.

2. *Light curve interpolation.* To use the smallest number of assumptions, this task is carried out using a non-parametric method. I chose *Gaussian processes*, used for a similar application by Faraway et al. (2014) and described in Chapter 7. This technique

[1] `www.python.org`

is also used in other field of astronomy, like modelling light curves from exoplanets transient (Gibson et al. 2012). The interpolation returns all of the corrected light curves on a time grid with 1 day steps. This turns out to be the most delicate step. Issues like over- or under-fitting, in this framework are reduced only by injecting some assumption on the light curve shape (see Section 7.4); this has to be carefully done to maintain the data-driven approach as much as possible. The interpolation has been performed using the `Python` language package `GPy` (The GPy authors 2012–2014).

3. *Light curve parameterisation.* Parameters are needed to easily identify clusters of similar objects. A data-driven technique to extract relevant parameters from the light curves is called *diffusion maps* (Coifman & Lafon 2006), ans is described in Section 8.2. The technique of diffusion maps has been used in Richards et al. (2009) and, for supernova classification, in Richards et al. (2012) with good results. The parameterisation was performed using the `R` language package `diffusionMap`, written by (Richards 2014).

4. *Build a classification model.* This step is performed using machine learning. The data set described in Section 4.4, contains a subset of light curves for which the type is known from spectroscopy. In machine learning jargon this is called the training set. Parameters describing these light curves are used to build a model to classify further light curves into type Ia, Ib/c and II. The machine learning technique I used is called *random forests* (Breiman 2001). This technique is widely used in present machine learning applications in many domains, from medical imaging to astronomy, such as in Goldstein et al. (2015). Using the packaged routine in the `R` language, `randomForest` (Liaw & Wiener 2002), building the classification model is fairly easy and fast, see Chapter 9.

The classification model built at the end of this four-step process has to be tested to determine its confidence. The test is carried out using light curves that did not take part in the training, but for which the type is known as well. The confidence of the model is assessed by comparing the true belonging class to the prediction by the classification model. The not encouraging results described in Section 9.2 are discussed in Section 9.3, with possible solutions.

4.4 A simulated data set

To develop this automatic classification tool, I used the simulated data set based on the one from the "Photometric SN Classifier Challenge" Kessler et al. (2010), and including bug-fixes, improvements, and keywords indicating the true type and redshift for each SN. The data set includes $\sim 21\,000$ supernovae brokedown in the 3 known types Ia, Ib/c and II as in Table 4.1. [2] The sample is generated using the SNANA software package by Kessler et al. (2009). SNANA has a routine designed to simulate a real supernova survey in many of its aspects, thus including the telescope set up, the filters, the observing site etc.. This means, also, that the simulated explosion time extends well outside the survey season, both before and after. Therefore the data set contains also incomplete light curves.

To the SNPhotCC, light curves were simulated as if collected during the Dark Energy Survey (DES) (Bernstein et al. 2009) run at Cerro Tololo Inter-American Observatory. Each

[2] The data set is freely available at `http://sdssdp62.fnal.gov/sdsssn/SIMGEN_PUBLIC/SIMGEN_PUBLIC_DES.tar.gz`

Table 4.1: Breakdown into supernova types and simulating algorithm of data set based on SNPhotCC.

SN type	Algorithm	No. SNe	No. Spec
Ia	MLCS2k2	2505	256
	SALT2	2583	
Ib/c	SALT2-like	2801	303
II	SALT2-like	13430	544

Notes. Breakdown into supernova types and simulating algorithm, of data set based on SNPhotCC. The simulating algorithm are respectively published by Jha et al. (2007); Kessler et al. (2009) (MLCS2k2) and Guy et al. (2007) (SALT2). These information are compiled combining information from ASCII files `SIMGEN_PUBLIC_DES.README` and `SIMGEN_PUBLIC_DES.DUMP`. The last column reports the number of spectroscopically confirmed supernovae, which are forms the training set.

Table 4.2: Integer codes for supernova types

SN type	code
Ia	1
II	2
	21
	22
	23
Ib/c	3
	32
	33

supernova is supposed to have flux measurements taken in four filters, the g, r, i and z, with effective wavelengths of 479.66 nm, 638.26 nm 776.90 nm 910.82 nm respectively Schlafly & Finkbeiner (from 2011, Table 6). Each supernova comes in an ASCII file divided in two parts: header and flux measures. The header reports many information on the simulated supernova, such as the reddening from the Milky Way, the redshift and the real type. Regarding redshift and real type, to be congruent with the information available in the SNPhotCC, I only considered the values of keywords `REDSHIFT_SPEC`, `HOST_GALAXY_PHOTO-Z` and `SNTYPE`. When `REDSHIFT_SPEC` has a valid value (not equal to -9) the supernova has associated a spectroscopic redshift; in such cases, `SNTYPE` is diverse from -9 and equals a number encoding the real type of the supernova, reported in Table 4.2. Supernovae with a spectroscopic redshift form what in machine learning jargon is called the "training set".

This characteristic is put in place to reflect what is the most important target of supernova surveys: to probe the accelerating Universe. In this view, the limited spectroscopic resources are much more likely to obtain observations of SNe Ia then of other supernova types. This feature turns out to be a problem when building a classification model with the ability to distinguish among the three classes of Ia, Ib/c and II. Since machine learning techniques learns from examples in the training set, having less examples of Ib/c and II will make supernovae of these two classes less recognisable lowering the classification model's accuracy.

4.5 Wrapping up

Supernovae are important events in astronomy, used to study phenomena from the evolution of the Universe to the star formation rate history. Central to both these application is the knowledge of the type of supernova. So far these explosive events have been classified using features in their spectra like the presence or absence of hydrogen lines (Figure 4.1). With the advent of the present and next future high data throughput photometric surveys, the spectroscopic followup and classification of all the candidate is not feasible by the available facilities. An automatic classifier exploiting only photometric data is advisable.

The automatic tool I developed uses a data-driven approach, by introducing the less assumptions as possible. It is organised into the four main steps of pre-processing, light curves interpolation via the non-parametric technique of Gaussian Processes, parameterisation using diffusion maps and definition of the classification model by means of random forest, a machine learning technique.

One important goal is to verify if the information extracted from light curves and used to build the classification model is able to reproduce the present classification scheme derived from spectra.

Chapter 5 will introduce and explain the key concepts of machine learning, with which we will go along to the end of the manuscript. Each of the four steps described above and in Section 4.3, will be explained in the following chapters. The results will be discussed in Section 9.2.

Chapter 5

Machine Learning

Most of the methods developed in the recent past for photometric classification of supernovae use, or are based, on probability; see for example the colour-colour diagram proposed by Poznanski et al. (2002), the probabilistic methods from Kuznetsova & Connolly (2007) or the template fitting of Sullivan et al. (2006). However, none of them have the "ability" to *learn* from data. Their results can improve only if we construct better and more detailed models once *we* have learned something new about supernovae from observations. Also, these models are based on heuristics methods.

With the advent of next future supernova survey, the number of supernovae discoveries per year will grow from two to three order of magnitude with respect to current discovery rate. Thus we will need automatic and reliable methods to classify this huge number of supernovae.

Using machine learning algorithms we can actually teach a computer about SN type's photometric characteristics in a rigorous way, using statistics and computational science. In this way we can use "learned" computers to classify the huge number of supernovae that will be discovered from near future surveys.

This chapter introduces machine learning basics. These concept will be of use in Chapters 6, 7 and 8 which describe different learning and machine learning methods, to respectively carry out light curve fitting, parameterisation (or dimensionality reduction) and classification.

5.1 What is Machine Learning?

The objective is to use a computer to classify supernovae. In the next future, photometric supernova surveys are expected to observe a large number of these objects in the form of light curves. Of key importance is to classify these events.This is a problem in which, apart from the obvious knowledge of the input (the light curve), it is also known what the output (the supernova type) should be. What we don't know is how to transform the supernova light curves into supernova types, that is, we don't have any algorithm able to match a given supernova light curve to the right supernova spectral type. This lack of knowledge can be filled up using data: from observations it is possible to build a rule to match supernova light curves to supernova types, that is to match the input to the output. The rule is the process through which we transform input into output; it may not be completely identified, but it could be a useful and good approximation. Machine learning allows us to shift the process of learning this rule from human scientists to much faster computers, using tools that comes

from statistics and computer science. Thus, answering to the question on what machine learning is:

> *Machine learning is programming computers to learn the rule that transforms input into output using example data or past experience* (Alpaydin 2010).

The rule could be learned by finding patterns or regularities into data; once identified, the patterns may be used both to understand the process of transforming input into output and to make predictions on future data. As an example take that of a model defined up to some parameter: the learning process is the execution of a computer program to optimise the parameters of the model using the examples contained in the training set (Section 5.3) or past experience.

To be able to make future prediction, the learning algorithm has to build a mathematical model. The core task of the model is to *make inference* from a sample. To make reliable inferences the model is built using the *theory of statistics*.

Computer science enters in machine learning in two ways: first during training, when we need efficient algorithms to solve the optimisation problem; then in the representation of the learned model and its algorithmic solution that need to be efficient as well.

To measure improvements in the learning of a machine, in a way that bypasses philosophical questions such as "what is learning?" or "how do we measure knowledge?", it is useful to link *learning* to *performance* rather then knowledge by saying that

> Things learn when they change their behaviour in a way that makes them perform better in future (Witten et al. 2011).

Performance is a quantity that can be measured by simply observing present behaviour and comparing it to past behaviour.

Machine learning is widely used in a large number of applications. Email services uses machine learning techniques to filter spam. The input words to an Internet search engine are stored in order to learn which are your interests and optimise advertising. The theories of machine learning are used also in recognising handwriting in OCR software. Banks record past transaction of their clients and classify them as a low- or high-risk costumers, a precious information when they ask for a loan.

5.1.1 Input Representation

Before proceeding in explaining some insights of machine learning, it is important to have a look at the different forms the input might take. It takes the form of *concepts*, *instances* and *attributes*. What has to be learned is called a *concept description*. In the case of classification of supernovae this is the class to which each light curve in the data set belongs; this is a description of the concept that is intelligible in that it can be understood, and operational in that it can be applied to actual examples. The information we give to the learner takes the form of a set of instances, in the case of supernovae classification these are the light curves. Finally, each instance is characterised by the value of the attributes that measures different aspects of the instance, in our case the set of attribute are the parameters that will be determined by diffusion map, described in Chapter 8.

5.2 Three machine learning flavours

Machine learning can be divided into the two main categories of unsupervised and supervised learning, depending on the absence or presence of the output, respectively. In the recent years, a third approach has emerged, half way between the previous two, called semi-supervised learning. These are the three "flavours" of machine learning.

5.2.1 Unsupervised learning

Learning algorithms using this approach, are employed when the input data is provided without knowledge on what the output should be. The aim is to find the regularities in the input; in statistic this is called *density estimation*.

One method for density estimation is *clustering* where the aim is to find clusters or groups of inputs such that within each group, instances are more closely correlated then with instances of other groups.

5.2.2 Supervised learning

For supervised learning algorithms, both input and output are known. The aim is to calculate the model that maps input into output. The general approach is to assume a parametric model

$$y = g(\boldsymbol{x}|\boldsymbol{\vartheta}), \tag{5.1}$$

where $\boldsymbol{x}$ is the input vector, y is the output and $\boldsymbol{\vartheta}$ is the parameters vector. The machine learning algorithm have to optimise the parameters' values in order to minimise the error.

Classification is a supervised problem, for which the input (in our case supernova light curves) and the output (the supernova types) are given, and the solution is the best rule (in term of error) that associate light curves to supernova types.

5.2.3 Semi-supervised learning

This kind of learning algorithms are halfway between unsupervised and supervised learning schemes. During the learning phase, the algorithm is provided with input for which the output is known (labelled input) and input with unknown output (unlabelled input). The labelled input represent a sort of supervised information. A semi-supervised approach is useful if the distribution of unlabelled inputs can help in solving the problem. As an example, when looking for the best set of coordinates able to separate supernova classes, using unlabelled light curves helps in giving a more complete representation of the population than considering only the labelled inputs, usually less numerous. This is the approach of the method described in Chapter 8. See Chapelle et al. (2006) for an introduction on semi-supervised methods.

5.3 Training, Validation and Test Sets

A learning algorithm needs example data on which to be trained on. These data form the so called *training set*. For an instance of the training set we know all the attributes describing it (a simplification fitting the majority of problems) and also the output, such as the class to which the instance belongs. It is because we know the output that we can train the learning scheme on the training set; from these data the algorithm learns what is the solution to the

problem. An important feature of the training set is that it has to be a representative sample of the data to be classified, otherwise the results of the training cannot be applied to make predictions.

The classifier performance in classification problems is measured in terms of error rate. The classifier predicts the class of each instance: if it is correct, than it is counted as a success; if not, it is an error. The error rate is just the proportion of errors made over a whole set of instances, and it measures the overall performance of the classifier. Of course, the performance which we are interested in is the likely future performance on new data. The error rate on the training set is not likely to be a good indicator of future performance; this is because the classifier has been learned by the very same data and so any error estimate will be optimistic.

The errors produced during the training phase are of no importance for future error estimates. To predict the performance of a classifier on new data, one needs to assess the error rate of the method on a data set that played no part in the formation of the classifier. This independent data set is called *test set*; of curse, to be able to estimate error rates, all the instances in this set are classified, that is to say that we know also the output. Again, we assume that both the training and test sets are representative of the underlying problem. It is important that the test data is not used in any way to create the classifier.

For learning schemes that involve two stages, one to come up with a basic structure (that is the rule that associate supernova light curves to supernova classes) and the second to optimise parameters involved in that structure, or in situations where we might try out several learning schemes on training data and then evaluate them, we need a third set separate from training and from testing sets. This is called *validation set*.

To summarise, the processing of the data is as follow:

1. the training data is used by one or more learning schemes to produce a classifier;
2. the validation data is used to optimise parameters of those classifier, or to select a particular one, if more then one are available;
3. the test data is used to calculate the error rate of the final, optimised, method.

Once the error rate has been determined the test set could be bundled back into the training in order to maximise the amount of data used to generate the classifier. Also the validation set, after it has been used to determine the best model, it can be bundled back into the training to retrain the learning scheme.

5.3.1 Error Decomposition

The error rate of a particular learning scheme and training set can be decomposed into two different contributions.

The first source of error is *inductive bias* and it is proper of any learning scheme. Inductive bias is a quantity that measures how well the learning method matches the problem. In other words, inductive bias measures the persistent error of a learning algorithm that can't be eliminated even taking an infinite number of training sets into account, because it is due to the method's assumption in solving the problem. In inductive bias is included also the effect of the *noise* component that is generally unknown in practice.

The second error source in a learned model derives from the particular training set used, which is inevitably finite and therefore not fully representative of the actual population of

instances. The expected value of this error component is called *variance* of the learning method for that problem, and measures the fluctuation of the predictions around the expected value.

During learning, when a model has to be selected, the method has to trade off between bias and variance trying to keep them as lower as possible in order to avoid over fitting (that is, modelling also the noise component) and under fitting, so to obtain a simple model. Said differently, the learning scheme has to find the right balance between the model's complexity and its fit to the data, because a complex model with error rate near zero on the training set usually isn't reliable in its predictions on new data.

5.4 Summary

This Chapter described what is machine learning, and the main principles of how it works. The three approaches of unsupervised, supervised and semi-supervised learning have been introduced. In every machine learning application the use of training and test set is mandatory, for the algorithm to learn the rule mapping the input to the output, and for this rule to be tested with previously unseen data. This chapter described a little fraction of the machine learning techniques; for a comprehensive introduction see Alpaydin (2010), Witten et al. (2011) and Zumel et al. (2014).

The following Chapter will describe the pre-processing of the data set, which takes care of cleaning the light curves from astrophysical effects, such as reddening by Milky Way dust and time dilation. The introduction of K correction will be discussed, especially the difficulties arising when the only information is provided by photometry, and the consequences on the data set composition. Chapter 7 and Chapter 9 will see the application of two supervised machine learning techniques, Gaussian processes for light curve interpolation, and random forest to produce the classification model. The light curves parameterisation, carried out using the semi-supervised learning technique of diffusion maps, will be explained in Chapter 8.

Chapter 6

Light curves pre-processing

During the travel from the supernova to us, several phenomena intervene on the radiation, with the net effect of reducing the intensity and changing the shape of the light curves. There are two main alterations:

- absorption by host galaxy and Milky Way dust,
- time dilation due to the expansion of the Universe.

These two astrophysical effects have to be removed, to clean the data before interpolation.

There is a third correction to introduce, the K correction; it depends on the object redshift and observing filter. Due to Universe expansion, radiation emitted at some wavelength λ_e, is redshifted and then observed in a redder region of the electromagnetic spectrum. This difference between emitted and observed frequencies increases with redshift. The K correction is needed to bring back an observed light curve to the object rest-frame.

As illustrated in Chapter 8, parameterisation is performed via a pairwise comparison of light curves from different objects. To include the corrections described above is essential to a meaningful comparison of light curves from objects at different redsfhits. In Section 6.2 will be explained why, with photometric data alone, the calculation of the amount of K correction is not a trivial issue; describe different approaches to work around the problem will be described.

6.1 Correcting for absorption and time delay

6.1.1 Interstellar extinction

Dust is responsible for absorbing and scattering light, especially from the UV through the infrared. The process is complex, involving radiation wavelength, dust grains shape and composition, and the line of sight direction. Correction for dust extinction should consider the effects from dust both in the host galaxy and in the Milky Way. The data set from SNPhotCC provided for each supernova only extinction due to the Galaxy.

Extinction in the Milky Way has been widely studied, and can be now addressed using the mean Galactic extinction curve, which is parameterised by the relation:

$$R_V = \frac{A_V}{E(B-V)}, \tag{6.1}$$

the ratio between total absorption in V band, A_V, and colour excess $E(B-V)$. The total absorption A_V, is the quantity for which the light curves have to be corrected; it is trivially the product of colour excess and R_V. The colour excess $E(B-V)$ is provided for each supernova of the SNPhotCC data set. Concerning R_V, its mean value has been found to be ~ 3.1 by many studies: Savage & Mathis (1979); Seaton (1979); Cardelli et al. (1989); O'Donnell (1994); Fitzpatrick (1999). The extinction law used by SNANA in simulating SNPhotCC light curves combines what published by Cardelli et al. (1989) and O'Donnell (1994); colour excess values are calculated using the dust maps from Schlegel et al. (1998) (see Kessler 2014). When calculating the correction, caution has to be taken, since R_V describes extinction in band V in the Johnson-Morgan photometric system (Johnson & Morgan 1953), while SNPhotCC light curves are measure using DES *griz* filters. Hence, the mean value of R_V has to be adapted to these different filters. The correct R_V values are listed in Schlafly & Finkbeiner (2011, Table 6).

Concerning extinction from the host galaxy, the SNPhotCC data set provides no information. The approach usually employed to estimate the host A_V is through template fitting, host galaxy extinction being among the parameters (for an example see Holwerda et al. 2014). Such a use of templates would vanish the effort for a data-driven approach. As a consequence, light curves has not been corrected for host A_V.

Consideration on host galaxy extinction

It is of interest to briefly discuss the different effects that host galaxy extinction has on classification and clustering results.

In classification, a predefined set of classes (output) is imposed, and the light curves (input) will have to belong to one of these classes. In supernova classification, the effect host A_V has is to introduce degeneracy on the classes of type Ia and type Ib/c. This happens because a reddened light curve from a type Ia is similar to a non-reddened light curve from a Ib/c supernova. This kid of degeneracy may partly be cause of results from the test, which are reported in Section 9.2.2 and Table 9.2.

When performing clustering, the algorithm looks for clusters in a parameter space, there is no imposed output. If host galaxy extinction plays an important role in the SNPhotCC data set, this would reflect into having clusters of reddened objects distinct from cluster of non-reddened input. This represent an interesting future development.

6.1.2 Time dilation

The effect of cosmological time dilation, caused by the Universe's expansion, have to be corrected as well. This is fairly easy. The redshift can be defined as:

$$1 + z = \frac{\Delta t_{\mathrm{o}}}{\Delta t_{\mathrm{e}}}; \tag{6.2}$$

Δt_{o} is the time interval between two subsequent observations in the light curve. Inverting the Equation we obtain the time intervals in the supernova rest-frame, Δt_{e}, hence correcting for time dilation.

6.2 The issue of K correction

Radiation emitted from astrophysical objects is redshifted, with respect to rest-frame, by the Universe expansion. Thus, the radiation collected in one specific filter has been emitted at bluer wavelengths. To compare light curves in the same band, is a key step to extract parameters using diffusion map, as will be explained in Section 8.3. To a meaningful comparison, flux measurements in light curves from objects at different redshifts, have to be corrected to report them at a common rest-frame. Section 6.2.2 explains how to approach this issue using only photometric data. Even though an approximate solution can be implemented, this has non trivial effects on data set composition, not least on the training set. Due to reasons discussed in Section 6.2.3 and Section 6.2.4, this approximation cannot be used on the SNPhotCC data set. In Section 6.2.5 is explained a possible approach to understand how influential is K correction on the machine learned classification model performances.

6.2.1 Classical approach

K correction is usually calculated by means of a template spectrum for the source. The spectrum is reported to the object rest-frame, and its flux integrated over the wavelength range of the specific photometric filter; in this process also the filter response function is taken into account. This is the way to calculate the flux emitted at rest-frame in the given filter. By comparing emitted flux against measured flux, a correction is possible. Different supernova types contain very different features in their spectra, whose contribution in the integrated flux can make a huge difference. Moreover, a supernova spectrum evolves in time, thus the K correction at different times would be different. As a result, the knowledge of supernova type and phase is critical in calculating the K correction. It is clear that K correction cannot be calculated using the approach described so far, if the only information available is from photometry and not spectroscopy.

Nonetheless, exploiting multiband photometry and without needing any prior knowledge on the supernova type, an alternative method to calculate K correction can be developed; this is described in following Section 6.2.2.

6.2.2 An alternative approach

When multiband photometry is available, K correction at different epochs can be inferred directly from the light curves. The simulated sample from SNPhotCC provides, for each supernova, light curves in the four *griz* Dark Energy Survey filters. As a result, at each epoch it is possible to have a maximum of four measures, each of which is referred to a different wavelengths (the effective wavelength of each filter, obtained from Schlafly & Finkbeiner 2011, Table 6). Using these four values, an artificial very low resolution spectrum can be built. Such a spectrum for a supernova at redshift z_{SN} is then reported to rest-frame by means of the equation:

$$1 + z_{\mathrm{SN}} = \frac{\lambda_{\mathrm{obs}}}{\lambda_{\mathrm{rf}}} \tag{6.3}$$

Intensities at intermediate wavelength are approximated by linearly interpolating the two adjacent measures. The value of K correction is then calculated by getting the intensity from the rest-frame artificial spectrum at the effective wavelength of each filter λ_{obs}, refer to the dot-dashed line in Figure 6.1.

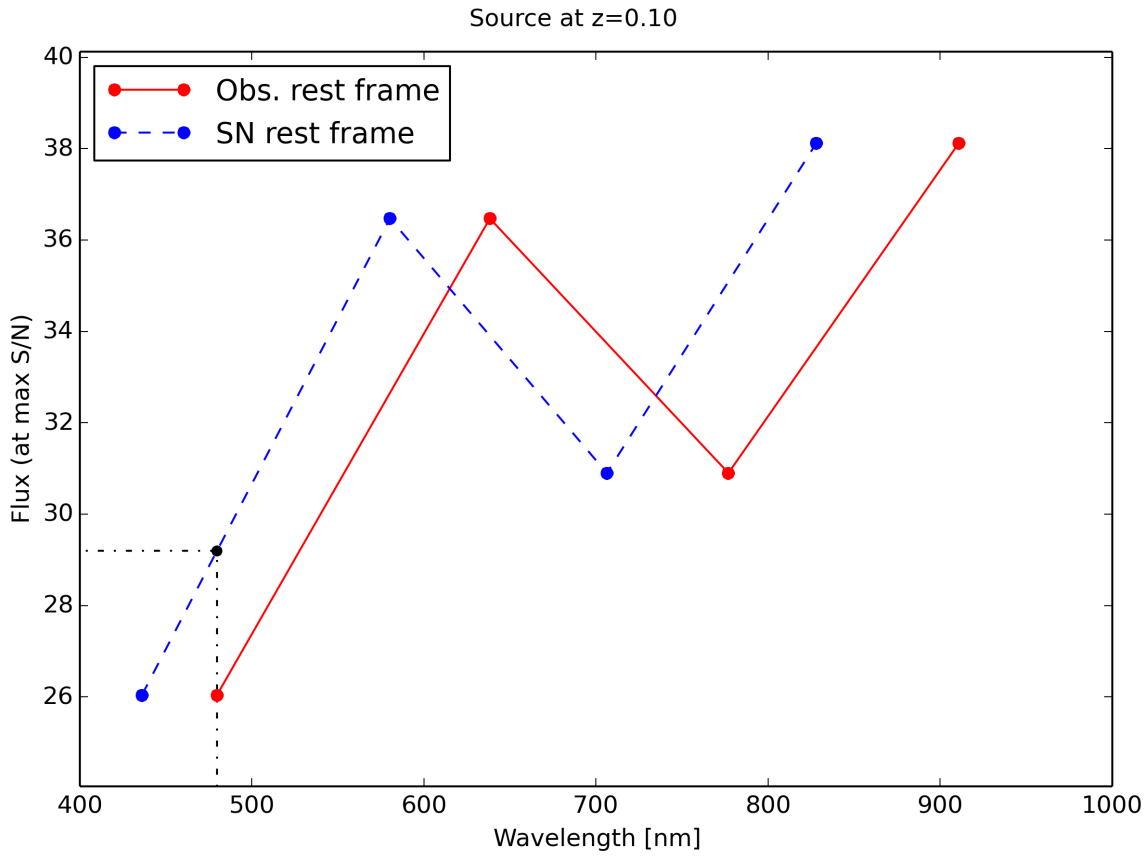

Figure 6.1: The alternative approach to K correction calculation is shown for an simulated object at redshift 0.10. Red solid line marks the spectrum, built using same day observations in the four *griz* bands. Blue dashed line is the same spectrum, blue shifted to the object's rest frame. Black dot-dashed line shows the flux emitted at the rest-frame g band, the quantity needed to infer K correction in that band, for that day.

In spite of being a raw approximation, this approach has the advantage to exploit the information on the supernova type encapsulated in each of those raw spectra.

6.2.3 Loss of data

The alternative method described above presents the critical issue of reducing the data set size in two ways. This is something to be concerned with in a data-driven project, since Shrinking the data set is reflected in a loss of information and eventually in a loss of accuracy in classification.

The first information loss occur when, at a specific epoch, an observation is not made in each of the available bands. As a consequence, for that epoch a raw spectrum cannot be built, thus no K correction can be calculated. Not K corrected flux measures cannot be included in the light curve, and hence data points are lost. A K corrected light curve with less data points then the original is reflected into a poorer interpolation, with higher uncertainties which will affect the classification process in a negative way.

The second cause of loss is connected to the objects redshift. Since the process of calcu-

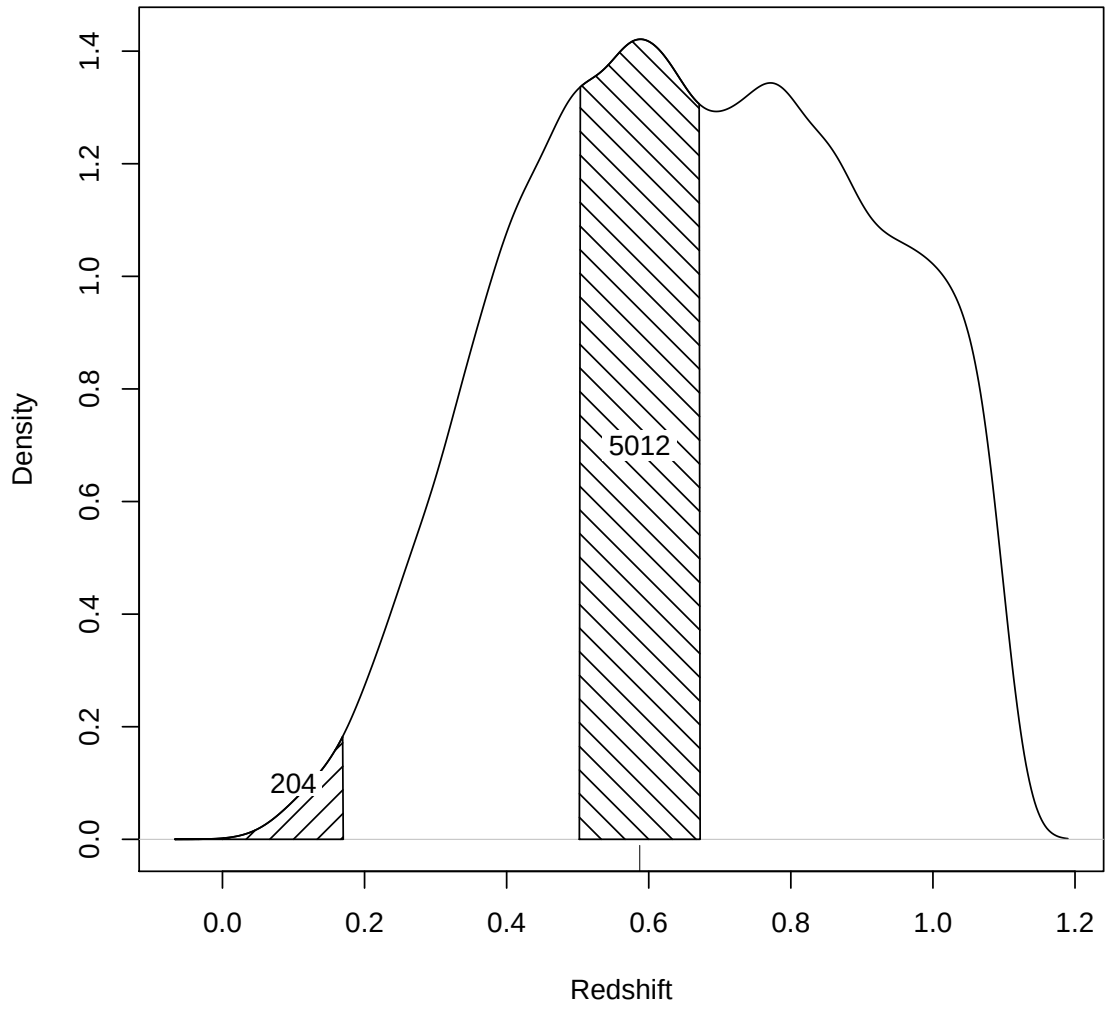

Figure 6.2: Redshifts distribution from supernovae in the SNPhotCC sample. The rug tick marks redshift at distribution's maximum. The shaded areas are intervals of 0.17 in redshift, the maximum redshift at which up to three bands can be K corrected.

lating K correction involves the shifting of a spectrum to bluer wavelengths, the information to correct the flux measured at a specific wavelength comes from a redder region (see Figure 6.1). Hence, due to the limited spectral coverage, the measures at the red endpoint of the spectra cannot be corrected. In applying the alternative approach to light curves in the SNPhotCC data set, it is not possible to calculate K correction for band z. Hence, all the information encapsulated in z band light curves is lost; this means that a quarter of the data set in terms of light curves is not exploited. Moreover, with increasing redshifts, the same problem will affect the other bands, to a limit where there would be no K correction for any band.

This second information loss can be limited by maximising the number of K corrected

bands. This is achieved by correcting light curves up to a specific redshift. In the DES *griz* setup, the maximum number of K corrected bands is three: g, r and i bands. This corresponds to a redshift interval of 0.17, obtained through Equation (6.3), where radiation emitted in rest-frame i band, λ_{rf}, is redshifted to z band (the observed wavelength λ_{obs}). As can be seen in Figure 6.2, an interval of 0.17 in redshift contains only ~ 200 supernovae, corresponding only to 1% of the SNPhotCC data set. Such a small subset is not statistically relevant both to obtain a meaningful parameterisation using diffusion maps, and to build a useful classification model through machine learning.

The problem can be mitigated by moving the observer rest-frame from redshift zero to a value near the maximum of the distribution in redshift, reported in Figure 6.2. Even if still small compared to the SNPhotCC data set, it is more statistically relevant than the previous K corrected subset. In what follows, this subset formed by $\sim 5\,000$ supernovae will be called $\mathcal{S}_2$.

Data set shrinkage has important effects also on the composition of the training set, which reflects on the machine learned classification model. The reduction in size described so far, excluding a fraction of the original data set, produces a new training set. For the classification model to distinguish among the three classes of Ia, Ib/c and II, the new training set have to be representative of the three types. This is explored in what follows.

6.2.4 Effects on the training set

Training set composition, in terms of number of examples per type, is extremely important: the machine learned classification model is built using training set light curves as ground truth. Hence, before shrinking the data set as described in Section 6.2.3, effects on the resulting training set have to be investigated. In particular, new training set size and composition have to ensure a good sampling of each class the model has to distinguish. The size has to ensure the training set is statistically significant. As for the composition, the three classes of Ia, Ib/c and II have to be well represented, so to give enough examples for the machine learning algorithm to identify the most important parameters necessary to build an effective classification model.

Regarding subset $\mathcal{S}_2$, the resulting training set size amounts to ~ 300 supernovae; compared to $\mathcal{S}_2$ size, ~ 5000 supernovae, this is a good total size. However the composition, reported in Figure 6.3, is an issue. Based on the experience gained in my master thesis (De Pascale 2011), the 21 Ib/c and 66 type II supernovae are likely to provide very little information to the machine learning algorithm. With so few examples, the machine learned classification model would not be able to correctly identify supernovae belonging those classes. This would result in a big mis-classification rate and consequent low confidence.

A resulting low-confidence classification model makes this alternative method not worth to be used. Nonetheless, it is possible to design an experiment to understand how relevant in K correction with respect to the classification model performances.

6.2.5 A data set at rest-frame

K correction is fundamental to a meaningful pairwise comparison among light curves at different redshift, a key step in parameterisation. Unfortunately, this correction can be neither calculated using template spectra nor derived from the light curves.

What follows describes an experiment with which to assess K correction influence on the performances of the classification model built by random forest, the machine learning

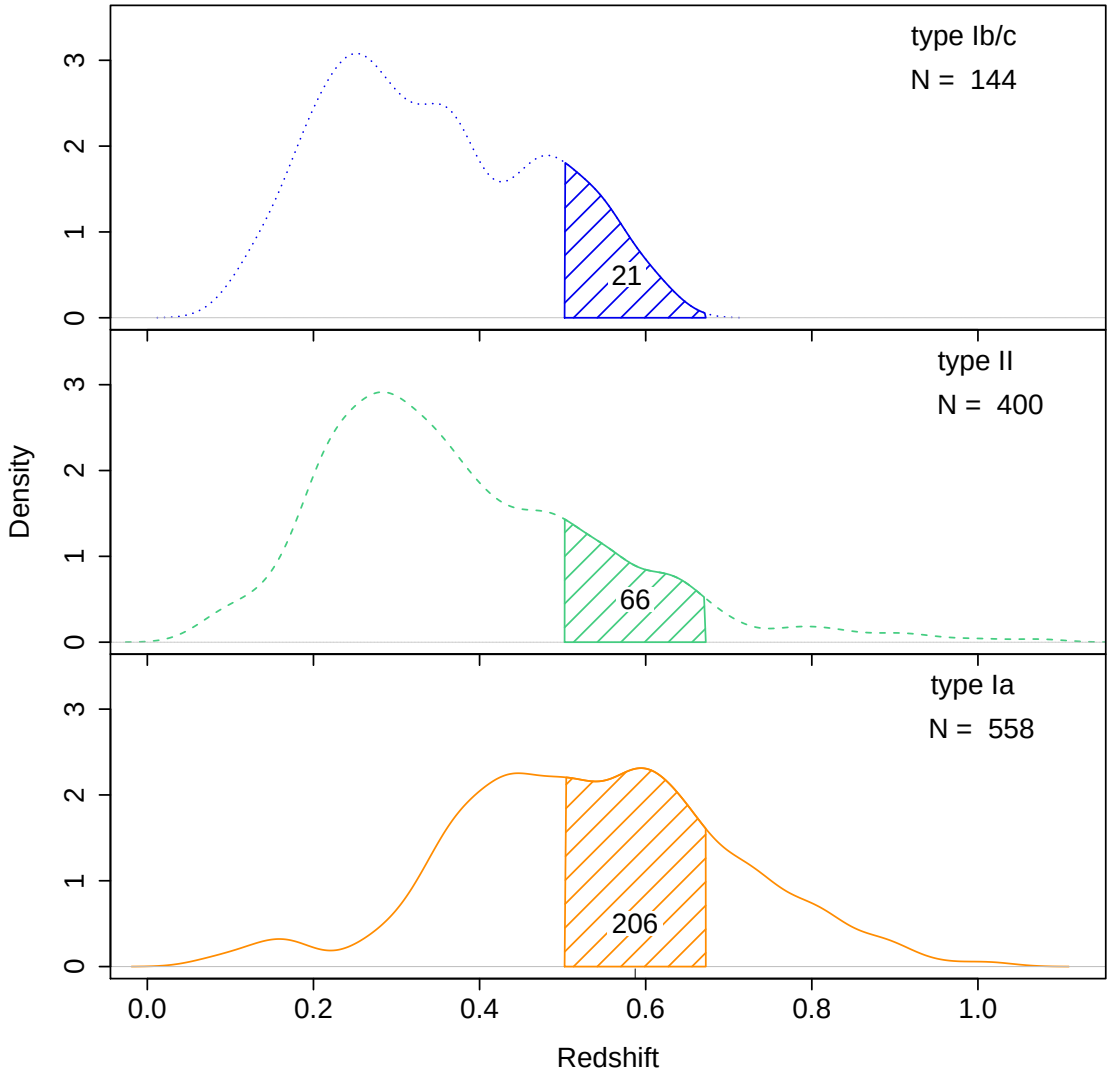

Figure 6.3: Distributions in redshift of supernovae belonging to the training set, distinguished by type. In each panel is show the total number of objects for each type. The number reported inside the shaded areas equals the supernovae within that redshift interval.

algorithm used.

To this extent, a comparison classification model has to be built using light curves for which K correction can be avoided. Stated differently, the supernovae have all to be in the same rest-frame. Comparison of performances from the two classification models, will give indication on K correction influence.

Same rest-frame supernovae data set, in what follows called $\mathcal{S}_{\mathrm{rf}}$, can be generated using the simulation software SNANA (Kessler et al. 2009). The composition of $\mathcal{S}_{\mathrm{rf}}$ has to reflect the composition of SNPhotCC as reported in Table 4.1; this is essential to assure, as much as possible, that differences arising from model comparison are caused by K correction.

6.3 Pre-processing conclusions

Parameterisation is obtained by comparing pairwise interpolated light curves. Hence, before performing interpolation, it is essential to remove from light curves the astrophysical effects of dust reddening and time dilation. Information provided in SNPhotCC data set, allows only to correct for Milky Way absorption. Contribution of host galaxy extinction could be visible only as a posterior result, after parameterisation, as separated clusters in parameter space. Time dilation is corrected using either spectroscopic redshift or photometric redshift from the host galaxy.

It would be also fundamental to report all light curves to the same rest-frame, introducing K correction. Unfortunately this is not possible. The classical approach needs the prior knowledge of the supernova type, and extracting K correction directly from the photometric observations would produce a catastrophic loss of data. However, using SNANA a new data set at the same rest-frame can be generated; a classification model build on these light curves could be used as comparison, to understand how influential is K correction.

Chapter 7

Light curves interpolation: non-parametric inference

The classification method proposed has been introduced as having the characteristic of being data-driven. This means all information necessary to classify supernovae is extracted from the observational data. This is in contrast with methods using what is called "prior knowledge", something considered true and useful to carry out the result. An example of such methods is template fitting, using light curves of known objects to recognise the type of a supernova.

The classification method proposed is organised in four main steps, described in Section 4.3. Light curves parameterisation, essential to build the classification model, is performed by pairwise comparison. To this extent, SNPhotCC data set poses the difficulty of being formed by SNe sampled on irregular time grids (as would happen in real survey). To circumvent this problem, light curves have to be expressed on the same, regular, time grid, via interpolation. In the attempt to not make use of prior knowledge, interpolation is carried out using a data-driven approach. In this way, light curves of unknown types of transients can be correctly interpolate, without being limited by the prior choice of templates.

Data-driven approaches to interpolation are gathered in the branch of non-parametric inference. From Wasserman 2006, pg. 1, non-parametric inference is used "*... to infer an unknown quantity while making as few assumptions as possible*", and keeping them as weak as possible. A better and more precise definition of non-parametric inference is difficult to give. Curve estimation is one of the problems that can be approached using non-parametric regression.

This Chapter describes light curve interpolation using Gaussian processes (GPs) a non-parametric modelling technique (Rasmussen & Williams 2006, MacKay 2003, Chapter 45, Bishop 2006, Section 6.4), among with the results obtained on the SNPhotCC data set.

A crucial complication, discussed in Section 7.2.4, has been the difficulty to obtain "pure" data-driven interpolation. The kind of assumptions to be introduced, to obtain smooth and continuous results, may be a problem to further identify new classes of transients.

7.1 Gaussian processes

Gaussian processes are non-parametric methods extensively used both for regression and classification in the machine learning community. Regression is, indeed, a task that can be preformed by supervised machine learning techniques, algorithms learning a model by

mapping inputs to outputs from examples. Recalling the definition given in Chapter 5, supervised means that the output is known for each input in the training set; for light curves the input is the epoch and the output is the measured (or simulated) flux. In this perspective, the set of N observations $\{x_n, t_n\}_{n=1}^{N}$ forming a light curve is seen as the training set, x_n denoting the input epoch, and t_n the output flux, called target value in the machine learning nomenclature.

Given the training set, through regression we wish to infer the function $y(x)$ underlying the data, so to make predictions at new epochs which are not among the training data. To do this, assumptions on the characteristics of $y(x)$ has to be introduced, otherwise any function consistent with the training data would be equally valid. Such assumptions can be introduced by giving a probability to every possible function, with high probability to functions considered more likely. Although this is a rather loose statement, especially because there are infinite possible functions, it is exactly what a GP does. Let see this in more detail.

The inference of $y(x)$ from a regression model, can be described by the posterior probability distribution given by the Bayes theorem as follows

$$P(y(x)|\boldsymbol{t}_N, \boldsymbol{X}_N) = \frac{P(\boldsymbol{t}_N|y(x), \boldsymbol{X}_N)P(y(x))}{P(\boldsymbol{t}_N|\boldsymbol{X}_N)}, \tag{7.1}$$

where $\boldsymbol{t}_N \equiv \{t_n\}_{n=1}^{N}$ denotes the set of target values, while $\boldsymbol{X}_N \equiv \{x_n\}_{n=1}^{N}$ is the set of input values. Focusing on the right-hand side, the term $P(\boldsymbol{t}_N|y(x), \boldsymbol{X}_N)$ expresses the probability of the target values given the function $y(x)$, which in regression problems is often assumed to be a Gaussian distribution (MacKay 2003). The term $P(y(x))$ is the prior distribution on functions assumed by the regression model. Gaussian process modelling defines the prior probability $P(y(x))$ directly on the function space. It can be thought of as a generalisation of Gaussian distribution over a finite vector space to a function space of infinite dimensions. This is how GPs define a probability for every possible function.

In analogy with Gaussian distributions, fully specified by a mean and a covariance matrix, a Gaussian process needs a mean function and a covariance function. In most applications, including this on supernova light curves, there is no prior knowledge on the mean function, hence this is set to zero. It results that all the prior information on $y(x)$ in expressed by the covariance function $k(x, x')$, which is evaluated at any two values of the input set $\boldsymbol{X}_N$. Mean and covariance functions are two expectation values of $y(x)$ expressed by

$$m(x) = \mathbb{E}\left[y(x)\right], \tag{7.2}$$

$$k(x, x') = \mathbb{E}\left[(y(x) - m(x))(y(x') - m(x'))\right]. \tag{7.3}$$

The function $y(x)$ is assumed to be a single sample from the Gaussian distribution defined by the Gaussian process

$$y(x) \sim \mathcal{GP}(m(x), k(x, x')), \tag{7.4}$$

following the notation in Rasmussen & Williams (2006). In is important to note that. as Gaussian distributions, Gaussian processes naturally take into account all possible functions.

Focusing on Equation (7.3), the covariance between training set inputs is defined in terms of the kernel function $k(x, x')$. The kernel function thus encodes all the prior knowledge on supernova light curves we want to include, and clearly has to be chosen consequently. The following sections introduce the essential prior information to be provided by the kernel function to the GP for supernova light curves regression, among with the two kernel functions employed. Results are discussed to understand how far can be pushed the proposed data-driven approach exploiting GP in this context.

7.2 The kernel function

The kernel functions defines the covariance for a Gaussian process; as such, it is imperative for $k(x, x')$ to be symmetric and to give rive to a positive semidefinite covariance matrix K.[1] Thus, any function satisfying these requirements, its a viable kernel function.

To perform regression of supernova light curves, the essential information $k(x, x')$ has to provide to GP are about smoothness and continuity; additionally can be included the presence of fast rise and slow decline.

These assumptions are enclosed in the kernel functions in term of their shape, which is controlled by a set of parameters, called *hyper parameters*, since not directly connected to the curve to estimate. Although it might seem a contradiction to introduce parameters in a data-driven approach, GP hyper parameters have a loose control on the estimated curve; in determining the regression result the data have much more relevance. Indeed, under the restrictions imposed by the kernel function, its the data governing the regression process. After hyper parameters optimisation, the estimated light curve will always follow the trend set by the observations; the only problems arising could be of over- or under-fitting.

Section 7.2.1 describes the two different kernel functions adopted to carry out light curve interpolation using Gaussian processes, namely the squared exponential and the rational quadratic. After illustrating the problems of optimisation in Section 7.2.3, the results obtained with the two kernels are discussed in Section 7.4, along with the issues posed on the identification of transients different from supernovae by the choice of kernel and constraints on hyper parameters.

7.2.1 Two kernel functions

Among the essential prior information listed in previous section, continuity and smoothness are both provided by the two kernel functions that have been tested. They achieve this by simply assigning a covariance proportional to the similarity between pair of input data (the epoch in the light curve): the near in time two flux measures are, the higher the covariance.

Regarding the third assumption to include, it says that the GP performing regression has to model an astrophysical process which is expected to have two different time scales, one short to describe the fast rise and a long one characteristic of the decline. We will see that this prior information in the one introducing more issues in regression and can be a potential cause for not identifying new classes of transient. What follows describes the two kernel functions used to interpolate the observed light curves, the set of training inputs is denoted by $\boldsymbol{X}$, as in Section 7.1.

Squared exponential

The squared exponential (SE) is the most used kernel function. It allows for the introduction of the two prior information of smoothness and continuity. Furthermore, this kernel assumes that the function to be estimated has just one characteristic length scale over which the flux undergoes a sizeable variation. The mathematical form for SE kernel is as follow, with $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ denoting two generic input set of training light curves:

$$k_{\mathrm{SE}}(\boldsymbol{X}_i, \boldsymbol{X}_j) = h^2 \exp\left(-\frac{|\boldsymbol{X}_i - \boldsymbol{X}_j|^2}{2\ell^2}\right). \tag{7.5}$$

[1] A positive semidefinite matrix satisfies $\mathbf{v}^\top K \mathbf{v} \geq 0$, for all vectors $\mathbf{v} \in \mathbb{R}^n$.

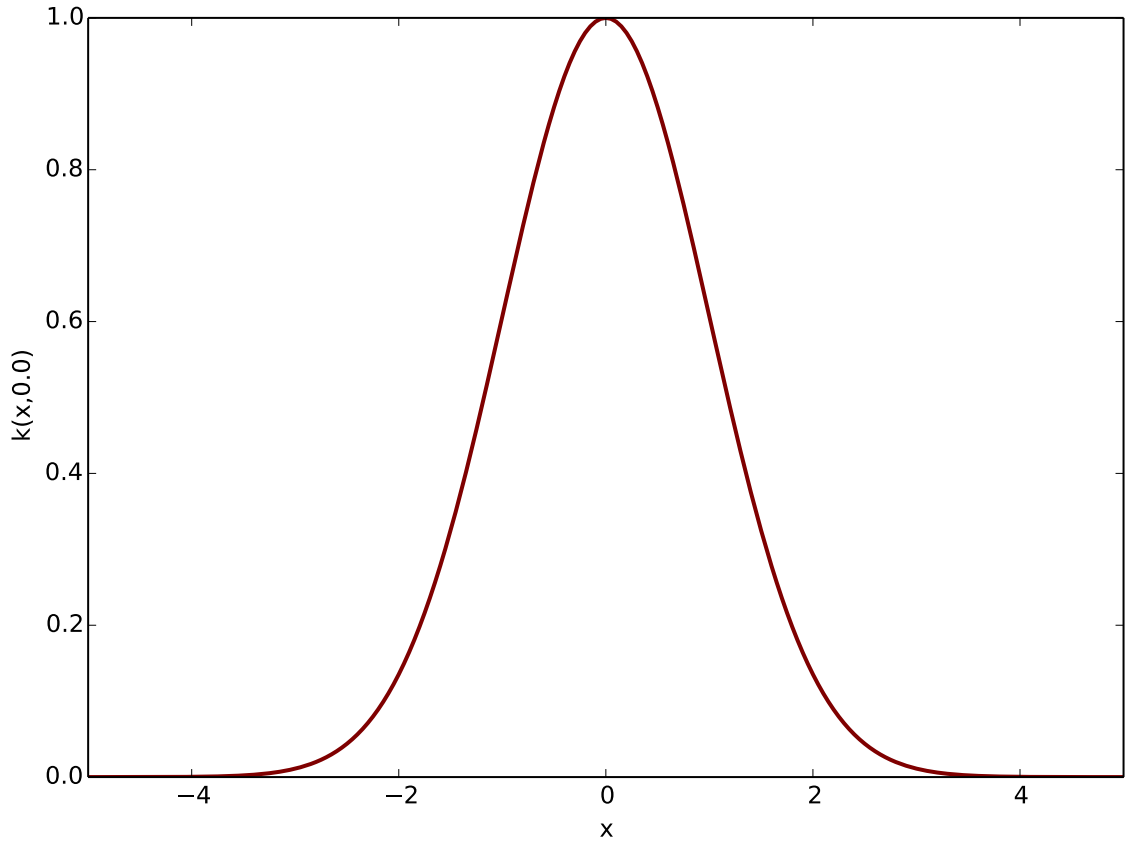

Figure 7.1: Shape of the squared exponential kernel, showing the correlation around input value $x = 0$. Length scale hyper parameter $\ell = 1$.

SE kernel has two hyper parameters, h and ℓ. The first controls the amplitude of the light curve, the range in flux over which the light curve vary significantly; the second controls the length scale of the flux variation. The correlations defined by the squared exponential between data points will be always positive. Being a negative exponential, function of the squared difference between two epochs of observation, the $|\boldsymbol{X}_i - \boldsymbol{X}_j|^2$ term, the correlation will be stronger for observations near in time, and weak otherwise. This assures the smoothness of the estimated light curve, avoiding discontinuities when predicting the flux at an epoch between two near observations.

The assumption of a single time length scale describing the significant variation in flux may seem too loose. As anticipated, supernovae light curves have shown to be characterised by two different length scales, connected to physical phenomena taking place during the explosion, such as the decay of ^{56}Ni through ^{56}Co to ^{56}Fe in type Ia. Nonetheless, avoiding to introduce more detailed assumptions favours the possibility to classify new types of transients. From this viewpoint, the SE kernel can be seen as a minimum assumption to interpolate transients light curves.

Rational quadratic

The rational quadratic (RQ) kernel brings in the GP regression the assumption that variations in flux in a light curve may take place on more then one length scale. This function can be seen as a sum of squared exponential kernels with different hyper parameters ℓ (Rasmussen & Williams 2006):

$$k_{\mathrm{RQ}}(X_i, X_j) = \left(1 + \frac{|\boldsymbol{X}_i - \boldsymbol{X}_j|^2}{2\alpha_{\mathrm{RQ}}\ell^2}\right)^{-\alpha_{\mathrm{RQ}}}. \tag{7.6}$$

This kernel hyper parameters are ℓ and the exponent α_{RQ}. For $\alpha_{\mathrm{RQ}} \to \infty$, this kernel becomes the squared exponential. Has for the SE kernel, the smoothness of the interpolated light curve

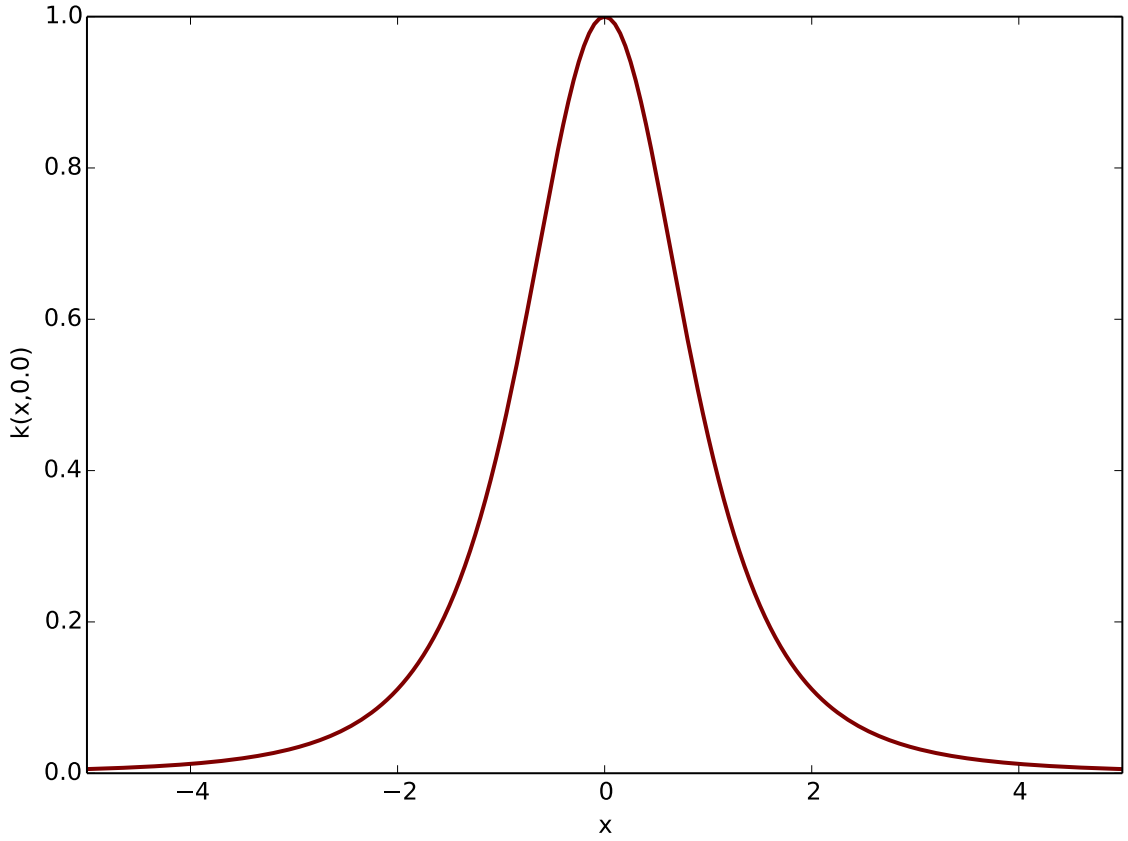

Figure 7.2: Shape of the rational quadratic kernel with $\ell = 1$ and $\alpha_{\mathrm{RQ}} = 1$. This example shows the correlation of $x = 0$ and other $x \in \mathbb{R}$ on the x-axis. Compared to the SE kernel in Figure 7.1, RQ is narrower with tails going to zero more slowly.

is assured by the kernel being function of the absolute squared difference $|\boldsymbol{X}_i - \boldsymbol{X}_j|^2$.

Thanks to the assumption of flux variations taking place on more then one length scale, the interpolating functions are expected to interpolate the data with more confidence than the SE kernel. However, we will see in Section 7.2.3 and Section 7.4 that RQ kernel introduces over-fitting issues, thus revealing to be not the correct kernel.

7.2.2 Code snippet

Light curve interpolation has been implemented in `Python` language exploiting the `GPy` package developed by Machine Learning Department of Sheffield University (The GPy authors 2012–2014). Listing 7.1 reports an example on how `GPy` is used to perform GP regression. The first step is to specify the kernel to use, then the regression model can be built and the hyper parameters optimised to best interpolate the data. Finally the optimised model is used to predict flux intensities over a regular time grid.

```
import GPy

"""Kernel setting
Radial Basis Function (RBF) kernel is the SE kernel
"""
kernel = GPy.kern.RBF(input_dim=1, variance=1., lengthscale=1.)
#kernel = GPy.kern.RatQuad(1)

"""Data read-in extracts from input ASCII file
- epoch
- flux
- fluxErr (assumed to be Gaussian)
"""

"""Gaussian process model setting up with read-in data.
Not constant ‘fluxErr‘ imposed the used of heteroscedastic
regression.
"""
gpModel = GPy.models.GPHeteroscedasticRegression(epoch, flux, kernel)
gpModel['.*Gaussian_noise'] = fluxErr

"""GP regression model hyper parameters optimisation. Method
‘optimize_restarts‘ selects random initialisation for the
parameter values, optimizes each, and sets the model to the
best solution found.
"""
gpModel.optimize_restarts(num_restarts=10, parallel=False,
                          robust=True)

"""Flux prediction on regular time grid
"""
predictedFlux, variance = gpModel._raw_predict(epochOnRegularGrid,
                                               full_cov=False)
```

Listing 7.1: Example code to set up kernel function and GP regression model using `GPy` functionalities.

7.2.3 Hyper parameters optimisation

To optimise kernel functions hyper parameters, means to find the best regression curve for a specific set of observations and assumptions. Hyper parameters optimisation is executed by `GPy` through the maximum likelihood estimation (MLE) technique. As function of the hyper parameter value, the likelihood states how well the model fits the given data set. Finding the values of the hyper parameters at which the likelihood is maximised means, in principle,

to have found the model that best fits the data. A description of MLE can be found in Section 3.4.3 of Feigelson & Babu (2012) and in Chapter 9 of Wasserman (2010).

As default, the method `gpModel.optimize_restarts(...)` provided by `GPy` package (see Listing 7.1) makes use of the Broyden–Fletcher–Goldfarb–Shanno (BFGS) optimisation algorithm. BFGS is an iterative method which approximates Newton's method in seeking stationary points (minima or maxima) for a real valued function; the position of the stationary point is to be read as the best value for the parameter to be optimised. To update the position of the stationary point, Newton's method uses the function's second derivative, or the Hessian matrix for functions with more then one parameter.

The approximation introduced by BFGS is to avoid to directly evaluate the Hessian, but rather it is updated with matrices specified by gradient evaluation. According to Nocedal & Wright 2006, pg.24, BFGS is one of the most popular members of the class of quasi-Newton's methods.

Compared to gradient descent algorithm, BFGS takes a more direct route to the stationary point.

The main issue in optimisation, is for the optimiser to get trapped in a local stationary point, which corresponds to a non optimal value for the parameter.

As example application, consider the length scale hyper parameter ℓ; the value identified by MLE through BFGS algorithm could be either too small or too big with respect to the best value. In the first case, the predicted light curve will "over-fit" the data; it will pass through every observed point, with consequent high oscillations and big uncertainties between each of them, as reported in Figure 7.3. Conversely, when the optimised value for ℓ results to be much bigger then the best one, the predicted light curve is "under-fitting" the data, passing far from every point, see Figure 7.4 for an example.

7.2.4 Preventing bad optimisation

As shown by Figure 7.3 and Figure 7.4, BFGS algorithm failed to converge to the best ℓ value for some light curves. These are problems of bad optimisation, and can be solved either by changing the optimisation algorithm, or by introducing prior information on the hyper parameter, such as a range of realistic values.

Tests on other optimisation algorithms, such us gradient descent, showed no improvement while increased computation time. Thus, it has been preferred to introduce some prior information.

Prior on hyper parameter

A prior on hyper parameters introduces assumptions on which are realistic values, in order to obtain a good interpolation function. Focusing on the length scale hyper parameter, a realistic value for supernova light curve would be in the interval between five and twenty days. It is not realistic to have ℓ set to values smaller then one day: the flux of a supernova does not change considerably on this scale.

The better way to introduce the prior information is by using of a "prior function". By doing so the optimiser would search the likelihood maximum in the region containing what is expected to be a reasonable values for the hyper parameter. In probability terms, the prior function is a probability density function multiplying the likelihood function, lowering the probability of unrealistic values.

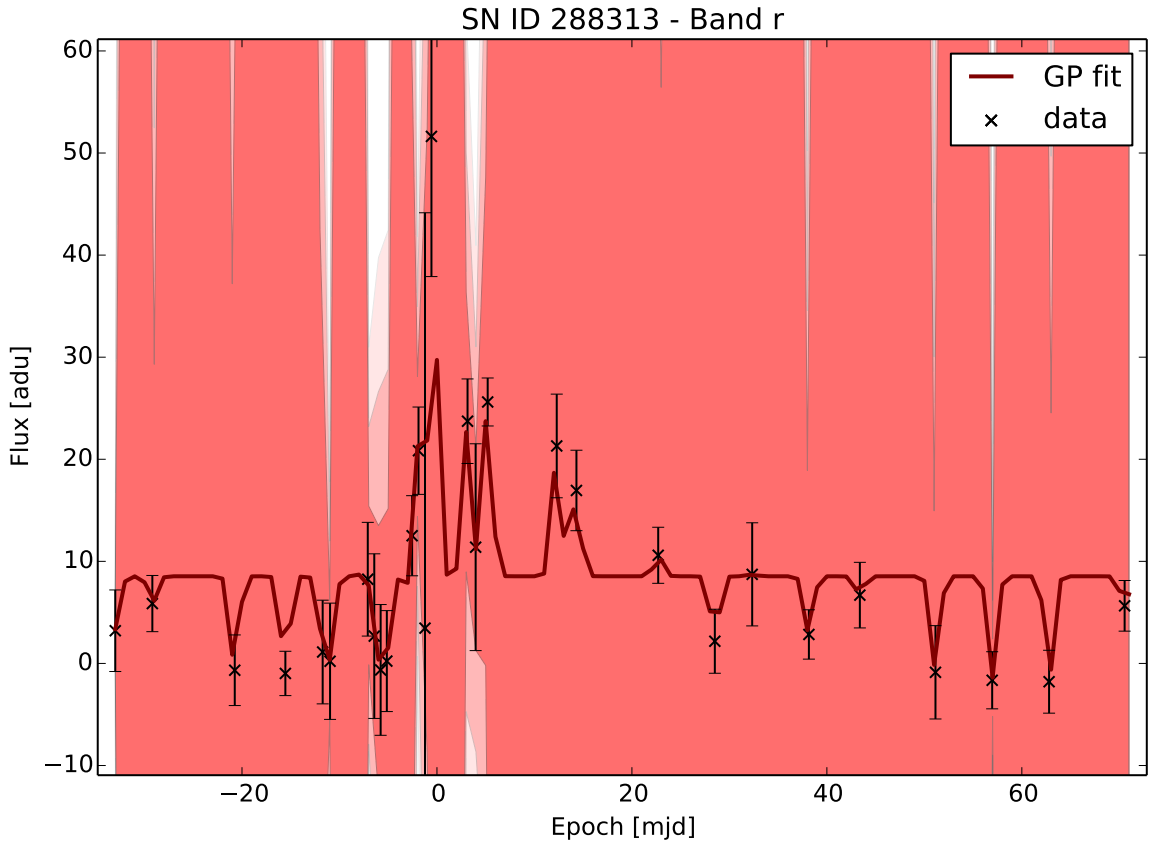

Figure 7.3: Example showing over-fitting problem. The estimated value for the length scale parameter ℓ has been optimised to a too lower value, giving a fit tending to pass through every data point. The kernel used is a squared exponential, with prior on ℓ. The light curve is from the SNPhotCC data set.

Prior function for length scale hyper parameter

Concerning characteristic flux variation time scales, values smaller then one day are not to be expected. Thus, a function approaching zero near the origin is a valid prior function for ℓ. Among the prior functions available in `GPy`, the most suitable in these terms is a Gamma probability density function (PDF):

$$f(x;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \tag{7.7}$$

with α the "shape" parameter, β the "rate" parameter, and x the epoch of observation. Parameters α and β have to be set so to push the length scale parameter ℓ away from small values.

Prior function for rational quadratic exponent

Regarding the rational quadratic hyper parameter α_{RQ}, tests showed that small values should be preferred. As for hyper parameter ℓ, also this prior function was a Gamma PDF, with parameters set to ensure high values near the origin.

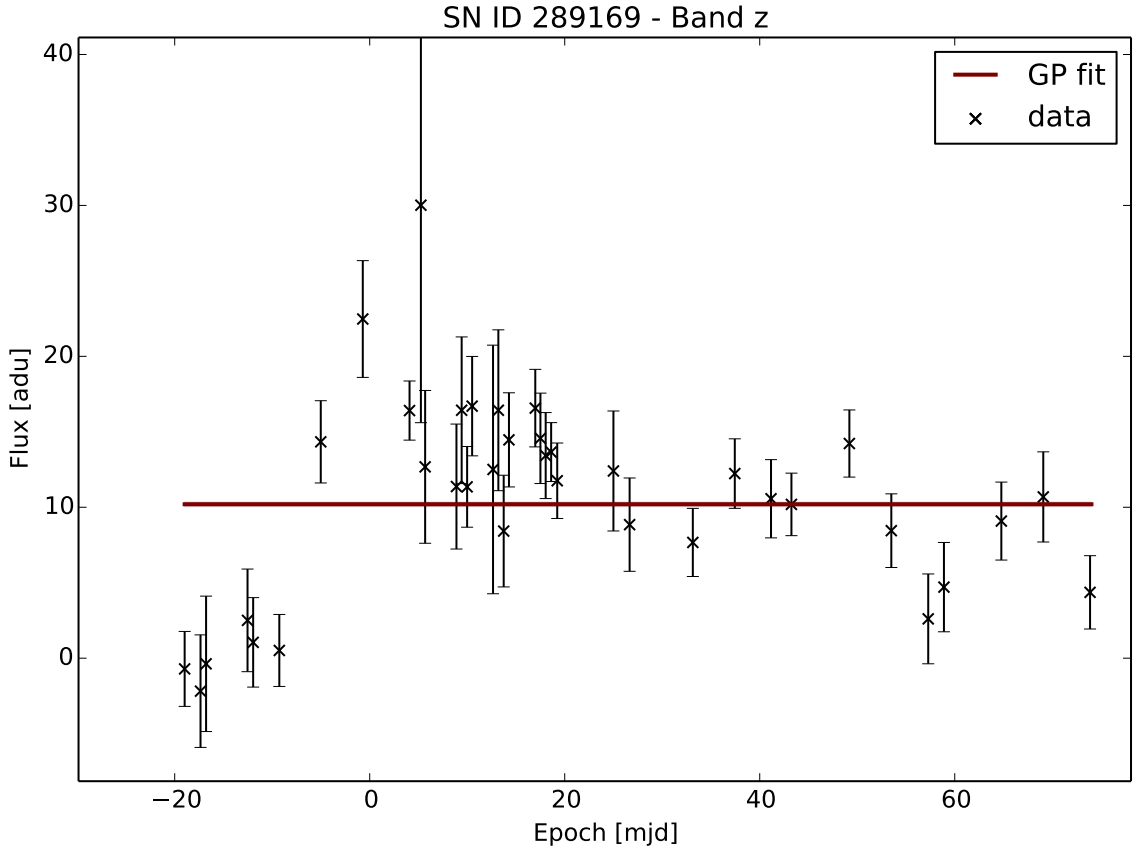

Figure 7.4: Example showing the under-fitting problem. Here the hyper parameter ℓ has a value too low to produce a light curve well representing the observations. The kernel used in this case is a rational quadratic, with prior on ℓ, as described in Section 7.2.4. The light curve is from the SNPhotCC data set.

Package `GPy` offers the possibility to associate a prior function to kernel's hyper parameters:

```
gpModel['.*lengthscale'].set_prior(GPy.priors.Gamma(alpha, beta))
```

Listing 7.2: Example code using `GPy` to set up a prior function for hyper parameter ℓ.

Constraining the hyper parameters

A second approach to avoid having the optimiser converging to a local stationary point, is to directly set the hyper parameter to a random value in the range containing realistic estimates, and from there starting the optimisation process. Regarding the length scale hyper parameter ℓ, this approach demonstrated to work better than introducing a prior function. To constrain the value of an hyper parameter in `GPy`, the command in Listing 7.3 as to be invoked before optimisation:

```
gpModel['.*lengthscale'].constrain_fixed(randomLength)
```

Listing 7.3: `GPy` instruction to constrain the hyper parameter ℓ to a random value, to avoid convergence to the wrong stationary point in the likelihood.

Final notes on optimisation

It is worth to note that these two techniques attempting to avoid bad optimisation, can introduce a different kind of bias in the recognition of the event, since they are biasing the Gaussian process to fit a supernova light curve. If this classification technique is to be extended to deal with events different from supernovae, such as AGN (which have much different length scales), these objects could potentially never be identified and correctly classified. The final result would be additional noise in the classification. Furthermore, the use of prior functions increases the number of parameters of a classification technique designed to be data-driven. For these reasons, and because the inclusion of prior functions did not bring any improvement (see Section 7.4), they have been discarded as a way to avoid bad optimisation, in favour of hyper parameter constraining.

7.3 Considerations on negative fluxes

Light curves in the SNPhotCC can contain negative flux measures. As reported by Kessler et al. (2010), simulated fluxes are defined as calibrated following the equation:

$$\texttt{FLUXCAL} = 10^{(-0.4 \cdot m + 11)} + \text{noise}, \tag{7.8}$$

where m is the modelled AB-magnitude. The noise contribution includes Poisson fluctuation, sky noise and CCD noise. The addition of noise can result in negative fluxes. These correspond to measures below the threshold limit of the optical system; in terms of flux, these are lower limits.

The problem of upper and lower limits (or censoring) is well known in statistics. Indeed, a whole category of statistical methods to recover information from censored data have been developed, and goes under the name of *survival analysis* (see Feigelson & Babu 2012, Chapter 10; Helsel 2005; Kleinbaum & Klein 2005). Negative measures should be included in the interpolation using the appropriate techniques from this branch of statistics. At the time of writing, the integration of survival analysis techniques in `GPy` is under development. Instead of ignoring the values or substituting them with a constant (an approach strongly discouraged by Helsel (2005)), the negative value were kept has they are, waiting for the survival analysis techniques to be available.

7.4 Overall results

Light curve interpolation was performed using the *heteroscedastic regression* model from `GPy` (see Listing 7.1). Adopting this kind of model is extremely important, since every point of a light curve, due to variable observing conditions, comes with a different error.[2] The strength of the heteroscedastic regression model is to give less weight to more uncertain measures, so for them to affect less the interpolation process.

[2]From the ancient Greek, "heteroscedastic" means "different dispersion".

Following the discussion in the previous sections, the light curves from SNPhotCC data set have been interpolated using three different Gaussian processes, in terms of kernel function and priors on hyper parameters:

1. GP with squared exponential kernel and prior function on the length scale hyper parameter ℓ,

2. GP with squared exponential kernel and constraining on starting value for ℓ, as described in Section 7.2.4,

3. GP with rational quadratic kernel and prior functions on both ℓ and α_{RQ}.

For each supernova, the light curve in r band was used to set the zero point to which to align light curves in other bands. For this reason from the analysis were discarded all the interpolation producing a flat r band light curve. All supernovae not having such a light curve were discarded too.

To assess which of the three Gaussian processes produced the better results, a small subset of the interpolated light curves was checked by eye. This revealed that the best results were obtained by the GP using the squared exponential kernel with constraining on starting value for the length scale hyper parameter. With such a Gaussian process, more light curves got interpolated, and thus can enter the next stage of the analysis: 21 200 against $\sim$ 19 500 with the rational quadratic kernel and $\sim$ 19 070 with the squared exponential including prior functions.

Gaussian processes with prior functions on hyper parameters, produced Regression curves in majority over-fitted and under-fitted, as is shown by central and bottom panel of Figure 7.5, Figure 7.6 and Figure 7.7. This shows that prior functions described in Section 7.2.4 were not strong enough. Indeed, the integral of a Gamma probability density function , is by definition of PDF equal to one. Since the Gamma PDF extends to infinity on the x axis, moving its maximum to high x acting on α and β, makes it its value on the y axis to decrease. In defining the prior on ℓ, the best interpolations were obtained by a very wide Gamma PDF, with a resulting very low maximum. Even though it achieved the best interpolations, with respect to other Gamma PDFs, the prior was not influential enough to avoid over-fitting.

Considering the rational quadratic hyper parameter α_{RQ}, the prior function was easier to set, since high probabilities at low x are much more simple to obtain with a Gamma PDF.

The Gamma PDF as prior on length scale hyper parameter proved to be difficult to handle, and did not provide improvements to GP regression, even with different kernels. For this reason, the use of priors on hyper parameters was discarded, and the regression was carried out using the SE kernel with constraints on length scale.

7.5 Normalisation and zero point estimation

Before proceeding with light curve parameterisation, described in Chapter 8, the interpolated light curves are normalised and aligned on the time axis. The alignment is necessary to measure with accuracy the similarities between light curves of different supernovae. Following the approach described in Richards et al. (2012), the zero point in time, x_0, is set to be the epoch of flux maximum in r band. For light curves where this maximum corresponds to a time endpoint (un-peaked light curves), x_0 is estimated by cross-correlation with peaked light

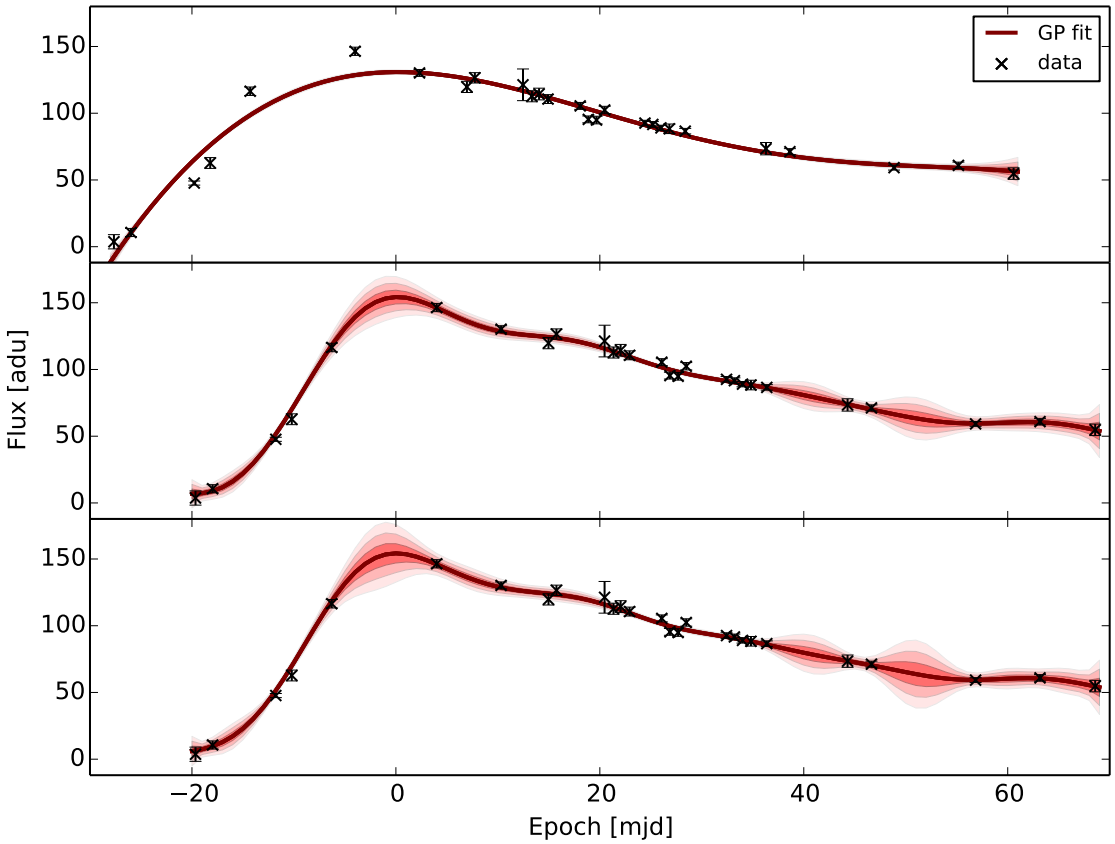

Figure 7.5: Comparison of interpolation of the same simulated light curve using the three different Gaussian processes. The GP with squared exponential kernel without prior function on ℓ, top panel, gives the best result. The GP based on squared exponential kernel with prior function, middle panel, or on rational quadratic kernel with prior on both hyper parameters, bottom panel, gives very similar over-fitted interpolations. The offset on the epoch is due to a different position of the maximum in flux.

curves. Given M peaked light curves, the sequence of estimates $(\mathbb{E}(x_{0,j}))_{j=1}^{M}$ is averaged to get x_0 estimate:

$$\mathbb{E}(x_0) = \frac{1}{M}\sum_{j=1}^{M}\mathbb{E}(x_{0,j}). \tag{7.9}$$

Normalisation is introduced to mitigate the effect that the observed brightness might have on the comparison between light curves. Calling the flux measures and the associated errors $\{F_{ik}^b, \sigma_{ik}^b\}$, where k indexes the number of measurements in a generic band b, the interpolated light curves will be denoted by $\{\hat{F}_{ik}^b, \hat{\sigma}_{ik}^b\}$. The normalisation is applied both to flux and error as shown below:

$$\begin{aligned}\tilde{F}_{ik}^b &= \frac{\hat{F}_{ik}^b}{\sum_{b\in griz}\max_k\{\hat{F}_{ik}^b\}},\\ \tilde{\sigma}_{ik}^b &= \frac{\hat{\sigma}_{ik}^b}{\sum_{b\in griz}\max_k\{\hat{F}_{ik}^b\}}.\end{aligned} \tag{7.10}$$

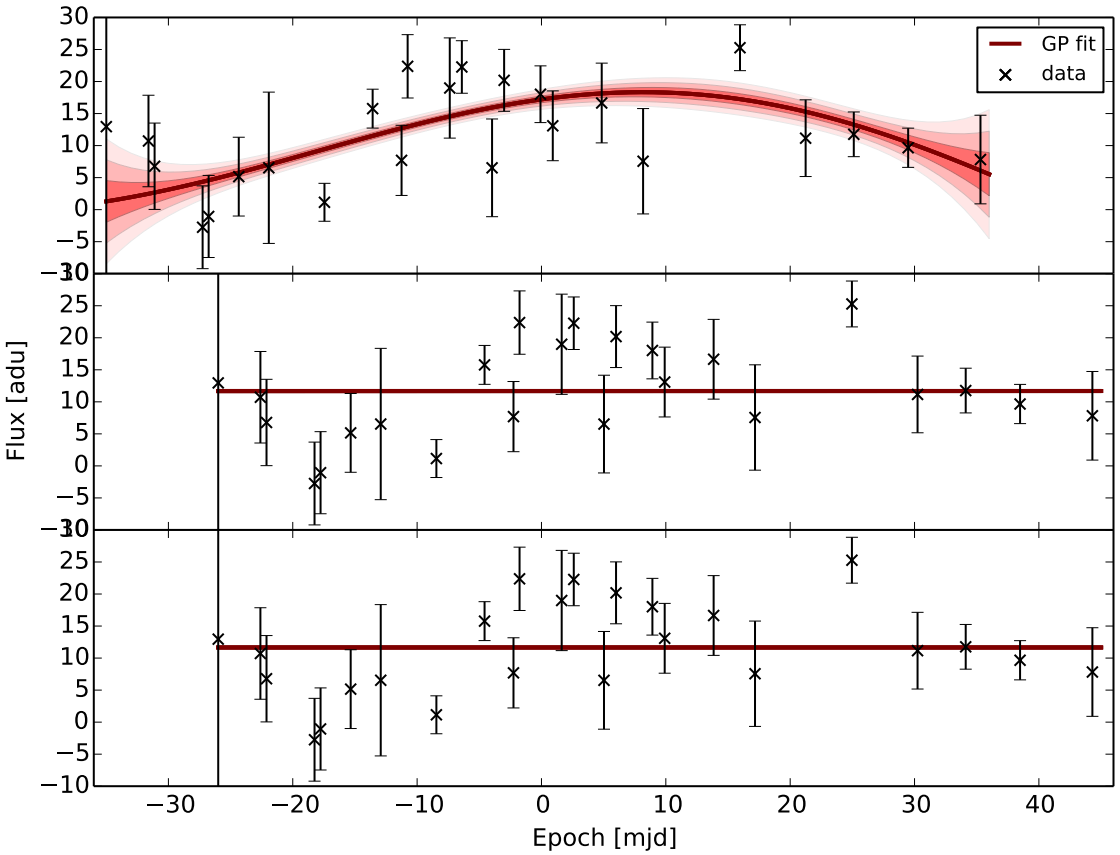

Figure 7.6: Comparison of interpolations by the three different Gaussian processes in the case of flux measures with large scatter and large errors. Using a squared exponential kernel without prior function, top panel, gives the best result. The middle e bottom panels, shows the regression function of SE with prior and RQ with priors, respectively.

Since the same denominator is used, the colour of each light curve is preserved. Normalising the light curve means that only the shape is used to classify the supernova; all information about relative luminosities is lost.

7.6 Wrap up

Observed light curves, corrected for astrophysical effects, needs to be interpolated to have them expressed on a regular time grid; this makes possible the pairwise comparison necessary to parameterisation.

To maintain the data-driven approach, the interpolation is carried out using a non-parametric technique called Gaussian processes, a generalisation of Gaussian distribution to a function space of infinite dimensions. As non-parametric, GP uses few and loose assumptions on the function to interpolate. These assumptions are encapsulated in the kernel function, which defines the covariance between data points. In this chapter the square exponential and the rational quadratic kernel have been introduced and tested. The first assumes only one characteristic length scale for the light curve, while the second brings in the GP the

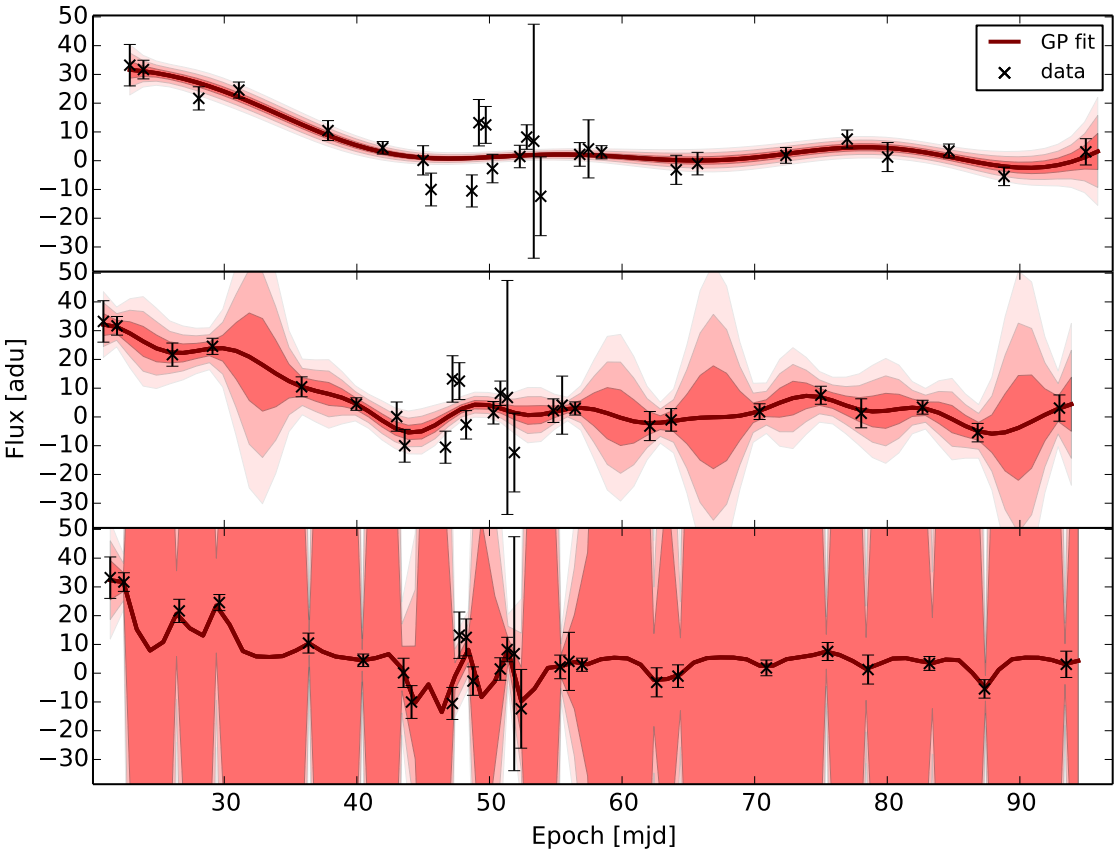

Figure 7.7: Comparison of interpolations carried out by the three different Gaussian processes. The top panel reports the result using the squared exponential kernel without prior function on the length scale hyper parameter. Middle panel shows the light curve recovered by a GP using squared exponential kernel with prior function on ℓ. Bottom panel reports the result using a rational quadratic kernel, with prior functions on both hyper parameters. Among the three, the top panel shows the best interpolation.

assumption of flux variation taking place over more then one length scale.

Kernel functions have parameters, called hype parameters, which have to be optimised to the specific light curve to be interpolated. Optimisation is implemented by seeking for the global stationary points of the likelihood function. Since likelihood has not a simple trend, optimisation algorithm, such as BFGS, may fail to converge to the best value. This reflects in over- or under-fitted light curves. To avoid this, either prior functions may be set on hyper parameters values, or a constraint can be imposed. Regression was performed using the two kernels separately, also including prior functions. The results showed that the SE kernel with constraint is the best choice.

Chapter 8

Light curves parameterisation: semi-supervised learning

Supernova light curves have been corrected for the astrophysical effects of Milky Way dust reddening and for time dilation. In order to standardise the raw data, Gaussian processes regression has been used to get a light curve with measures on a regular time-grid; in this way light curves from different supernovae can be compared. How comparison in accomplished is argument of this Chapter. The comparison is used to gain a parametric view; light curves are then moved in a low-dimensional parameter space, where the classification model is built.

Following the data-driven approach, the parameterisation, or dimensionality reduction, is achieved employing diffusion map, a semi-supervised learning technique developed by Coifman & Lafon (2006).

8.1 Why a low-dimensional space?

A low-dimensional representation of light curves is required because of the way machine learning algorithms works, and is related to the so called *curse of dimensionality*, a well-known problem in many statistical and machine learning applications.

A machine learning algorithm learns how to map input to output using *features*, or parameters, which describes examples in the training set (see Chapter 5).

At the simplest level, a light curve is parameterised by its own points, since at a given day from maximum, every light curve has a different flux. Thus, in principle flux values could be used has features to carry out the classification. However, in machine learning, the complexity of the problem is directly proportional to the number of features; this is the curse of dimensionality. Since the median number of flux predictions in each light curve is ~ 300, the complexity can be very high, increasing the computational time. Using a low-dimensional representation is thus a way of simplifying the problem for the machine learning algorithm.

One way to to reduce a data set dimensions is called "feature extraction" (Alpaydin 2010). Said n the number of predicted data points in a smoothed light curve, the new $m \ll n$ dimensions are some combination of the original n. Since dimensionality reduction is in place to simplify the problem, the n original features are combined so that the m new features will contain information useful to solve the problem. Into the specific of the classification problem, the m feature will have to be designed to discriminate between supernova classes.

The data set of interpolated supernova light curves is complex, with individual objects

showing important differences. In the master thesis De Pascale (2011), I grew confident that to reduce the dimensionality of such a complex data set, a non-linear approach would bring good results. *Diffusion maps* is a non-linear dimensionality reduction technique developed by Coifman & Lafon (2006). It belongs to the domain of semi-supervised learning algorithms, since, to achieve its task, it exploits the whole data set. In this way the method samples the complete supernova population, to understands which are the important characteristics identifying each supernova type. The parameterisation will reflect such findings.

This technique has been shown by several authors (de la Porte et al. 2008; Richards et al. 2009) to perform better then other well known feature extraction methods, such as principal component analysis.

8.2 Introduction to non-linear dimensionality reduction

The key idea to non-linear dimensionality reduction, is that a high-dimensional data set lies around a low dimensional structure, or manifold. On this manifold, similar objects are expected to be grouped together, and far apart from objects showing different characteristics. Thus, identifying the low dimensional structure and expressing the light curves in terms of coordinates on this manifold, makes the identification of classes easier.

A simple example of how diffusion map works is reported in Figure 8.1. The points are scattered in a Euclidean space. Yet, the manifold best describing the distribution is the one-dimensional spiral. Indeed, to know how far is point $\mathbf{x}$ from point $\mathbf{y}$, the most informative distance is measured along the spiral, while the common Euclidean distance is not going to give a useful answer. The method of diffusion map is able to identify the spiral, by providing as output the coordinates of the points on the spiral.

The following Sections will explain how diffusion map works. The outline is as follows. Measures of similarities are calculated between every light curve using a Euclidean distance introduced in Section 8.3. Section 8.4 explains how light curves are organised on a graph taking into account Euclidean distances calculated in the previous step. Through a random walk on the graph is possible to identify the lower dimensional structure, by means of the diffusion matrix, built from Euclidean distances. On this manifold, so called diffusion distances are calculated as in Section 8.5. Finally, the light curves low dimensional representation, expressed as coordinates on the manifold, are obtained by mapping the diffusion distances back into the Euclidean space, and keeping the most informative. The parameters obtained are called *diffusion coordinates*, and the process to calculate them is explained in detail in Section 8.6.

8.3 Pairwise distances: measuring similarities

To describe the Euclidean pairwise distance, is introduced a formal mathematical notation as follow. As in Section 7.5, flux measurements of light curve i are noted as F_{ik}, with k indexing the single measure; a hat on top refers to GP predictions. The errors on the fluxes are therefore σ_{ik}. In what follows, the generic interpolated light curve i in the generic band b will identified by using the formalism $\hat{\boldsymbol{X}}_i^b$, which groups the two sequences $\hat{F}_{ik}^b$ and $\hat{\sigma}_{ik}^b$. The distance calculated in the generic band b between two normalised interpolated light curves

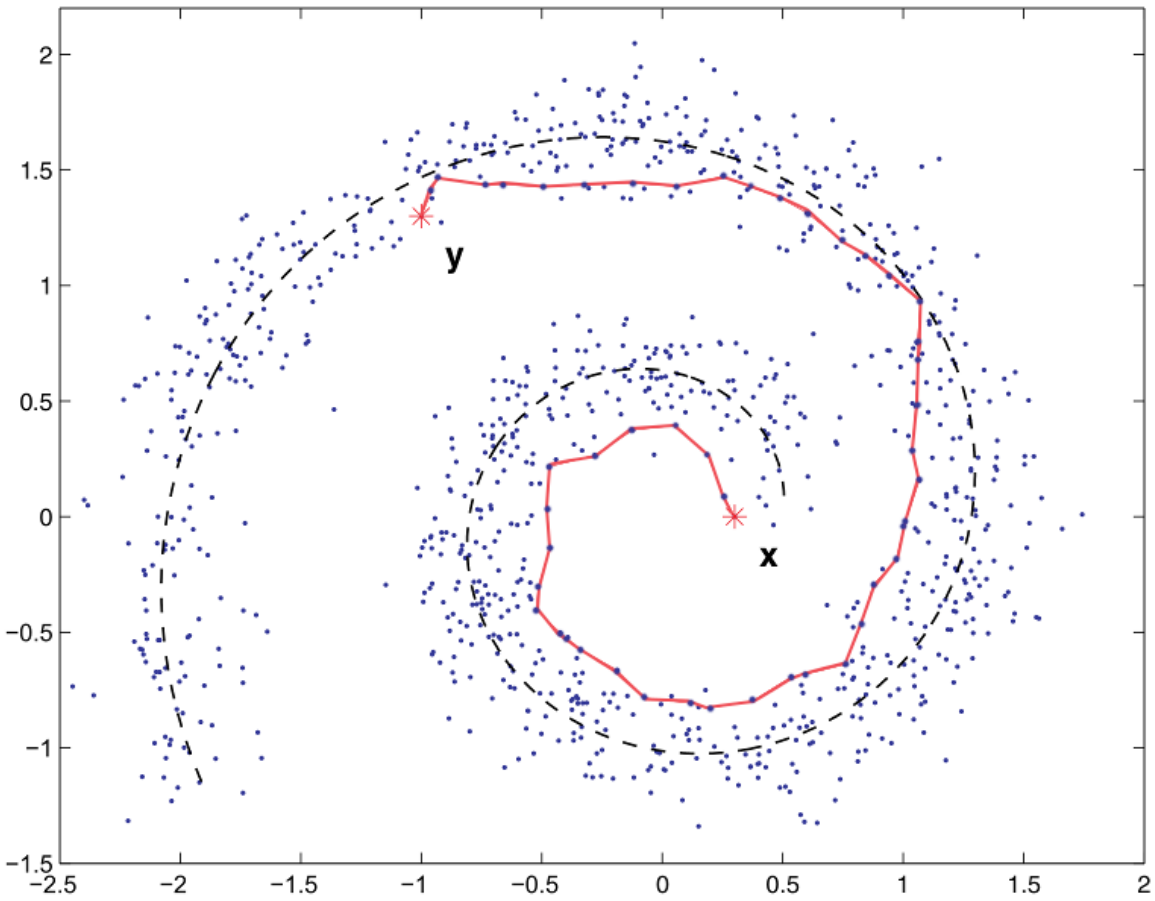

Figure 8.1: Example of using diffusion map on a distribution of points scattered along a spiral. The best metric to measure how far is point **x** from **y**, in the distribution, is along the dashed spiral line. The Euclidean metric does not give a useful information. The dimensionality of the data set is reduced from two coordinate in the Euclidean space to one coordinate on the spiral. Figure from Richards et al. (2009)

$\hat{\boldsymbol{X}}_i$ and $\hat{\boldsymbol{X}}_j$ is defined as:

$$s_b(\hat{\boldsymbol{X}}_i^b, \hat{\boldsymbol{X}}_j^b) = \frac{1}{\hat{x}_u - \hat{x}_l} \sqrt{\sum_{k \in [\hat{t}_l, \hat{t}_u]} \frac{\left(\hat{F}_{ik}^b - \hat{F}_{jk}^b\right)^2}{(\hat{\sigma}_{ik}^b)^2 + (\hat{\sigma}_{jk}^b)^2}}, \tag{8.1}$$

where b ranges over photometric filters, which in the SNPhotCC data set are the *griz* bands employed by the Dark Energy Survey, and k indexes the time grid, with time steps of one day.

This measure of similarity is a weighted Euclidean distance, calculated over the overlapping time coverage of the two light curves. Indeed, x_u and x_l are the upper and lower time bounds of the overlapping region. The definition given in Equation (8.1) is the same used by Richards et al. (2012). When there is no overlap, the two light curves cannot be compared; in such cases the distance is set to a large value. The distances calculated in each of the four

bands are then summed to obtain the total distance

$$s(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j) = \sum_{b \in griz} s_b(\hat{\boldsymbol{X}}_i^b, \hat{\boldsymbol{X}}_j^b). \tag{8.2}$$

A small distance s would mean the two light curves are similar, and they would probably belong to the same class. It will be shown that no important information is provided by distance s when it is bigger then a fixed value. Equation (8.2) will be widely used in the following Sections.

Distance s can also be interpreted as a *local similarity* measure. It is local in that, if small, it well approximate small distances on the manifold in which the data set is embedded. It is a similarity measure because a small distance on the manifold means the two light curves are similar. An example is again Figure 8.1. The short segments connecting adjacent points are good approximations of small distances along the spiral. On the opposite, if s is large, as the distance separating points x and y, it is not approximating a distance on the manifold, thus it cannot be read as a measure of similarity.

To be used by diffusion map, the pairwise distances are collected in the symmetric matrix, with zero diagonal, called *distance matrix*.

Notes on execution time

The calculation of the distance matrix has been implemented in native `Python` to achieve high speed computation. Indeed, this is the bottle neck of the whole classification process. Considering the symmetry of the matrix, and distributing the calculation on eight 2.4 GHz Intel® Xeon® cores, to output a $20\,000 \times 20\,000$ matrix the code takes 10 hours. This time, multiplied by the number of photometric bands gives a total of almost 40 hours of computation. The execution could gain a significant speedup by parallelising the code using a Graphical Processing Unit.

8.4 Random walk on a graph

The distance matrix is used by the diffusion map algorithm to organise light curves on a *graph*. A graph is a mathematical structure, a collection of *nodes* connected by *edges*, see Figure 8.2 for an example.

The weight on the edges

Each node represents a light curve, while to each edge is associated a *weight*, which is defined as function of the pairwise distance s of Equation (8.2):

$$w(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j) = \exp\left(-\frac{s(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j)}{\epsilon}\right). \tag{8.3}$$

It is worth to note how Equation (8.3) sets the limit up to which distance s is still valid as a local measure of similarity. This limit is determined by the constant ϵ. Outside the region defined by ϵ, thus for $s > \epsilon$, the weight goes to zero quickly. This means that when distance $s > \epsilon$ the less reliable as similarity measure it becomes. In these cases, the two light curves are likely to belong to different classes. Thus, the constant ϵ sets the limit up to which light curve are recognised as similar, and this has effects on the classification model.

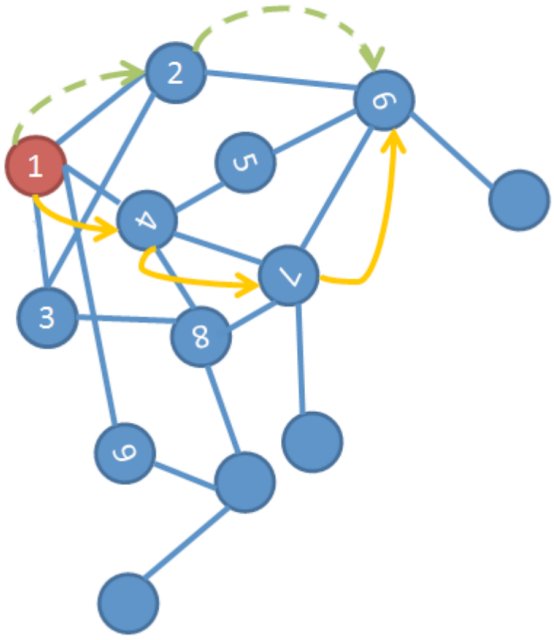

Figure 8.2: An example of graph. The circles are called nodes while the lines connecting them are the edges. From node one, the arrows show directions of two possible random walks. The probability of the single jumps between two nodes are proportional to the weight w defined in Equation (8.3). The higher w, to higher the probability for that jump, the higher the similarity between the light curves represented by the nodes. Adapted from Figure 3 in de la Porte et al. (2008).

From weight to connectivity and probability

The weight $w(\cdot, \cdot)$, or strength, is used to define a quantity called *connectivity*, which is a different way to call the strength between two nodes:

$$\text{connectivity}(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j) \propto w(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j). \tag{8.4}$$

Now, suppose to take one step of a random walk on the data set, jumping between nodes on the graph, as depicted by arrows in Figure 8.2. A random walk is a way to approximate a diffusion process on the data, from which the name of the technique. In a random walk, is more likely to jump between two light curves which are nearby in the low dimensional structure, than jumping to another that is far away. This concept is encapsulated in the connectivity. Indeed, two nearby light curves will be "separated" by a small distance s; from Equation (8.3), small s means the edge connecting the two nodes has high weight w. Consequently, by Equation (8.4), between the two nodes there is high connectivity. The opposite is true for light curves separated by big distances. In this view, it is possible to relate connectivity (hence distance) to the probability of jumping between nodes:

$$\text{connectivity}(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j) \equiv p(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j). \tag{8.5}$$

Combining Equation (8.4) end Equation (8.5), probability can be expressed as proportional to the weights in Equation (8.3). Since, by definition, probability has to sum up to one, the

normalisation constant will be the sum of all the weights:

$$p(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j) = \frac{w(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j)}{\sum_l w(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_l)}, \quad l = 1, \dots, N, \tag{8.6}$$

where N is the total number of light curves; Equation (8.6) defines the one-step pairwise probabilities. As for distances s, the $p(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j)$ are collected together to form an $N \times N$ symmetric matrix called *diffusion matrix* P:

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{bmatrix}, \tag{8.7}$$

with N equal to the number of light curves, or nodes, in the graph. The diagonal contains the probabilities of jumping to the same nodes, thus those elements are all equal to one.

Matrix P collects the information on the connection strength between every pair of nodes. Considering that high pairwise probabilities p_{ij} identifies similar objects, diffusion matrix P can be said to encapsulates the degree of clustering.

This can be better seen calculating the diffusion matrix for a random walk with two steps. All it has to be done is to raise diffusion matrix P in Equation (8.7) to the second power. Each element P^2_{ij} will be equal to the sum of all the paths connecting node i to node j in two jumps. As an example, consider a 2×2 diffusion matrix, along with its second power:

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix},$$
$$P^2 = \begin{bmatrix} p_{11}p_{11} + p_{12}p_{21} & p_{12}p_{22} + p_{11}p_{12} \\ p_{21}p_{12} + p_{22}p_{21} & p_{22}p_{22} + p_{21}p_{12} \end{bmatrix}.$$

Focus on p_{12}, the probability of jumping from node one to node two, or from light curve one to light curve two, in one step. If this probability is high, therefore also probability $p_{12}p_{22} + p_{11}p_{12}$, jumping from node one to node two in two steps, will be high, because combination of high probabilities (by definition p_{11} and p_{22} are both equal to one). This strengthen the similarity between the two associated light curves. Using words from the graph jargon, $p_{12}p_{22} + p_{11}p_{12}$ identifies an "high probability path". Raising diffusion matrix P to higher powers (taking more steps on the graph), the high probability paths will more and more identify paths on the geometric structure around which the data set lies. This happens because, along that geometric structure, points are dense and therefore highly connected.

8.5 Diffusion distance

A fictitious diffusion process can be used to reveal the global geometry of a data set. This is achieved by calculating the diffusion matrix P in Equation (8.6). To further evaluate similarities between light curves, distances on this manifold, the *diffusion distances* have to be measured.

When raising diffusion matrix P of Equation (8.7) to powers t grater then one, we are taking t steps in the random walk. In doing so, the most important and global geometric structures are revealed, to the expenses of the smaller and more local structures. Hence,

the diffusion distance depends on which geometry the diffusion matrix P, or its powers, has revealed. Thus, the formal expression for the diffusion distance between two smoothed light curves $\hat{\boldsymbol{X}}_i$ and $\hat{\boldsymbol{X}}_j$ is:

$$D_t(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j)^2 = \sum_u \left| p_t(\hat{\boldsymbol{X}}_i, u) - p_t(\hat{\boldsymbol{X}}_j, u) \right|^2 \tag{8.8}$$

$$= \sum_{k=1}^{N} \left| P_{ik}^t - P_{kj}^t \right|^2 . \tag{8.9}$$

Consider the term $p_t(\hat{\boldsymbol{X}}_i, u)$, the probability of jumping from $\hat{\boldsymbol{X}}_i$ to u in t steps. It is obtained by summing the probabilities of all possible paths of length t between $\hat{\boldsymbol{X}}_i$ and u, as explained in Section 8.4. This term gains an high value on the paths along the geometric structure underlying the data. It follows that, for the diffusion distance separating $\hat{\boldsymbol{X}}_i$ from $\hat{\boldsymbol{X}}_j$ to remain small, the path probabilities from both light curves to *any* u have to be almost equal. In other words diffusion distances are small if there are many high probability paths of length t between two nodes (de la Porte et al. 2008, Section 3.3). Which means that the distance is small if the light curves are similar.

As anticipated in Section 8.4, when considering many steps on the graph, local geometries are discarded to reveal the more global structure. To not discard local structures, the most simple setting of $t = 1$ was employed; thus diffusion distances were calculated on the geometry revealed by a one-step diffusion process. Equation (8.9) becomes therefore:

$$D(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j)^2 = \sum_{k=1}^{N} \left| P_{ik} - P_{kj} \right|^2 . \tag{8.10}$$

Once such diffusion distances are calculated, to obtain the light curves low-dimensional representation expressed in terms of diffusion coordinates, a mapping to a Euclidean space have to be performed; this is presented in Section 8.6. Then, light curve parameterisation will be achieved and supernovae will be ready to build the classification scheme with random forest (see Chapter 9.1).

8.6 Diffusion coordinates and dimensionality reduction

The diffusion distance D_t in Equation (8.9) measures distances along the structure around which the data set lies. Hence, D_t is the measure of similarity, between two light curves, valid both locally and globally over the data set; this is in contrast with distance s given in Equation (8.2), which measures only local similarities. To calculate D_t is computationally expensive, even for $t = 1$. Less expensive is to map the light curves from the manifold in which the data set is embedded, to a Euclidean space, according to the diffusion metric D_t. This Euclidean space is called *diffusion space*. By doing so, for each light curve we get a set of coordinates in the diffusion space, called *diffusion coordinates*. Following the mapping, the diffusion distance is mapped to a Euclidean distance calculated using the diffusion coordinates, thus preserving the measure of similarity.

The diffusion coordinates are expressed in terms of the right eigenvalues and eigenvectors of diffusion matrix P noted as λ_j and ψ_j respectively. This takes much less computational

time. The mapping from data space to diffusion space is operated by the *diffusion map* $\boldsymbol{\Psi}$:

$$\boldsymbol{\Psi} : \hat{\boldsymbol{X}}_i \mapsto \begin{bmatrix} \lambda_1\psi_1(\hat{\boldsymbol{X}}_i) \\ \lambda_2\psi_2(\hat{\boldsymbol{X}}_i) \\ \vdots \\ \lambda_N\psi_N(\hat{\boldsymbol{X}}_i) \end{bmatrix}. \tag{8.11}$$

Finally, the dimensionality reduction is achieved by retaining the $m \ll N$ diffusion coordinates associated with the dominant eigenvectors, so to ensure the best approximation for the diffusion distance $D(\hat{\boldsymbol{X}}_i, \hat{\boldsymbol{X}}_j)$.

8.7 Implementation and results

All the calculations described in this Chapter are performed by `diffusionMap` (Richards 2014), a package written in `R` language (R Core Team 2015). The distance matrix of Section 8.3 is calculated by `Python` code and then fed to the `diffuse` routine along with a value for constant ϵ of Equation (8.4) and the number m of eigenvalues to keep, as exemplified in Listing 8.1.

```
library(diffusionMap)
message('Calculate diffusion coordinates...')
dmap <- diffuse(distMatrix, eps.val=epsilon, neigen=m)
```

Listing 8.1: Example code to calculate diffusion coordinates using `diffusionMap` functionalities.

Regarding ϵ value, since it defines the region outside which distance s is no more reliable as similarity measure, it has been set equal to the default big distance used when the light curves cannot be compared, (see Section 7.5). The number m of eigenvalues is chosen by validating the tests on different classification models obtained with different m. The performance of the models does not improve for $m > 50$.

Figure 8.3 reports a projection of the diffusion space. The figure shows the coordinates of light curves forming the training set, the subset for which a spectroscopic classification is provided. The training set is used by the random forest machine learning algorithm to build the classification model. From the relative distribution of light curves belonging to different classes, it is possible to have an idea of how good the model will be in discriminating between them. Supernovae type Ia and supernovae type II are sufficiently well separated, thus we can imagine that, on the plane of this two coordinates, the classification model will have good performances in recognising the two types. On the other end, supernovae type Ibc are distributed in the same region as type Ia. This means that the model does not give a good representation of type Ibc in terms of diffusion coordinates. As a result this type of supernovae are likely to be misclassified.

Figure 8.4 shows the same projection but contains all the light curves in the data set. There is no more the clear separation between type Ia and type II, while type Ibc show no different distribution. To make this plot we exploited the fact that for the simulated data set, the type is known for all the light curves.

These two Figures are just examples of how well the diffusion coordinates do in separating different classes. Since the diffusion space has 50 dimensions, other coordinates will give additional information to the machine learning model (as examples, see Figure A.1 and Figure A.2). For this reason, the only way to investigate how good is the representation given by the diffusion map, is to build the classification model and test its results.

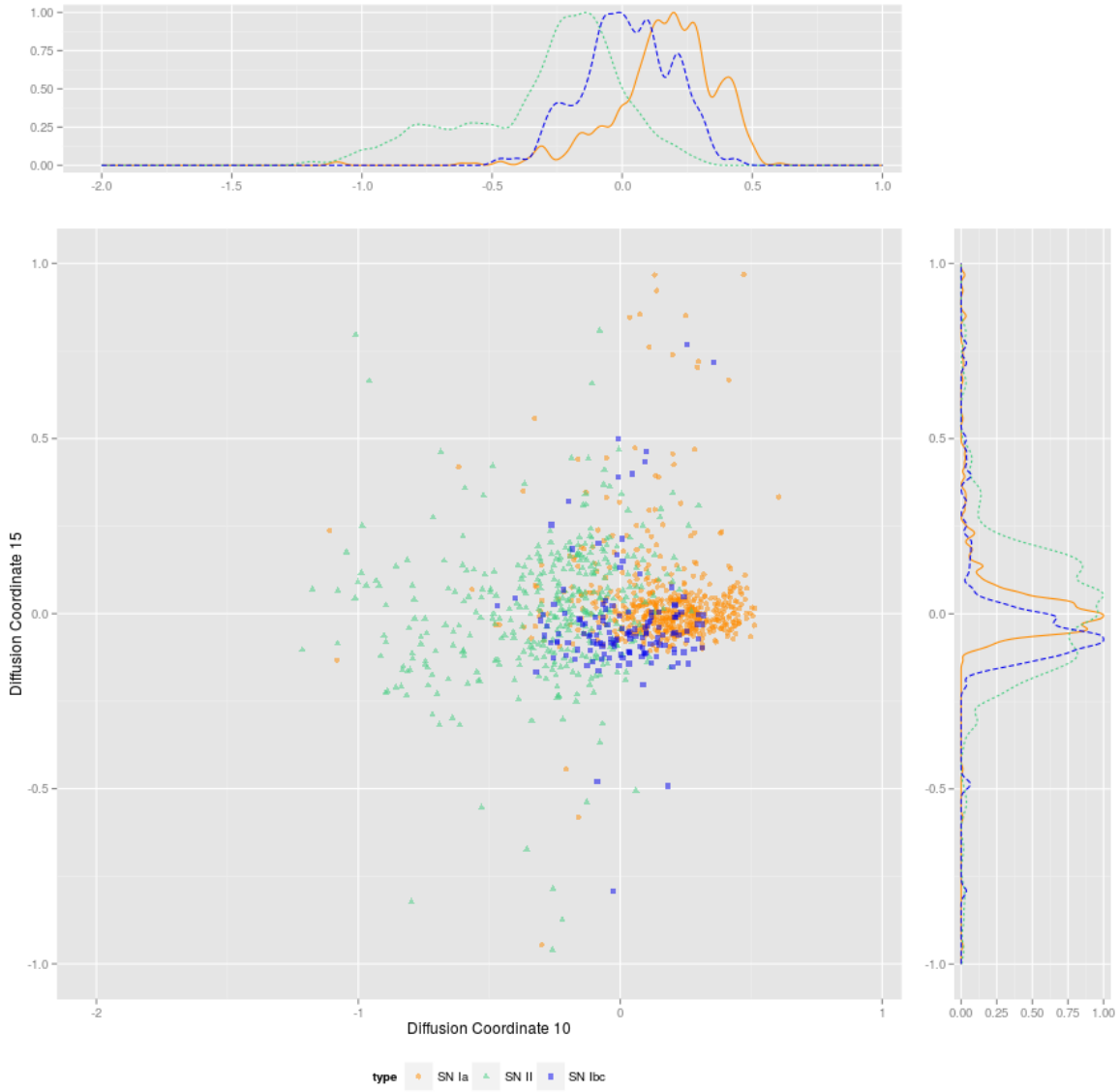

Figure 8.3: Diffusion coordinates of light curves in the training set. Clusters of supernovae Ia (dots and solid line) and supernovae II (triangles and dotted line) are discretely separated as shown by the marginal density distributions. On the other hand, supernovae type Ib/c (squared and dashed line) cluster is distributed both in Ia cluster and in type II cluster. The coordinates displayed are the most important to the machine learning algorithm building the classification model (Figure 9.2).

8.8 Wrap up and conclusions

The machine learning technique responsible for building the classification model, maps the input light curves to the output classes by means of features describing instances in the training set. In principle, a light curve is parameterised by its own points. However, the number of points of an interpolated light curve is high (~ 300), and would make too complex building the classification model. Light curve parameterisation is thus put in place to reduce the problem dimensionality.

Parameter extraction is performed using a semi-supervised machine learning technique, diffusion map, developed by Coifman & Lafon (2006) to explicitly implement non-linear dimensionality reduction.

The basic idea of non-linear dimensionality reduction, is that a high-dimensional data set, such as the one formed by interpolated SNPhotCC light curves, lies on a low-dimensional structure. The identification of the manifold is achieved using a local similarity measure calculated between each light curve pair. The parameters extracted are function of coordinates on the manifold.

Diffusion map extracts the information it needs from the data set, indeed calculating the local similarity measure is done with no assumptions; thus it can be thought as a data-driven method. However to provide light curve parameters, diffusion map needs the specification of an external parameter, ϵ, setting the limit up to which two light curves are deemed as similar. A too small value could prevent diffusion map from identifying any similarity in the data set. On the other hand, a too high and all the light curves could be seen as belonging to the same cluster on the manifold. In this application, the value of the threshold ϵ has been set to the default value of similarity measure s used when two light curves are known to be different.

The results obtained, showed in Figure 8.3 and Figure 8.4 in a 2-D projections, are not revealing a sharp separation between the three supernova classes; this is reflected in the low performances of classification model presented in Chapter 9.

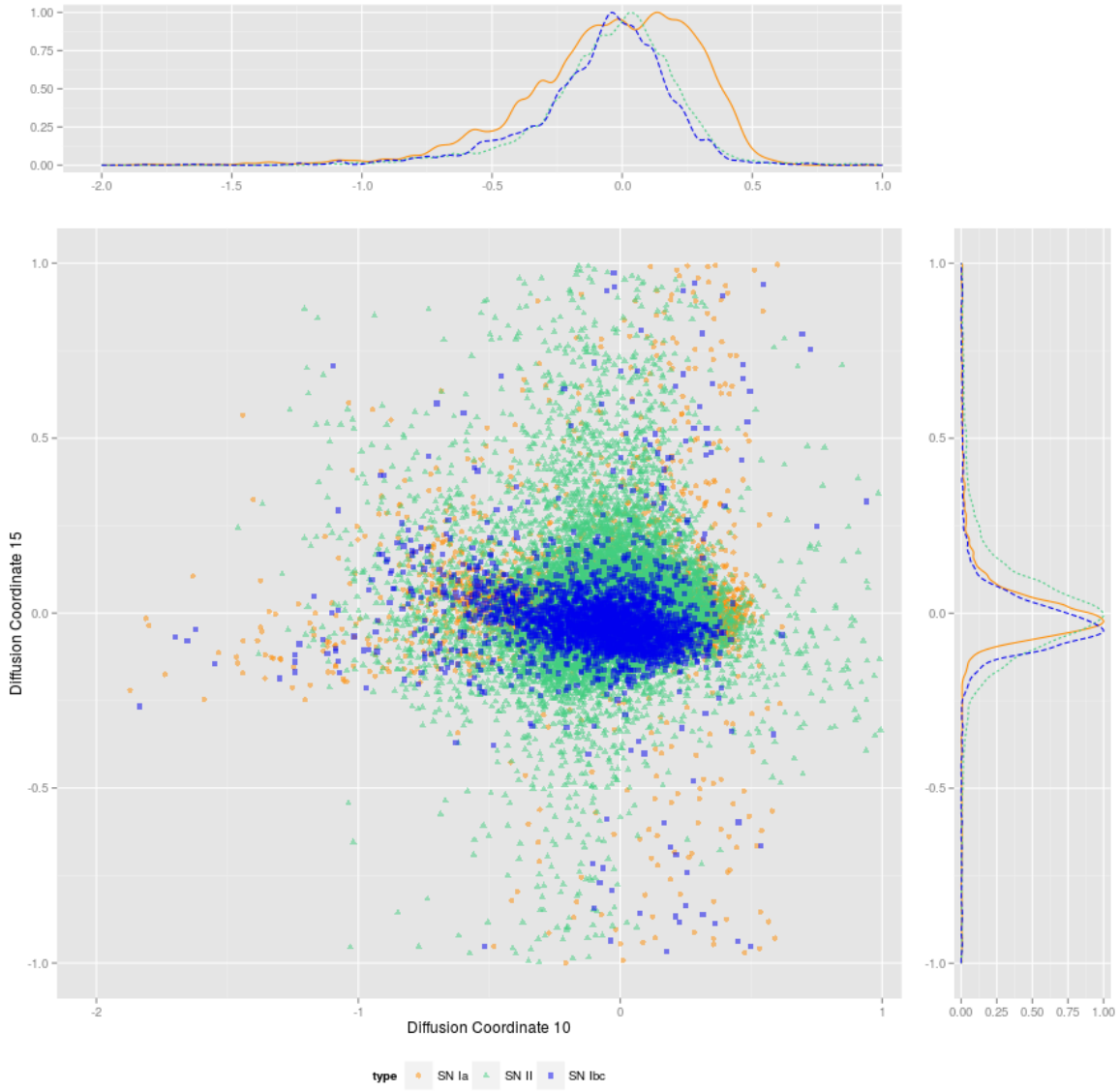

Figure 8.4: Diffusion coordinates of light curves in the test set. The three clusters do not exhibit the same separation as in Figure 8.3.

Chapter 9

Supernova classification: supervised machine learning

The observed light curves have been corrected for astrophysical effects, continuous light curves have been determined through the regression method of Gaussian Processes, and finally a low dimensional representation has been provided using the diffusion map technique. The last stage is to build a classification model.

The classification model is the output of a supervised machine learning algorithm. The rule to classify supernova light curves in three known types, Ia, II and Ib/c, is learned using instances from the training set. This is a subset of light curves for which the supernova type is provided through spectroscopic followup.

The machine learning algorithm employed is called random forest, introduced by Breiman (2001), and briefly described in Section 9.1.

To evaluate the classification rule performances, the model is tested on light curves that did not take part in the learning phase. The SNPhotCC simulated data set provides the supernova type for each light curve, both spectroscopically confirmed and only with photometric observations; this second subset is used as test set.

Results of the test reported in Section 9.2 show how the model fails in providing good classification, the most striking problem being SNe Ib/c misclassification. Possible causes of this result are discussed.

The classifier performances are also measured using the figure of merit for type Ia classification provided by Kessler et al. (2010). The comparison with what scored by Richards et al. (2012) with a similar classification method, supports the method presented, in supernova Ia classification.

9.1 Random Forest

Random Forest is a machine learning method formalised by Breiman (2001). It is widely used for many applications, from gene selection in biology to image recognition in medical applications and video games. It is also used by the Dark Energy Survey automated transient identification software, with great success in recognising supernovae Ia (only 1% of their simulated sample is lost, see Goldstein et al. (2015)).

Random Forest is an *ensemble learning* technique, which employs structures called *decision trees*.

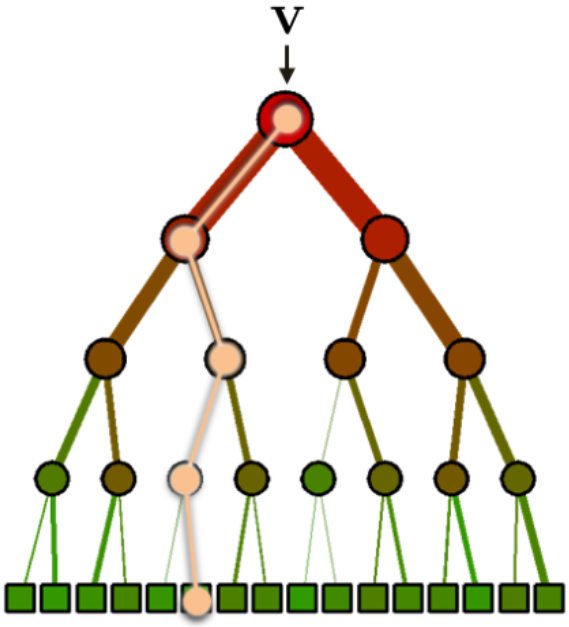

Figure 9.1: Example of decision tree with a possible decision path (in pink). Adapted from Figure 2.2 in Criminisi et al. (2012).

The term "ensemble learning" means that the model produced by the learning algorithm is some sort of combination of other models, obtained by less sophisticated learning algorithms (or learners) all using the same training set. Models produced by the learners are combined in a majority vote fashion, strengthening the most common vote, which is assumed to be the correct answer.

9.1.1 Decision Tree

A decision tree is a structure modelling the way into which humans take a decision. It is a collection of hierarchical decisions that brings to a set of possible answers.

More formally, a decision tree is a collection of nodes and edges (or branches) organised in hierarchical levels in a flowchart-like structure, as in Figure 9.1. Each node represent a test on an attribute of the incoming data. In the case of supernovae classification, the incoming data is a light curve, while the attributes are the diffusion coordinates calculated by diffusion map, Section 8.6. The branch into which data flows next depends on the result of the test. The last nodes of the tree are called leaves, and represents the answers the tree can give.

Tests at each node are equivalent to cuts in the data set, since at each node the sample is divided in two parts. Which attribute is tested and which is the best cut is based on the maximisation of the so called information gain. This is a highly technical issue, and will not be covered here; however, see Criminisi et al. (2012) for a good introduction to this topic. The issue of building decision trees is solved by software packages dedicated to the use of these structures.

In random forest, every decision tree answers to a classification problem, thus in what follows, the terms decision tree and classification tree will be equivalent.

9.1.2 Ensemble learning in random forest

As anticipated, in ensemble learning classification models produced by single classification trees can be combined together to obtain a more robust model. The aim is to provide acceptable predictions even in parameter space regions where the classes are not well separated. Random forest is one technique for ensemble learning, among with bagging (Breiman 1996) and boosting (Schapire 1990).

The essential point in these techniques is that the classification models to combine have to be *independent* (or de-correlation)one from the other. To satisfy this condition, each tree in the ensemble is trained on a different training set. This is essential for the trees not to make all the same mistakes, in which case the aggregation of their models would bring no additional improvement.

Random forest achieves de-correlation in two ways. First, the algorithm builds a different training sample for each tree in the ensemble. These training sets are drawn with replacements (bootstrapping) from the original training set. Second, at each node of each tree, only a different *random* subset of features is available. In this way, considering the same node for each the trees in the ensemble, the best feature and the best split will be different. This manner of achieving de-correlation is what makes random forest a powerful learning algorithm.

Nonetheless, as pointed out by Hastie et al. (2001) in their Chapter 15, random forest are not perfect learning algorithms; indeed, they can produce overfitted models.

The resulting ensemble of trained trees in then combined giving the same weight to all trees. This means that every tree casts a unit vote to determine the classification of one input light curve.

9.2 Classification results

Light curves parameterised using diffusion map are divided into training and test set. The SNPhotCC data set is simulated, so for elements of both sets, the supernova type is known. Training set light curves are used by the random forest algorithm to build the classification model, which will be validated using the test set.

Liaw & Wiener (2002) `R` package `randomForest` is a porting of the original `Fortran` software by Breiman and Cutler, available at `http://www.stat.berkeley.edu/~breiman/RandomForests/`. The package `randomForest`, provided with training set, output classes and the number of trees in the ensemble, takes care of building the classification model. Such model is also automatically tested if the test set is provided, see Listing 9.1.

```
sn.ranFor <- randomForest(x=training.Set, y=sn.classes,
                          xtest=test.Set, ytest=sn.classes,
                          ntree=300,
                          importance=TRUE)
```

Listing 9.1: Example code to build a classification model employing the `randomForest` package.

Predictions provided by a random forest are relatively robust to the choice of tuning parameters such as `ntree`. Tests with different values for `ntree` showed it could be lowered from the default 500 to 300 without any loss in performances.

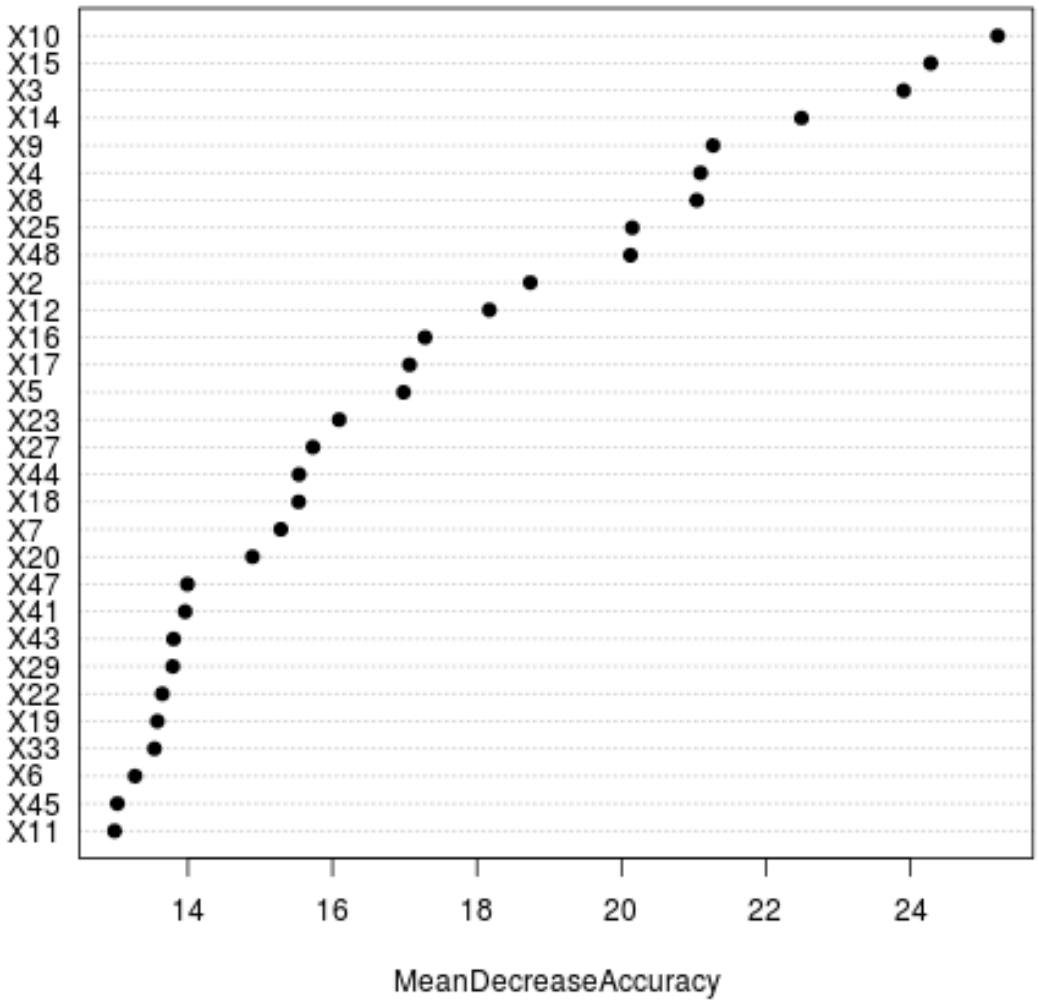

Figure 9.2: Dot chart of features importance from the random forest, expressed in terms of decrease in accuracy when not considering the feature.

9.2.1 Parameter importance

Specifying `importance=TRUE` the algorithm outputs the importance of each feature in the classification process. Figure 9.2 reports how important is each parameter in describing the light curves; this quality is measured in terms of accuracy decrease when the classification model is built without considering one parameter at time.

Using parameter importance it would be possible to investigate how the shape of light curves changes as function of the most important parameters. Thus, it would be feasible to understand which specific feature of the light curve the parameters are describing. This has been not implemented, but represents an interesting way forward.

9.2.2 Model performances

As anticipated, model performances are assessed by matching the predicted class to the actual class to which each supernova belongs to. The results of this comparison are summarised in the so called *confusion matrix*, organised with a row and a column for each class; rows identify the actual classes, while columns the prediction given by the model. Table 9.1 and Table 9.2 are the confusion matrices for training and testing phase, respectively; they are discussed in the following Sections.

The elements forming the diffusion matrices can be grouped in *true positives* (TPs) and *false positives* (FPs), the quantities needed to check the goodness of the model. Consider as example the class of type Ib/c, the true positive is the number of light curves classified as Ib/c that are indeed light curves from supernovae Ib/c; that is the correctly classified light curves. On the opposite, the false positive is the number of light curves classified as Ib/c that actually are not light curves from supernovae Ib/c, the incorrectly classified light curves (Zumel et al. 2014). Using the confusion matrix is possible to estimate how good is the model in recognising light curves belonging to a specific class.

Training phase

As explained in Chapter 5, the training phase, is required by the machine learning algorithm to build the classification model. The model performances can be directly tested on the training set data. The results of this test have to be interpreted taking into account that by definition, they always overestimate the goodness of the model. This kind of test can be used to understand if the algorithm learned at least to separate the different classes, and to spot potential issues.

During training, the random forest algorithm exploits the different training sets by defining, for each tree, an estimate of the error rate. This is obtained by testing each of the single models with light curves excluded from the training set. The aggregation of these estimates is called out-of-bag (OOB) estimate of the error rate. This gives an overall idea of the model performances on the training set. In our case, The OOB estimate obtained is $\sim 12\%$, indicating the single models were good in recovering the supernovae type.

This is somewhat in contrast with the results shown by the confusion matrix in Table 9.1. Numbers listed in the column "classification error" tell that the model learned to separate supernovae Ia from supernovae type II. The low number of Ia false positives in type II row supports this conclusion. This was expected, the projection of diffusion space reported in Figure 8.3 already showed the two classes are discretely well separated.

On the other hand, the model fails in classifying supernovae type Ib/c; the classification error is worse than what would score a random classifier. Reasons for this bad score are multiple. The first, already suggested by Figure 8.3, is the parameter obtained by diffusion map are not good in separating type Ib/c from the other two types. Combining the superimposition of Ib/c cluster with both clusters of type Ia and type II shown in Figure 8.3, with the higher number of Ia false positives belonging to Ib/c actual class, the misclassification is likely caused by the interpolated light curves of Ib/c being similar to type Ia. On more reason for the high misclassification is that the SNPhotCC training set is unbalanced with respect to type Ib/c supernovae. The number of supernova Ib/c cases provided is much lower with respect to the other two types, only 144 against the 559 and 500 of type Ia and type II. Thus the model is not as trained on type Ib/c light curves as well as for the other two types, and fails in recognising Ib/c supernovae.

Table 9.1: Confusion matrix for training phase.

		Predicted class			
		snIa	**snIb/c**	**snII**	**class. error**
Actual class	**snIa**	528	11	20	0.055
	snIb/c	59	70	15	0.513
	snII	18	8	374	0.065

Notes. Row's labels states the true class, while on columns is noted the class inferred by the model.

Table 9.2: Confusion matrix for test phase.

		Predicted class			
		snIa	**snIb/c**	**snII**	**class. error**
Actual class	**snIa**	3340	846	285	0.252
	snIb/c	1111	1116	386	0.572
	snII	4213	1047	7756	0.404

The possible actions to undertake to solve this issue, will be explained in Section 9.3. Lets now examine the results produced by the test on the proper test set.

Testing the model

By testing the model on light curves that didn't take part in the learning phase, it is possible to measure the performances of the model on unknown data. This measures the goodness of the classification method in solving the problem. Table 9.2 summarises the results of the test.

Compared to what obtained in the training phase, the classification errors increased for all classes, as well as the number of false positives. From Table 9.1 was already clear a tendency for the model to misclassify type Ib/c as Ia; this is now even more evident by almost half of Ib/c being recognised as Ia.

More striking are the classification errors for type Ia and II, which became 5 and 6 times bigger than in Table9.1. A special concern is the number of type Ia false positives in type II row. These results confirm that the training set is not representative of the population of supernova transients in the simulated survey. Indeed it is biased towards supernovae Ia, since these are the most targeted objects by spectroscopic facilities, because of their characteristic of cosmological distance indicator.

The high number of type Ia false positives belonging to type II class, is further understood by the superimposition of the corresponding clusters in Figure 8.4. In fact, although that in the Figure is just a slice of the 50-dimensional diffusion space, it is made using what are the most important parameters to the classification model. Thus, in the 50-dimensional diffusion space type Ia and type II clusters are not well separated (see Appendix A).

One interesting feature is the predicted class of type Ia contains the largest number of false positives. By combining this observation with what already discussed, the classification model appears to be biased, as the training set is, towards supernovae Ia. Equivalently, the model can be described as overfitting on type Ia.

9.3 Conclusion and future development

Planned and ongoing photometric surveys are expected to produce a large supernova observations. The present spectroscopic facilities do not match the demand to follow and confirm each event. Thus there is a strong pressure for classification techniques relying only on photometric data. Classification of supernova transients is central to gather information on astrophysical phenomena spanning from the evolution of star formation rate to the Universe expansion.

In this thesis was explored an automatic classification method based on a data-driven approach, extracting all the necessary information directly from the data, using as few assumptions as possible. Such approach has been implemented employing algorithms from the machine learning domain. Although the classification model produced is designed to classify supernovae in the three known types of Ia, Ib/c and II, the data-driven approach is intended to let open the possibility to identify new classes of transients, including events not triggered by stellar explosions.

The aim of this work was dual: to see how far a data-driven approach could be pushed; namely how many assumptions have to be introduced and how strong they have to be. The second aim was to understand if a classification model based on photometry is able to reproduce the spectroscopy-based supernova classification scheme.

Data-driven approach and assumptions

The classification method proposed results not purely data-driven, because it turned out that to produce the classification model some assumptions are unavoidable. The first assumptions are introduced when interpolating light curves with Gaussian processes. In addition to continuity and smoothness, specified by the kernel function, the strongest assumption introduced has been on the length scale hyper parameter. This had been a challenging point in designing the method; the assumption is necessary otherwise the optimisation method does not converge to values resulting in over-fitting. On the other hand, an assumption biased towards supernova light curve characteristics, would inhibit the method in identifying new types of transients. The assumption adopted revealed to be successful in terms of interpolation results; however it has yet to be tested on not supernova transients.

Parameterisation using diffusion maps, also needs an assumption: the threshold parameter ϵ, defining the maximum distance (Equation (8.1)) up to which two light curves are deemed as similar. In this work the value of ϵ had been hardwired to a reasonable value. However, a cross validation approach is to be implemented, where different classification models are built as function of different ϵ. The classification model with better performances would define the value of ϵ. Such an application would depend much more on the data set at hand.

Finally, also the supervised machine learning algorithm building the classification model introduces one assumption: the number of trees in the ensemble `ntree`. However, provided the number of trees is large enough, the assumption has not a marked influence, since the aggregated model is robust with respect to changes in this parameter.

Reproducing spectroscopic classification

Unlike recently published supernova photometric classification methods, like Richards et al. (2012), Ishida & de Souza (2013), du Buisson et al. (2014) and Varughese et al. (2015), focused on separating supernovae Ia from non-Ia, the method proposed aimed to reproduce

the spectroscopy-defined classes of supernovae Ia, Ib/c, II. Such a classification is useful to the study of supernova rates. Results from the test presented in Section 9.2.2, show the model fails in this respect. True positives for predicted class Ia and class II are fairly good, especially for Ia. Nonetheless, the false positives polluting class Ia and Ib/c are high. Considering Ia supernovae, this means that, although the model is good in recognising true Ia, in that class it places also supernovae belonging to other classes. For this reason it can be concluded that the model, based only on photometric data and missing K-correction, is not able to reproduce the spectroscopic classification. Future developments are sketched in Section 9.3.1.

Comparison with independent results

Classification results can be evaluated also using Kessler et al. (2010) figure of merit (FoM), targeted to classification between Ia and non-Ia:

$$\begin{aligned} C_{\mathrm{FoM-Ia}} &= \text{efficiency} \times \text{pseudo-purity} \\ C_{\mathrm{FoM-Ia}} &= \frac{N_{Ia}^{\mathsf{true}}}{N_{Ia}^{\mathsf{tot}}} \times \frac{N_{Ia}^{\mathsf{true}}}{N_{Ia}^{\mathsf{true}} + W_{Ia}^{\mathrm{false}} N_{Ia}^{\mathsf{false}}}, \end{aligned} \tag{9.1}$$

where the factor W_{Ia}^{false} weights false positives in predicted Ia; following Kessler et al. (2010) is set equal to three. The values calculated with Equation (9.1) are reported in Figure 9.3 as function of redshift; the higher the FoM the better are the results from the classifier. Figure 9.3 reports FoM calculated using both training set data and test set data. The solid line can be compare with dashed line in Figure 9.4, reporting the FoM of a similar classification method published in Richards et al. (2012) (R2012). The comparison favours the method proposed in this work, whose FoM reach a score of ~ 0.9 above redshift 0.4, while R2012 FoM stays always below 0.4. Two further comparisons can be performed on quantities as Ia-efficiency, the ratio between true type Ia over the total of the same type (see Equation (9.1)), and Ia-purity, the fraction of true positives among all supernovae classified as Ia:

$$\mathrm{Purity}_{Ia} = \frac{N_{Ia}^{\mathrm{true}}}{N_{Ia}^{\mathrm{true}} + N_{Ia}^{\mathrm{false}}} \tag{9.2}$$

They are reported in Figure 9.5 and Figure 9.6 respectively.

Both comparisons again favours the method developed in this work. In conclusion the combination of Gaussian processes, diffusion map and random forest, builds a competitive method to the identification of supernovae Ia from non-Ia.

9.3.1 Future development

The main failure of the presented method is with the classification model not succeeding to classify supernovae Ib/c. In the attempt to solve this issue, some improvement can be introduced. They are listed following the order of task the method executes to provide the classification model.

To understand the influence of K-correction on the final model as explained in Section 4.4 is of key importance, since the comparison of light curves at different redshifts can give a blurred representation of the reality.

The goodness of light curve interpolation should be evaluated in a more rigorous way then eyeball check of a subsample. Light curves not passing the evaluation would be excluded from the diffusion map parameterisation, since they would represent only noise in identifying the

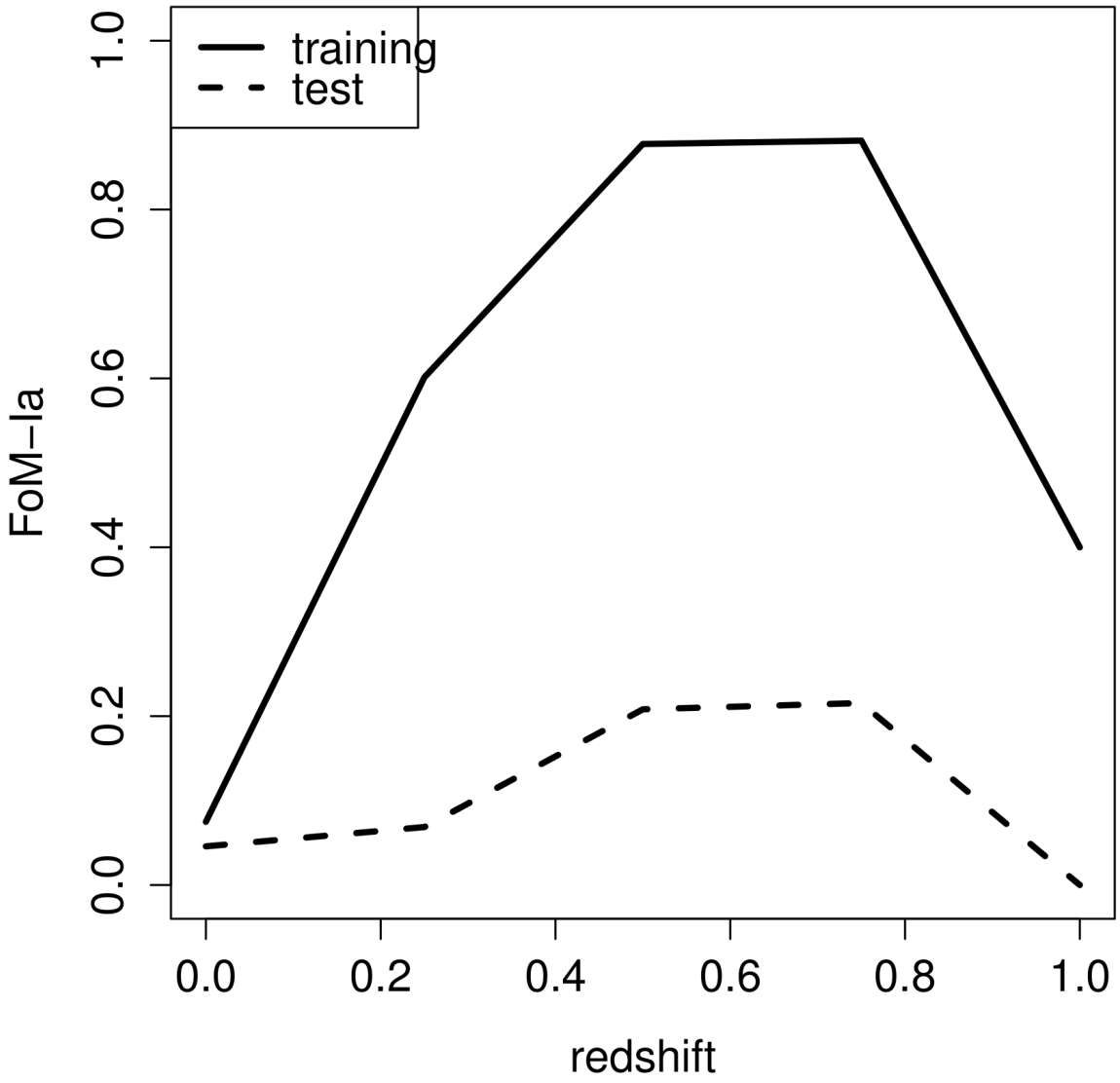

Figure 9.3: Values for the FoM-Ia as function of redshift. Results calculated on both training set and test set data are reported as in legend.

manifold from which parameters derive. The exclusion of bad interpolations would ease the extraction of the best parameters, allowing a good classification model to be built.

The information included in relative light curves intensities can play an important role in further definition of similarities. Thus it is worth to avoid normalisation of interpolated light curves.

Along with the improvements presented, also the training set composition has to change. The SNPhotCC training set is not representative of the population to be classified. An example to show this is the broad class of type II, which contains type IIP, IIn, IIL and IIb subclasses. In terms of type II the test sample is much more populated than the training set. This means that the variety, in terms of light curves shapes, enclosed in the test is not sampled by the small training set (see Figure 8.3 and Figure 8.4). Apart from adding more examples, a way to expand the training set without relying on spectroscopic facilities, is to

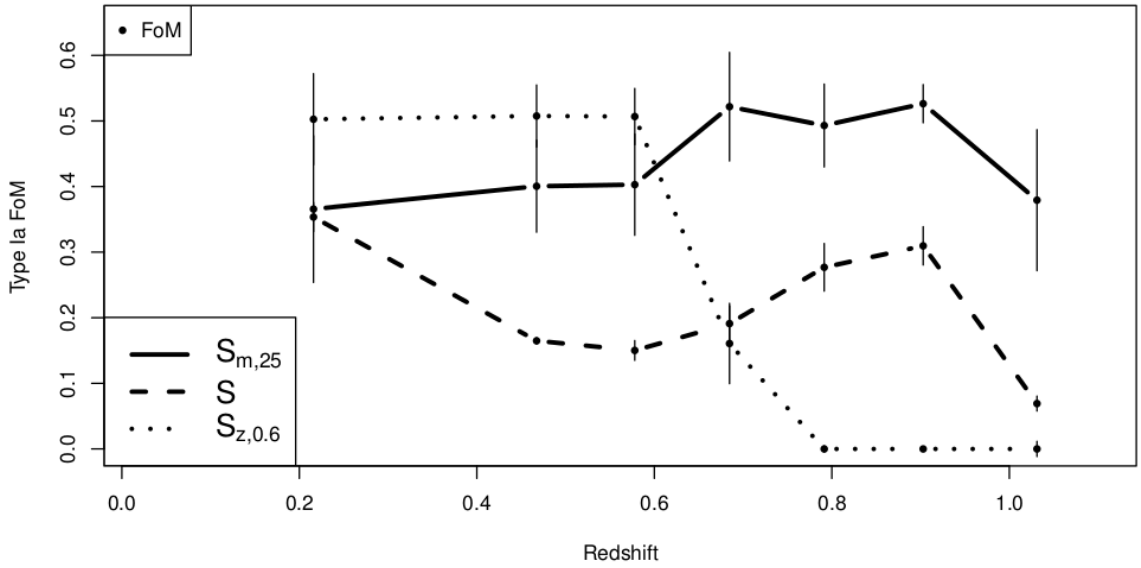

Figure 9.4: Values for the FoM-Ia as function of redshift from (Richards et al. 2012, Figure 11). Curves are reported for different training sets. Dashed curve is relative to SNPhotCC trainig set. Is has to be compared with continous curve in Figure 9.3.

add light curves measured with redder bands. At these wavelengths supernova Ib/c light curves are much more different from Ia light curves, thus the distinction between the two type could be easier.

As a final consideration, it is possible that photometric classification cannot reproduce the classical scheme based on spectroscopic features. It is then worth to investigate which clusters are formed in parameter space, since photometric information could just produce a different classification system. Such a classification scheme could then be related to the spectroscopic one. This kind of investigation can be done using random forest, as explained in (Criminisi et al. 2012, Chapter 5).

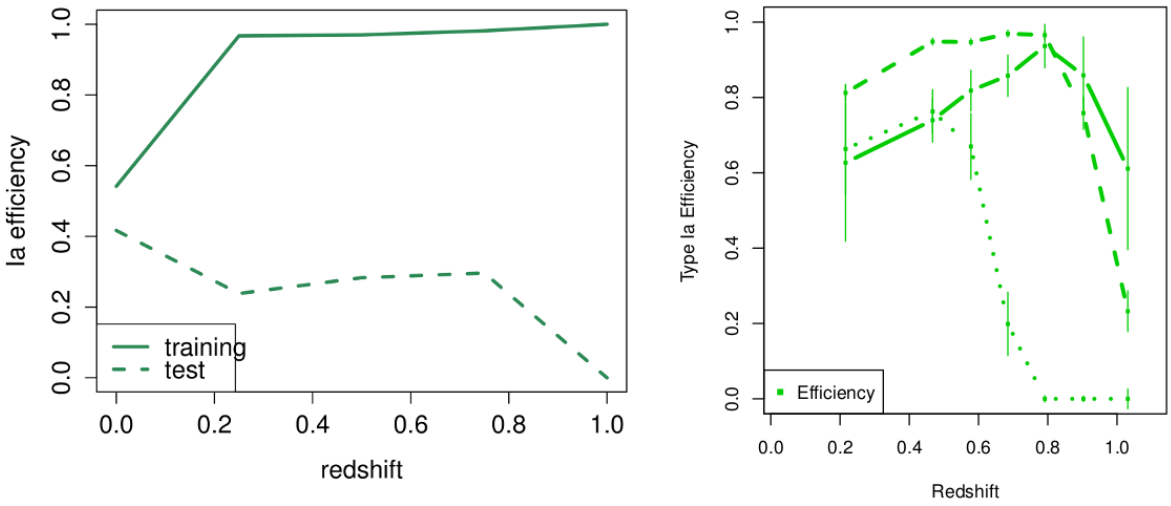

Figure 9.5: Comparison of type Ia efficiency for the presented model (left panel) and R2012 (right panel, dashed line, Richards et al. 2012, Figure 11).

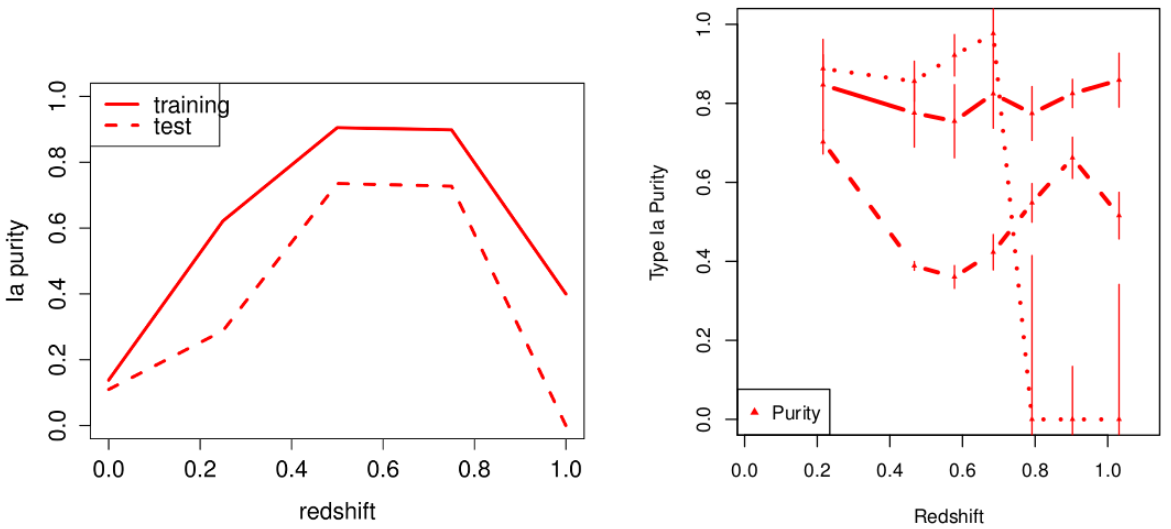

Figure 9.6: Comparison of type Ia purity for the presented model (left panel) and R2012 (right panel, dashed line, Richards et al. 2012, Figure 11).

Appendix A

Diffusion coordinates: 3D distribution

This appendix contains two plots showing the three dimensional distribution of interpolated light curves using the three most important diffusion coordinates: 15, 10 and 3. Figure A.1 reports the distribution for the training set. With respect to Figure 8.3, the clusters formed by type Ia and type II are more distinguished, thus the good results reported in Table 9.1. On the other hand, Figure A.2 shows the distribution of light curves from the test set. In this bigger data set, the two clusters of Ia and II merges. Also the cloud of points from Ib/c class can be seen to follow Ia cluster and merging at the same time with type II cluster. Additionally, the three cluster are much wider distributed with respect to the training set, showing an horizontal structure not visible in Figure A.1. This partially explains the results of the test in Table 9.2.

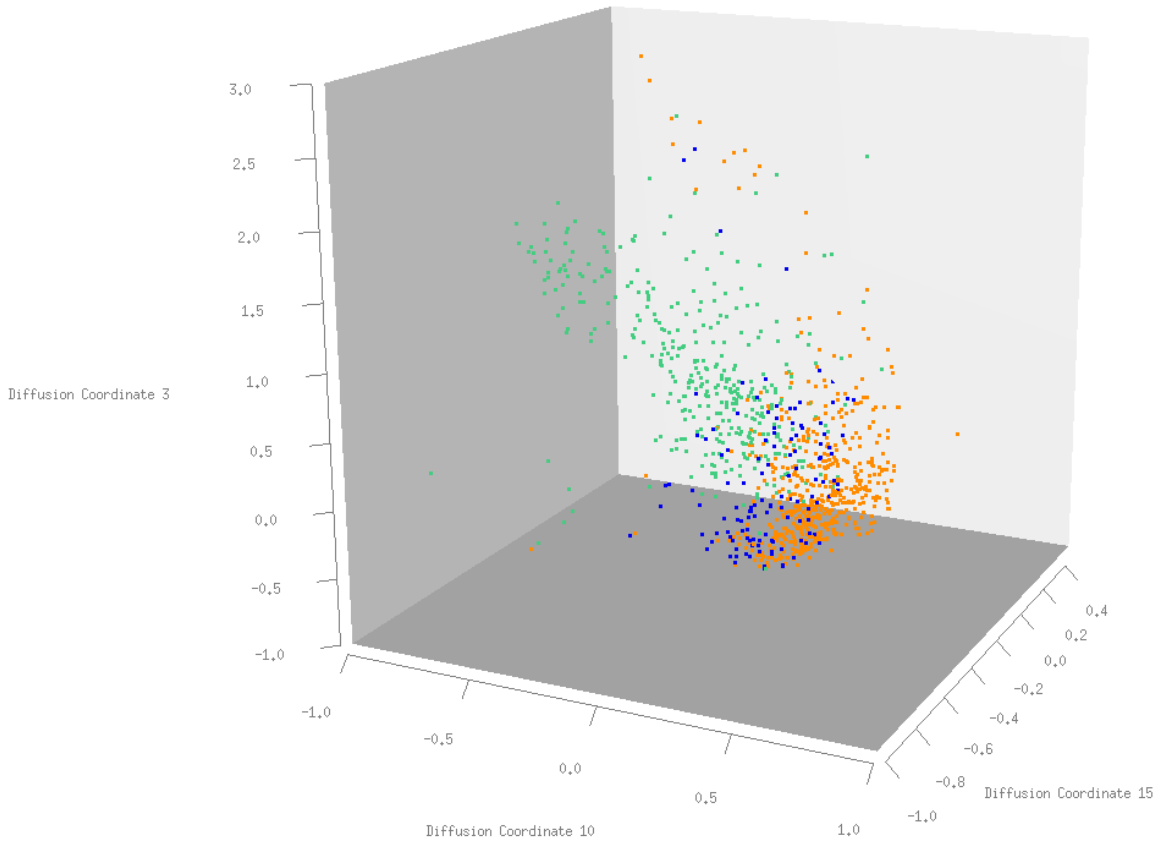

Figure A.1: Diffusion coordinates for the light curves in the training set. Supernovae type are color coded, Ia in orange, II in green and Ib/c in blue.

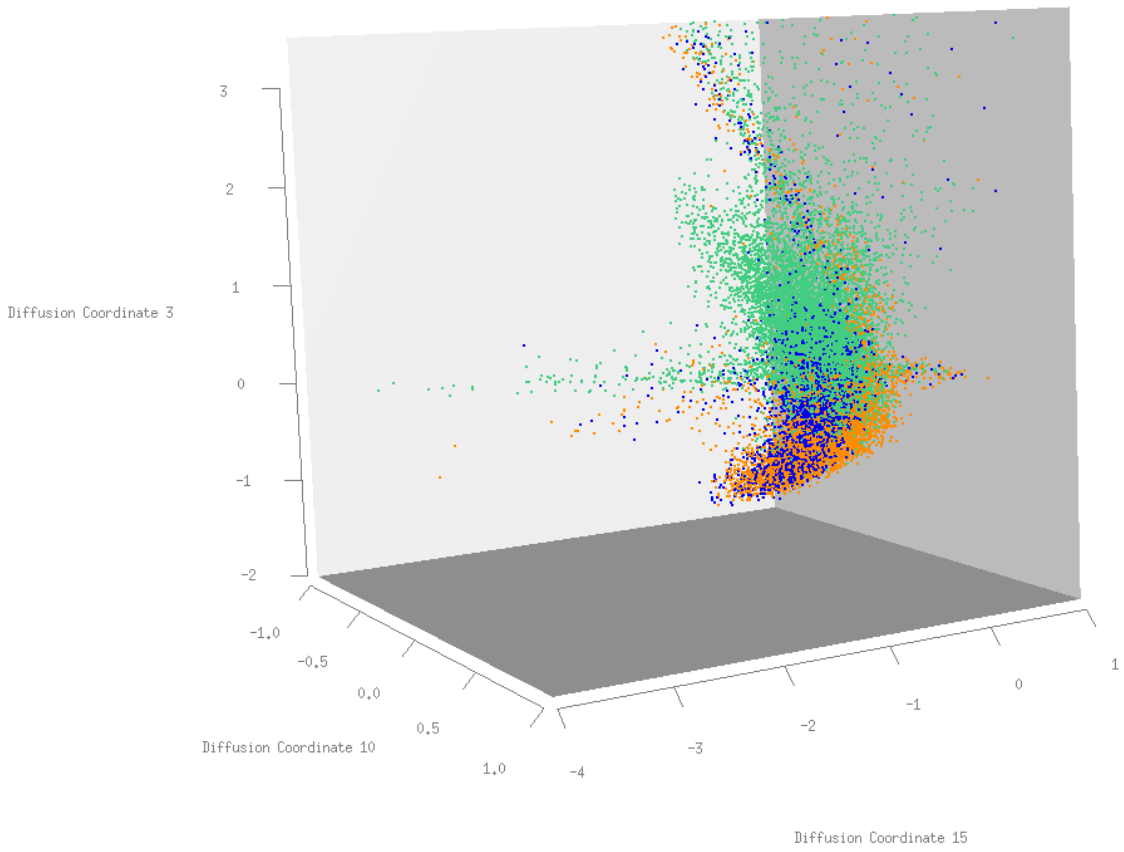

Figure A.2: Diffusion coordinates for the light curves in the test set. Supernovae type are color coded, Ia in orange, II in green and Ib/c in blue.

Bibliography

Adibekyan, V. Z., Sousa, S. G., Santos, N. C., Delgado Mena, E., Gonzlez Hernndez, J. I., Israelian, G., Mayor, M., & Khachatryan, G. 2012, A&A, 545, 32

Allende Prieto, C., Beers, T. C., Wilhelm, R., Newberg, H. J., Rockosi, C. M., Yanny, B., & Lee, Y. S. 2006, The Astrophysical Journal, 636, 804

Alpaydin, E. 2010, Introduction to machine learning, 2nd edn. (The MIT Press)

Bensby, T., & Feltzing, S. 2010, Proceedings of the International Astronomical Union, 5, 300

Bernstein, J. P., Kessler, R., Kuhlmann, S., & Spinka, H. 2009, ArXiv:astro-ph.CO/0906.2955

Bishop, C. M. 2006, Pattern Recognition and Machine Learning (Springer), 738

Branch, D., & Tammann, G. A. 1992, Annual Review of Astronomy and Astrophysics, 30, 359

Breiman, L. 1996, Machine Learning, 10, 262

—. 2001, Machine Learning, 45, 5

Cardelli, J. A., Clayton, G. C., & Mathis, J. S. 1989, The Astrophysical Journal, 345, 245

Carroll, S. M., Press, W. H., & Turner, E. L. 1992, Annual Review of Astronomy and Astrophysics, 30, 499

Chapelle, O., Schölkopf, B., & Zien, A., eds. 2006, Semi-Supervised Learning (Cambridge, MA: MIT Press)

Coifman, R. R., & Lafon, S. 2006, Applied and Computational Harmonic Analysis, 21, 5 , special Issue: Diffusion Maps and Wavelets

Criminisi, A., Shotton, J., & Konukoglu, E. 2012, Foundations and Trends® in Computer Graphics and Vision, 7, 81

de la Porte, J., Herbst, B. M., Hereman, W., & van der Walt, S. J. 2008, in Proceedings of the Nineteenth Annual Symposium of the Pattern Recognition Association of South Africa, ed. F. Niccols, Pattern Recognition Association of South Africa

de Laverny, P., Recio-Blanco, A., Worley, C. C., De Pascale, M., Hill, V., & Bijaoui, A. 2013, The Messenger, 153, 18

de Laverny, P., Recio-Blanco, A., Worley, C. C., & Plez, B. 2012, A&A, 544, 126

De Pascale, M. 2011, Master's thesis, Astronomy Dept., University of Padova

De Pascale, M., Worley, C. C., de Laverny, P., Recio-Blanco, A., Hill, V., & Bijaoui, A. 2014, Astronomy & Astrophysics, 570, A68

du Buisson, L., Sivanandam, N., Bassett, B. A., & Smith, M. 2014, arXiv e-print 1407.4118, 11

Faraway, J., Mahabal, A., Sun, J., Wang, X., Yi, Wang, & Zhang, L. 2014, ArXiv e-prints

Feigelson, E. D., & Babu, G. J. 2012, Modern Statistical Methods for Astronomy (Cambridge University Press)

Fitzpatrick, E. L. 1999, Publications of the Astronomical Society of the Pacific, 111, pp. 63

Gazzano, J.-C., et al. 2010, A&A, 523, A91

Gibson, N. P., Aigrain, S., Roberts, S., Evans, T. M., Osborne, M., & Pont, F. 2012, Monthly Notices of the Royal Astronomical Society, 419, 2683

Gilmore, G., et al. 2012, The Messenger, 147, 25

Goldstein, D. A., et al. 2015, arXiv e-print 1504.02936, 24

Gustafsson, B., Edvardsson, B., Eriksson, K., Jørgensen, U. G., Nordlund, Å., & Plez, B. 2008, A&A, 486, 951

Guy, J., et al. 2007, Astronomy & Astrophysics, 466, 11

Hastie, T., Tibshirani, R., & Friedman, J. 2001, The elements of statistical learning: data mining, inference, and prediction (Springer Verlag)

Helsel, D. 2005, Nondetects and data analysis: statistics for censored environmental data, Statistics in practice (Wiley-Interscience)

Hinkle, K., Wallace, L., Valenti, J., & Harmer, D. 2000, Visible and Near Infrared Atlas of the Arcturus Spectrum 3727-9300 Å (Astronomical Society of the Pacific)

Holwerda, B. W., Reynolds, A., Smith, M., & Kraan-Korteweg, R. C. 2014, Monthly Notices of the Royal Astronomical Society, 446, 3768

Ishida, E. E. O., & de Souza, R. S. 2013, Monthly Notices of the Royal Astronomical Society, 430, 509

Ivezic, Z., et al. 2008, ArXiv e-prints

Jha, S., Riess, A. G., & Kirshner, R. P. 2007, The Astrophysical Journal, 659, 122

Jofr, P. et al. 2014, A&A, 564, A133

Johnson, H. L., & Morgan, W. W. 1953, The Astrophysical Journal, 117, 313

Kessler, R. 2014, SNANA User's Manual: Simulation, Lightcurve Fitters & Cosmology Fitters, Department of Astronomy & Astrophysics - University of Chicago

Kessler, R., Conley, A., Jha, S., & Kuhlmann, S. 2010, eprint arXiv:1001.5210

Kessler, R., et al. 2009, The Astrophysical Journal Supplement Series, 185, 32

Kessler, R., et al. 2009, PASP, 121, 1028

—. 2010, PASP, 122, 1415

Kleinbaum, D. G., & Klein, M. 2005, Survival Analysis: A Self-Learning Text (Springer Science & Business Media), 590

Kordopatis, G., Recio-Blanco, A., de Laverny, P., Bijaoui, A., Hill, V., Gilmore, G., Wyse, R. F. G., & Ordenovic, C. 2011, A&A, 535, A106

Kuznetsova, N. V., & Connolly, B. M. 2007, ApJ, 659, 530

Leibundgut, B. 2007, General Relativity and Gravitation, 40, 221

Liaw, A., & Wiener, M. 2002, R News, 2, 18

Lo Curto, G. 2011, HARPS User Manual, 2nd edn., ESO (European Southern Observatory)

MacKay, D. 2003, Information Theory, Inference and Learning Algorithms (Cambridge University Press)

Majewski, S. R., et al. 2007, in Bulletin of the American Astronomical Society, Vol. 39, American Astronomical Society Meeting Abstracts, 962

Mayor, M., et al. 2003, The Messenger, 114, 20

Nocedal, J., & Wright, S. 2006, Numerical Optimization (Springer Science & Business Media), 636

O'Donnell, J. E. 1994, The Astrophysical Journal, 422, 158

Perlmutter, S., et al. 1999, ApJ, 517, 565

Poznanski, D., Gal-Yam, A., Maoz, D., Filippenko, A. V., Leonard, D. C., & Matheson, T. 2002, in Riess et al. (1998), 833–845, 833

Prieto, C. A. 2010, Proceedings of the International Astronomical Union, 5, 304

Prugniel, P., Soubiran, C., Koleva, M., & Borgne, D. L. 2007, arXiv:astro-ph/0703658

R Core Team. 2015, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria

Rasmussen, C. E., & Williams, C. K. I. 2006, Gaussian Processes for Machine Learning (Massachussets Institute of Technology)

Recio-Blanco, A. 2014, Proceedings of the International Astronomical Union, 9, 366

Recio-Blanco, A., Bijaoui, A., & de Laverny, P. 2006, Monthly Notices of the Royal Astronomical Society, 370, 141

Recio-Blanco, A., et al. 2014, Astronomy & Astrophysics, 567, A5

Reddy, B. E., Lambert, D. L., & Prieto, C. A. 2006, Monthly Notices of the Royal Astronomical Society, 367, 1329

Richards, J. 2014, diffusionMap: Diffusion map, R package version 1.1-0

Richards, J. W., Freeman, P. E., Lee, A. B., & Schafer, C. M. 2009, ApJ, 691, 32

Richards, J. W., Homrighausen, D., Freeman, P. E., Schafer, C. M., & Poznanski, D. 2012, MNRAS, 419, 1121

Riess, A. G., et al. 1998, AJ, 116, 1009

Savage, B. D., & Mathis, J. S. 1979, Annual Review of Astronomy and Astrophysics, 17, 73

Schapire, R. E. 1990, Machine Learning, 5, 197

Schlafly, E. F., & Finkbeiner, D. P. 2011, The Astrophysical Journal, 737, 103

Schlegel, D. J., Finkbeiner, D. P., & Davis, M. 1998, The Astrophysical Journal, 500, 525

Seaton, . 1979, Monthly Notices of the Royal Astronomical Society, 187

Sousa, S. G., Santos, N. C., Israelian, G., Lovis, C., Mayor, M., Silva, P. B., & Udry, S. 2011a, A&A, 526, 99

Sousa, S. G., Santos, N. C., Israelian, G., Mayor, M., & Monteiro, M. J. P. F. G. 2007, A&A, 469, 783

Sousa, S. G., Santos, N. C., Israelian, G., Mayor, M., & Udry, S. 2011b, arXiv:1108.5279

Sousa, S. G., et al. 2008, A&A, 487, 373

Steinmetz, M., et al. 2006, AJ, 132, 1645

Sullivan, M., et al. 2006, AJ, 131, 960

The GPy authors. 2012–2014, GPy: A Gaussian process framework in python, `http://github.com/SheffieldML/GPy`

Tsantaki, M., Sousa, S. G., Adibekyan, V. Z., Santos, N. C., Mortier, A., & Israelian, G. 2013, A&A, 555, 150

Valenti, ., & Piskunov, . 1996, Astronomy and Astrophysics Supplement

Varughese, M. M., von Sachs, R., Stephanou, M., & Bassett, B. A. 2015, arXiv e-print 1504.00015, 14

Wallace, L., Hinkle, K., & Livingston, W. 1998, An atlas of the spectrum of the solar photosphere from 13,500 to 28,000 cm-1 (3570 to 7405 A) (National Optical Astronomy Observatories)

Wasserman, L. 2006, All of Nonparametric Statistics, Springer Texts in Statistics (Springer)

—. 2010, All of Statistics: a concise course in statistical inference, Springer Texts in Statistics (Springer)

Witten, I. H., Frank, E., & Hall, M. A. 2011, Data Mining: Practical machine learning tools and techniques, 3rd edn. (Elsevier Ltd)

Worley, C. C., de Laverny, P., Recio-Blanco, A., Hill, V., Bijaoui, A., & Ordenovic, C. 2012, A&A, 542, 48

Worley, ., deLaverny, ., Recio-Blanco, ., & Hill, . 2014, International Workshop on Stellar Spectral Libraries ASI Conference Series, 11

Yanny, B., et al. 2009, AJ, 137, 4377

Zumel, N., Mount, J., & Porzak, J. 2014, Practical data science with R (Manning)