University of Cape Town

Thesis Presented for the Degree of Doctor of Philosophy in the Department of Mathematics and Applied Mathematics

Aspects of Bayesian inference, classification and anomaly detection

Author:
Ethan Roberts

Supervisor:
Prof. Bruce Bassett

March 2021

Declaration of Authorship

I, Ethan Roberts, declare that this thesis titled, "Aspects of Bayesian inference, classification and anomaly detection" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed: Signed by candidate Date: 15 March 2021

Declaration of Publications

Here, publications resulting from work that went into this thesis are listed in the order in which they were accepted for publication below. Publication 3 is an extended version of publication 2, which we were invited to submit after the Intelligent Systems Design and Applications conference 2019.

Publication 1

Roberts E., Lochner M., Fonseca J., Bassett B.A., Lablanche P.-Y., Agarwal S. (2017) zBEAMS: a unified solution for supernova cosmology with redshift uncertainties. In: Journal of Cosmology and Astroparticle Physics, vol 2017. DOI: 10.1088/1475-7516/2017/10/036.

Publication 2

Roberts E., Bassett B.A., Lochner M. (2021) Bayesian Anomaly Detection and Classification for Noisy Data. In: Abraham A., Siarry P., Ma K., Kaklauskas A. (eds) Intelligent Systems Design and Applications. ISDA 2019. Advances in Intelligent Systems and Computing, vol 1181. Springer, Cham. DOI: 10.1007/978-3-030-49342-4_41.

Publication 3

Roberts E., Bassett B.A., Lochner M. (2020) Bayesian anomaly detection and classification for noisy data. In: International Journal of Hybrid Intelligent Systems, vol 16. DOI: 10.3233/HIS-200282.

Acknowledgements

The journey through a PhD is one that is long and arduous, and is characterised by the lonesome tackling, jousting and wrestling with a series of problems all located in a particular nook of human knowledge. In spite of this lonesome undertaking, my PhD journey would not have been possible without the help and contributions of a number of people. Here, I acknowledge these contributions.

The first and certainly most notable acknowledgement is to my supervisor, Bruce Bassett, whose ideas formed the backbone of this thesis, and whose continuous input and guidance were invaluable to me. Bruce has, over the course of my post-graduate studies, relayed a massive corpus of knowledge to me through what I can only describe as a large number of uncannily applicable contributions to my work. A massive thanks as well to Michelle Lochner, who in addition to sharing her penchant for science and scientific programming, contributed a lot to both the zBEAMS and BADAC publications, particularly with the generation of the data used in the analysis. I am also hugely grateful to José Fonseca, Pierre-Yves Lablanche and Shankar Agarwal, who in addition to Bruce and Michelle coauthored the zBEAMS publication, and helped with the derivations of equations and the generation of the simulated catalog data, in what was one of the most rewarding collaborations I have been a part of.

A huge thanks as well to Max Hayes, Evander Nyoni, Nadeem Oozeer, Emmanuel Sekyi, Cong Ma, Alireza Vafaei Sadr and Chris Finlay for comments on this thesis.

Abstract

The primary objective of this thesis is to develop rigorous Bayesian tools for common statistical challenges arising in modern science where there is a heightened demand for precise inference in the presence of large, known uncertainties. This thesis explores in detail two arenas where this manifests.

The first is the development and testing of a unified Bayesian anomaly detection and classification framework (BADAC) which allows principled anomaly detection in the presence of measurement uncertainties, which are rarely incorporated into machine learning algorithms. BADAC deals with uncertainties by marginalising over the unknown, true value of the data. Using simulated data with Gaussian noise as an example, BADAC is shown to be superior to standard algorithms in both classification and anomaly detection performance in the presence of uncertainties. Additionally, BADAC provides well-calibrated classification probabilities, valuable for use in scientific pipelines. BADAC is therefore ideal where computational cost is not a limiting factor and statistical rigour is important. We discuss approximations to speed up BADAC, such as the use of Gaussian processes, and finally introduce a new metric, the Rank-Weighted Score (RWS), that is particularly suited to evaluating an algorithm's ability to detect anomalies.

The second major exploration in this thesis presents methods for rigorous statistical inference in the presence of classification uncertainties and errors. Although this is explored specifically through supernova cosmology, the context is general. Supernova cosmology without spectra will be an important component of future surveys due to massive increases in data volumes in next-generation surveys such as from the Vera C. Rubin Observatory. This lack of supernova spectra results both in uncertainty in the redshifts and type of the supernova, which if ignored, leads to significantly biased estimates of cosmological parameters. We present a hierarchical Bayesian formalism, zBEAMS, which addresses this problem by marginalising over the unknown or uncertain supernova redshifts and types to produce unbiased cosmological estimates that are competitive with supernova data with fully spectroscopically confirmed redshifts. zBEAMS thus provides a unified treatment of both photometric redshifts, classification uncertainty and host galaxy misidentification,

effectively correcting the inevitable contamination in the Hubble diagram with little or no loss of statistical power.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The rapid growth of the machine learning over the last few decades has meant that machine learning algorithms are now used in many applications, even in the physical sciences. The number of papers released on the application of machine learning in science has grown exponentially in recent years (see [5; 63; 81; 123; 131; 139], to list a few). This growth is driven by the availability of huge volumes of data, the exponential growth of available computing power and the continuing development of new algorithms.

This rapid growth in machine learning has helped facilitate the undertaking of large 'big data' experiments, where traditional data analysis methods would be on their own insufficient to fully utilise the volume of data created. One example of this is in astronomy, where large surveys such as those undertaken by the Vera C. Rubin telescope in Chile and the Square Kilometre Array in South Africa and Australia will observe orders of magnitude more data than any experiments that pre-date them [91; 92; 66]. While these projects have many fundamental science objectives, they also afford the opportunity for serendipitous discovery. The classification of transients, and the discovery of new/anomalous types of objects in this large volume of data is an important problem. Anomaly detection is critical at this junction since manual inspection of this data volume is not possible at a meaningful scale, and historically this would have been the primary mode of discovering unexpected scientific or experimental phenomena. Machine learning is a vital tool in this context.

Examples of work in this area include [142; 32; 34; 114], though more are shown in section 1.3.

In these cases, results returned by machine learning algorithms need to be well calibrated to ensure biases are not introduced into the analysis pipeline. This is an important consideration since most machine learning algorithms do not return calibrated probabilities as standard.

A problem encountered with the use of machine learning algorithms in scientific applications that is not typically encountered in other areas, is that the data often have associated measurement uncertainties, which are rarely incorporated into machine learning algorithms. Failing to account for these measurement uncertainties can lead to inaccurate predictions, particularly in cases where the measurement uncertainties are large. Additionally, failing to correctly account for statistical errors can result in difficulty constraining systematic errors that arise elsewhere in the scientific analysis pipeline. This is an important consideration, since accounting for these systematic errors comprises a large portion of many analytical studies [76].

These large surveys are not just a challenge from a machine learning perspective, they will have an impact on fundamental science as well. Surveys undertaken by the Vera C. Rubin telescope will record photometric data from of order 10^5 type Ia supernova candidates [91]. Observations of type Ia supernovae have been, and are still, a vital probe for cosmology. Using type Ia supernovae to better constrain cosmological parameters is difficult without spectroscopic follow-up, primarily since the resulting photometric redshift uncertainties are large [72; 144]. Doing so is however necessary if research of type Ia supernovae is to stay competitive with other methods, such as studies of the Cosmic Microwave Background (CMB) and Baryon Acoustic Oscillations (BAO) [148]. Additionally, supernova type is not unambiguous for photometric observations, and constraining errors on supernova classifications is also an important step in ensuring that type Ia supernovae remain a useful probe for cosmology. Moreover, frameworks developed for cosmological inference in the context of photometric supernova observations need to deal with statistical, classification and systematic errors in a unified way in order to maintain a high degree of statistical rigour.

The problems outlined in this section motivate the development of robust statistical tools. These problems are to a large extent unsolved, and finding applicable solutions has the potential to make a large impact on research in the respective areas of

fundamental science.

1.2 Problem Formulation

As outlined in section 1.1, several of the problems in the field of astronomy are strongly related to the fields of machine learning and Bayesian inference. This thesis tackles two of these interdisciplinary problems.

The first is the problem of anomaly detection and classification in the presence of measurement uncertainties on the data. This poses a unique challenge: how can one perform these machine learning tasks when the exact values of the features are unknown, or not known with certainty? This thesis tackles this problem using Bayesian hierarchical modelling. This approach allows for statistically rigorous inference of class labels in the case where measurement uncertainties are large relative to the inter/intra-class variability. This is important since in this case determining the true class of an object can be ambiguous even to an expert.

This technique can also be used to address the second problem this thesis tackles. The second problem is that of Bayesian inference, applied to cosmology for photometric supernova observations. This is achieved using Bayesian hierarchical modelling, which allows for the marginalisation over uncertainty in observed variables.

Bayesian hierarchical models are an ideal tool for these kinds of problems, since they offer a means of combining the analysis of observed and model parameters in a statistically rigorous way.

1.2.1 Aims and Objectives of this Study

Taking into account the motivation for this work, as well as the two problems discussed in the problem formulation, the aim of this work is to explore and develop some rigorous Bayesian tools for use in astronomy and related fields, primarily based on the framework of Bayesian hierarchical models. In the development of these tools, the following objectives are proposed:

1. Develop a statistically rigorous framework for anomaly detection and classifi-

cation in the presence of measurement uncertainties.

2. Test the proposed framework against existing approaches.
3. Create several hierarchical models for the observations of type Ia supernovae in the context of a cosmological analysis under differing observational paradigms:
   - Spectroscopic observations
   - Photometric observations with host galaxy spectra
   - Purely photometric observations
4. Determine the impact of increased redshift error in photometric observations on our ability to constrain cosmological parameters.

1.3 Related Work

Classification and anomaly detection are well studied topics, both in general, as well as in the field of astronomy. Some examples of transient and supernova classification in astronomy are [105; 146; 101; 118; 111]. More general frameworks for anomaly detection in astronomy have also been investigated [88], which rely on active learning. A notable project on machine classification in astronomy was the Photometric LSST Astronomical Time-Series Classification Challenge (PLAsTiCC) [60; 73; 95]. PLAsTiCC was a competition/challenge to the astronomy and data science communities to help develop classification and anomaly detection algorithms for astronomical transients. This competition was won using Gaussian process augmentation [9].

Another area in which the control of uncertainties is incredibly important when deploying machine learning algorithms is in the medical field. An example of this is in computer vision for medical images [120]. More general work has also been done on this topic in the context of Bayesian neural networks [70].

The most notable work in supernova cosmology is the discovery of the accelerated expansion of the universe due to dark energy [113; 43]. The volume of research in this field has since grown substantially, with most research focusing around better constraining cosmological parameters. This has been achieved to some extent through better control of systematics [97; 127; 74; 132; 93; 68; 8; 20]. These systematic uncertainties include supernova redshift and type uncertainty introduced by pure-photometric observations.

1.4 Thesis Outline

The following chapters are organised into two literature review chapters, covering the topics of Bayesian statistics and machine learning, as well as two studies that deal with two independent applications of Bayesian hierarchical modelling.

1.4.1 Chapter 2: Bayesian Statistics

This chapter presents an introduction to Bayesian statistics and also introduces notation and terminology on probability theory that is central to the research presented in chapters 4 and 6. This chapter also includes a section on Markov Chain Monte Carlo (MCMC), a numerical sampling technique often used in Bayesian statistics, and is applied in chapter 6 as well.

1.4.2 Chapter 3: Machine Learning

This chapter presents a broad review of topics within the field of machine learning and reviews a few specific algorithms that are used as benchmark algorithms in chapter 4. It also reviews some commonly used metrics in machine learning and presents a novel metric, the rank-weighted score (RWS), for use in anomaly detection.

1.4.3 Chapter 4: Bayesian Anomaly Detection and Classification

This chapter presents the Bayesian Anomaly Detection and Classification (BADAC) formalism for noisy data. BADAC is a statistically rigorous joint anomaly detection and classification scheme for use when there are measurement uncertainties on the associated data. This chapter proceeds by focussing on a specific case of the general formalism, the case with Gaussian measurement uncertainties. This chapter also presents a series of experiments that demonstrate the performance of BADAC on simulated data against a series of benchmark algorithms.

1.4.4 Chapter 5: Breaking BADAC

This chapter expands on chapter 4, by presenting a series of experiments that test how robust the performance of BADAC is on data which violates the BADAC noise model. This chapter illustrates how the performance of BADAC decreases in these cases, and presents ways to address/mitigate this.

1.4.5 Chapter 6: Bayesian Estimation Applied to Multiple Species for Photometric Data

This chapter presents an application of Bayesian hierarchical models in the field of supernova cosmology. Bayesian Estimation Applied to Multiple Species for photometric data (zBEAMS) is an extension of the original BEAMS formalism. The zBEAMS formalism jointly models observational and cosmological parameters in a hierarchical framework, in order to fit for cosmological parameters in the case where there are photometric redshift errors. This chapter then demonstrates that the zBEAMS formalism is able to recover a fiducial cosmology using a series of catalogue simulations. Parameters are fit for using Markov Chain Monte Carlo (MCMC) modelling.

Chapter 2

Bayesian Statistics

This chapter presents an outline on the fundamentals of Bayesian statistics. Section 2.1 introduces some of the notation that will be used throughout this thesis. While this may seem like over-clarifying for the most part, doing so accurately is necessary since chapters 4 and 6 are notationally complex. Additionally, this leads into a description of Bayes' law that is intuitive to understand. Chapters 4 and 6 show original work that builds from this.

2.1 Introduction to Bayes' Law

The field of Bayesian statistics owes its name to the Reverend Thomas Bayes (b. 1701), an early pioneer on the subject [7]. The first written account of Bayes' law in the form we know it today was however given by Pierre-Simon Laplace in 1814 [86]. Bayes had previously only described the law in words. Bayesian statistics remains an active and fruitful area of research in spite of the fact it has been around for almost three centuries. The field is coming to the fore because (1) it is theoretically sound and therefore attractive in subjects with lots of data trying to get optimal results, but (2) it is computationally demanding, and has only become possible with the advent of Moore's law. Another driver of the growth of Bayesian statistics is its far reaching applications, especially those where results differ from or are more interpretable than those produced by orthodox (or "frequentist") statistics. Bayes' law was historically used to combine scientific measurements with differing associated confidence levels, and this is where it still sees a lot of use today. An

intrinsic feature of Bayesian statistics is that theories/models and data are viewed on an equal footing. This footing is probability-centric, and as such, this section presents an introduction to Bayesian statistics from a probabilistic viewpoint.

Consider two independent random events (or variables), A and B. The probabilities of a specific outcome for each of these events can be written as $P(A = a)$ and $P(B = b)$ respectively. The probability of both events A and B occurring is written as $P(A, B)$. Since it has been posited that A and B are independent, $P(A, B)$ is simply $P(A)P(B)$, since the probability of independent events is just the product of the probabilities of each event. In the more general case, events A and B need not be independent. In this case, $P(A, B)$ can be evaluated using the product rule:

$$P(A, B) = P(A|B)P(B) \tag{2.1}$$

Here, $P(A|B)$ is the probability of event A *given* event B. Even when events A and B are not independent, there is a simple way to evaluate $P(A, B)$, as long as $P(A|B)$ can be evaluated.

The probability of events A and B occurring, is the same as the probability of events B and A occurring:

$$P(A, B) = P(B, A) \tag{2.2}$$

$$P(A|B)P(B) = P(B|A)P(A) \tag{2.3}$$

Here the order in which we write the variables in the joint distribution does not matter. Equation 2.3 can be reordered such that $P(A|B)$ is the subject of the formula:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.4}$$

Equation 2.4 is a statement of Bayes' law in its most simple form. It shows a simple relation between $P(A|B)$ and $P(B|A)$. Specifically, the probability of A given B is equal to the probability of B given A, multiplied by the ratio of the probability of A over the probability of B. Section 2.2.1 describes the significance of this relation, as well as why it is important for inference.

2.1.1 A Simple Example

Consider a simple example[1], inspired to some extent by the current global pandemic. Consider an antibody test for COVID-19, with accuracy 90%. Assuming that the prevalence of COVID-19 among the South African population is 2%, what is the chance that Jack, a randomly selected member of the South African public, has COVID-19, given a positive antibody test result?

Here, $P(+|\text{infected}) = 0.9$, the likelihood of a positive test result given infection, is not the probability that Jack has COVID-19, but rather the probability he would test positive for COVID-19 if it was known he was infected. The probability that Jack is infected can be expressed using Bayes' law:

$$P(\text{infected}|+) = \frac{P(+|\text{infected})P(\text{infected})}{P(+)} \tag{2.5}$$

where:

$$P(+) = P(+|\text{infected})P(\text{infected}) + P(+|\text{not infected})P(\text{not infected}) \tag{2.6}$$

is the evidence, found by summing the probabilities of all the circumstances under which one may observe a positive test result. This means:

$$P(\text{infected}|+) = \frac{0.9 \times 0.02}{0.9 \times 0.02 + 0.1 \times 0.98} \tag{2.7}$$

$$P(\text{infected}|+) = 0.1551... \tag{2.8}$$

$$P(\text{infected}|+) \approx 15.5\% \tag{2.9}$$

Why is this? It seems strange that Jack has a $\sim 15.5\%$ chance of being infected given a positive test that is 90% accurate. The reason becomes clear if we consider Jack as part of a population. For a population of 1000, with a 2% COVID-19 prevalence, 20 people will be infected, and 980 will not. If every member of this population were to be tested with an antibody test that was 90% accurate, 18 out of 20 infected people would test positive, and 98 out of 980 not infected people would also test positive. This means 18 of the $(18 + 98)$ people who tested positive were actually infected, corresponding to $\sim 15.5\%$.

This shows how prior knowledge about disease prevalence is an important aspect to

[1] Versions of this example are commonly used to introduce the idea of Bayes' law.

solving this problem. The above example highlights that in fact prior knowledge is often an important aspect to making predictions, and demonstrates how this can be done using Bayes' law.

2.2 A Probabilistic View of Statistics

The use of probability is ubiquitous within all sub-fields of statistics. Differences between these sub-fields often arise from which probability or probability distribution is considered in a statistical analysis and how it is interpreted [3]. This section expands on the introduction to probability theory and Bayes' law presented in section 2.1, and illustrates the need for a Bayesian approach in certain statistical applications.

2.2.1 Bayes' Law for Inference

Equation 2.4 showed Bayes' law in the most simple form. This section presents a formulation of this rule that is more useful for inference - "a conclusion reached on the basis of evidence and reasoning" [134]. Consider a generic experiment. If there are some observed data, D, and a model, M, with associated parameters, θ, that is fitted to the data, then the probability distribution over these parameters can be found using Bayes' law:

$$P(\theta|D, M) = \frac{P(D|\theta, M)P(\theta|M)}{P(D|M)} \tag{2.10}$$

This formulation of Bayes' law is important since the conditional probability of $\theta|D, M$ takes into account that there is always some degree of subjectivity inherent to the choice of model used to describe/make inferences from data. A great chapter on Bayes' law in the context model fitting is presented by [94].

Here, and throughout the rest of this thesis, the M is dropped from the equation for notational simplicity, as shown in equation 2.11. It should however always be assumed.

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} \tag{2.11}$$

Each of the terms in equation 2.11 have commonly used names: $P(\theta|D)$ is the

posterior, $P(D|\theta)$ is the likelihood, $P(\theta)$ is the prior and $P(D)$ is the evidence (also referred to as the "marginal likelihood"), which is also sometimes denoted by Z. The evidence term is calculated by *marginalising* over all possible parameter values, as shown in equations 2.12 and 2.13 for discrete and continuous parameter spaces respectively. Marginalisation is discussed in section 2.2.3.

$$Z \equiv P(D) = \sum_{\theta} P(D|\theta)P(\theta) \tag{2.12}$$

$$Z \equiv P(D) = \int_{\theta} P(D|\theta)P(\theta)d\theta \tag{2.13}$$

It should be noted that the evidence term is just a summation or integral over the numerator in equation 2.11. What this means is that the term, $P(D)$, acts as a normalisation constant. This ensures that $\int P(\theta|D)d\theta = 1$. For some applications, such as those discussed in section 2.3, the calculation of the evidence is not required since it is independent of model parameters, and we are only interested in the best fitting parameters. These parameters are found using a ratio of probabilities, both of which have the evidence in the denominator, which cancels out. The evidence is just a scaling factor in this case that is not needed.

2.2.2 Naïve Bayes

Naïve Bayes is the set of supervised learning classification algorithms that utilise the conditional independence assumption commonly used in Bayes' law. For data with n features, x_i, where $i \in [1, n]$ and class label, y, the posterior over y given the features is expressed with Bayes' law:

$$P(y|x_1, x_2, ..., x_n) = \frac{P(x_1, x_2, ..., x_n|y)P(y)}{P(x_1, x_2, ..., x_n)} \tag{2.14}$$

The likelihood term, $P(x_1, x_2, ..., x_n|y)$, can be rewritten using the chain rule:

$$P(x_1, x_2, ..., x_n|y) = P(x_1|x_2, ..., x_n, y)P(x_2|x_3, ..., x_n, y)...P(x_n|y) \tag{2.15}$$

Now assuming the features are independent of one another, this can be rewritten:

$$P(x_1, x_2, ..., x_n|y) = P(x_1|y)P(x_2|y)...P(x_n|y) \tag{2.16}$$

$$P(x_1, x_2, ..., x_n|y) = \prod_i P(x_i|y) \tag{2.17}$$

Naïve Bayes classifiers make predictions under the assumption stated in equation 2.17. Despite the fact the independence assumption inherent in using Naïve Bayes is rarely accurate, it is widely utilised in many applications, and is often effective in practice [145]. Investigations about why this is have also been done [155].

2.2.3 Marginalisation

For a multivariate problem, it is often the case that we wish to learn information about the posterior distribution of a particular parameter without the conditional dependence of another parameter/parameters. In this case, these parameters are referred to as nuisance parameters. More specifically, the joint distribution is not what is of interest, since the nuisance parameters are not key. We do not care about them other than that they correlate with parameters we do care about. We only care about the posterior for the parameters of interest. The nuisance parameters are eliminated by a process called *marginalisation*. Marginalisation is the integration of a multivariate probability distribution w.r.t. at least one of the variables:

$$P(A|B) = \int_i P(A|B, C_i)P(C_i|B)\ dC_i \tag{2.18}$$

where C_i is the continuous nuisance parameter. Or:

$$P(A|B) = \sum_i P(A|B, C_i)P(C_i|B) \tag{2.19}$$

where C_i is the discrete nuisance parameter. These are the same equations as are used to calculate the model evidence in equations 2.12 and 2.13, since the process of calculating the model evidence is similar to that of marginalisation, though over *all* the model parameters (hence why the evidence is also referred to as the marginal likelihood).

For the case where the posterior distribution has been obtained with MCMC methods, numerical marginalisation over a particular parameter is trivial, since one only

need histogram the points in the axis of the parameter of interest to obtain the marginal distribution [80]. MCMC is discussed in detail in section 2.3.

2.2.4 Priors

Prior probability distributions (commonly called priors) in Bayesian statistics incorporate beliefs about which values of the model parameters are probable before (prior to) the measurement of any data. This can seem like a cause of confusion since it seems to be a highly subjective choice. The incorporation of prior distributions is however vital in determining the posterior distribution, and choosing a prior that encapsulates the researcher's true prior beliefs is crucial to do so [41]. More fundamentally though, we wish to obtain a posterior distribution for science, and the only way to do this mathematically is through a prior.

An additional point about priors becomes apparent when one considers the simulation of some data. The values taken on by the variables that comprise this simulation are selected by drawing them randomly from a probability distribution. Using a prior distribution over a corresponding physical quantity for this is a useful tool to ensure the data are representative of what one expects to observe in reality. Prior distributions are therefore central in considering the mechanics of a given simulation, or a statistical analysis in general.

2.2.4.1 Proper Priors

A proper prior is a prior probability distribution that integrates to one:

$$\int_{\theta} P(\theta) d\theta = 1 \tag{2.20}$$

An improper prior, is one that does not integrate to one. A prior that does not integrate to one or cannot be normalised to integrate to one doesn't make particular sense as a probability distribution, since the probability of all possible outcomes should sum to one. Using improper priors, particularly in hierarchical models, can lead to biases [46]. This is discussed in some detail in chapter 6.

A corollary of this is that one should not use uniform unbounded priors. Particularly, $P(\theta) = 1$ should not be specified as a prior distribution, since this integrates to

∞. This would indicate that one's prior knowledge about the variable θ is that it is equally likely to take on any value, which is generally not the case for model parameters. This can be mitigated by using a top hat prior that is uniform inside the range $[\alpha, \beta]$, and zero everywhere else. Here a glimpse of why Bayesian methods are sometimes more suitable than their frequentist counterparts appears: failing to specify an informative prior means inappropriate model fits can be applied to data, which can lead to biased parameter estimation.

2.2.4.2 Informative and Uninformative Priors

An informative prior is one that expresses specific information about a variable [80; 104]. An uninformative prior does not. Weakly informative priors express partial information about a variable, and often 'uninformative' priors would be better descried as 'weakly informative' since they do express some objective information about the variable. For example, using a prior to constrain a variable to be positive or below a particular limit is considered uninformative.

The choice about whether an informative or uninformative prior should be used is often determined by whether it is appropriate to encode subjective or objective information about the uncertainty of a variable [104]. Subjective priors can better incorporate the true uncertainty associated with a particular variable into a statistical analysis. Uninformative priors are also referred to as objective priors, since the information they express objectively encapsulates uncertainty on they variables they describe. Another way to say this, is that objective priors encode information in a way that can be rigorously justified. The choice about whether to specify a subjective or objective prior is however a philosophical one, and the use of either is commonplace in practice [42]. A motivation for the use of objective priors appears when viewing the prior and posterior relationship as: initial belief + data = updated belief. An objective prior will typically lead to larger information gain when going from prior to posterior distributions. This is useful when designing an experiment where the objective is to yield the largest information gain given some observed data. Something to note is that if we change our parameters to a new set of parameters, $\theta' = f(\theta)$, an uninformative prior on θ may become a highly informative prior on θ'.

2.2.4.3 Conjugate Priors

A conjugate prior is one that when paired with a particular likelihood function, leads to a posterior with the same form of probability distribution as the prior [80; 104]. Cases where a conjugate prior is appropriate make the job of determining the posterior (and evidence) easier, since the the form of the posterior is known and can often be solved for analytically [38]. Similarly, the integration required to calculate the model evidence is typically tractable in cases where a conjugate prior is used. An example of this is using a Gaussian likelihood and Gaussian prior, the product of which given Bayes law yields a posterior that is Gaussian as well. Another example is when the likelihood can be assessed using the binomial distribution:

$$P(x, n|p) = \binom{n}{x} p^x (1-p)^{n-x} \tag{2.21}$$

where n is the number of trials, x is the number of times a specified outcome occurs within a series of n trials with two possible outcomes, and p is the probability of a single one of these outcomes. In this case, the beta distribution is a conjugate prior:

$$P(p|\alpha, \beta) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha, \beta)} \tag{2.22}$$

where α and β are the hyperparameters and $B(\alpha, \beta)$, the beta function, is a normalisation constant. The beta function is defined by the integral:

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1} dt \tag{2.23}$$

2.2.5 Graphical/Causal Models

Complex problems have, by definition, many variables. A large subset of these variables are likely to be inter-related in some way, and modelling the dependencies and causal relations between the variables can be difficult. Graphs are a useful tool to deal with this, since they offer an expressive means of visualising the relationships between variables in these problems.

A graph, in this context, is a pair $G = (V, E)$, where V is a finite set of distinct vertices and E is a set of edges between these vertices [77]. A causal model, also commonly called Directed Acyclic Graph (DAG), influence diagram or probabilistic

network model, is a type of graph where the vertices represent variables and the edges represent relations between these variables [77]. An example of a graphical model is shown in figure 2.1.

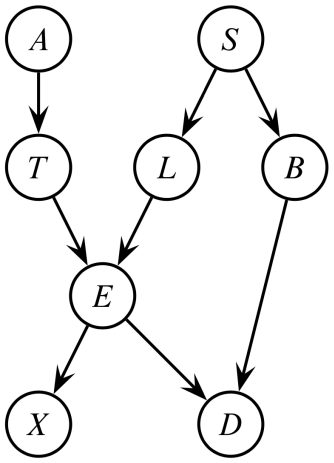

Figure 2.1: Figure taken from [77]. The vertices are represented by the encircled letters, and the arrows indicate the relationship between these vertices (the vertices represent variables). The direction of the arrows shows the causal relationship between variables/vertices.

In the example shown in figure 2.1, the left diagram (a) shows arrows between a subset of the variables. These arrows indicate a particular direction of causality between variables, which is the reason DAGs are referred to as "directed". DAGs are also acyclic: any variable that is causally influenced by another variable, cannot itself have a causal influence on that variable. This makes sense, since a particular variable should not have a causal relationship with itself.

While DAGs are not a central component of this thesis, they are used in chapter 4 to visualise the model that is implemented.

2.2.6 Hierarchical Bayesian Models

For many multivariate statistical problems, the variables are connected in some way determined by the structure of the problem [38]. Here the case where one variable is hierarchically distributed w.r.t. another is considered. That is to say that the hierarchically distributed variable is more easily expressed in relation to another

variable than on its own. In such cases, the hierarchy formed by the model should reflect the form and structure of the random variables and how they connect to observables.

A simple example of a hierarchical model is when a noisy observation of a particular quantity is made, though the true (unknown) value of this quantity is actually the quantity of interest. In such cases a latent variable for the true value can be introduced such that the observed value is hierarchically distributed around it given the noise distribution:

$$P(\theta|y_o, y_t) \propto P(y_o, y_t|\theta)P(\theta) \tag{2.24}$$

$$P(\theta|y_o, y_t) \propto P(y_o|y_t, \theta)P(y_t|\theta)P(\theta) \tag{2.25}$$

where y_o is the observed quantity, y_t is the true value (model parameter) and θ are the remaining model parameters. Note the introduction of the prior, $P(y_t|\theta)$, which incorporates the hierarchical structure of the data into this probabilistic model. Here, one can say that the observed variable, y_o, is hierarchically distributed with respect to the underlying true value, y_t.

Hierarchical models, however, need not be defined in a mathematically rigorous way or in terms of statistical random variables, though this is the context where they see most use throughout this thesis. A hierarchical model is any model that makes use of some multilevel structure. For the most simple two-level hierarchy, a hierarchical model makes use of a higher level model, and lower level sub-models, the control/fine-tuning of which is often governed by the higher-level model. Broadly speaking, there are two cases in which one may want to do this. The first is when there is some inherent known hierarchical structure to the data. Some examples of this are:

- Modelling disease prevalence and transmission characteristics among different communities [109]
- Studying the effect of child support programs at different schools [39]
- Estimating cosmological parameters using supernovae of differing type [83]

In these examples, how an individual is modelled depends on a higher level parameter (i.e. which school or community they belong to) as well as the low-level observed

variables (data on their test scores, for example). The second case where one may want to use hierarchical modelling is when the modelled data are too complex to be accurately portrayed by a single model. Ensemble models are an example of this: multiple models (which are likely suited to different data sub-spaces) are used to make predictions, and their predictions averaged over to produce an overall final result that is often better than any of the individual models. This second case is discussed further in the context of heterogeneous data in section 2.2.6.1 below.

2.2.6.1 Modelling Heterogeneous Data

An area where the use of a hierarchical modelling approach is useful is in the modelling of heterogeneous data [39]. Heterogeneity refers to the variation of structure within data. In the context of Bayesian hierarchical models, heterogeneity can more generally refer to the variation of structure within any of the random variables, even if they are not observed variables (data). This variation can manifest as clustering in random variables [39]. In some cases, these clusters are not well described by a single model, and employing multiple models to describe each cluster or a subset of clusters is a better approach. These models comprise a sort of 'ensemble'. For this method of hierarchical modelling the ensemble of models typically displays a similar hierarchical structure to that of the data/variables being modelled.

2.3 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a way to do inference. It is a numerical sampling technique used to approximate posterior probability distributions that are not analytically solvable.

Consider a mathematical system consisting of a single random statistical variable, θ. A Markov chain is such a system where the future state of the system, or next value of θ, is dependent only on the current state, and is independent of previous states. This means that all the information required to predict the future state is encoded in the current state only. MCMC utilises this property of Markov chains in order to sample from a probability distribution. The samples made by the chain would ideally be independent, but may however be correlated in some cases, and this needs to be tested for. In order to ensure the samples are independent, every alternate sample

can be dropped from the chain, or every third sample kept, or every fourth sample and so on, until it is found that the samples in the chain are statistically independent of one another. This process is called *thinning*, and is commonly applied to most problems that are tackled using MCMC. More sophisticated (optimal) approaches to implement thinning have been studied [121]. The 'Monte Carlo' part of MCMC is so named after the area in the Principality of Monaco famous for gambling, because it makes use of stochastic processes to determine the next state of the Markov chain [94].

How exactly the MCMC algorithm works to return the desired posterior probability distribution depends on the method of sampling that is used. This section provides a description of MCMC using the Metropolis-Hastings [99; 58] sampling algorithm. The algorithm works in the same way for the case with many parameters, or if θ is multidimensional, but here the single parameter case is considered for simplicity.

Pseudocode for this algorithm is summarised in Algorithm 1, though a full description of the algorithm follows.

Algorithm 1: Pseudocode for MCMC using the Metropolis-Hastings algorithm

```
input  : θ_init, n, step size
output: chain
chain=[ ];                    // 'chain' is initialised as an empty list
for i ← 1 to n do
    if i=1 then
        θ_current = θ_init;
        chain← θ_current;          // θ_current is appended to list, 'chain'
    else
        Δθ ∼ N(0, step size);              // Δθ is drawn from a normal
         distribution
        θ_proposed = θ_current + Δθ;
        r = P(θ_proposed|D)/P(θ_current|D);
        u ∼ U[0, 1];          // u is drawn from a uniform distribution
        if r>u then
            θ_current = θ_proposed;
        end
        chain← θ_current;          // θ_current is appended to list, 'chain'
    end
end
```

The MCMC algorithm is used to deliver an approximation of the posterior distribu-

tion, but can also be used to find a maximum a posteriori (MAP) estimate, the value of θ that maximises the approximate posterior. In the most general case, this is not easily done, although it should be possible to calculate the posterior probability for any given θ for MCMC to be viable.

MCMC approximates the posterior by making use of a random walk in the θ-space. An initial point in the θ-space, θ_{init}, is chosen from where to begin the random walk. The value of the posterior is then calculated for θ_{init}, which after the initial step is called $\theta_{current}$. This is found using Bayes' law and the relevant data. Then, a new point in parameter space is proposed, $\theta_{proposed}$, given a randomly drawn value from a proposal probability density function (PDF). A PDF expresses the probability distribution of a continuous random variable. It is defined over the sample space of the variable, and allows us to draw at random a value for that variable. The PDF does not provide the probability of drawing a particular value, but rather the probability that the drawn value will lie in a particular interval. The PDF is > 0 everywhere, and integrates to 1. The value drawn from the proposal PDF specifies the size of the jump to the next value of θ. The proposed point is then:

$$\theta_{proposed} = \theta_{current} + \Delta\theta \tag{2.26}$$

where $\Delta\theta \sim \mathcal{N}(0, \alpha)$ is the proposal PDF, and α is a hyperparameter for the step size. The proposal PDF needn't be Gaussian, though is should have mean zero and should take hyperparameters that tune the location and stretch of the distribution [25]. If the proposed point has a higher associated posterior probability than the current point, it is accepted, and appended to the Markov chain. If the proposed posterior probability is lower than the current one, then the Metropolis-Hastings acceptance criterion is used to determine which point is appended to the chain. If the point is accepted, again the proposed point is appended to the Markov chain. Now the proposed point is used as the current point for the next step of the algorithm. If the point is rejected, the current point instead is appended to the chain.

The Metropolis-Hastings acceptance criterion states that if the posterior probability of the proposed point is lower than that of the current point in parameter space, then the proposed point is accepted with probability [94]:

$$r = \frac{P(\theta_{\text{proposed}}|D)}{P(\theta_{\text{current}}|D)} = \frac{P(D|\theta_{\text{proposed}})P(\theta_{\text{proposed}})}{P(D|\theta_{\text{current}})P(\theta_{\text{current}})} \tag{2.27}$$

This means that if the proposed point has a posterior probability of 0.01 times that of the current point, there is a 1% chance the point will be accepted. Similarly, if the proposed point has a probability of 0.99 times that of the current point, there is a 99% chance the point will be accepted.

The choice of θ_{init} is usually made by drawing a random sample from the prior distribution of θ:

$$\theta_{init} \sim P(\theta) \tag{2.28}$$

In some cases this initial choice may be far away from the peak or the bulk of the posterior distribution, in which case the chain will not be stationary. A Markov chain is described as stationary if its distribution does not change in time/with iterations. If the chain is not stationary, it does not deliver a good approximation to the posterior. A way to mitigate this is by using 'burn-in' [104; 100]. Burn-in is when an initial number of steps in the random-walk are removed from the chain. Since it is not in general known how many steps need to be removed, one approach is to discard all points in the chain before it hits 50% MAP value for the first time. This is often required for cases where the chain is initialised in an area of very low posterior probability. One thing to note is that this would not be an issue if the algorithm could run for infinite time, since the portion of the chain after what would have been discarded through burn-in is infinitely longer than the initial portion.

The distribution of points in the Markov chain gives the approximation to the posterior for sufficiently many steps, n. The value of n cannot be determined in general, but one can check whether the chain has converged under certain metrics using methods discussed in section 2.3.3. An example using MCMC to sample a one-dimensional posterior is shown in figure 2.2, first for 100 samples and then for 10000 samples. It is not trivial to see why the Metropolis-Hastings algorithm converges, though distinct proofs are given by [125] and [140].

How quickly the algorithm converges, or how efficient the sampler is, is largely determined by the step size (or width of the proposal PDF). If the chosen step size is too small, any step proposed by the random walk will have approximately the same associated posterior probability as the current step. This will result in all proposed jumps being accepted, and the algorithm will take a long time to converge to the true posterior. Conversely, if the chosen step size it too large, any step proposed by the algorithm will yield a proposed posterior probability that is far from the bulk of the posterior distribution. In this case, most steps will be rejected, and the

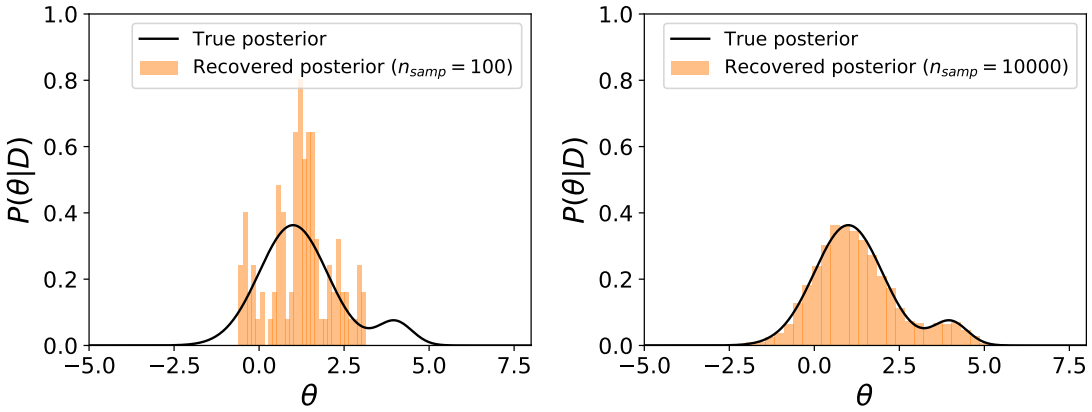

Figure 2.2: Left panel: Histogrammed MCMC chains (shown in orange) of the recovered posterior plotted alongside the true posterior (black solid line). Here the MCMC algorithm has run for 100 steps. Right panel: Histogrammed MCMC chains (shown in orange) of the recovered posterior plotted alongside the true posterior (black solid line). Here the MCMC algorithm has run for 10000 steps. The right panel illustrates the algorithm's recovered posterior distribution converging toward the true posterior after a large number of steps. The Metropolis-Hastings acceptance criterion means that the random walk can explore local minima. This also allows the random walk to explore all modes of a multi-modal distribution. This is illustrated in the right panel where convergence to the true posterior in the region of the smaller mode is just as good as at the peak of the posterior.

algorithm will also take a long time to converge to the true posterior. These two outcomes can be tested for by checking the acceptance ratio of the sampler. The acceptance ratio is the number of accepted steps over the total number of steps. Ensuring the step size is suitable for a particular application is necessary in order to ensure an efficient sampler.

Approaches to learn the correct step size as the sampler runs can increase the efficiency. Additionally, some samplers can vary their hyperparameters (step size, covariance structure) conditioned on θ i.e. depending on where the sampler is in parameter space. These are called adaptive MCMC techniques, and can be useful when sampling posteriors with highly irregular shapes. A few examples of these techniques are [78; 54; 55; 49; 103]. One should note however, that the convergence guarantees do not apply to these methods, since the proposal PDFs they implement are not reversible, a requirement given by the proofs by [125; 140].

2.3.1 Block Metropolis-Hastings

The block (also called block-wise or component-wise) Metropolis-Hastings algorithm is a modification of the standard Metropolis-Hastings algorithm that is used in cases where the parameter space is very high-dimensional. This algorithm differs from the standard one in that 'jumps' are not made in all dimensions at once, but rather only a few at a time (in blocks). This aims to mitigate the curse of dimensionality: in very high-dimensions it is much easier to make a jump to a proposed point that has a very low posterior probability. In these cases, the random walk used by the standard algorithm struggles to converge to the true posterior. The choice of block size, the number of parameters/dimensions to adjust at once, is dependent on the particular problem. Adding blocks decreases the efficiency of the sampler [141], while having the potential to improve the acceptance ratio in very high dimensions. A sampler with k blocks will have to generate k times more samples than an equivalent sampler with one, in order to yield the same effective sample size [126]. Approaches to automatically determine the ideal blocks (optimised for sample efficiency) have been investigated [141], and are shown to be effective at increasing sampler efficiency relative to naïve MCMC methods. These methods are particularly suited to problems where a subset of the variables are correlated. While they do not guarantee improvements in sampling efficiency in all problems one may tackle with MCMC, they offer significant advances for a subset of these problems that are often encountered.

A simple approach to deal with which parameters to vary, however, is randomly selecting n parameters at a time, where n is the block size. While this is not optimal in terms of sampling efficiency, it is effective on a broad range of problems.

The block Metropolis-Hastings algorithm is used to sample a 1000-dimensional posterior in Chapter 6.

2.3.2 Gibbs Sampling

Gibbs sampling is a MCMC algorithm that can be used to sample a multivariate probability distribution. The Gibbs algorithm aims to approximate the joint posterior (the posterior over all parameters), by iteratively sampling the individual conditional posteriors. It works by sampling only one parameter at a time, while the others are held fixed [37; 19; 94]. This means that the Gibbs algorithm is suited to

cases where the joint posterior is difficult to sample, but the conditional distributions are known.

For the simple case of two parameters, θ_1 and θ_2, the joint distribution, $P(\theta_1, \theta_2|D)$, can be found using Gibbs sampling given conditional distributions $P(\theta_1|\theta_2, D)$ and $P(\theta_2|\theta_1, D)$, which can be sampled from. Additionally, if only the marginal distributions are of interest (consider here the marginal distribution on θ_1, $P(\theta_1|D)$), then performing a numerical marginalisation on the joint posterior is a simple histogramming of the sample points in the θ_2 direction.

2.3.3 Convergence

There are generally two main challenges to deal with when using MCMC in any application, namely burn-in and convergence. Section 2.3 introduced the idea of burn-in, which is a way to mitigate the fact that the Markov chain may have been initialised in a low posterior probability region of parameter space. Convergence is a condition met by the Markov chain, when it is said to have a distribution that is the same as (or sufficiently similar to) that of the posterior it is sampling. Testing for convergence is of the utmost importance in terms of validating any statistical analysis using MCMC. It is however not trivial to test for, since for any real example one would not have access to the 'true' posterior against which to compare the distribution of the chain. The two challenges of burn-in and convergence are subtly related, as will be made clearer in this section.

A good way to validate any statistical analysis using MCMC is to run multiple chains from different random seeds, and compare the results. One of the ways used to test for convergence is the $\hat{R}$ statistic, initially introduced by Gelman & Rubin [40]. This statistic has since been updated [16; 38; 143], though the general idea remains the same. The idea is that after burn-in, the variance and mean estimates of the multiple chains should be the same. Additionally, the chains should be stationary. This is checked for by comparing mean and variance estimates of the entire chain against with-in chain estimates of the same quantities.

Calculating $\hat{R}$ for a given parameter requires splitting each of the chains used to sample the distribution. This ensures the first and second halves of the chains are stationary (they have the same mean). For a given parameter, θ, with m chains (after splitting) each of length n (again, after splitting), the between-chain variance

(B) and within-chain variance (W) are calculated using the equations below [38]:

$$B = \frac{n}{m-1}\sum_{j=1}^{m}(\bar{\theta}_j - \bar{\theta})^2 \tag{2.29}$$

$$W = \frac{1}{m}\sum_{j=1}^{m}\left[\frac{1}{n-1}\sum_{i=1}^{n}(\theta_{ij} - \bar{\theta}_j)^2\right] \tag{2.30}$$

Here, $\bar{\theta}$ is the average θ calculated across all chains, and $\bar{\theta}_j$ is the average θ of the j^{th} chain, where $j \in [1, m]$.

The total variance, $\hat{\text{Var}}^{+}(\theta|y)$, is then calculated by a weighted sum of the between-chain and within-chain variances. The '$^{+}$' in $\hat{\text{Var}}^{+}(\theta|y)$ indicates that this quantity overestimates the 'true' posterior variance.

$$\hat{\text{Var}}^{+}(\theta|y) = \frac{n-1}{n}W + \frac{1}{n}B \tag{2.31}$$

In the limit as $n \to \infty$, W and $\hat{\text{Var}}^{+}(\theta|y)$ both tend towards the true variance. The (square root of the) ratio of these quantities is therefore monitored for convergence, with an $\hat{R} \simeq 1$ indicating the chains have converged:

$$\hat{R} = \sqrt{\frac{\hat{\text{Var}}^{+}(\theta|y)}{W}} \tag{2.32}$$

Figure 2.3 shows an example of multiple chains sampling a posterior distribution. It shows an illustrative example of how convergence is determined using the method discussed here.

If the MCMC algorithm could be run for an infinite amount of time, then there would be no need for burn-in since the posterior mass would be infinitely more dense than the initial region of the chain. See in figure 2.3 for example, chain 1 was initialised at '-5.0' (a region of very low posterior probability), yet after a few steps this region is not explored again by any of the chains. For this example, the sampler converged to the true posterior very quickly. For more complex examples, this quick convergence is not guaranteed, and assessing the stationarity of the chains may be required to determine how many iterations should be discarded with burn-in.

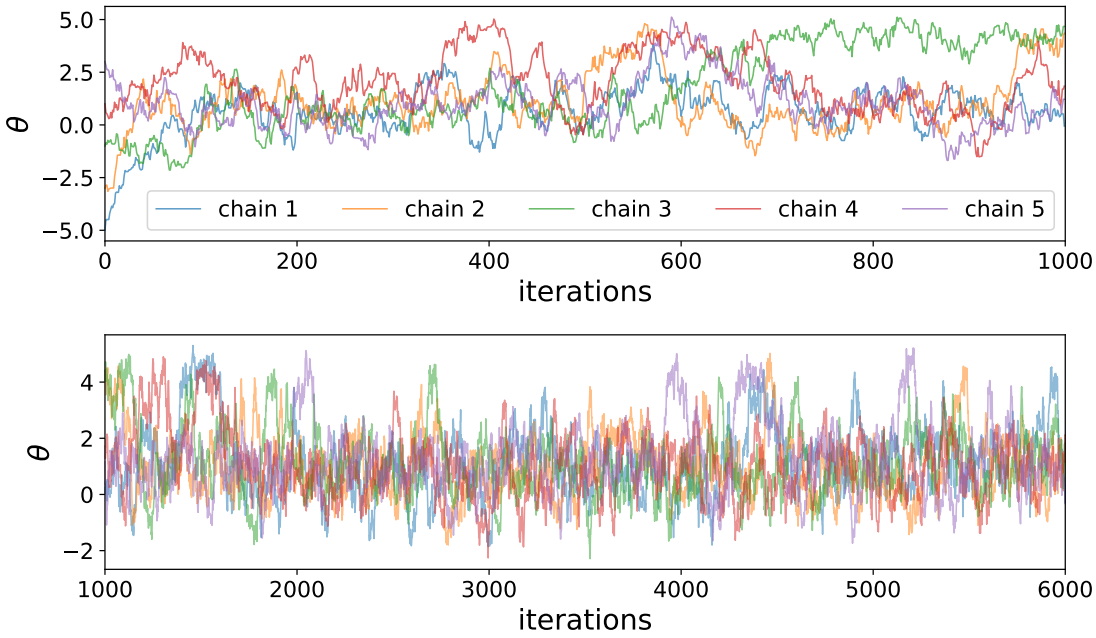

Figure 2.3: Plot showing 5 MCMC chains with different initialisations on the same posterior used in the illustrative plot in figure 2.2. Top pane: the first 1000 iterations of the chains. Bottom pane: the next 5000 iterations of the same chains. The first $\sim$ 100 iterations in this case show the burn-in phase of the chains. This is best illustrated by chain 1 (shown in light blue), which has an associated θ_{init} that is furthest from the bulk of the posterior mass. The chains in the top pane have an associated $\hat{R} = 1.24$ (not yet converged), and the chains in the bottom pane have an associated $\hat{R} = 1.01$, which indicates the 5 chains have converged over 1000-6000 iterations. For this example, convergence was reached after approximately 6000 iterations. The distribution of these chains in the θ-direction can therefore be said to have converged to the posterior of θ, $P(\theta|D)$.

2.3.4 Covariant Proposal Functions

Section 2.3 briefly introduced the idea of adaptive MCMC techniques. The idea behind these techniques is that the sampler can vary its hyper-parameters conditioned on the posterior as it samples. These techniques help with regard to sampling efficiency and convergence. This section deals with one such example: covariant proposal probability density functions (PDFs), see for example [50; 51; 21].

A covariant proposal PDF is defined in more than one dimension, where its basis vectors are not independent of one another. That is to say, when a jump is proposed in a particular axis, it has an impact on the jumps made in the other axis/axes.

Covariant proposal PDFs need not be adaptive, though they often are for practical reasons: for any real application, the covariance matrix of the sampler given posterior samples needs to be learned. Alternatively, an initial MCMC algorithm can be run, just to find the covariance matrix of the chains that result from sampling the posterior. Here the focus is on the mechanism behind covariant proposal PDFs and how and why they work, rather than how they are determined in practice.

Consider the proposal function shown in equation 2.26 (stated here again):

$$\theta_{proposed} = \theta_{current} + \Delta\theta$$

For the multi-dimensional case, this can be generalised as follows:

$$\mathbf{\Theta}_{proposed} = \mathbf{\Theta}_{current} + \Delta\mathbf{\Theta} \tag{2.33}$$

where $\mathbf{\Theta} \equiv (\theta_1, \theta_2, ..., \theta_n)^T$ for n dimensions. Here, $\Delta\mathbf{\Theta} \equiv (\Delta\theta_1, \Delta\theta_2, ..., \Delta\theta_n)^T$ is a sample drawn from the proposal PDF.

A typical choice for the proposal PDFs in each i^{th} dimension is $\mathcal{N}(0, \alpha_i)$. This choice may fail or be inefficient if the posterior, $P(\mathbf{\Theta}|D)$, has a strong covariance structure. That is to say that at least 2 of the dimensions θ_i are strongly positively or negatively correlated with one another. In this case proposed jumps in the correlated dimensions are more likely to land in an area of parameter space with low associated posterior probability if a conditionally independent proposal PDF is used. This will result in the sampler being less efficient, and in some cases it will fail to converge. This is demonstrated by the illustrative plot shown in figure 2.4.

This can be solved by multiplying the 'jumps' proposed by a univariate Gaussian proposal PDF (like the proposal PDF for the independent case discussed above) by a matrix, $\mathbf{C}$. The matrix, $\mathbf{C}$, can be found by estimating the covariance matrix of the chain, and then using the Cholesky decomposition of the estimated covariance matrix. The Cholesky decomposition is given by:

$$\mathbf{C}\mathbf{C}^T = A \tag{2.34}$$

where A is a positive definite matrix, and the estimated covariance matrix in this case, and $\mathbf{C}$ is a lower triangular matrix.

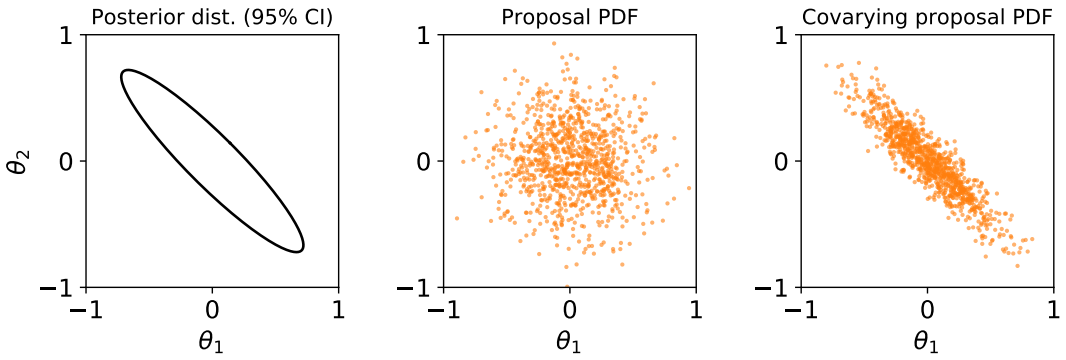

Figure 2.4: Left pane: the ellipse represents a 95% credible interval (CI) for a posterior distribution, in this case a 2-dimensional correlated Gaussian. Centre pane: 1000 samples drawn from a 2-dimensional Gaussian proposal PDF. Right pane: the same proposal PDF as that shown in the center pane, though linearly transformed by covariance matrix, $\mathbf{C}$. Here the posterior and both proposal PDFs are centred at $(0, 0)$. If the proposal PDF from the centre pane was used to propose a point from the centre of the posterior, many of the proposed points would lie far outside the 95% CI. Conversely, using the covariant proposal PDF shown in the right pane for the same jump would result in approximately 95% of proposed points lying inside the 95% CI. This serves as motivation that using the proposal PDF shown in the right pane would be a more suitable choice in this case.

Chapter 3

Machine Learning

This chapter presents an introduction to machine learning, with a specific focus on supervised machine learning. Since an aim of this thesis is to produce a rigorous algorithm for classification and anomaly detection, we need to compare the standard tools that are used to do this. The chapter begins by providing a broad overview of the field of machine learning. It also focuses on some key areas of machine learning, which are particularly aligned with the scope of this thesis. It then provides detailed information on some specific algorithms that are used in the remainder of this thesis. Section 3.5 shows detailed explanations of a number of existing algorithms, which are used as benchmark algorithms in chapters 4 and 5. Section 3.7 presents a selection of applicable metrics for use in machine learning, and introduces a novel metric, the Rank-Weighted Score (RWS), designed for use in anomaly detection tasks.

3.1 Introduction to Machine Learning

Machine learning is an area of study that bridges the fields of statistical inference and computer science. Classical programming involves explicitly coding a machine to produce a desired output, given inputs. In machine learning however, the machine is programmed to learn, given some external information that can not easily be included in an explicit manner. This information is data. Machine learning algorithms learn to automate a decision making process based on data, much like humans learn to do things from experience. The ability of an algorithm to take over tasks that would typically be thought of as human (or even super-human), is one

of the most valuable contributions by the field of machine learning. Some examples of this are in marketing [28], driving autonomous vehicles [130; 36; 22] and even for government [52], with some researchers suggesting we will develop technologies capable of doing any human job within the next 100 years [44].

Research in the field of machine learning is more prolific now than it has ever been. The primary drivers for the growth of machine learning in the past few decades have been the exponential increase in computational power, as well as the growing volume of available data. While this growth is driven by applications in industry, it will benefit in the fundamental sciences as well. Modern upcoming scientific experiments (for example, the Vera C. Rubin Telescope and the Square Kilometer Array) will collect orders of magnitude more data than any experiments that pre-date them. The tools afforded by machine learning are essential in maximally utilising this volume of data.

Machine learning algorithms are often categorised as either supervised, or unsupervised [45]. Further explanation on these two sub-fields is given in sections 3.2 and 3.3 respectively.

3.2 Supervised Learning

Supervised learning is a sub-field of machine learning where the data that are fed to the algorithm are labelled [57]. Usually we call these labels in classification and not regression problems, since the values in regression problems don't take on labels, y, but rather values, y. Here, and in section 3.2.2, both types of problem are expressed in a similar way. To express this mathematically: The data have features, X, and associated labels, y. The algorithm learns a mapping from $X \to y$, f. I.e. $f : X \to y$. When some new data, X_{new}, are observed, the algorithm can predict labels for the data given by: $y_{\text{pred}} = f(X_{\text{new}})$. The process whereby the algorithm learns f is called *training*, and the process where an algorithm generates predictions given unseen unlabelled data is called *testing*. Training reduces to minimising (or maximising) some cost function in terms of the data, which is analogous to optimisation. The 'algorithm' is also sometimes referred to as the 'model' in this chapter. Further explanation on training and testing is presented in section 3.2.1.

Some examples of the applications of supervised machine learning are:

- Facial recognition
- Hand-writing recognition
- Identifying spam e-mails
- Weather forecasting
- Predicting housing prices

Supervised learning is the focal area of the remainder of this chapter, as well as Chapter 4.

3.2.1 Training, Testing and Validation

Given a dataset of features, X, and associated labels, y, a machine learning algorithm, f, can be trained to make predictions on some new data X_{new}. Here the question arises: how does one know if the results returned by the algorithm are satisfactory? In this example, X, y are the training data. The quality of the predictions made on new data, $f(X_{\text{new}})$, cannot be quantified since there is no 'known' correct answer.

The most common approach to dealing with this is to split the data, X, y, into two sets: a training dataset, $X_{\text{train}}, y_{\text{train}}$, and a test dataset, $X_{\text{test}}, y_{\text{test}}$. Now only a subset of the original data is used to train the model, and the rest is held out to evaluate the model's predictive power. If the dataset is large, this approach may work well, and a 50/50 split of training/test data would be appropriate. If the dataset is small, a 60/40, or even a 70/30 train/test split is a common choice, since it is desirable to have as much data as possible to use to train the algorithm. One of the main reasons to do this train/test split is to avoid *overfitting*.

Overfitting is when an algorithm has a significantly superior predictive performance on the data on which it was trained than on some new unseen data, assuming the training data are representative of this new data. It typically occurs when either the model is made too complex in order to 'explain' the data, or if insufficient data is used during training, which leads to the model not being able to generalise well.

Another approach to model training and evaluation is to use validation, or cross-validation. These approaches address some challenges that arise when using a single

test set, particularly on small datasets [82]. Validation is the process where an additional split is made in the train/test split, such that there are three sub-datasets (training, validation, test), and the validation set is used to tune hyperparameters on an already trained model.

Cross-validation (more specifically k-fold cross-validation) is the process of splitting the entire dataset into k disjoint partitions, where k is an integer usually in the range [3,10]. Choosing a higher value of k helps guard against overfitting, however, this comes at the cost of computational resources since k algorithms need to be trained. An additional point worth noting for cases where the volume of training data is small, is that choosing a high value for k can penalise the accuracy of the evaluation step because set held out for evaluation is small. Once the value for k has been selected, k model training and evaluation steps are performed iteratively. For each step, the k-th set is held out for evaluation, and the model is trained on the remaining $k-1$ sets. The result is k trained algorithms. The model parameters from each algorithm can then be averaged to achieve a final predictive model. Alternately, the k predictive models can be used as an ensemble of predictive algorithms that 'vote' if averaging of model parameters is not a suitable technique for a particular algorithm.

3.2.2 Classification and Regression

Classification and regression are two common types of problems in supervised machine learning. Both classification and regression algorithms learn a mapping from the X-space to the y-space. I.e. both types of algorithm find a mapping, $f : X \to y$, that minimises some error/loss.

For classification problems, the values of y are always discrete, as well as finite. This means that a classification algorithm, f, outputs values that are elements of a finite set, determined by the number of classes present in the training data.

For regression problems, the values of y are continuous. Thus, a regression algorithm, f, outputs any value in the continuous y-space, even if that exact value has not been encountered in training.

There is a subtle difference between classification and regression problems: every regression problem can be made into a classification problem by binning the continuous values, but the reverse is not true. Class labels can be categorical (e.g. cat or

dog), which means they have no ordering. With continuous numerical values, one can say $1.1 < 1.3$, but obviously not cat<dog or cat>dog.

3.3 Unsupervised Learning

In unsupervised learning, the data are not labelled. This means an unsupervised learning algorithm takes only the data as input, and has to find some underlying pattern to that data [45; 3]. Some examples of the application of unsupervised machine learning are [45]:

- Clustering
- Segmentation
- Dimensionality reduction
- Feature selection

Unsupervised learning algorithms have the potential to be incredibly powerful, since most available data are unlabelled. The trade-off is that unsupervised learning methods can be computationally intensive on large datasets. The primary reason for unsupervised methods typically requiring more computational resources than their supervised counterparts is that there are no human-annotated labels present. The lack of labelled data means that no context/structure for the data is provided, and this structure must instead be learned as well.

See for example figure 3.1 from [138], where the task is to cluster similar news stories. In order to read the figure properly, two concepts need to be introduced, namely the concepts of Markov stability and Sankey diagrams. A Sankey diagram represents the flow of information within a system using arrows. The width of the arrows corresponds to the volume of information that is transferred through the portion of the system that the arrow represents. In figure 3.1, the number of arrows shown at each step in the clustering process indicates the number of clusters, and the width of each arrow represents the volume of information in that cluster. Markov stability is a measure of the quality of clusters over time (or through iterations on the clustering process) [30]. The clustering process splits the data into distinct partitions and the Markov stability (analogous to the variation of information, $VI(t)$, shown in

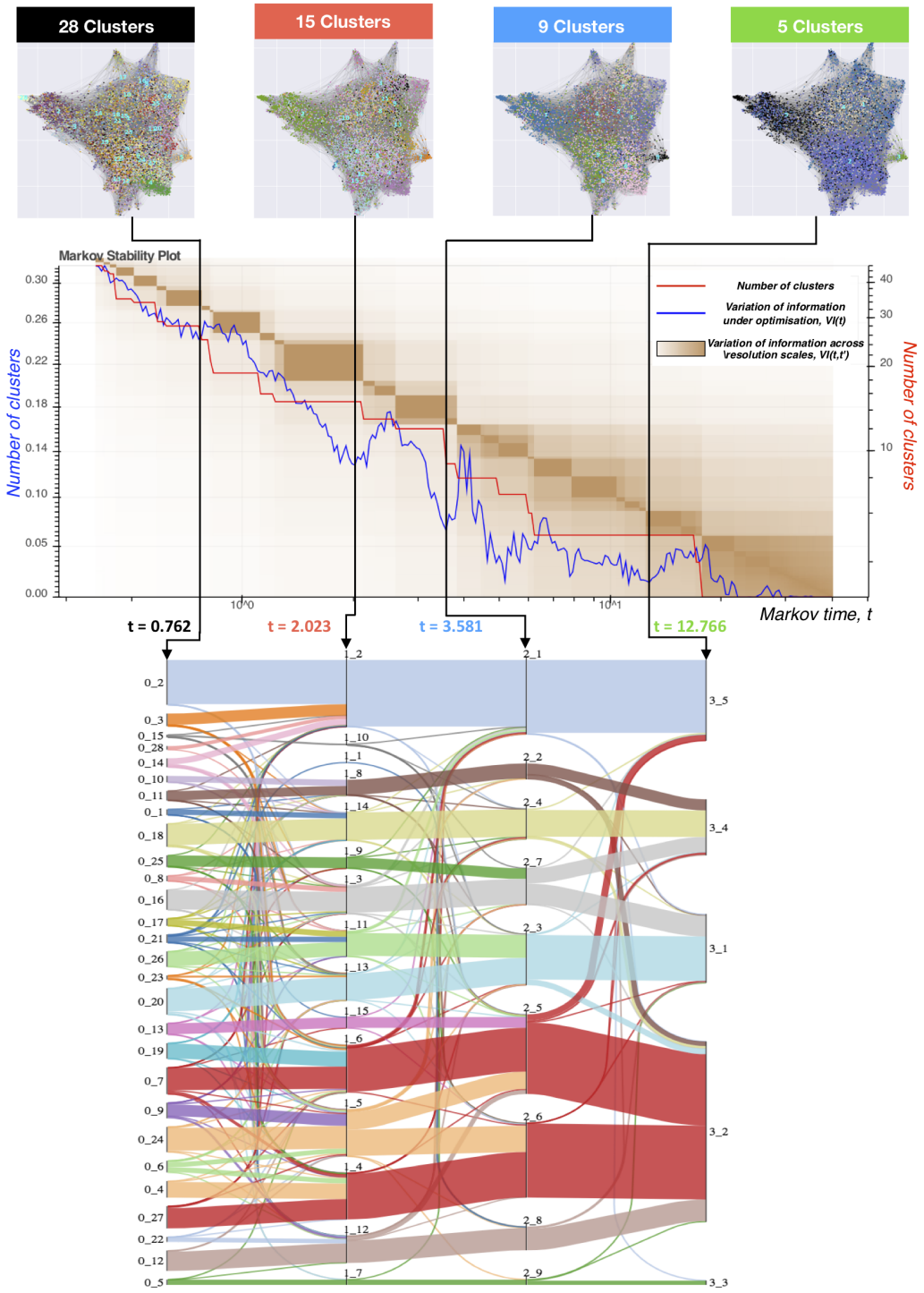

Figure 3.1: Figure taken from [138]. The top four plots illustrate the data in a two-dimensional space, where different numbers of clusters are assumed (28, 15, 9 and 5 respectively). The middle plot illustrates the stability of the clustering, using a Markov stability plot (decreasing variance of information or increased stability from left to right). The bottom plot is a Sankey diagram, showing the connection/flow between clusters of different size.

figure 3.1) provides a measure of probability that a particular item is likely to stay in the partition in which it started. These distinct partitions can be thought of as states of a discrete-time Markov chain. The transition matrix that governs this discrete-time Markov chain is what is used to compute the Markov stability for a given clustering. Clusters that are "stable" under the variation of information metric comprise of individual stories which are unlikely to transition to a different cluster throughout iterations of the clustering process.

For the example in figure 3.1, the Markov stability plot and Sankey diagram illustrate how similar stories are clustered together in sub-topics, which are then clustered into broader topics in a bottom-up manner. A common issue with many clustering problems is that it is not clear how many clusters 'should' be present in the data, so having an idea of the quality of the clusters (given by the Markov stability in this case) is important.

The large volume of data as well as the complexity of the data makes the clustering process difficult, since in order to create clusters in a lower dimensional space than the raw data, distances between the data instances need to be computed, i.e. if one wants to say that one story is similar to another. Even though the words/stories have been converted into a vector space, this space is high-dimensional, which when coupled with the volume of data makes clustering computationally intensive.

3.4 Anomaly Detection

Anomaly detection is an important area of research in the field of machine learning that has gained huge traction owing to its large range of applications [110]. A few examples of these applications include:

- security [135]
- fraud detection
- defect finding [96]
- data cleaning
- serendipitous knowledge discovery [27]

These are all problems where there is some standard behaviour that can be modelled (or learned) given that there is a large amount of representative data, and there is also a chance of aberrant behaviour with a small probability. This aberrant behaviour is typically modelled as a deviation from the standard by some threshold [128; 3].

Anomaly detection can be thought of as a special case of classification, where data from the anomaly class is only present in the test data. Thus, the algorithm has not seen any objects from the anomaly class during training. This is however not always the case, since there may be contamination from anomalies in the training data. In this case, unsupervised anomaly detection methods are ideal, since they focus only on the features of the data, and do not require training.

In this case of unsupervised anomaly detection, the algorithm would explore the data in order to find examples/instances of data that do not conform to the rest of the data. Distinctions between anomaly/outlier detection and novelty detection are sometimes made in the literature. The difference between these is that in novelty detection, instances encountered in testing that are similar anomalous instances encountered in training will not be flagged as 'novel', since the algorithm has already seen them before. However, throughout this thesis, the terms 'anomaly' and 'outlier' are used interchangeably to describe any data instance that is found by an algorithm to deviate from a common rule by some threshold.

3.5 Existing Algorithms

This section presents detailed descriptions on a number of classification and anomaly detection algorithms that are used as benchmark algorithms throughout this thesis. These algorithms are Random Forest in section 3.5.1, Local Outlier Factor in section 3.5.2 and Isolation Forest in section 3.5.3. This is by no means an exhaustive list, but each algorithm has been chosen given its suitability to the required tasks. This section also presents justifications for the choice of algorithms discussed here.

3.5.1 Random Forest

Random forest (also sometimes referred to as random forests or random decision forests) is a decision tree-based method for classification or regression tasks [62; 13; 57]. An in-depth investigation into the workings of ensemble decision trees and random forests is presented by [90].

A decision tree can be thought of as a set of sequential decision nodes - a means of splitting information that enters the node through a single input into an arbitrary number of outputs given a condition. In this case, the decision nodes are designed in such a way as to split the data based on training labels. An example of a decision tree is shown in figure 3.2.

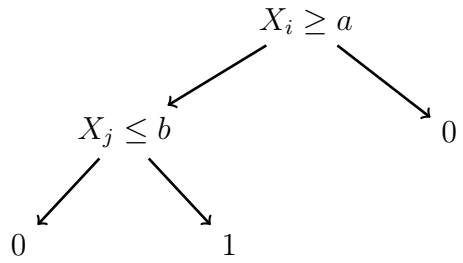

Figure 3.2: Decision tree for generic features X and associated class labels y, where $y \in \mathbb{Z} \in [0, 1]$ in this case. I.e. this is an example decision tree for a binary (two-class) classification problem. Here the branch down to the left corresponds to a *true* condition, and the branch down to the right corresponds to a *false* condition. In the case of classification, the terminal points on the decision tree are always the class labels, $y_i \in y$.

The random forest algorithm is an ensemble of many decision trees. Each decision tree makes a prediction about the value (for regression) or class (for classification), and the mean or most popular prediction is returned given the predictions by the ensemble (mean in the case of regression, mode in the case of classification). The random forest algorithm can therefore also be described as a 'voting' of randomly generated decision trees. For an ensemble with K trees, K random vectors Θ_k for $k \in [1, K]$ are generated that are independent and identically distributed (i.i.d) [13]. Each tree in this ensemble we then write as $h(\mathbf{x}, \Theta_k)$, where $\mathbf{x}$ is the input. Here, $\mathbf{x} \equiv x_1, x_2, ..., x_m$ is a particular instance of data with m features, which, along with class label, y, is drawn from a distribution (that can be thought of as the full dataset), $\mathbf{X}, \mathbf{Y}$ such that $\mathbf{x}_i, y_i \sim \mathbf{X}, \mathbf{Y}$. The random subset of $\mathbf{X}, \mathbf{Y}$ that is used for training given Θ_k for $k \in [1, K]$, is denoted by $\mathbf{X}_\Theta, \mathbf{Y}_\Theta$.

The ensemble of trees is written as:

$$\mathbf{h} \equiv \{h(\mathbf{x}, \Theta_1), h(\mathbf{x}, \Theta_2), ..., h(\mathbf{x}, \Theta_K)\} \tag{3.1}$$

The parameters Θ_k (which have been randomly drawn) determine the structure of the k-th tree, and which features $x_i \in \mathbf{x}$ are used. An empirical margin function is defined:

$$m(\mathbf{X}, Y) = \hat{P}(h_k(\mathbf{X}) = Y) - \max(\hat{P}(h_k(\mathbf{X}) = j)) \tag{3.2}$$

where $j \neq y$, $\hat{P}(h_k(\mathbf{X}) = Y)$ is the proportion of classifiers that correctly classify the subset of features, $\mathbf{X}_\Theta$ as Y_Θ, $\max(\hat{P}(h_k(\mathbf{x}) = j)$ is the proportion of trees that classify features, $\mathbf{x}$ as j, where j is the most popular label predicted by the ensemble, unless y is the most popular prediction. In this case, j is the second most popular label predicted. The empirical margin function, m, is the extent to which the number of votes for the correct class outnumber the votes for the next best class.

Here there is still the question of at which values should the splits on the randomly selected features occur? Approaches to deal with this vary with different implementations, but the most common approach is to split the data among every value $x_i \in X_\Theta$, and pick the solution that yields the largest information gain, or Gini criterion [12]. The Gini criterion describes the level of variation within the predicted classes corresponding to the set X_Θ, yielding 0 for no variation (all the same class) or 1 for the maximum amount of variation possible (all different classes). The approach of using the Gini criterion is encapsulated in the empirical margin function in equation 3.2. One thing to note is that this approach is intractable for very large X_Θ.

How does random forest deal with overfitting, particularly in the case where a large ensemble is used? The use of a large number of trees does not overfit the data. Take for example the generalisation error on the ensemble as $K \to \infty$:

$$\lim_{K \to \infty} e = P_{\mathbf{x},y}\left[P_\Theta(h(\mathbf{x}, \Theta) = y) - \max(P_\Theta(h(\mathbf{x}, \Theta) = j)) < 0\right] \tag{3.3}$$

where $j \neq y$, and $\mathbf{x}, y$ is treated as a random variable. The subscripts $\mathbf{x}, y$ and Θ denote probability in the $\mathbf{x}, y$-space and Θ-space respectively. This surmises that for random variables, $\mathbf{x}, y$, the generalisation error of the ensemble is less than 0 as $K \to \infty$. Here this result is simply stated, though a proof is given by [13].

The randomness in the algorithm comes from the fact that each decision tree is

generated on a random subset of features, as determined for the k-th tree by Θ_k. As such, a random forest classifier can be sensitive to the initial random seed, since there is no guarantee which subset of features will be selected. However, as the number of trees is increased, this variance drops given that the generalisation error of the ensemble is guaranteed to drop, which yields more stable predictions.

Random forest is used as a benchmark algorithm for classification of sequential data in Chapter 4. The specific implementation used is that from scikit-learn [112] *(version 0.20)*.

3.5.2 Local Outlier Factor

Local outlier factor (LOF) is an anomaly detection algorithm, that outputs a 'degree of anomalousness' (or outlier factor) relative to the local neighbourhood [14]. LOF differs from anomaly detection methods that pre-date it, in that no thresholds are used, and instead only the degree to which a data instance is clustered relative to its local neighbours is considered. Comparisons to the data are made for the k-nearest objects only. While the value of k should be at least 10 in order to avoid statistical fluctuations, using a small number of neighbours against which to measure anomalousness allows for reduced computational complexity, without impacting significantly on algorithm performance.

The LOF is calculated given k-nearest objects, to the object of interest, p. The distance between the object of interest and the k-th nearest object is called the k-distance. The k-nearest objects to object p make up a neighbourhood $N_k(p)$. The LOF for object p is then:

$$\mathrm{LOF}_k(p) = \frac{\sum_{o \in N_k(p)} \frac{\mathrm{lrd}_k(o)}{\mathrm{lrd}_k(p)}}{|N_k(p)|} \tag{3.4}$$

for objects o in $N_k(p)$, where $|N_k(p)|$ is a the number of objects that constitute a local neighbourhood of object p, found by how many objects are closer to p than the k-distance (the k-distance is defined below). The quantities k and $|N_k(p)|$ need not be the same if there are objects, $o \in N_k(p)$, which are not unique. Here, lrd_k is

the *local reachability density*, given by:

$$\mathrm{lrd}_k(p) = \frac{|N_k(p)|}{\sum\limits_{o \in N_k(p)} reach{-}dist(p, o)} \tag{3.5}$$

and $reach{-}dist$ is the reachability distance of object p w.r.t. object o, given by:

$$reach{-}dist = \max(k{-}\mathrm{distance}(o), d(p, o)) \tag{3.6}$$

where $d(p, o)$ is just the distance between objects p and o. The k-distance of an object p is defined for positive integers, k, as the Euclidean distance between points p and o, $d(p, o)$, where o is selected such that the distances from p to at least k other objects are less than or equal to $d(p, o)$, and the distances from p to at most $k - 1$ objects are less than $d(p, o)$.

LOF is an unsupervised algorithm as standard, though it can be used as both a unsupervised and semi-supervised learning method, depending on whether there is suitable training data available for the problem being considered. In implementations of the semi-supervised algorithm [112] there is no change in the computation of the LOF, rather, a new threshold is learned for what LOFs are considered anomalous. The standard (unsupervised) algorithm assumes most inlier instances have an LOF of around 1 and outliers have an LOF that is much larger [14]. The semi-supervised implementation learns, based on computation of the LOF for the training data, which LOF values correspond to inliers/outliers for a given problem. It is important to note that this semi-supervised approach fails when there is contamination from anomalies in the training data, since the algorithm learns to flag these (which have high LOFs) as inliers. Anomalies with similar LOFs to those seen in training will therefore not be picked up in testing. In these cases the unsupervised algorithm should be used.

LOF is used as a benchmark algorithm for anomaly detection in Chapter 4. The specific implementation used is the supervised implementation from scikit-learn [112] *(version 0.20)*.

3.5.3 Isolation Forest

Isolation forest is a decision tree-based anomaly detection algorithm, that isolates outlying data instances from the rest of the instances [87]. The algorithm works by iteratively creating nested decision trees on randomly selected features, which split the data about random numerical values that are selected between the minimum and maximum values for that feature. In the original Isolation Forest algorithm, these splits were only made parallel to the axes used, though the algorithm has been extended to allow for linear splits in any direction [56]. This is illustrated in figure 3.3, which shows an example of extended Isolation Forest on some 2D data.

For a given decision tree, the path length (measured by the number of nodes passed through in the decision tree) required to isolate an instance from the rest of the data is used in order to compute an anomaly score. Anomalous instances will typically require fewer splits in the decision tree (or a shorter path length) in order to become isolated from the rest of the data instances, when compared with inlier instances.

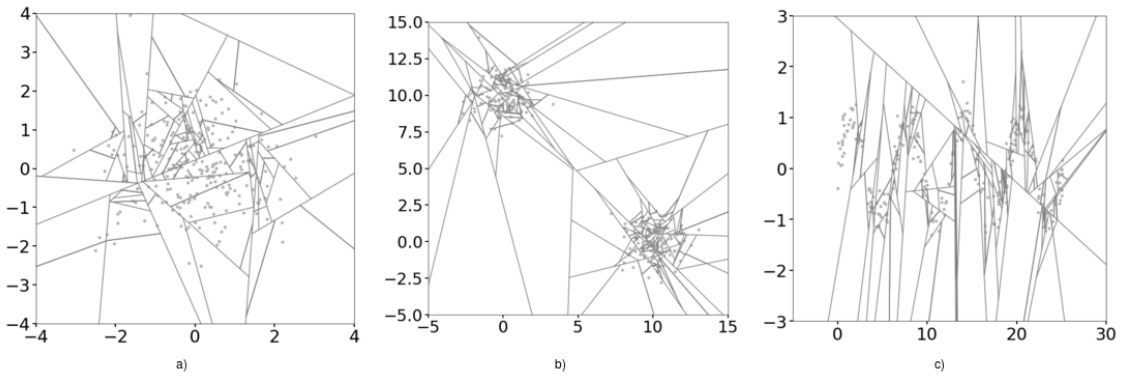

Figure 3.3: Figure taken from [56]. Illustration of extended Isolation Forest on some 2D data. The splits made at each node of the decision tree are represented with straight lines, which seek to find the shortest path to isolate the data instances (shown by the solid black points).

The anomaly score is given by:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{3.7}$$

where $h(x)$ is the path length of observation x, and $c(n)$ is the average path length followed by the inlier/normal data. Scores close to 1 correspond to anomalies and scores close to 0 correspond to inlier objects.

Isolation forest is used as a benchmark algorithm from anomaly detection in Chapter 4. In the supervised implementation of Isolation forest, there exists the possibility to provide the algorithm with labelled anomaly instances during training. However, this violates one of the assumptions present in many anomaly detection problems: that examples of anomalies are not known *a priori*. In Chapter 4, examples of anomalies are not included in training.

3.6 No Free Lunch Theorem

The no free lunch theorem was first introduced in the field of statistical inference [150]. The theorem is however also applicable more generally to optimisation, as well as to the field of machine learning [151].

The no free lunch theorem states that there is no single algorithm that works best on all possible problems. It was originally stated that "any two algorithms are equivalent when their performance is averaged across all possible problems" [151].

The no free lunch theorem is often cited as justification for the fact that some domain knowledge is needed when implementing a machine learning algorithm. This argument is also used in this thesis. In particular, it is not surprising that an algorithm developed for a specific task outperforms more generally applicable algorithms on that task. Similarly, an algorithm developed for a specific task may perform worse than a more general algorithm when evaluated by average performance over a range of tasks.

3.7 Metrics

The choice of metric is of utmost importance when evaluating machine learning algorithms [10; 84; 117], and the study of metrics for learning algorithms is a subfield of research on its own. Different metrics can produce different results in terms of ranking algorithms on a particular machine learning task. For this reason, it is necessary to know whether a particular metric is applicable to a given problem.

This section outlines a number of metrics that can be used to quantify algorithm performance for both classification and anomaly detection tasks. The Rank-Weighted

Score, introduced and described in section 3.7.4, is a novel metric for use in gauging algorithm performance on anomaly detection tasks.

Before the individual metrics are presented in detail, it is useful to introduce the terms True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). These terms apply in classification problems, and more generally in any problems that make use of an algorithm in order to produce discrete predictions, where the true/correct labels are known. In such problems, each item/datum in the dataset can be termed either a TP, FP, TN or FN once predictions over the dataset have been made. Whether the item is termed 'positive' or 'negative' depends on the output of the algorithm. The terms 'true' and 'false', rather intuitively, tell us whether the prediction was correct or not. The number of TPs, FPs, TNs and FNs in the dataset sums to the total number of items in the dataset, and these numbers are usually presented in a table called a confusion matrix. One subtlety here is that this description of the confusion matrix only applies to cases where there are two possible prediction outcomes (positive or negative). For multiclass classification, the confusion matrix is instead a $n \times n$ matrix, where n is the number of classes.

3.7.1 Accuracy

Accuracy is a metric that is used to gauge algorithm performance on classification problems. Accuracy is defined as the number of True Positives (TP) divided by the total number of objects for binary classification problems, but can be generalised for multi-class problems as the number of correctly identified objects divided by the total number of objects. Accuracy can be an inappropriate choice of metric if the data are imbalanced. The task of anomaly detection is an extreme case of this. In anomaly detection, there are typically far fewer anomalies than inliers, so an algorithm that predicts only inliers could score a high accuracy, despite predicting no anomalies (the task at hand). For this reason, accuracy is used to quote algorithm performance in classification tasks only in the following chapters.

3.7.2 Area Under the Curve

The Area Under the Curve (AUC) metric is suitable for gauging performance in any binary classification problem, though it has been generalised to the multi-class

case as well [53]. It is found by calculating the area under the Receiver Operating Characteristic (ROC) curve for the class of interest. The ROC curve is found by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) for different threshold settings. In order to do this, some sort of probability of belonging to the class of interest is required for the objects being classified. An example ROC curve is shown in figure 3.4.

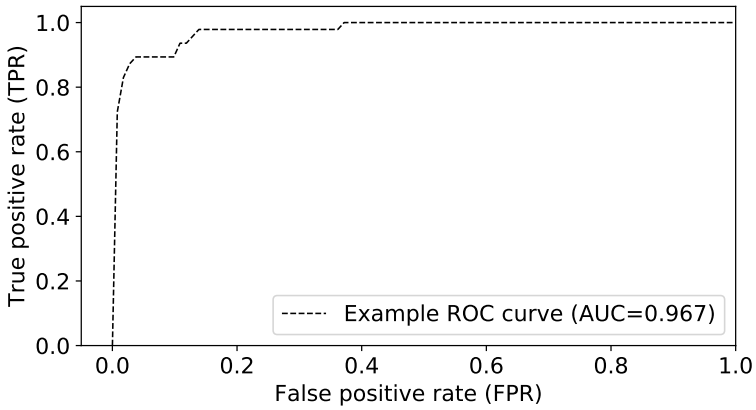

Figure 3.4: An example ROC curve. The true positive rate (TPR) is plotted against the false positive rate (FPR). The AUC is found by calculating the area under the plotted curve over the plotted range. In this case, the AUC is 0.967.

The AUC has a range of $[0, 1]$, where 0 represents all true objects belonging to the class of interest being given lower probabilities than other objects, and 1 represents all true objects belonging to the class of interest being given higher probabilities than other objects. An algorithm performs perfectly under this metric with a score of 1, and the worst possible performance is 0. Random guessing for classification would on average yield an AUC score of 0.5 for either balanced or unbalanced data. One consideration is that the AUC of the ROC curve is known to be problematic for imbalanced data (of which anomaly detection is the extreme case) [29].

3.7.3 Matthews Correlation Coefficient

The Matthews Correlation Coefficient (MCC) [98] is a metric for assessing the performance of a machine learning algorithm on a binary classification task (though it has been generalised to the multi-class case as well).

The MCC is defined as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.8}$$

where TP is the number of true positives, TN the number of true negatives, FP the number of false positives and FN the number of false negatives in a test dataset. The MCC takes on a value in the range $[-1, 1]$, where -1 is the worst possible score and 1 is the best. Random guessing would on average yield an MCC score of 0.

The MCC is useful for assessing algorithm performance on datasets with unbalanced classes. When compared with accuracy for example, an algorithm could achieve an accuracy score of 99% on an anomaly detection task if there were 1% outliers and the algorithm predicted inliers only. The same algorithm would score an MCC of 0.

3.7.4 Rank-Weighted Score

This section introduces a new metric for quantifying an algorithms' performance on an anomaly detection task, which we call the Rank-Weighted Score (RWS). In addition to being insensitive to class imbalance - an important consideration for gauging algorithm performance in anomaly detection - the RWS is sensitive to relative ranking of anomalous objects. This means that algorithms which rank anomalous objects as being more anomalous are preferred under the RWS. Compare this for example to the Spearman's rank correlation [136], which does not give higher weight to more anomalously scored objects.

The RWS is calculated given a ranked list (most anomalous to least anomalous) of the N most anomalous objects identified by an algorithm. It is defined as:

$$S_{RWS} = \frac{1}{S_0} \sum_{i=1}^{N} w_i I_i \tag{3.9}$$

where:

$$w_i = N + 1 - i \tag{3.10}$$

The weight term, w_i, gives linearly more weight to objects that the algorithm ranks as more anomalous. The term I_i is an indicator variable: it takes on a value of $I_i = 1$ if the i-th object is a true anomaly, and $I_i = 0$ if it is not. S_0 is a normalisation

term, and is calculated by: $S_0 = \frac{N}{2}(N+1)$. This means the RWS has a score in the range [0,1], where 0 implies that no true outliers were found in the N most anomalous objects ranked by the algorithm, while a score of 1 would mean that all N most anomalous objects identified by the algorithm were in fact outliers. The value of N must be chosen on a per problem basis, and kept consistent across the various algorithms being considered to allow fair comparison.

Chapter 4

Bayesian Anomaly Detection and Classification

The tasks of automated classification and anomaly detection are becoming more and more important, even in the fundamental sciences. An example of this is in large surveys in Astronomy. Experiments such as those that will be undertaken by the Vera C. Rubin Observatory (previously named the Large Synoptic Survey Telescope (LSST), though it will still produce the Legacy Survey of Space and Time) will record millions of transients every night [92; 89]. Classifying these objects accurately will enable much higher quality science and open the door to new discoveries, but this is difficult due to systematic bias, contamination and noise.

This chapter presents Bayesian Anomaly Detection and Classification (BADAC), a unified statistically robust approach to dealing with these challenges. Section 4.2 presents the formalism of the BADAC algorithm. Section 4.3 then presents a series of simulations, which illustrate the performance of BADAC against a number of benchmark algorithms. Finally, concluding remarks are shown in section 4.4.

4.1 Introduction

In any fully rigorous or scientific analysis, uncertainties must be quantified and propagated through the full analysis pipeline. This is difficult to do with traditional machine learning algorithms that do not explicitly take into account uncertainties on the data or features. As machine learning is increasingly given authority for

making more important and high-risk decisions, (e.g. in self-driving cars), and with the potential for adversarial attacks [1], there is an increasing need for interpretable models and rigorous statistical uncertainties on machine learning predictions.

In classification problems, class labels are typically inferred through the use of a separation boundary that is learned from training data [35] and is based on a score combined with a threshold. This threshold is often arbitrary or learned as a hyper-parameter to minimise some chosen loss function [107]. Any resulting class "probabilities" are systematically distorted in ways unique to the classification algorithm used and are not true probabilities, though in some cases these can be calibrated in a frequentist sense with more training data using isotonic regression or Platt scaling for example [107]. Isotonic regression and Platt scaling are both a means of transforming the probabilities output by an algorithm, such that the algorithm's calibration curve produces a desired response, a topic that is covered in some detail in section 4.3. Isotonic regression achieves this by fitting a monotonic increasing function to the output probabilities [4]. Platt scaling fits a logistic regression model the classifiers output probabilities [116].

However, particularly in the physical sciences, we desire an algorithm that automatically outputs unbiased, accurate probabilities, since knowing the probabilities of an object belonging to various classes is typically more useful than the class label alone. The classification process is often just one step in a multi-stage pipeline, and it is important to propagate class uncertainties through the additional steps. This need is especially true in cases where the true class labels of the training data are noisy or subjective, or the training data are not representative of the test set. An example in astronomy is provided by the photometric classification of type Ia supernovae which are subsequently used for studies of dark energy. Hard label classification leads to contamination from non-Ia supernovae that leads to biases in dark energy properties while fully propagating class probabilities instead allows for unbiased results at the end of the pipeline [83; 61].

In this context Bayesian methods are ideal [31], as they have been proven optimal for classification for certain loss metrics, e.g. [33], and allow the option of both supervised or unsupervised classification [24]. In the context of astronomy, Bayesian techniques have been applied to the classification of transient objects such as supernovae [26]. A common limitation in the classification of noisy data however, is that the classes in the training data are typically represented by a single template with

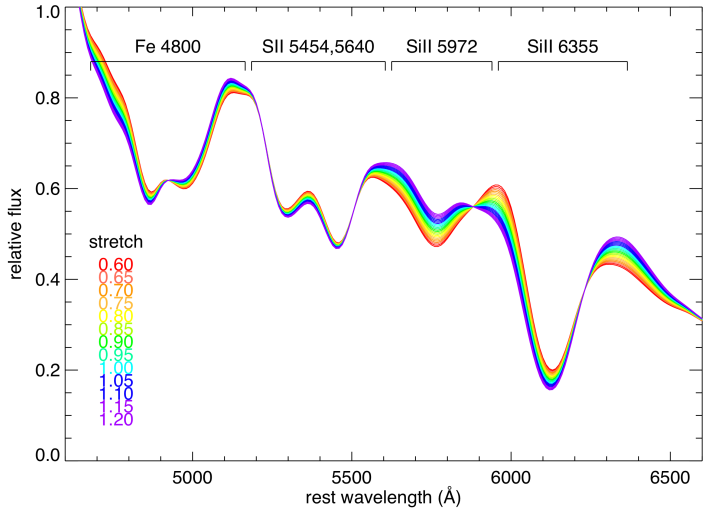

Figure 4.1: Figure taken from [64]. Spectral template compiled using a series of supernova observations (28 type Ia supernovae). Here, the variability shown by changing colour accounts for stretch: stretching the lightcurve in the time direction causes changes within the resulting spectrum. Crucially, the templates have no errors on them, and are assumed to be noise free.

zero variability (e.g. [129]). An example of this is illustrated in figure 4.1 from [64], where a single template (albeit with some variability to account for stretch) is used to fit supernovae observations. This allows straightforward Bayesian methods to be applied but does not apply if there is significant intraclass variability. Ignoring this intraclass variability also makes principled anomaly detection challenging: how unlikely is an example if one doesn't know the underlying distribution within a class? Examples of recent work in this area include [147], [65] and [152].

The following sections address these limitations, constructing a statistically robust supervised Bayesian method that can simultaneously be used for both anomaly detection and classification in the presence of measurement uncertainties on all data. This method works directly with raw data, requiring no feature extraction, and requires minimal assumptions about the nature of the anomalies or classes.

4.2 Formalism

Here, the problems of supervised classification and anomaly detection in the presence of Gaussian measurement uncertainties are considered. For this case, it is assumed

that the data are a set of one-dimensional time series[1] with features, y_o, associated measurement uncertainties, σ_{y_o}, and class labels, τ_{y_o}. Given some newly observed data without an associated label, it is possible to formulate a model for determining whether the new observed data 'belongs' to one of the known classes (classification), or none of the known classes (anomaly detection).

Since there are no perfect measurements of the training and test data, it is useful to introduce a model parameter, y_t, which is the true (unknown) value of the data that will be marginalised over in order to perform classification and anomaly detection. A graphical representation of this causal model is shown in figure 4.2. (Refer to section 2.2.5 for more on causal/influence diagrams).

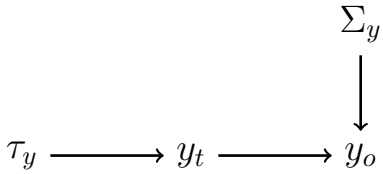

Figure 4.2: Directed acyclic graph illustrating the causal relationship between variables in the standard BADAC hierarchical model. Here, Σ_y is the noise on the underlying signal, y_t, giving rise to the observed variable y_o.

In figure 4.2, τ_y and y_o are the observed variables (data), though we are actually interested in the model parameter y_t. For the tasks of classification and anomaly detection, the test data, d_o, interacts with this model through a probabilistic link that is shown in figure 4.3. An important note is that this link is not a causal link.

Here the training data are always denoted by y. The test data are always denoted by d ('d' for data), just because it is useful to differentiate between training and test data in this formalism. A subscript o and subscript t are used to differentiate "observed" (data) and "true" (model parameters) variables.

The measurement uncertainty (also sometimes referred to as noise), is given for the training and test data by:

$$y_o = y_t + \Sigma_y$$

$$d_o = d_t + \Sigma_d$$

As can be seen from the causal model shown in figure 4.3, there is no direct causal

[1] One should note though that nowhere is the "time" nature important. The data can equally be thought of as a one-dimensional spectrum, or any generic spatial data.

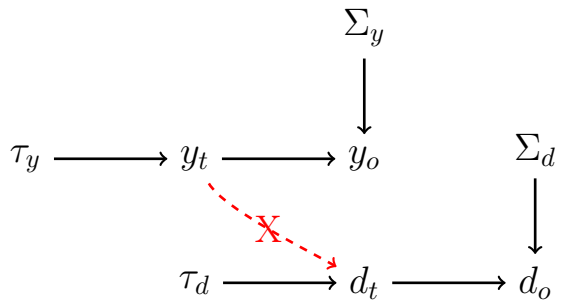

Figure 4.3: Directed acyclic graph illustrating the causal relationship between variables in the standard BADAC hierarchical model, as well as a probabilistic link between training and testing data, which allows us to generate a classification scheme. Here the probabilistic link is shown by 'X' with a red arrow. Noise on the training data is shown as Σ_y, and the noise on the test data is shown as Σ_d.

link between training and test data. However, it is well established that information from a large dataset can be used to make inferences about some new data, given a few assumptions about the causal mechanisms behind the generation and observation of the data. Logical inferences can work equally well in either direction, even if causal influences only propagate forward in time [67]. In this case, the probabilistic link 'X' between each instance of training data and the test data is used to infer the class label. This differs from an approach where test instances are compared to a class mean/average. This is important since the averaged instances for a particular class may look nothing like the actual individual instances that make up the class.

The posterior distribution for τ_d, the class label for the test data d, is shown in equation 4.1. The expression for the posterior distribution is set up using Bayes' law, given observed variables y_o and d_o:

$$P(\tau_d|d_o, y_o) = \frac{P(d_o, y_o|\tau_d)P(\tau_d)}{P(d_o, y_o)} \tag{4.1}$$

In general, the likelihood term $P(d, y_o|\tau)$ in equation 4.1 cannot be evaluated since both d and y_o have associated measurement uncertainty. However, if we assume our data have a known uncertainty distribution, then we can marginalise over the uncertainty on the training data. In some special cases, such as the Gaussian case we discussed below, this can be done analytically (as shown in equation 4.9). Using the latent variables, y_t, the likelihood in equation 4.1 can be written as the marginalisation over these latent variables, assuming d and y_o are statistically independent

of one another:

$$P(\tau_d|d_o, y_o) \propto P(\tau_d) \int \mathrm{d}y_t P(d_o, y_o, y_t|\tau_d) \tag{4.2}$$

$$P(\tau_d|d_o, y_o) \propto P(\tau_d) \int \mathrm{d}y_t P(d_o, y_o|y_t, \tau_d) P(y_t|\tau_d) \tag{4.3}$$

$$P(\tau_d|d_o, y_o) \propto P(\tau_d) \int \mathrm{d}y_t P(d_o|y_o, y_t, \tau_d) P(y_o|y_t, \tau_d) P(y_t|\tau_d) \tag{4.4}$$

$$P(\tau_d|d_o, y_o) \propto P(\tau_d) \int \mathrm{d}y_t P(d_o|y_t, \tau_d) P(y_o|y_t, \tau_d) P(y_t|\tau_d) \tag{4.5}$$

Here, equation 4.2 is obtained by introducing the model parameter, y_t, which is then marginalised over in the integral. Marginalisation is discussed in some detail in section 2.2.3. Equations 4.3 and 4.4 are then obtained using the product rule for probabilities. Equation 4.5 is then found since the term $P(d_o|y_o, y_t, \tau_d)$ should have no dependence on y_o. This independence relation is illustrated in figure 4.3.

Once again, thus far this has been for a single one-dimensional training instance in each class. The approach when considering the general case (multiple multidimensional training instances) is however the same. For the general case, the features for the training and test data are now $\{d\}$ and $\{y_o\}$ respectively. The likelihood for a new test instance $\{d\}$ belonging to class τ - assuming the instances in the training data are uncorrelated - is then given by:

$$P(\{d\}, \{y_o\}_\tau|\tau) = \int d\{y_t\}_\tau P(\{d\}|\{y_t\}_\tau, \tau) P(\{y_o\}_\tau|\{y_t\}_\tau, \tau) P(\{y_t\}_\tau|\tau) \tag{4.6}$$

Assuming n m-dimensional training instances, indexed by i and j respectively, the multidimensional probability distributions in equation 4.6 can be expressed as follows:

$$P(\{d\}, \{y_o\}_\tau|\tau) = \int d\{y_t\}_\tau \left[\frac{1}{n}\sum_{i=1}^{n} P(\{d\}|\{y_t\}_\tau, \tau)\right] \times \prod_{j=1}^{m} P(\{y_o\}_\tau|\{y_t\}_\tau, \tau) \prod_{j=1}^{m} P(\{y_t\}_\tau|\tau) \tag{4.7}$$

For notational simplicity, the i and j indices have been dropped from the variable names in equation 4.7. For completeness sake, the variables $\{d\}$, $\{y_o\}_\tau$ and $\{y_t\}_\tau$ should read $\{d^j\}$, $\{y_o^{i,j}\}_\tau$ and $\{y_t^{i,j}\}_\tau$ respectively. Here $P(\{d\}|\{y_t\}_\tau, \tau)$ is the likelihood of observing the data $\{d\}$, conditioned on both the class type τ and the

unknown true values of the training data. $P(\{y_t\}_\tau|\tau)$ is the prior on the true value $\{y_t\}_\tau$ given the class τ. Due to the uncertainties in the training data, the classification of just a single scalar data point requires an n-dimensional integral over the n^2 instances in the training data of each class τ. Section 4.2.1 focusses on the case where this integral can be solved analytically, which fortunately corresponds to many datasets in physical sciences.

In order to simultaneously perform anomaly detection and to normalise the posterior probabilities in equation 4.9, the Bayesian evidence for K known classes, $P(\{d^i\}, \{y_o\}_{\tau\in K})$ is computed over the entire training data $\{y_o\}_{\tau\in K}$, and for each test data instance, i, giving:

$$P(\{d^i\}, \{y_o\}_{\tau\in K}) = \sum_k^K P(\{d^i\}, \{y_o\}_\tau|\tau_k)P(\tau_k) \tag{4.8}$$

where the likelihood is given by equation 4.9. The evidence in equation 4.8 is used as an anomaly score: lower evidence values imply a data instance is more anomalous than test instances with higher evidence for one of the known classes.

If one has some prior knowledge of the anomalies, then a better alternative is to create a $K+1$-th class with no training data but with a prior $P(\tau_{K+1})$ that encodes this knowledge. This is however more sensitive to model misspecification: for example, using an anomaly prior performs worse when the noise is assumed uncorrelated Gaussian but is actually either correlated or non-Gaussian. The anomaly results in section 4.3 are therefore reported using equation 4.8 to rank instances.

4.2.1 Gaussian distributed data

This section deals with the case where the measurement uncertainties on all data are Gaussian. The significance of this case is twofold: (1) to good approximation this is actually the case that will be encountered in many astronomical experiments, and (2) the resulting solution affords significant computational speed ups.

The posterior can be analytically evaluated in the special case of uncorrelated Gaussian distributed test and training data, and for (improper) flat priors on $\{y_t\}_\tau$. The

[2]Strictly speaking this should be n_τ since the number of samples in each class will be different, but this is suppressed to keep the notation relatively simple.

two terms that then make up the likelihood are:

$$P(\{d\}|\{y_t\}_\tau, \tau) = \frac{1}{\sqrt{2\pi\sigma_d^2}} \exp\left(\frac{\{d\} - \{y_t\}_\tau}{\sigma_d}\right)^2$$

and

$$P(\{y_o\}_\tau|\{y_t\}_\tau, \tau) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(\frac{\{y_o\}_\tau - \{y_t\}_\tau}{\sigma_y}\right)^2,$$

where σ_d is the measurement uncertainty on d, and σ_y is the uncertainty on y_o. Equation 4.7 can then be solved analytically. The intermediate steps are omitted here, but the result is:

$$P(\{d\}, \{y_o\}_\tau|\tau) = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{m}(2\pi\sigma_d\sigma_y^i)^{-1}\left[\frac{\pi}{\frac{1}{2}(\Gamma_d + \Gamma_y^i)}\right]^{1/2} \times \exp\left(-\frac{1}{2}\left(\Gamma_d\{d\}^2 + \Gamma_y^i\{y_o^i\}_\tau^2 - \frac{(\Gamma_d\{d\} + \Gamma_y^i\{y_o^i\}_\tau)}{\Gamma_d + \Gamma_y^i}\right)\right) \quad (4.9)$$

where $\Gamma_d \equiv \sigma_d^{-2}$ and $\Gamma_y \equiv \sigma_y^{-2}$ are the precisions of the data, n is the total number of training instances and m is the number of datapoints per instance. Figure 4.4 demonstrates using BADAC for classification. The figure is a schematic example that shows how BADAC generates classification probabilities for a binary classification problem. The main figure represents the underlying known classes as error envelopes, and the adjacent subplots show how the value for the "true" new observed data is marginalised over. The figure shows this for the case where the data has two features, though the idea is fully general for n-dimensional features.

Equation 4.9 is used in section 4.3 to evaluate BADAC on data with uncorrelated Gaussian noise.

4.3 Experiments

To illustrate and test the performance of BADAC, a number of one-dimensional datasets are simulated, and results compared with those from a series of benchmark algorithms using multiple metrics. These metrics include the Rank-Weighted Score (RWS), introduced in section 3.7, that is optimised for anomaly detection.

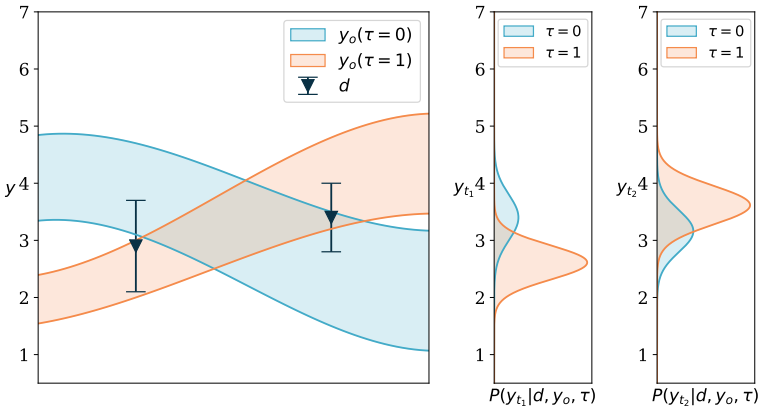

Figure 4.4: Schematic representation of BADAC as a classifier. *Left*: a single test example consisting of just two data points (black triangles with error bars). The training data comes from two classes shown schematically as the blue ($\tau = 0$) and orange ($\tau = 1$) 1-σ error envelopes. Which of these two classes does the test data come from? *Middle and Right*: panels showing the unnormalised posterior probability for the true value, y_t^i, for the first (middle panel) and second (right panel) data point, marginalised over the true value of the other point and conditioned on belonging to either class (class 0 - blue or class 1 - orange). The relative area of the corresponding Gaussian distributions in the middle and right panels gives the probability for the data to belong to either class. As can be seen, the data is more likely to come from class 1 (the orange class), in this case with a probability of 73%.

4.3.1 Simulations

For all the following experiments, data are simulated from arbitrary mathematical functions. Two base mathematical functions are used to build two "normal" classes and three other functions are used as anomalies. Each function has parameters which, when generating the data, are randomly drawn from a Gaussian distribution. The class functions and their corresponding parameter distributions are given in table 4.1.

Label	Type	Functional form	Parameter distributions
0	Inlier	$y = \sin(\omega \mathrm{x})$	$\omega \sim \mathcal{N}(5, 2)$
1	Inlier	$y = \alpha x^2 + \beta x + \gamma$	$\alpha \sim \mathcal{N}(0.5, 0.2)$ $\beta \sim \mathcal{N}(0.5, 0.2)$ $\gamma \sim \mathcal{N}(0, 0.2)$
2	Outlier	$y = h$ if $x \leq x_0$, else $y = 0$	$h \sim \mathcal{N}(1, 0.3)$ $x_0 \sim \mathcal{N}(0.5, 0.2)$
3	Outlier	$y = A \exp\left(-(\frac{x-\mu}{w})^2\right)$	$A \sim \mathcal{N}(0.5, 0.2)$ $\mu \sim \mathcal{N}(0.1, 0.05)$ $w \sim \mathcal{N}(1, 0.5)$
4	Outlier	$y = \frac{1}{5}\Sigma_{i=1}^{5} \sin(\omega_i x)$	$\omega_i \sim \mathcal{N}(30, 20)$

Table 4.1: Description of the functions used to create the simulated data. 99% of the test objects in the dataset are of the type "inlier" and 1% are "outliers". Each class has the corresponding functional form with parameters drawn randomly for each instance from Gaussian distributions with hyperparameters specified in the table.

For each experiment, 15000 curves are generated with roughly equal number of objects from class 0 and 1 as training data. In the test data, 1% outliers from classes 2, 3 and 4 are added. Figure 4.5 illustrates some randomly drawn objects from the training and test sets.

4.3.1.1 Experiment 1: Gaussian errors

The framework presented in section 4.3.1 is used to create a variety of experiments to test BADAC in both anomaly detection and classification. Here the data are simulated as described in section 4.3.1 with uncorrelated Gaussian errors on all data points. The standard deviation of the underlying noise distribution depends on the class, and is given by: $\sigma_0 = \sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 0.1$. This experiment is the ideal case in which the noise distribution used for generating the simulated data is the same as that in the mathematical formulation of equation 4.9.

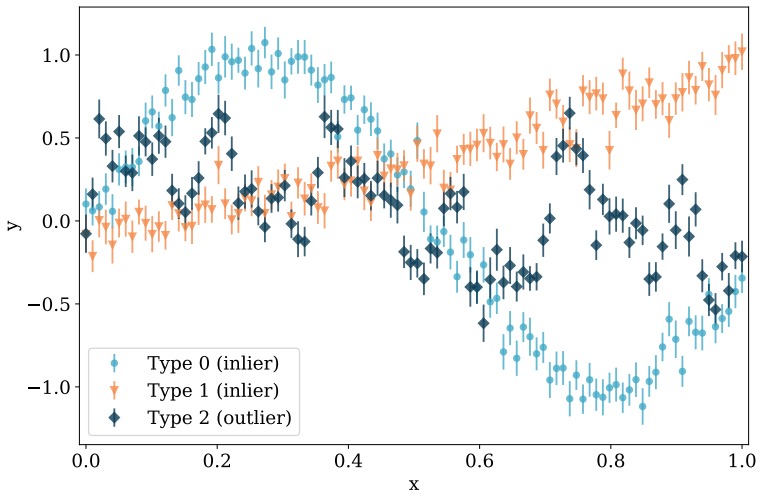

Figure 4.5: Illustrations of example objects from the simulated data. The plotted error bars correspond to 1σ error of Gaussian noise. The functional form and distribution of hyperparameters used to generate these examples is shown in table 4.1. The points are coloured by true type, where light blue circles correspond to a type 0 object, orange triangles to type 1 and dark indigo diamonds is an outlier. Only type 0 and type 1 curves are used during the training phase.

4.3.1.2 Experiment 2: Compact Anomalies

This experiment tests BADAC's ability to detect curves with a compact anomaly embedded somewhere in them. The anomalous objects are constructed by placing a narrow Gaussian on top of an object generated from the base class 0 (the sine curve) as described in experiment 1. The parameters of the sine curve are randomly drawn from the distribution described in table 4.1, and the parameters of the compact anomalies are randomly drawn as described in table 4.2. An example of a compact anomaly is shown in figure 4.6.

4.3.2 Comparison of algorithm performance

This section assesses the performance of BADAC on the simulated data discussed in section 4.3.1. This performance is compared to that of a series of benchmark algorithms, namely IsolationForest [87] and Local Outlier Factor (LOF) [14] for anomaly detection, and random forests [13] for classification.

The `sklearn` [112] implementations of all of these algorithms are used for this com-

Label	Type	Functional Form	Parameter dists.
0	Inlier	$y = \sin(\omega \mathrm{x})$	$\omega \sim \mathcal{N}(5, 2)$
1	Inlier	$y = \alpha x^2 + \beta x + \gamma$	$\alpha \sim \mathcal{N}(0.5, 0.2)$ $\beta \sim \mathcal{N}(0.5, 0.2)$ $\gamma \sim \mathcal{N}(0, 0.2)$
2	Outlier	$y = \sin(\omega \mathrm{x}) + \mathrm{A} \exp\left(-\left(\frac{\mathrm{x}-\mu}{\mathrm{w}}\right)^2\right)$	$\omega \sim \mathcal{N}(5, 2)$ $A \sim \mathcal{N}(1.5, 0.5)$ $\mu \sim \mathcal{U}(0, 1)$ $w \sim \mathcal{N}(0.03, 0.01)$
3	Outlier	$y = \sin(\omega \mathrm{x}) - \mathrm{A} \exp\left(-\left(\frac{\mathrm{x}-\mu}{\mathrm{w}}\right)^2\right)$	$\omega \sim \mathcal{N}(5, 2)$ $A \sim \mathcal{N}(1.5, 0.5)$ $\mu \sim \mathcal{U}(0, 1)$ $w \sim \mathcal{N}(0.03, 0.01)$

Table 4.2: Description of the functions used to create the compact anomaly simulated data. 99% of the test objects in the dataset are of the type “inlier" which are the same as class 0 and 1 in table 4.1. The remaining 1% are drawn from one of two compact anomaly classes. These are narrow Gaussians added to a randomly generated function of class 0. The parameters of the Gaussian are drawn randomly for each object from a distribution with hyperparameters as specified in the table.

	BADAC			IsolationForest			LOF		
	MCC	AUC	RWS	MCC	AUC	RWS	MCC	AUC	RWS
Gaussian	**0.95**	**0.99**	**0.99**	0.00	0.89	0.02	0.83	0.97	0.96
Com.Anom.	0.41	**0.91**	0.59	0.11	0.80	0.14	**0.44**	0.90	**0.63**

Table 4.3: Result summary for both the Gaussian error and compact anomaly (com. anom.) experiments using three metrics (MCC, AUC and RWS) discussed in section 3.7. The best performer is shown in bold. Note the particularly poor performance of IsolationForest in the MCC and RWS metrics. BADAC significantly outperforms the other algorithms in the Gaussian case.

parison. For anomaly detection, all algorithms receive only the input training data, and the percentage of outliers of 1%. For classification with random forests, the input parameter `n_estimators=1000` is used. There are unsupervised implementations of IsolationForest and LOF, but here the supervised methods only are considered.

Tables 4.3 and 4.4 below summarise the algorithms' performance for the 2 experiments presented in the previous section. Sections 4.3.2.1 and 4.3.2.2 present a more detailed assessment of these experiments.

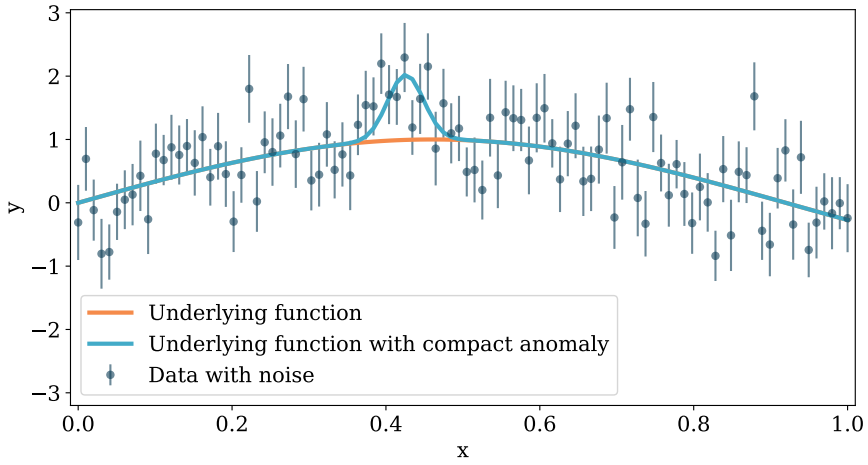

Figure 4.6: Example of the compact anomaly simulations. The underlying function from which the data were generated is shown as an orange solid line. The underlying function with the compact anomaly superposed is shown as the light blue solid line. The final data with noise are shown by the dark indigo scatter where the errorbars represent the 1σ Gaussian measurement error.

	BADAC	Random Forests
Gaussian Noise	**99.02**	98.66
Compact Anomalies	**95.51**	95.18

Table 4.4: Comparison of BADAC's *classification* performance to that of random forests using average accuracy across both inlier classes.

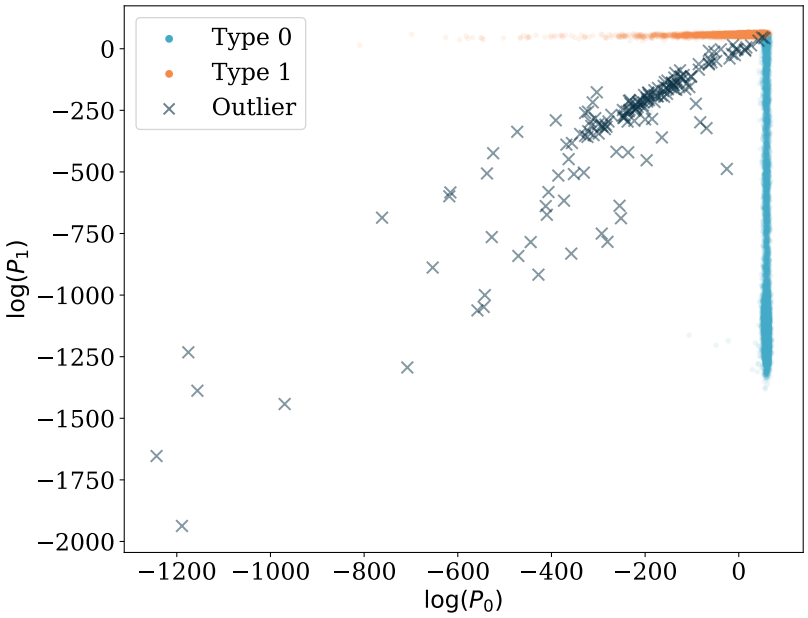

Figure 4.7: Probability scatter plot showing the computed log-probabilities returned by BADAC for the data with Gaussian measurement error. Each point corresponds to a test object, which is shown in the log(P_0)-log(P_1) space. Points that appear high on the y-axis have a high likelihood of being type 1. Points that appear higher on the x-axis (further to the right) have a high likelihood of being type 0. The points are coloured by their true type, where light blue corresponds to type 0, orange is type 1 and the dark crossed are anomalies.

4.3.2.1 Gaussian error performance

This section illustrates the performance of BADAC, as well as the benchmark algorithms, on the data discussed in section 4.3.1.1 with Gaussian measurement error. The formalism shown in section 4.2 is used to provide two probabilities, P_0 and P_1, which are the un-normalised probabilities of belonging to class 0 and class 1 respectively. These probabilities are plotted in figure 4.7.

Plotting the unnormalised probabilities is useful for visualising the decision boundary that separates both the known classes and anomalies. It also does not require us to make any assumptions about the nature of the anomalies we expect to see. However, to make use of these probabilities in an analysis pipeline, they must be normalised. In order to normalise the probabilities, the Bayesian evidence must be computed. In the case where one is interested in classification only, the evidence would be $P_0 + P_1$. In the case where anomaly detection is of interest as well, the

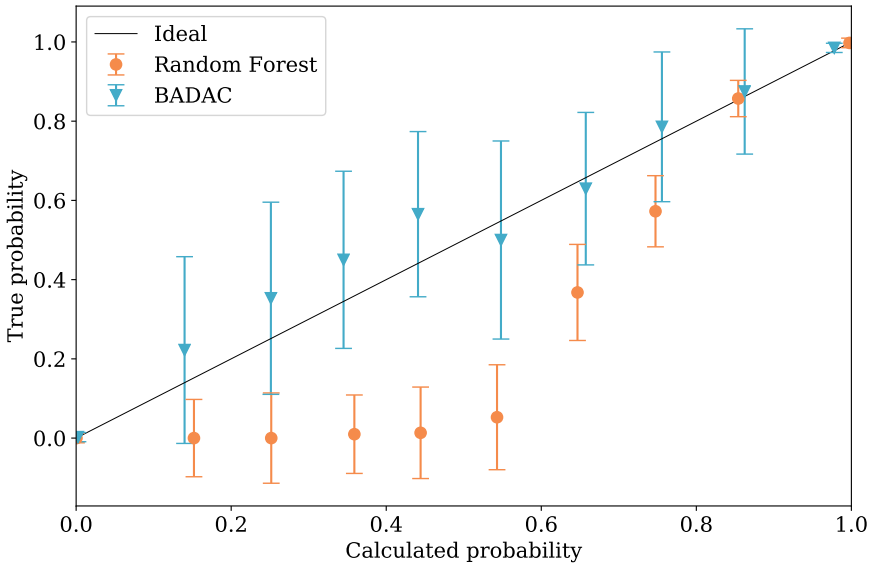

Figure 4.8: Probability calibration curve for the Gaussian case for BADAC and random forests (in classification only). A perfectly calibrated algorithm on a particular problem would return probabilities on the line $y = x$. This plot shows the probability, as returned by the respective algorithms, that each object in the test set is a type 1. The true probability is found by measuring the fraction of true type 1 objects in a particular bin of calculated probabilities. The errorbars show the Poisson uncertainties given by the number of objects in each bin. The x-coordinate for each bin is given by the mean calculated probability in that bin. Random forest gives poorly calibrated probabilities, while BADAC automatically returns well-calibrated probabilities.

evidence is $P_0 + P_1 + P_{\text{anomaly}}$, where in this case, P_{anomaly} is evaluated using a top-hat likelihood equal to $1/(b-a)$ over the range $[a, b]$, and equal to 0 otherwise. Here a and b are chosen to cover twice the observed range of the input data.

By binning the normalised probabilities for a single class, we can measure whether or not they are calibrated. It is a well known problem that many machine learning algorithms give uncalibrated probabilities that do not correspond to the true probability of an object belonging to a certain class [107; 154]. The reliability of probabilities can be investigated by plotting a probability calibration curve: the output probabilities from the algorithm for a selected class only are binned and compared with the actual fraction of objects in that bin belonging to the class. This result is shown for classification only for type 1 objects in figure 4.8, where the calibration curve of BADAC and those of random forests are compared.

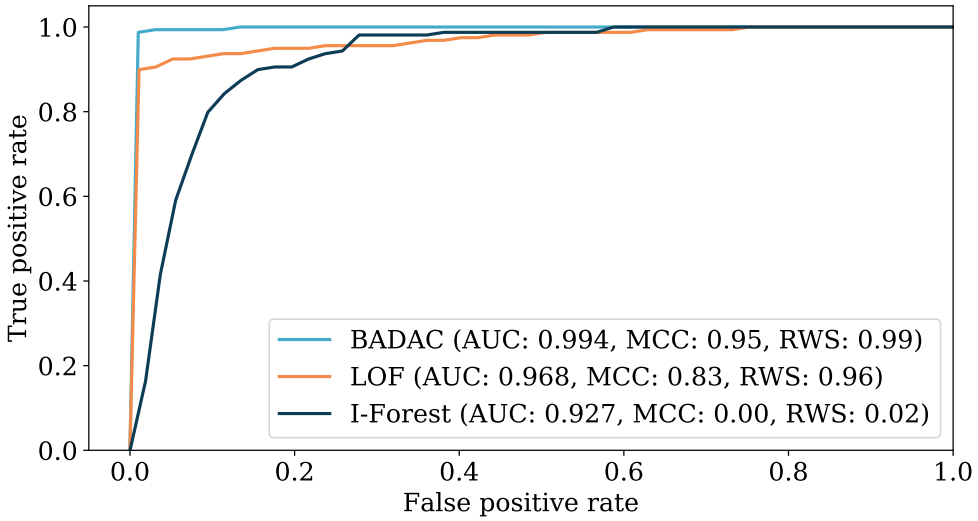

Figure 4.9: ROC curves for BADAC, LOF and IsolationForest on the dataset with uncorrelated Gaussian error for anomaly detection only. BADAC performs best under the AUC metric shown in the legend. An 'ideal' algorithm would have a ROC curve that reached the [0,1] vertex (in the top left corner), which would correspond to an AUC of 1.

ROC curves (see section 3.7 for a description of ROC curves) for BADAC as well as LOF and IsolationForest are shown in figure 4.9 in order to compare algorithm performance in anomaly detection. A summary of algorithm performance on all the datasets considered here in both anomaly detection and classification is shown in tables 4.3 and 4.4 respectively.

4.3.2.2 Compact anomaly performance

This section illustrates the performance of BADAC and the benchmark algorithms on the compact anomaly data discussed in section 4.3.1.2. It should be noted that the compact anomaly data is generated with Gaussian noise, which is the type of noise assumed in this implementation of the BADAC formalism, and is also the same kind of noise as the data described in section 4.3.1.1. This means one would expect the algorithms to have similar performance in *classification* in this section as in section 4.3.2.1. For this reason, the classification performance of any of the algorithms on the compact anomaly dataset is not discussed further here. This section proceeds in the exact same manner as section 4.3.2.1, except here we are interested in how robust the algorithms are to different types of anomalies (compact

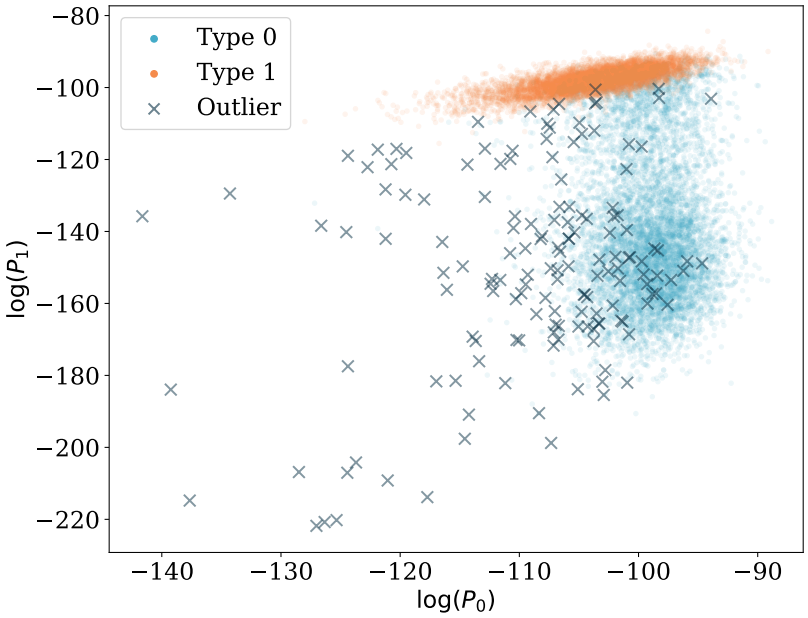

Figure 4.10: Scatter plot showing the computed log-probabilities for the test data discussed in section 4.3.1.2. Each point corresponds to a test object, which is shown in the $\log(P_0)$-$\log(P_1)$ space. Points that appear high on the y-axis have a high likelihood of being type 1. Points that appear higher (to the right) on the x-axis have a high likelihood of being type 0. The points are coloured by true type, where light blue corresponds to type 0, orange is type 1 and the dark crosses are outliers.

ones).

The importance of an algorithm being able to detect compact anomalies is twofold. Firstly, compact anomalies are often interesting in science when one wishes to measure or detect aberrant behaviour of known sources. Secondly, an algorithm's ability to detect compact anomalies demonstrates its overall sensitivity in measuring small variations within data.

The probabilities, P_0 and P_1, generated by the formalism discussed in section 4.2, are shown in figure 4.10.

As can be seen from figure 4.10, the outlier data has significant overlap with type 0 data. This is because the compact anomalies were generated on top of type 0 data only. The varying scale/amplitude to the anomaly is responsible for where the outlier data is positioned on the $\log(P_0)$-axis (further left indicates a more anomalous object). Outlier points with high $\log(P_0)$ values are likely associated with compact anomalies with very low amplitudes.

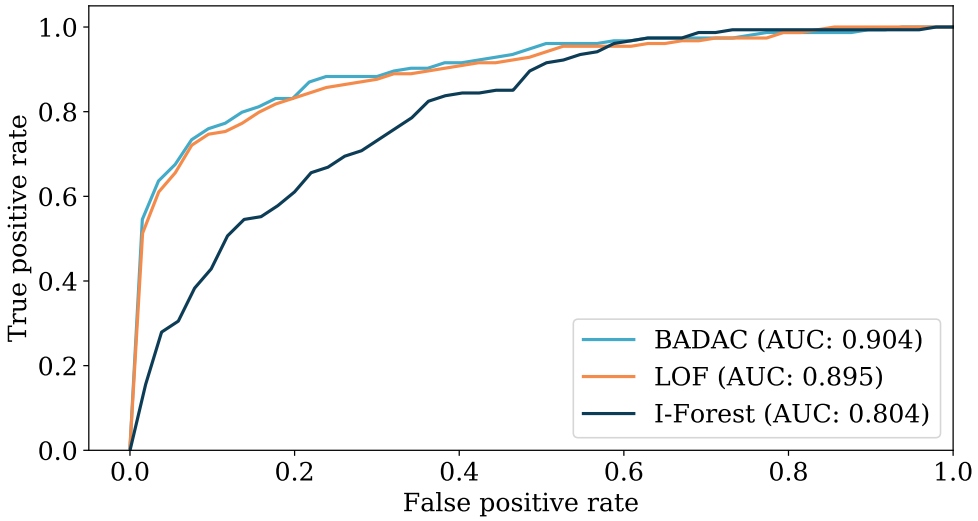

Figure 4.11: ROC curves for anomaly detection with BADAC, LOF and IsolationForest on the dataset with compact anomalies. BADAC performs best under the AUC metric, whose values in each case are shown in the legend.

Here the ROC curves for BADAC as well as LOF and IsolationForest are shown in figure 4.11, in order to compare algorithm performance in anomaly detection. Under the AUC metric, BADAC performs the best in this case. LOF is almost as good, and actually performs better under the MCC and RWS metrics. A summary of algorithm performance on all the datasets considered in both anomaly detection and classification is shown in tables 4.3 and 4.4 respectively.

4.3.3 Computational performance

It is difficult to give a “fair" comparison of computational performance between BADAC, random forests, LOF and IsolationForest, since unlike the benchmark algorithms, the BADAC algorithm has no distinct training and testing phases. This means that the computational complexity of these algorithms scales very differently (depending on the amount of training and test data available). For example, for a dataset with m training and n test examples, the computational time required for random forests, IsolationForest and LOF would increase as $f(m) + f(n)$. For BADAC, the computational time required increases as $f(n \times m)$. In fact the computational time increases linearly as a function of $n \times m$.

For an even comparison of computational performance, the same number of training

Algorithm	Training time (s)	Testing time (s)	Total time (s)
Random Forests	96.30	2.94	99.24
IsolationForest	1.62	1.21	2.83
Local Outlier Factor	13.21	27.25	40.46
BADAC	-	-	1281.82

Table 4.5: Comparison of the computational performance between the three algorithms compared in section 4.3. All measurements were made on the dataset used in experiment 1 (Gaussian noise) with 15000 training and 15000 test curves. There are no values shown for testing and training times for BADAC, since there are no distinct training and testing phases. Measurements were made on a 2.9GHz processor, where each algorithm was limited to use a single core.

and testing samples as were used in section 4.3.2 are considered here (15000 training samples and 15000 test samples). The total time (training time + testing time) is quoted in table 4.5. It should be noted, however, that there is ample room for optimisation and parallelisation in the BADAC code and the timings could be considerably improved.

As is evident in table 4.5, BADAC has a computational cost of around an order of magnitude more than any of the competing algorithms that are considered. Ways of mitigating this are discussed in the next chapter (section 5.7.2).

4.4 Concluding Remarks

This chapter has presented a novel statistically robust joint anomaly detection and classification method, Bayesian Anomaly Detection And Classification (BADAC), that is designed to take advantage of any knowledge of the underlying noise distribution in the training and test data. Although the tests performed in this study were for the case of Gaussian distributed data, the formalism is general.

This study presents a test of the classification and anomaly detection capabilities of BADAC using simulated one-dimensional data. Several metrics are used to gauge algorithm performance, including the novel Rank-Weighted-Score that rewards algorithms for ranking more anomalous objects above those that have been commonly seen. In the case where the correct noise model is known, BADAC outperforms random forests at classification and both IsolationForest and local outlier factor (LOF) at anomaly detection, due to its ability to correctly exploit uncertainty informa-

tion. In the case of compact anomalies, which could emulate noisy spikes in data, BADAC's performance is comparable to LOF and superior to IsolationForest on the simulated data. This study also demonstrates how BADAC produces calibrated classification probabilities, which is crucial if a machine learning algorithm is to be incorporated into a precise, scientific analysis pipeline.

While BADAC provides excellent performance by exploiting the extra information about the underlying noise distributions, the computational limitations discussed in section 4.3.3 mean that it does not scale well to large training datasets. In this case one must either use prototype templates to represent the classes (e.g. through Gaussian processes) or parametrise the data, to speed up classification and anomaly detection with BADAC.

We find ourselves in an era of exponentially increasing data volume, driving the need for machine learning algorithms. However, in the physical sciences there is equal need for accurate propagation of uncertainties from all parts of an analysis pipeline, including any machine learning algorithms. With its statistically principled approach to both classification and anomaly detection, BADAC is able to provide believable and interpretable probabilities in the presence of measurement uncertainties, as required by high precision scientific analysis.

Chapter 5

Breaking BADAC

The previous chapter showed that for Gaussian errors, BADAC outperformed random forests, LOF and IsolationForest under most metrics considered. This is perhaps not surprising since BADAC was designed to use the extra information available, namely that there are uncorrelated errors on the data that are Gaussian distributed. This chapter presents a series of more challenging tests where the uncorrelated Gaussian BADAC formalism is used, but on data that do not obey this model. Three experiments are presented, using the same framework as in the previous chapter: an experiment where the data have non-Gaussian noise in section 5.1, an experiment where the data have correlated Gaussian in section 5.2 and an experiment where the data have varying inter-class noise in section 5.4. This chapter concludes with concluding remarks and future work in sections 5.6 and 5.7 respectively.

5.1 Experiment 3: Non-Gaussian errors

For this experiment, and the experiments that follow, the data are simulated exactly as in Experiment 1. A description of the functions used to create the data is shown in table 4.1. For this experiment non-Gaussian errors are used instead of the Gaussian errors of Experiment 1. For 80% of the y values (randomly selected) of any given simulated object, the noise is drawn from a Gaussian distribution with standard deviation as described in section 4.3.1, meaning the scatter matches the error bar. However, for the remaining 20% of the values, the noise is drawn from a Gaussian

distribution of five times the width, resulting in scatter dramatically underestimated by the reported error bar.

5.2 Experiment 4: Correlated Gaussian Noise

This section presents a test of the sensitivity of BADAC to the uncorrelated noise assumption, by using it on data generated with correlated Gaussian noise. Here, only data from Class 0 is correlated, according to a "wedding cake" covariance matrix (based on [75] and [79]):

$$C_{ij} = \sigma_i \sigma_j \delta_{ij} + V_{ij}, \tag{5.1}$$

where

$$V_{ij} = \sum_{k=1}^{n_{i,j}} s_k \tag{5.2}$$

where i and j are indices of the data (in order of x value) and $n_{i,j}$ is the bin to which the object belongs. To produce the step-like structure, $n_{i,j} = \lfloor \frac{min(i,j)}{N/5} \rfloor + 1$ (where "$\lfloor\rfloor$" indicates the floor function, rounding down to the nearest integer). Here, $s_k = 0.1$ is used for each k. The result is that the data are correlated in such a way that the points at higher x-values are more correlated than the lower ones.

5.3 Results for Experiments 3 and 4

This section presents results for both classification and anomaly detection for the data discussed in sections 5.1 and 5.2 with both non-Gaussian and correlated Gaussian noise. It should be noted that equation 4.9 was used to determine classification/anomaly detection probabilities, despite the fact that the data does not have Gaussian uncorrelated noise as equation 4.9 assumes. Thus the performance of BADAC must be expected to decrease; the question is by how much?

In order to normalise the probabilities shown in figure 5.2, the evidence, $Z = P_0 + P_1 + P_{anomaly}$, is computed as in the previous chapter. As before, $P_{anomaly}$ is evaluated in order to determine the Bayesian evidence, using a top-hat likelihood equal to $1/(b-a)$ over the range $[a, b]$ and equal to 0 otherwise. In this case, naively choosing the width of the top-hat does not work, as the model used to determine P_0 and P_1 is incorrect, and hence returns low probabilities. As a result, $P_{anomaly}$ is a much higher

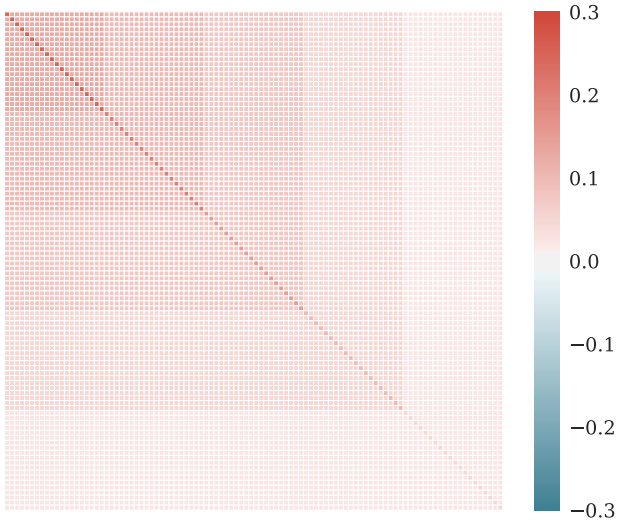

Figure 5.1: The covariance matrix used for correlating the class 0 data for experiment 3. This is a "wedding cake" covariance matrix, the form of which is shown in equation 5.1. The data are ordered by x-value starting at the top left corner (so values near the beginning of a given curve would be more highly correlated than those near the end). Class 1 and anomaly data remain uncorrelated.

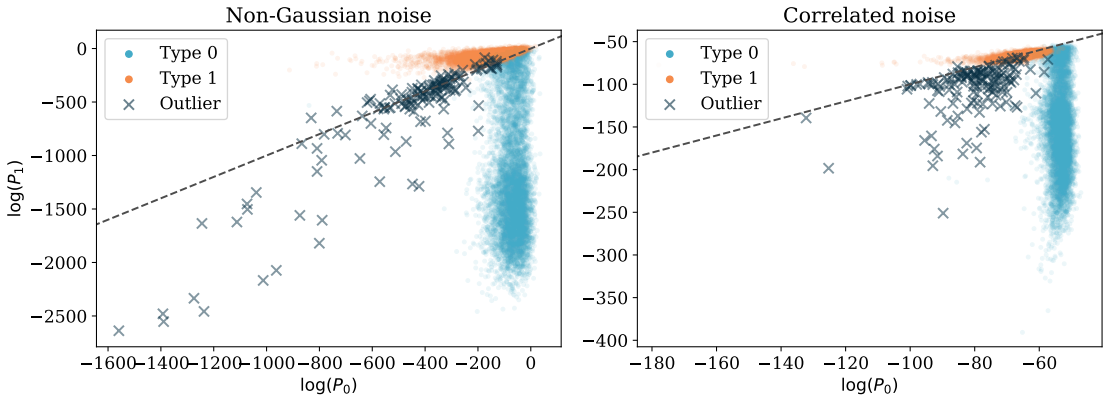

Figure 5.2: Probability scatter plot for the dataset with non-Gaussian noise (left panel), and the dataset with correlated Gaussian noise (right panel). Each point corresponds to a test curve, which is shown in the log(P0) − log(P1) space. The line $y = x$ has been added to each plot to highlight the bias introduced by using the wrong model for the noise with BADAC. Here the bias is only visible in the correlated noise case since only class 0 was correlated.

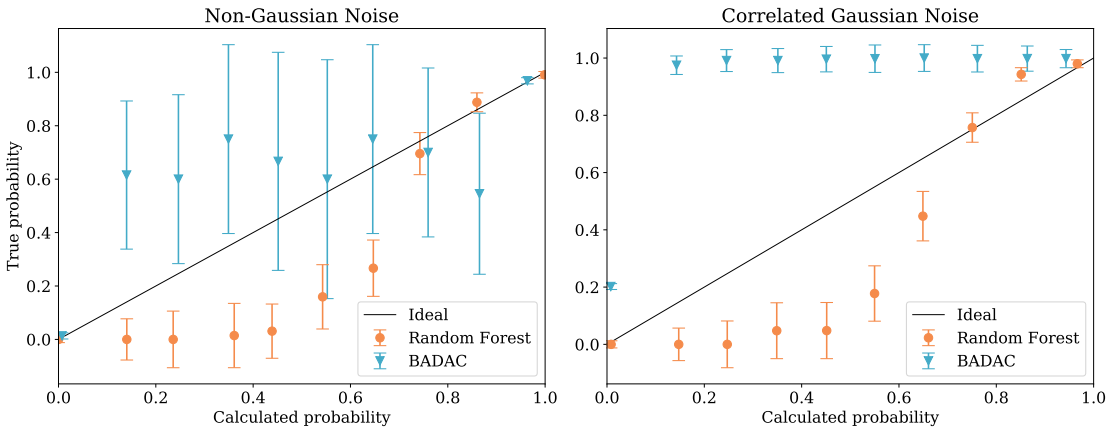

Figure 5.3: Probability calibration curves showing the degree to which the probabilities returned by each algorithm (in classification only) are calibrated for the non-Gaussian case (left panel) and the correlated Gaussian case (right panel). Perfectly calibrated probabilities would lie on the line $y = x$. Here the probability of an algorithm classifying an object as type 1 is considered. All objects within a particular probability range are binned, and the fraction of correct predictions plotted. The errorbars show the Poisson uncertainties given by the number of objects in each bin. While non-Gaussian noise does not distort the probabilities dramatically, correlated noise has a strong effect due to a fundamentally incorrect noise model assumption. Despite this, BADAC outperforms both Isolation Forest and LOF in all metrics considered for anomaly detection, but not surprisingly struggles with the classification in the correlated noise case.

probability than P_0 and P_1, even for inlier data, when the incorrect model for the noise is used. To get around this the height of the top-hat likelihood, $1/(b-a)$, is equated to $P_0 + P_1$ computed for the object corresponding the 99th percentile. The values of a and b can then be solved for. This enforces that the algorithm labels the most anomalous 1% of objects (as determined by the algorithm) as outliers. This is still a fair comparison with the benchmark algorithms, as both IsolationForest and LOF receive the percentage contamination of 1% as an input parameter. Methods to extend the BADAC formalism to be suitable for modelling data with different types of noise are discussed in section 5.7.5.

As can be seen from the scatter of classification probabilities shown in figure 5.2, there is more overlap of different object types than in the uncorrelated Gaussian noise case in Experiment 1. Additionally, in the correlated case, there is a significant bias introduced due to the noise from only one of the classes being correlated. Since the model does not favour fitting this class, these classification probabilities are not

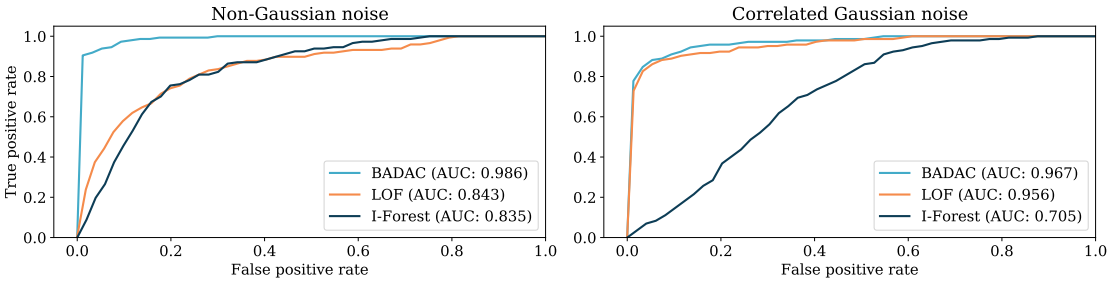

Figure 5.4: ROC curves for anomaly detection with BADAC, LOF and IsolationForest on the dataset with non-Gaussian error (left pane), and the dataset with correlated Gaussian error (right pane). BADAC performs best in both cases under the AUC metric (values shown in the legend).

	BADAC			IsolationForest			LOF		
	MCC	AUC	RWS	MCC	AUC	RWS	MCC	AUC	RWS
Non-Gauss.	**0.84**	**0.99**	**0.96**	0.06	0.84	0.10	0.16	0.84	0.18
Corr.Gauss.	**0.68**	**0.97**	**0.84**	0.01	0.70	0.03	0.61	0.96	0.76

Table 5.1: Results for anomaly detection only: Non-Gaussian (Non-Gauss. in table) and correlated Gaussian (Corr.Gauss. in table) noise. BADAC produces the best performance in both experiments, showing some robustness to incorrectly choosing the model of the noise. In the non-Gaussian case both IsolationForest and LOF perform poorly in terms of MCC and RWS due to the wide tails in the data, which allow for large noise fluctuations.

reliable. This is illustrated both by figure 5.2, where the diagonal dashed line shows where type 0 and type 1 clusters should be separated, and figure 5.3, where the classification probabilities are far from the calibrated line, $y = x$. This is due to the fact that an uncorrelated Gaussian model for the noise is used, despite the fact this model is wrong.

The ROC curves for BADAC as well as LOF and IsolationForest are presented in figure 5.4 in order to gauge performance in anomaly detection. In these two cases, it is surprising BADAC performs best, since the model for the noise is incorrect. Random forests however achieves a higher accuracy in classification in these two cases. A summary of algorithm performance on all the datasets considered in both anomaly detection and classification is shown in tables 5.1 and 5.2 respectively.

	BADAC	Random forests
Non-Gaussian noise	97.71	**98.14**
Correlated Gaussian noise	68.88	**96.72**

Table 5.2: Here the average accuracy of BADAC for *classification* over all classes is compared to that of Random forests. BADAC performs reasonably in the case of non-Gaussian noise but poorly on the correlated noise case, due to the incorrect model assumption in the BADAC formalism. Random forests is more robust as it can learn a model from the training data, while BADAC insists on interpreting the fluctuations as coming from an uncorrelated Gaussian distribution. This relatively poor performance of BADAC can be rectified by using, or learning, the right noise model.

5.4 Experiment 5: Variable Inter-class Noise

A case where the classification performance of BADAC drops is when there is systematic difference in the noise between the different classes considered. In order to demonstrate this drop in performance, several datasets with varying inter-class are simulated using the same framework as in the previous chapter. Details of this simulation are described in section 4.3.1, and more specific details shown in table 4.1. The difference for this experiment is that the noise on class 0 and class 1 objects, σ_0 and σ_1 respectively, is varied in order to see how sensitive BADAC's performance is to systematically different inter-class noise.

This experiment is performed using a grid of simulations, each with different $[\sigma_0, \sigma_1]$. The grid used is $\sigma_0 \equiv (0.1, 0.5, 0.9)$ and $\sigma_1 \equiv (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$. Diverging from the previous experiments, no anomalies are included in this experiment, since the effect it aims to illustrate does not have a huge impact on anomaly detection performance.

An important detail to note is that the drop in classification performance this experiment aims to illustrate increases with data of higher dimension. As a result of this, the proceeding analysis is done for the case where the data have 10-dimensional features, as well as the case where the data have 100-dimensional features (as in chapter 4).

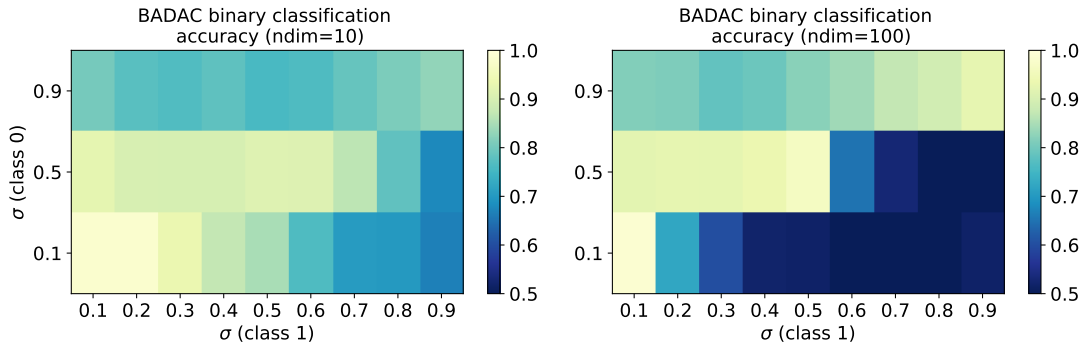

Figure 5.5: Classification accuracy for BADAC on 10-dimensional data (left pane) and on 100-dimensional data (right pane). In both cases, 27 different datasets are considered on a 3×9 grid, where each grid item corresponds to a case where the noise of class 0 and the noise of class 1 differ systematically. Each block on the grid is coloured by accuracy, where light yellow corresponds to 100% accuracy, and dark blue corresponds to 50% accuracy. The classification performance is worst in the case where $\sigma_1/\sigma_0 \gg 1$, though it is also low when $\sigma_0/\sigma_1 \gg 1$ (or any case where $\sigma_0 \approx \sigma_1$ is not true). Additionally, this drop in performance becomes worse in higher dimensions, as shown in the right pane.

5.5 Results for Experiment 5

This section presents the results for the experiment described in section 5.4. It begins by illustrating the degradation in classification performance caused when the noise on the data from the two classes is different, and then presents a number of viable solutions.

The classification accuracy of BADAC for this experiment is illustrated in figure 5.5 for the case with 10 dimensions (left panel), as well as with 100 dimensions (right panel).

What causes this poor performance? Plotting the classification probability scatter returned by BADAC gives insight into this, as shown in figure 5.6. Here, the probability scatters for both the test and training sets are shown. The probability scatter for the training set is generated using an $n - 1$ scheme: for n training instances, each instance is classified using the remaining $n - 1$ instances as 'training data'.

As can be seen in figure 5.6, the misalignment between the decision boundary and probability scatter is relatively consistent between training and test data. This insight is important, since an additional step can simply be introduced to learn a

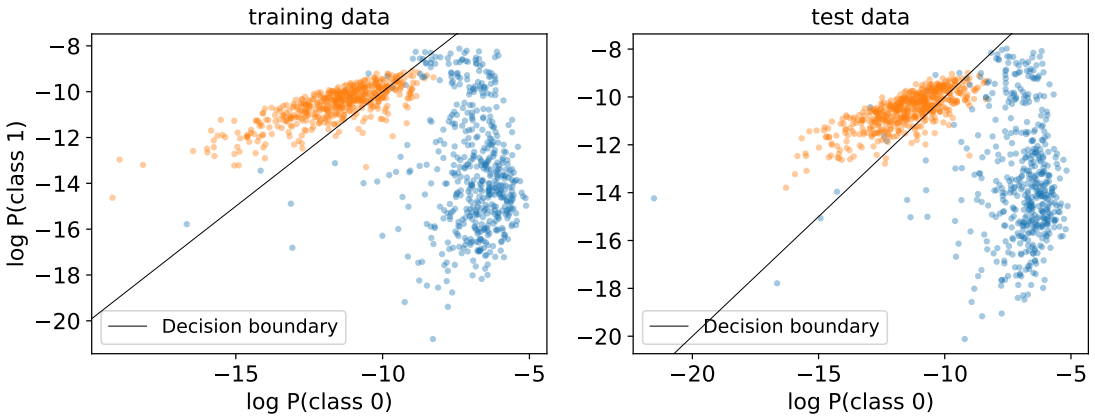

Figure 5.6: Probability scatter returned by BADAC on the training set (left pane), and on the test dataset (right pane). The dataset has 1000 training instances, 1000 test instances, $\sigma_0 = 0.3$ and $\sigma_1 = 0.5$. Class 0 objects are shown as blue points and class 1 objects are shown as orange points. The decision boundary used to classify test instances is shown by the solid black line. As can be seen, there is still clear separation between the two classes, the classes are just misaligned with respect to the decision boundary. This misalignment is consistent between training and test data. This decision boundary yields an accuracy of 90.4% in this case.

new decision boundary. Here this is done using an implementation of Support Vector Machines (SVMs) [116; 23]. However, there are many algorithms that can be used, and an alternative is discussed below. Figure 5.9 shows how the SVM classifier finds a new decision boundary on the training data, which can then be used to classify instances in the test set.

This method of utilising SVMs to 'calibrate' the decision boundary given the training data can be used on the same data as considered in figure 5.5. The result of this is shown in figure 5.8. Figure 5.8 illustrates that this method of calibration improves the performance of BADAC drastically in cases where the noise on the different classes is systematically different. The classification accuracy is now near perfect, with the exception of the top right corner in the low dimensional ($n_{\text{dim}} = 10$) dataset, where the magnitude of the noise is now comparable to that of the underlying signal.

If there is some reason why a linear decision boundary is preferred, instead a method such a Linear Discriminant Analysis (LDA) (generalised by [119]) can be used for this calibration step. This is illustrated in figure 5.9, where the scikit-learn [112] implementation of LDA is used to learn the new decision boundary. Here, using LDA to learn the new decision boundary does not work as well as one may naïvely

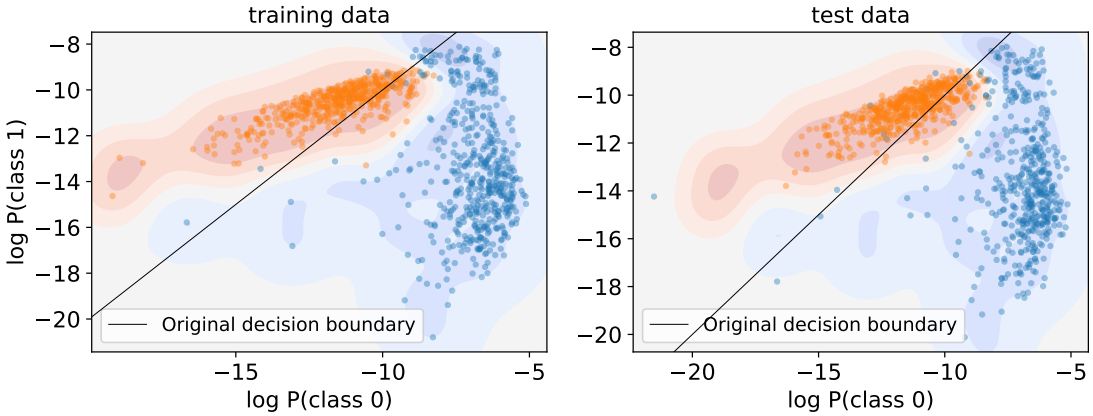

Figure 5.7: Probability scatter returned by BADAC on the training set (left pane), and on the test dataset (right pane). This scatter is the same as that shown in figure 5.6 on the same dataset. Here the blue contours define the region of this 2D parameter space where the SVM classifier determines class 0 objects. The orange contours define the region of this 2D parameter space where the SVM classifier determines class 1 objects. The solid line shows the original decision boundary. The SVM decision boundary is learned from the training data (left pane), though the same decision boundary is plotted alongside with the test data (right pane). The SVM decision boundary is clearly a more apt choice than the original in this case. Using the SVM decision boundary, the accuracy is now 98.2% on the test data, where before it was 90.4%.

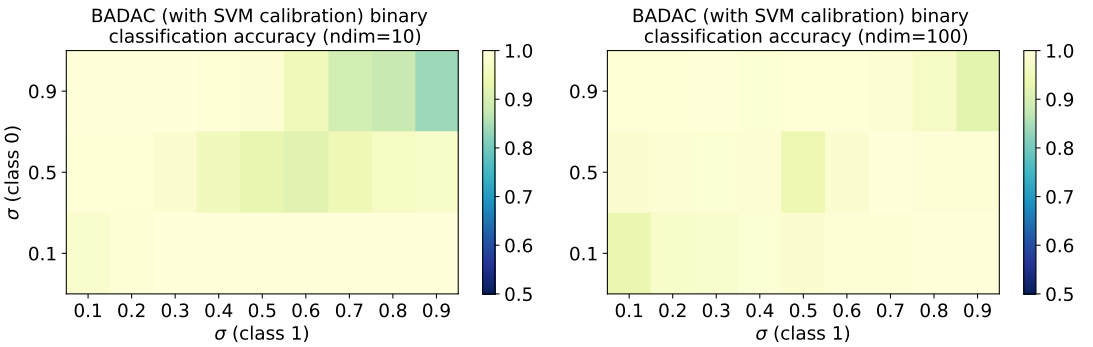

Figure 5.8: Classification accuracy for BADAC on 10-dimensional data (left pane) and on 100-dimensional data (right pane). This analysis is done on the same dataset used to produce figure 5.5. Each block on the grid is coloured by accuracy, where light yellow corresponds to 100% accuracy, and dark blue corresponds to 50% accuracy. The accuracy with the learned SVM decision boundary is now near perfect on the datasets considered in this experiment. In particular, classification accuracy in the cases where the noise on the two classes is systematically different is vastly improved.

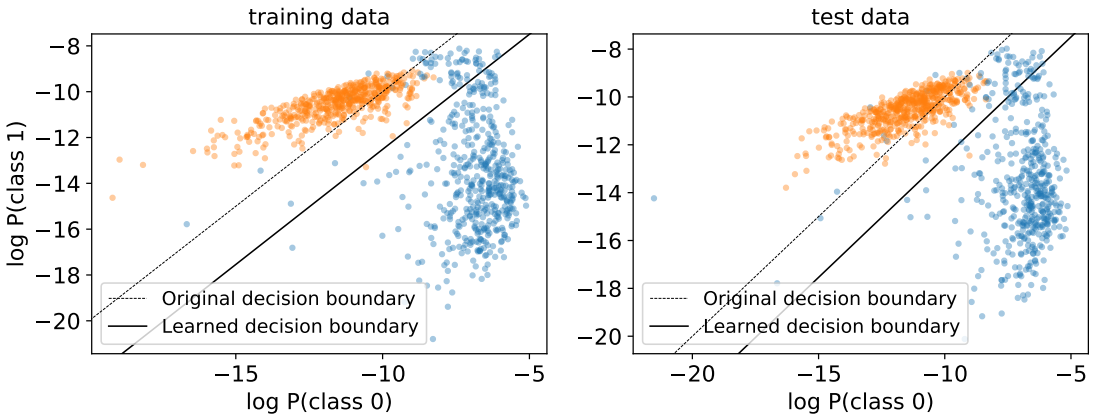

Figure 5.9: Probability scatter returned by BADAC on the training set (left pane), and on the test dataset (right pane). This scatter is the same as that shown in figure 5.6 on the same dataset. Here the dashed line shows the original decision boundary, and the solid black line shows the new decision boundary learned by LDA. Using the decision boundary learned by LDA, the accuracy score is now 96.8% on the test data.

expect. The new decision boundary does not correctly classify a large number of the class 0 (blue points) objects in figure 5.9. This is likely because LDA assumes that each of the clusters are Gaussian, and in this case the blue cluster in particular is clearly non-Gaussian. This is substantiated by the fact the blue cluster seems to comprise of two distinct sub-clusters. Though it is not clear here, this can be ascribed to shot noise, as it is not always a feature that appears when running this analysis with different random seeds.

Regardless of which algorithm is used to learn the decision boundary, this approach to calibrating BADAC incurs significant computational overhead when compared with the implementation used in chapter 4. This is because the classification probabilities for the training data need to be calculated as well - a step that is omitted in the original implementation. For a balanced train/test sample (50% training 50% test data), this corresponds to around double the computational cost, though this worsens with very large training sets. This somewhat limits the scalability of BADAC, which is an important consideration in cases where minimising computational complexity is important.

5.6 Concluding Remarks

This chapter builds on the work in chapter 4 by presenting a number of experiments in order to stress-test BADAC, as well as to investigate the degradation of performance if the assumptions made in the BADAC formalism are violated. Three independent experiments are conducted in order to determine how sensitive BADAC is to the incorrect modelling of the noise distribution. These experiments are: (1) non-Gaussian noise, (2) correlated Gaussian noise and (3) varying inter-class noise. This chapter illustrates exactly how the performance of BADAC decreases in these cases, and also presents ways to mitigate this drop in performance.

Interestingly, for the experiments on non-Gaussian and correlated Gaussian noise, BADAC still outperforms the other anomaly detection algorithms despite assuming an uncorrelated Gaussian noise model. However the classification performance degrades, especially in the correlated case. It should be noted that with an incorrect noise model, the probabilities of BADAC are no longer guaranteed to be calibrated as standard. An approach to mitigating this using a self calibration step is shown in the varying interclass noise experiment. This approach is shown to be effective, though at additional computational cost, an area in which the standard BADAC algorithm struggles already. However, if the noise distribution is known, the correct noise model can be incorporated into the BADAC likelihood. This will however incur additional computational overhead for most noise distributions as well.

The findings presented in this chapter mean that, even when the assumptions made in the mathematical formalism do not accurately represent the data, BADAC still performs well on the tests considered. As in the previous chapter, BADAC is suited to use-cases where statistical rigour is important and computational cost is not a limiting factor.

5.7 Future Work

This section presents possible extensions, of which there are many, to the work presented in this chapter, as well as chapter 4.

5.7.1 Dealing with Missing Data

The experiments thus far have assumed the idealised case, where there is data at the same points for all training and test data. This is clearly unrealistic and an important limitation, and must be dealt with for use on real data.

There are two approaches to do this. The first more conservative approach is to sample from the prior distribution with the error also given by the prior distribution for that class. If the data is missing from test data, the missing data can be sampled in the same way, but in each case the prior for the class that it is being considered must be used.

The second approach is to use some form of interpolation. A natural approach is to use Gaussian processes, since these give both an expected value and Gaussian error at the missing data. Gaussian processes need a covariance function, which encodes how rapidly the underlying class varies. As a result each class will have its own Gaussian process and covariance function which should be learned from the training data. Test data should then be compared to training classes using the appropriate Gaussian process for each of the training classes.

5.7.2 Template Construction

As shown in table 4.5 the full BADAC calculation is much slower than other classification or anomaly detection algorithms. This stems from the pairwise comparison of all data in the test dataset with all data in the training set, something which becomes computationally infeasible for very large amounts of training data.

Fortunately in the limit of large training data, the class distribution can be well sampled, and a single template for each class (or as an intermediate step, each subclass) can be created. This will dramatically speed up BADAC, though at the cost of having a non-Gaussian spread in general.

How should the class or sub-class templates be constructed? An elegant solution is to fit a single Gaussian process to the data of each class [149]. This has the advantage of automatically dealing with any missing data, but will not deal with non-Gaussian or multi-model intra-class variability. To get around this limitation one could use a Kernel Density Estimate summed over the training data in each class at each value of the independent variable (or in bins). However, since this is

still a sum over all the training data examples it will be slow. For computational speed up, an approximation to the KDE sum should be used.

Probably the simplest approximation - which also preserves the Gaussian distribution - is to use the inverse-variance estimator $\hat{y}$ with standard deviation $\hat{\sigma}$:

$$\hat{y} = \hat{\sigma}^2 \sum_i y_i/\sigma_i^2 \tag{5.3}$$

$$\hat{\sigma}^2 = \left(\sum_i 1/\sigma_i^2 \right)^{-1} \tag{5.4}$$

If the intraclass variability is highly non-Gaussian then it would be better to fit a more appropriate low-dimensional distribution to describe this to create the template.

5.7.3 Intraclass Variability

In the BADAC formalism, it was assumed that the variability in the observed data for a given class was small relative to the measurement errors. If this is not the case one can build more complex models for the intra-class variability. The simplest is to fit for a global standard deviation, σ_*, at training for each class (for example by using a validation subset of the training data). The intra-class variability model can be made arbitrarily complex and the Bayesian evidence could be used to select the best model.

5.7.4 Calibration and Zero-point Issues

In applying BADAC to real examples there may be systematic differences in the data between test and training. This could for example be because the data comes from different instruments or is taken under different conditions. As an example, consider applying BADAC to images where there may be large-scale calibration differences across the images. How can one deal with such effects which will invalidate the use of the simple versions of the BADAC formalism presented earlier, along with most anomaly and classification algorithms?

In the spirit of the Bayesian approach, one way to deal with such large-scale artefacts is to model their effects and introduce nuisance parameters φ, with their own prior

distributions $P(\varphi)$, which are then marginalised over before classification or anomaly ranking. Intuitively this means that the algorithm will exploit the freedom implicit in the calibration model to try to fit each test curve to the training data and will only highlight as outliers those data which are poor fits no matter the calibration freedom.

A related problem is the issue of zero-points, which occurs if the examples in training and test data are not all aligned on the x-axis. This is common when working with time series data. In principle this can be dealt with in a similar way, by allowing each data example to have an extra translation parameter which allows one to shift all points in the example left or right. One must then marginalise over this nuisance parameter when doing the fits.

Depending on the exact nature of the data these translation parameters may be well-constrained. For example, one may be able to align all examples approximately, in which case one can put priors on the translation parameters. However, the zero-point issue does raise significant complications. For each pair in the training and test sets one should in principle allow a translation parameter. This leads to $n \times M$ new nuisance parameters where n, M are the number of training and test set examples respectively. Unless the marginalisation can be performed analytically this will typically be prohibitively expensive.

A cheaper alternative is to pre-align all the training data by class. Now there is only one translation nuisance parameter per class and per instance in the test set. However, the alignment of the training data will not be perfect in general. This can be handled by adding an x-error bar to each data point in the training data, corresponding to small errors in the alignment of the data. These x-errors are perfectly correlated however (since the translation affects all data in the same way) and the BADAC formalism would need to be extended to account for such correlations, as done in e.g. [59; 124].

5.7.5 Non-Gaussian data

Often, the standard deviation is used as a proxy for the error distribution on an observation, even when the distribution is non-Gaussian. Sections 5.1 and 5.2 tested how the algorithm developed in the previous chapter (section 4.2) performs while assuming a Gaussian error distribution, even when the error distribution is non-

Gaussian. However, if the error distribution is known, the forms of the likelihood can be replaced with the known non-Gaussian distribution. These could be the binomial distribution in the case of count data, or the Poisson distribution in the case of certain time series. Any appropriate distribution that can be modelled can be used in this formalism. In the case of the binomial distribution, one would do a summation rather than an integration over the latent variables. For any distribution that doesn't yield an analytically integrable form for $P(\tau|d, y_o^1, ..., y_o^n)$, one can do the marginalisation numerically, though at increased computational cost.

5.7.6 Online Learning of New Classes

Once it has been confirmed that an anomaly represents a new class (i.e., if the Bayesian evidence for the anomaly class is higher than that of any of the existing classes) it can automatically be added to the training data as a new class (with a single example) to be compared with. This provides an online-learning version of the BADAC algorithm. Any future data belonging to the new anomaly class will be automatically assigned the new anomaly class label.

This process in no way limits us to a single new anomaly class. The BADAC formalism allows for the automatic addition of new classes as indicated by the data. If a new kind of anomaly is different from any previously identified anomalies it will be assigned to a new class, assuming the Bayesian evidence favours the addition of another class.

Chapter 6

Bayesian Estimation Applied to Multiple Species for Photometric Data

The discovery of the accelerated expansion of the universe has dominated the field of cosmology since the 1990s. Upcoming experiments such as the Vera C. Rubin telescope in Chile will observe of order 10^5 type Ia supernova (SNIa) candidates with photometric lightcurves only. Constraining cosmological parameters (such as dark energy content) precisely is difficult in this case, since the resulting systematic uncertainty is comparatively large.

This chapter begins by introducing the topic of supernova cosmology in section 6.1, as well as describing how inference is typically done in the field in section 6.2. Some current challenges in the field, as well as approaches to deal with these challenges are then discussed in section 6.3. Finally, section 6.4 demonstrates how these challenges can be dealt with using a series of experiments.

6.1 Introduction to Supernova Cosmology

Much of modern cosmology has been dominated by the study of the accelerated expansion of the universe due to dark energy [148]. Type Ia supernovae (SNe Ia) are a vital probe into the expansion history of the universe, since they are what are known as standard (or more accurately standardisable) candles. The physical

mechanisms behind the explosion of SNe Ia result in them having some intrinsic emitted luminosity/brightness, which we can determine, given their environment. We can thus create a distance measure, in this case the luminosity distance, for how far away the SNe appear given their observed brightness. The luminosity distances of these SNe can be compared to their redshifts, the speed at which objects move away from us due to cosmic expansion, in order to gauge how fast the universe is and has been expanding.

Studies of Type Ia supernovae led to the dark energy breakthrough in cosmology [113; 122], but one can argue that they have been supplanted by Baryon Acoustic Oscillations [6; 148] as the most precise way of constraining cosmology today. To be competitive in the era of LSST [91], EUCLID and the SKA, supernova cosmology faces several big challenges. One is that better control of systematics is required and a number of sophisticated approaches are being developed to improve the control of systematics (e.g. [97; 127; 74; 132; 93; 68; 8; 20]). Another critical problem is that next-generation supernova surveys will be severely spectroscopy limited: LSST will deliver over 10^5 Type Ia Supernova (SNIa) candidates with photometric lightcurves only. The lack of spectroscopy introduces a number of challenges. First, the true identity of any candidate without spectroscopic follow-up is ambiguous - photometric colours only provide a probability for an object to be a SNIa, as opposed to a Type Ibc or II supernova or other transient [129]. This is illustrated to some extent in figure 6.1, taken from [69]. Secondly, the precise redshifts of the supernovae are unknown. Photometric redshifts are fairly good if the candidates are known to be SNIa, yielding RMS errors of $\sigma_z \sim 0.04(1+z)$, depending on exact assumptions [91; 72; 144]. The problem is that we are exactly in the case where we are not sure whether each candidate is a SNIa or not, and the photometric redshift error is much larger if the object is not a SNIa [102], precisely because they are not standard candles.

A promising approach that dates back to the SDSS II supernova survey [61; 18; 108], is to obtain spectroscopic redshifts for the host galaxies of the supernova candidates and use this as a proxy for the supernova redshift. This will be particularly attractive in the era of big redshift surveys such as 4MOST, SKA and Euclid, where huge numbers of galaxy redshifts will be known. This has the potential to help remove biases [108] and yield improved constraints [153].

However, even this approach has a serious problem: identifying the host galaxy is

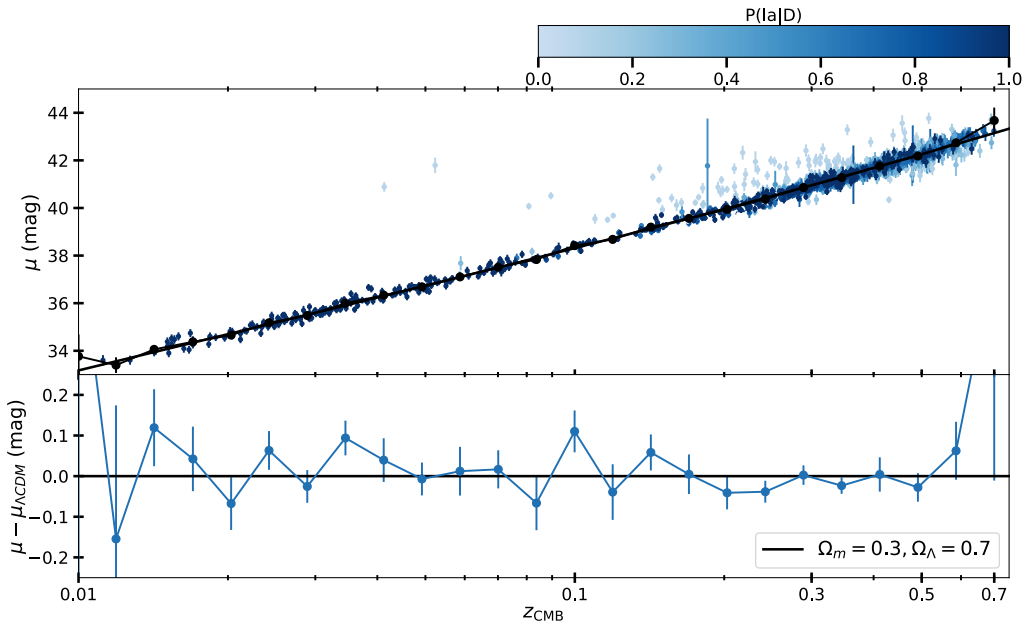

Figure 6.1: Figure taken from [69]. Top pane: Hubble diagram for a SN sample. The opacity of the points is plotted corresponding to the probability of being a type Ia, $P(\text{Ia}|\text{D})$, as per the colour-bar. Bottom pane: Hubble residuals of the sample assuming a flat ΛCDM universe with $\Omega_m = 0.3$ and $\Omega_\Lambda = 0.7$. Contamination of the sample by non-Ia SNe is illustrated by the light scatter points, which deviate from the Hubble diagram.

also not unambiguous. The supernova can appear to lie in between two or more galaxies or may live in a host that is too faint to be detectable ("hostless" galaxies). In general, it should therefore be assumed that instead we have probabilities for the supernova to belong to each of the nearby galaxies on the sky or to be hostless. Current matching algorithms can accurately match the correct host galaxy about 91% of the time when applied to data, potentially increasing to 97% by using machine learning techniques [47]. However, even a 3% contamination may cause significant biases on cosmological parameter inference and must be dealt with.

6.1.1 Photometric vs. Spectroscopic Astronomy

Photometry and spectroscopy are the two primary modes of observing electromagnetic (EM) radiation in astronomy. Photometry makes use of all EM radiation that the instrument is sensitive to, typically in order to build a spatial image of a particular region of the sky. Spectroscopy spreads the light from a particular point on the sky into components of varying wavelength. Crucially, spectroscopic mea-

surements yield a spectrum of the object being observed, from which, information about absorption/emission lines can be used to calculate the redshift of the object precisely.

This section does not aim to describe these two modes in detail, but rather highlights how photometric or spectroscopic SN observations change the way in which one should approach the resulting cosmological analysis. Traditional SN cosmological analyses (e.g. [122; 113; 2; 71]) proceed under the assumption that there are both multi-band photometric observations, as well as spectroscopic follow-up measurements. In this case, the photometric observations yield the total observed flux in multiple bands, which can be used to determine the distance modulus of the SN. The spectroscopic follow-up observations reveal absorption/emission lines that indicate the presence of certain elements within the SN. These are used for two reasons: (1) to determine the SN type (for cosmology we are largely only interested in type Ia SNe), and (2) to accurately estimate the redshift. Many upcoming experiments will observe thousands of SNe Ia with photometry only. Without spectroscopic follow-up, these two vital pieces of information about SN type and redshift cannot be obtained directly. In this case, producing a precise cosmological analysis is difficult. The following sections provide an approach to deal with these problems.

6.2 Standard Inference for Supernova Cosmology

Traditional supernova cosmological analyses make use of spectroscopically confirmed type Ia supernovae only, in order to make inferences about cosmological parameters, θ. The light[1] emitted from distant objects is subject to dilation along the line of sight (LoS) between the source and the observer. Additionally, the source may be moving away from the observer at the time the light was emitted, given by the expansion of the universe. These phenomena result in the redshifting of light between the source and the observer. The observed (redshifted) light has wavelength:

$$\lambda_{\text{obs}} = (1 + z)\lambda_{\text{emit}} \tag{6.1}$$

where z is the redshift. The redshift of distant objects is one of the measures used in any cosmological analysis.

[1]In this case, 'light' refers to electromagnetic radiation in any wavelength.

Another important measure is the luminosity distance, d_L. The luminosity distance is defined in terms of cosmological parameters in equation 6.5. More intuitively, it is the distance calculated to an object based on its luminosity, and the observed flux:

$$d_L = \sqrt{\frac{L}{4\pi F}} \tag{6.2}$$

where L is the luminosity, and F is the observed flux. This is useful in the case of type Ia supernovae which have some intrinsic luminosity [11]. The luminosity distance differs from a standard measure of distance for two reasons. Firstly, the distance between the source and the observer increases between the time of source emission and the time the light is observed. Secondly, the observed flux dilates along the LoS due to the expansion of space. The luminosity and comoving distances are related by:

$$d_L = (1+z)d_M \tag{6.3}$$

where d_M is the transverse comoving distance, or simply the comoving distance for a flat universe (when $\Omega_k = 0$).

The Hubble parameter for a given redshift in a ΛCDM universe is calculated by:

$$H(z) = H_0\left(\Omega_m(1+z)^3 + \Omega_{\mathrm{DE}}(1+z)^{3(1+w)} + \Omega_k(1+z)^2\right)^{1/2} \tag{6.4}$$

where H_0 is the Hubble constant, Ω_m is the energy density of matter, Ω_{DE} is the energy density of dark energy, Ω_k is the curvature parameter and w is the dark energy equation of state where $w = -1$ corresponds to Λ, the cosmological constant [133].

$$d_L(z) = \frac{c(1+z)}{H_0\sqrt{-\Omega_k}}\sin\left(H_0\sqrt{-\Omega_k}\int\frac{dz'}{H(z')}\right) \tag{6.5}$$

The distance modulus for a type Ia supernova is usually estimated from its observed light curve using the SALT2 model [48].

The distance modulus is defined as:

$$\mu(z) = m - M = 5\log_{10}\left(\frac{d_L}{1\mathrm{Mpc}}\right) + 25 \tag{6.6}$$

where m is the apparent magnitude of the object, M is the absolute magnitude of the object and d_L is the luminosity distance to the object in Mpc. Equations 6.4 to 6.6 provide a distance modulus-redshift relation in terms of cosmological parameters

that can be fit for.

This section presented a supernova cosmological analysis for the case where all SNIa data have well-measured spectra. However, this may not always be the case since many SNe Ia are only measured photometrically.

In sections 6.3 and 6.4.2 that follow, the experiment that deals with the instance where all SNe redshifts are spectroscopically confirmed is referred to as the 'spectroscopic case', and the experiment that deals with the instance where there are photometric measurements only is referred to as the 'photometric case'.

6.3 Inference with Type and Redshift Uncertainties

This section presents the experiment referred to throughout this chapter as the 'spectroscopic case'. It examines the case where the supernova spectra are not measured, but where the supernova redshifts are spectroscopically confirmed, given the host galaxy. In this case, photometric observation of the supernova yields the peak flux measurement, and the proximity of the supernova to its host galaxy, the redshift of which is assumed to be spectroscopically measured independently of the photometric observation.

The source of this uncertainty is that it is not possible to say with complete certainty that a given supernova is part of a given galaxy.

6.3.1 Inference in the Presence of Redshift Uncertainties

This section provides a stepping stone to the fully general case, and describes how redshift uncertainties impact on a supernova cosmological analysis. Spectroscopic measurements yield very small redshift uncertainties, δz. When performing a cosmological model fit using only spectroscopically confirmed SNe Ia, these uncertainties are dealt with by converting them into additional distance modulus uncertainty, $\delta\mu$, by adding them in quadrature. While this is not statistically correct, it works for very small δz. This fails for typical photometric redshift uncertainties, as will be shown later.

Here a general framework for incorporating the redshift uncertainty in a statistically

rigorous way is presented. The posterior distribution over cosmological parameters, θ, given data, $D \equiv (z_{\rm obs}, \mu_{\rm obs})$, can be expressed using Bayes' law. The arguments of the posterior can then be expanded in a hierarchical model to include the true redshift, z, and true distance modulus, μ. These latent parameters are then marginalised over, since we don't know their true values[2]:

$$P(\theta|D) \propto P(D|\theta)P(\theta) \tag{6.7}$$

$$P(\theta|D) \propto \int P(D, z, \mu|\theta)P(\theta)\ dz\ d\mu \tag{6.8}$$

Repeated application of the product rule allows for the expansion of this integral:

$$P(\theta|D) \propto \int P(D|z, \mu, \theta)P(z, \mu|\theta)P(\theta)\ dz\ d\mu \tag{6.9}$$

$$P(\theta|D) \propto \int P(D|z, \mu)P(\mu|z, \theta)P(z|\theta)P(\theta)\ dz\ d\mu \tag{6.10}$$

Now it is assumed that the true redshift values are independent of cosmological parameters, and that the distribution $P(\mu|z, \theta)$ is a delta function since μ is a deterministic function of z and θ. This allows for the elimination of the μ-integral since $\int f(x)\delta(x - x_0)dx = f(x_0)$:

$$P(\theta|D) \propto \int P(D|z, \mu)\delta(\mu - \mu(z, \theta))P(z)P(\theta)\ dz\ d\mu \tag{6.11}$$

$$P(\theta|D) \propto \int P(D|z, \mu(z, \theta))P(z)P(\theta)\ dz \tag{6.12}$$

Equation 6.12 shows the expression for the posterior distribution with general redshift uncertainties. For the case where the data are an estimate of the redshift and distance modulus from the extracted SN lightcurve, this becomes:

$$P(\theta|D) \propto P(\theta) \int_0^\infty P(z_{\rm obs}, \mu_{\rm obs}|z, \mu(z, \theta))P(z)\ dz \tag{6.13}$$

Assuming $z_{\rm obs}$ and $\mu_{\rm obs}$ are independent, this simplifies to:

$$P(\theta|D) \propto P(\theta) \int_0^\infty P(z_{\rm obs}|z)P(\mu_{\rm obs}|\mu(z, \theta))P(z)\ dz \tag{6.14}$$

Here an important role is played by the redshift prior, $P(z)$, which has no parallel in

[2]Hierarchical models and marginalisation are discussed in detail in Chapter 2.

the usual supernova analysis where the redshift is known spectroscopically. In the case of uncertain supernova redshifts, the Eddington bias must also be considered: a supernova discovered with a 4m telescope with a redshift estimate of $z_{\rm obs} = 0.75$, is much more likely to have a *true* redshift of $z = 0.6$ than $z = 0.9$.

The use of the incorrect prior, $P(z)$, (i.e. not the prior from which the data were drawn in the case of simulated data) leads to biased parameter estimation. This was first shown by [46], and illustrated with a simple example.

The inclusion of this prior is crucial in any practical application. Though it was stated that $P(z)$ is independent of cosmological parameters, there is actually a weak dependence on cosmological volume. Additionally, there is a dependence on SN rates. These issues can be dealt with by introducing latent parameters, φ, into the prior, $P(z, \varphi)$, which can then be fit for or marginalised over. Here the problem can actually be turned around, and the cosmological parameters marginalised over, such that the posterior over the SN rates can be solved for. Doing this is trivial in the case where all variables are simultaneously fit for using a numerical sampling technique such as MCMC (as is done here).

While this will be an important feature for more realistic simulations, it is not considered further here. Two special limiting cases are considered in the following sections: (i) spectroscopic galaxy redshifts but unknown host galaxy, and (ii) photometric supernova redshifts alone.

6.3.1.1 Unknown Host Galaxy

Studies and surveys of galaxies produce galaxy catalogues that are useful for photometric SN analyses as well. Photometric SN observations can use spatial information to ascertain whether a SN belongs to a certain host galaxy. In this case, the host galaxy redshift can be considered the same as the supernova redshift, which can greatly reduce redshift uncertainty if there are spectra for the host galaxy. This can however produce a new problem: how can we say with certainty that a photometrically observed SN belongs to a particular galaxy? Galaxies close to one another can cause confusion as to which is the true host. Additionally, the SN may lie in a galaxy that is too faint to resolve in the photometric SN observation.

This problem can be dealt with by specifying a redshift prior, which incorporates the probabilities of belonging to each candidate galaxy, which is then marginalised

over:

$$P(z) = \sum_{\gamma} P(z|\gamma)P(\gamma) \tag{6.15}$$

Here $P(\gamma)$ is the probability of belonging to a particular galaxy, and $P(z|\gamma)$ is the value of the redshift prior as in equation 6.12 evaluated at the redshift of galaxy γ, z_γ. I.e. $P(z|\gamma) = \delta(z - z_\gamma)$.

$$P(\theta|D) \propto P(\theta) \int_0^\infty P(z_{\rm obs}, \mu_{\rm obs}|z, \mu(z,\theta)) \sum_{\gamma} P(z|\gamma)P(\gamma)\ dz \tag{6.16}$$

6.3.1.2 Photometric Redshifts

A useful second limiting subcase is the case where there is no spectroscopic host information but instead only a photometric redshift estimate from the supernova itself or from the host galaxy in the case where the host is unambiguous. In this case we have an estimate for $P(z_{\rm obs}|z)$.

For simplicity it is assumed that the resulting photometric redshift distribution is Gaussian. This turns out to be a good assumption, though the generalisation to an arbitrary distribution is in principle simple since marginalisation needs to be performed numerically in this case. The formalism remains unchanged with the new photometric redshift distribution.

Here we can continue from equation 6.13, a restatement of which is:

$$P(\theta|D) \propto P(\theta) \int_0^\infty P(z_{\rm obs}, \mu_{\rm obs}|z, \mu(z,\theta))P(z)\ dz$$

If $z_{\rm obs}$ and $\mu_{\rm obs}$ are correlated, for example if they both come from the lightcurve, then equation 6.13 becomes:

$$P(\theta|D) \propto P(\theta) \int \frac{1}{2\pi\sqrt{\det|C|}} \exp\left(-\frac{1}{2}\Delta^T C^{-1}\Delta\right) P(z)dz\,, \tag{6.17}$$

where $\Delta = \begin{pmatrix} \mu_{\rm obs} - \mu \\ z_{\rm obs} - z \end{pmatrix}$ and $C = \begin{pmatrix} \sigma_\mu^2 & \sigma_{\mu z} \\ \sigma_{\mu z} & \sigma_z^2 \end{pmatrix}$ is the covariance matrix.

Assuming independent $z_{\rm obs}$ and $\mu_{\rm obs}$ for simplicity, equation 6.13 reduces to:

$$P(\theta|D) \propto P(\theta) \int \frac{1}{2\pi\sigma_z\sigma_\mu} \exp\left(-\frac{(z_{\rm obs}-z)^2}{2\sigma_z^2}\right) \exp\left(-\frac{(\mu_{\rm obs}-\mu(z,\theta))^2}{2\sigma_\mu^2}\right) P(z)dz \tag{6.18}$$

In the experiments shown in section 6.4.2, the marginalisation over redshift in equation 6.18 is performed numerically without considering correlations between $z_{\rm obs}$ and $\mu_{\rm obs}$, which is not generally true. However, such correlations can be included by modelling the covariance function in terms of some hyperparameters that can be marginalised over as well.

6.3.2 Contamination from non-Ia Supernovae

The previous section showed a derivation of the posterior in the presence of redshift uncertainties, but assumed that all objects considered were SNIa. However, contamination of supernova types will be a problem for photometric surveys as well. This has been addressed by the BEAMS (Bayesian Estimation Applied to Multiple Species) formalism for the case of spectroscopic redshifts [83; 79; 127; 61]. This section presents the key ideas from these works using the same hierarchical approach as the previous section. This will allow for a simpler task of combining these solutions when tackling the fully general case (both redshift and type uncertainty).

Here it is assumed that the redshifts of the supernovae are known exactly, but the type of the supernova is unknown. The supernova type is shown by the discrete variable τ. The variable τ takes on one of two values in this case, $\tau = \text{Ia}$ for type Ia supernovae, or $\tau = \text{nIa}$ for non-Ia supernovae.

As before, it is assumed that the data are estimates of the distance modulus and redshift extracted from the lightcurve. Therefore, for data, $D \equiv (z_{\rm obs}, \mu_{\rm obs})$, we have:

$$P(\theta|D) \propto P(D|\theta)P(\theta) \tag{6.19}$$

$$P(\theta|D) \propto \sum_\tau P(D,\tau|\theta)P(\theta) \tag{6.20}$$

$$P(\theta|D) \propto \sum_\tau P(D|\tau,\theta)P(\tau|\theta)P(\theta) \tag{6.21}$$

Here the latent parameter, τ, is summed over since it is a discrete parameter. Equa-

tion 6.21 is a result of the application of the product rule. Finally, the prior on τ is independent of cosmological parameters, θ, which yields the following result:

$$P(\theta|D) \propto P(\theta) \sum_{\tau} P(D|\tau, \theta) P(\tau) \tag{6.22}$$

For $\boldsymbol{\tau} \equiv [\mathrm{Ia}, \mathrm{nIa}]$, where $\tau_i \in \boldsymbol{\tau}$, this simplifies to the BEAMS [83] result:

$$P(\theta|D_i) \propto P(\theta) \Big[P_{\mathrm{Ia}} P(D_i|\tau_i = \mathrm{Ia}, \theta) + (1 - P_{\mathrm{Ia}}) P(D_i|\tau_i = \mathrm{nIa}, \theta) \Big] \tag{6.23}$$

where P_{Ia} is the probability of the SNe being type Ia, $P(\tau_i = \mathrm{Ia})$. Here it is assumed that there are only 2 groups of SNe that need to be considered for a cosmological analysis: SNe Ia and SNe non-Ia. SNe Ia are informative, and SNe non-Ia are not. In general, one may have to also account for type-II SNe, which are weakly informative of the cosmological parameters of interest. This can be accounted for by instead summing over 3 possible SN classes in equation 6.23. Additionally, this would require an extra model such that $P(D_i|\tau_i = \mathrm{II}, \theta)$ can be evaluated. While this may be a factor of more realistic simulations, it is not considered further here.

6.3.3 General Case: Type and Redshift Uncertainty

The preceding sections have presented approaches to dealing with type and redshift contamination independently of on another. Here the general case is considered, where both type and redshift uncertainty is modelled. Here it is assumed that the type, τ, of the SN is unknown and that we have either a photometric redshift estimate for the SN or other redshift information (either photometric or spectroscopic) of potential host galaxies.

Here, as in section 6.3.1, the host galaxy information is treated as a prior, $P(z)$, on the SN redshift:

$$P(z) = \sum_{\gamma} P(z|\gamma) P(\gamma) \tag{6.24}$$

The redshift information needn't be spectroscopic, which would just mean $P(z|\gamma)$ is a broader distribution than those assumed in section 6.3.1.1.

As in the previous cases, the posterior is expressed using Bayes' law. The hidden model parameters for the true redshift, distance modulus and SN type, z, μ and τ

are then introduced and marginalised over. Application of the product rule allows for the expansion of the posterior distribution as follows:

$$P(\theta|D) \propto \int P(D,\tau,z,\mu|\theta)P(\theta)\,d\tau\,dz\,d\mu \tag{6.25}$$

$$\propto \int P(D|\tau,z,\mu,\theta)P(\tau,z,\mu|\theta)P(\theta)\,d\tau\,dz\,d\mu \tag{6.26}$$

$$\propto \int P(D|\tau,z,\mu)P(\mu|\tau,z,\theta)P(\tau,z|\theta)P(\theta)\,d\tau\,dz\,d\mu \tag{6.27}$$

$$\propto \int P(D|\tau,z,\mu)P(\mu|\tau,z,\theta)P(\tau|z,\theta)P(z|\theta)P(\theta)\,d\tau\,dz\,d\mu \tag{6.28}$$

$$\propto \int P(D|\tau,z,\mu)\delta(\mu-\mu(\tau,z,\theta))P(\tau|z)P(z)P(\theta)\,d\tau\,dz\,d\mu \tag{6.29}$$

$$\propto \sum_{\tau}\int P\left(D|\tau,z,\mu(\tau,z,\theta)\right)P(\tau|z)P(z)P(\theta)\,dz \tag{6.30}$$

where the last step accounts for the fact that τ is a random variable. Substituting the prior on redshift given by the information from potential host galaxies gives:

$$P(\theta|D) = P(\theta)\int dz \sum_{\tau} P(\tau|z)P(D|\tau,z,\mu(\tau,z,\theta))\sum_{\gamma} P(z|\gamma)P(\gamma) \tag{6.31}$$

Equation 6.31 is the main result of this section, showing the posterior distribution for a single SN with both type and redshift uncertainty. Note that since the true redshift of the SN is unknown, the type-redshift dependency, $P(\tau|z)$, must be modelled. This is different to the approach taken by BEAMS [83], where this is assumed to be constant.

Thus far, the formalism presented has been for a single SN. To compute the full posterior for N SNe, collectively referred to as $\mathbf{D}$ where $D_i \in \mathbf{D}$ for $i \in [1, N]$, the same approach as before can be taken. Assuming the SN are independent of one another, the N posteriors can be multiplied together:

$$P(\theta|\mathbf{D}) \propto P(\theta)\prod_{i}^{N}\left[\int dz \sum_{\tau_i} P(\tau_i|z_i)P\left(D_i|\tau_i,z_i,\mu(\tau_i,z_i,\theta)\right)\sum_{\gamma_i} P(z_i|\gamma_i)P(\gamma_i)\right] \tag{6.32}$$

Here there are now $2N$ nuisance parameters to marginalise over: τ_i and z_i for each SN. Correlations between SNe can however also be considered, as in [79].

In practice, the marginalisation over redshift required by equation 6.32 can be

achieved efficiently through MCMC by allowing the redshift of each SN to be a free nuisance parameter that is varied along with the cosmological parameters, θ. If the SNe are correlated, then the type of each SN must also be introduced as a nuisance parameter. This does not significantly alter the MCMC analysis [79].

At this point one can again ask what the data, D_i, is for each supernova? At the most basic – and correct – level this would be the lightcurve measurements in various colour bands as a function of time. Extracting distance modulus and redshift information from each SN in this case is not independent of cosmological parameters, and would need to be fit for simultaneously using MCMC, requiring a not insubstantial addition in computational time. This does however mean that the zBEAMS formalism can be applied when working with the raw lightcurves. An added complexity arises when one considers quality cuts performed on lightcurve data or selection effects, such as those considered in [61; 17; 127]. These quality cuts may however introduce additional sources of bias, since it is likely catastrophic redshift outliers and type miss-classifications (which are often clipped from the analyses for practical reasons) are at the faint end of the SNe population. This has the potential to introduce another source of Malmquist bias, though the effect is small compared with typical selection effects. The idea behind the zBEAMS formalism, however, is to consider all the data, and marginalise over the uncertainty associated with catastrophic outliers. Additionally, for the case of selection effects, the unknown number of undetected SNe needs to be marginalised over as well, using for example simulated telescope observations. This is, from a Bayesian perspective, more satisfactory than performing quality cuts from the outset since the uncertainty as to which SNe are excluded from the sample is marginalised over rather than specified upfront.

However, a convenient simplification is to consider $D_i = (z_{\rm obs}, \mu_{\rm obs})$, *assuming* that the object is a SNIa. In general this would be inappropriate if more than one type of supernova contained useful information about the cosmological parameters θ. However, since in the non-Ia case the derived redshift and distance modulus give almost no useful cosmological information, one can simply take $P(z_{\rm obs}|z, \tau = \text{nIa})$ to be a wide uniform distribution or very wide Gaussian and use $z_{\rm obs}, \mu_{\rm obs}$ derived assuming the object is a SNIa [83; 127]. Then the fact that one is using the "wrong" values for $z_{\rm obs}, \mu_{\rm obs}$ has no impact. This is only an issue when the type of the supernova is unknown.

In the proceeding experiments, selection bias and the impact of performing quality

cuts is ignored since including them would required working at the level of the lightcurves, which has been avoided for simplicity.

6.4 Experiments

This section presents an illustrative set of simulations to show how zBEAMS, as described in section 6.3, recovers the correct cosmological parameters by marginalising over unknown supernova types and redshifts. This section considers 2 cases, one where there are spectroscopic redshift estimates from potential host galaxies, hereafter referred to as the *spectroscopic case*, and one where there are only photometric measurements, hereafter referred to as the *photometric case*. Conceptually, it is trivial to combine these cases when dealing with a dataset with mixed spectroscopic and photometric measurements, however these two cases are kept distinct here.

For both cases, it is assumed that all objects are detected, no matter how faint. Both cases also demonstrate the bias on cosmological parameters that is introduced when using the standard likelihood, ignoring redshift and type uncertainties. This section shows that zBEAMS is able to correctly marginalise over these uncertainties, recovering the fiducial cosmology.

For all simulations, we assume a flat ΛCDM universe with a fiducial cosmology given by the latest results from the Planck collaboration [115], i.e., $H_0 = 67.74$ km/s/Mpc, $\Omega_m = 0.31$ and $w = -1$. Inference over the parameters is done using Markov Chain Monte Carlo (MCMC) methods, specifically the Metropolis-Hastings [99; 58] algorithm for low-dimensional sampling and block Metropolis-Hastings when numerically marginalising over redshift in the photometric case. Detailed descriptions of both cases follow below.

6.4.1 The Spectroscopic Case

For the spectroscopic case, 1000 supernovae are simulated over a redshift range of $z \in [0.015, 1]$ and with a uniform redshift distribution. A reference dataset is created without any redshift of type uncertainties, herein referred to as the unbiased dataset. A dataset with both host galaxy redshift and non-Ia contamination is also created, and is hereafter referred to as the biased dataset.

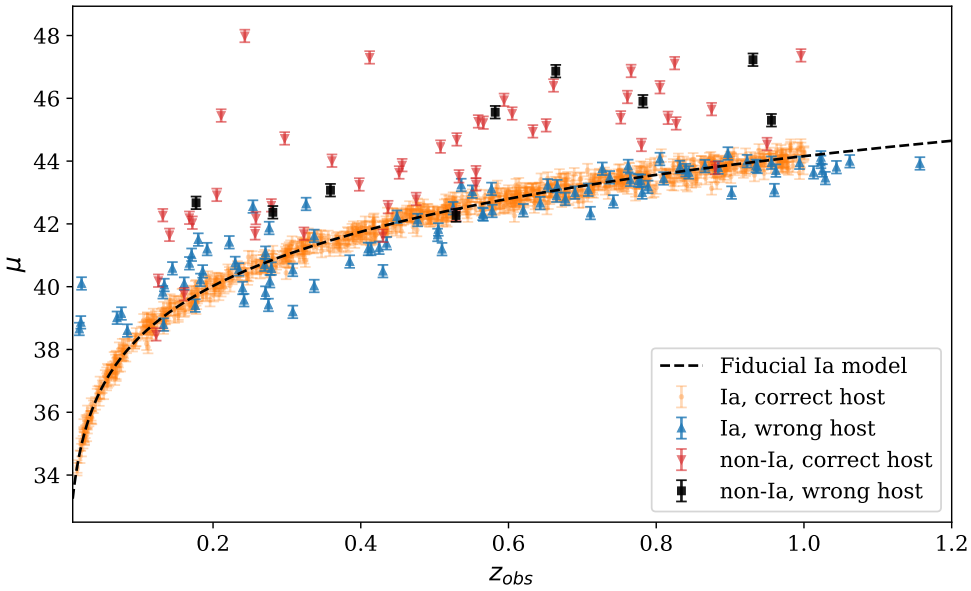

Figure 6.2: Distance moduli for the 1000 SNe as described in the spectroscopic case with two types of contamination: ∼9% host galaxy mis-identification and ∼5% non-Ia contaminants. The black square datapoints represent SNe that have both the wrong host (and hence incorrect redshift) and are non-Ia. The fiducial Ia distance modulus is shown by the black dashed line. Figure 6.3 shows how zBEAMS is able to untangle both forms of contamination with little to no increase in error contour size, while applying the standard MCMC approach and ignoring contamination leads to significant biases.

The unbiased SNe data is generated with a dispersion of 0.2 mag. For the biased dataset, the host galaxies are assumed to have spectroscopically confirmed redshifts, but the supernova is observed using photometry, hence the supernova type is not known and it is not always clear which galaxy the supernova belongs to if multiple galaxies lie within a small angular distance of one another. Additionally, it is assumed that supernovae with $z < 0.1$ will be identified spectroscopically, and will therefore not have associated type uncertainty. Here, a 5% type misidentification is assumed where the non-Ia population is offset from the Ia population by 2 mag, and has a Gaussian dispersion of 1.5 mag (similar to [61]). A more realistic distribution can be used with the same method, provided the form is known. A 9% host misidentification is assumed [47], where the misidentified host redshift is drawn from a normal distribution $z \sim \mathcal{N}(z_{\text{true}}, 0.1^2)$.

Figure 6.2 shows the distance moduli of the contaminated SNe Ia dataset. The biased dataset is analysed using both the standard likelihood (which does not take

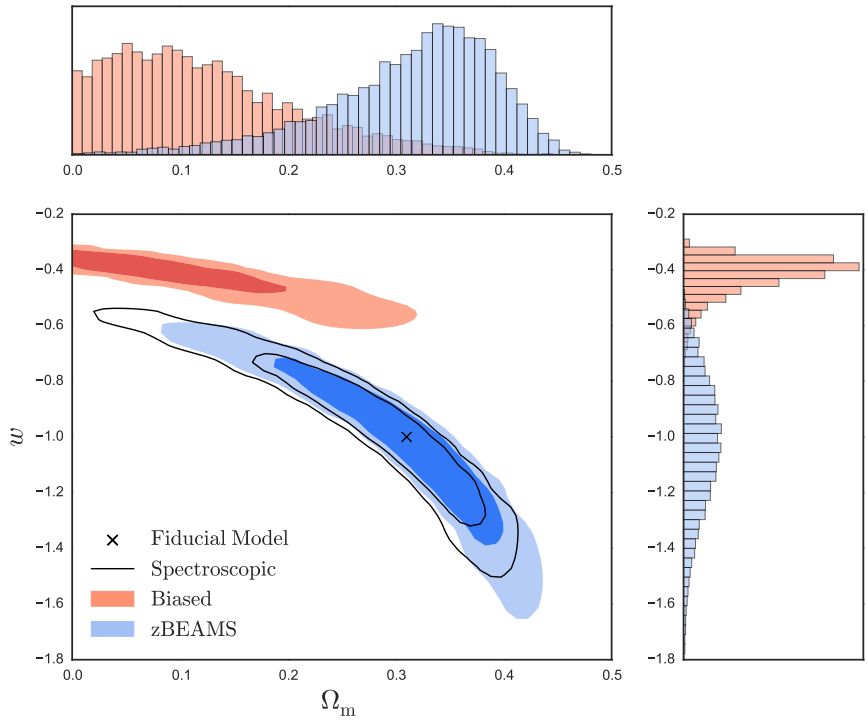

Figure 6.3: Contour plots for w and Ω_m showing the 68% and 95% credible intervals for the spectroscopic case. The black cross shows the fiducial model from which the data were generated. The black contours show the posterior distribution when there is no type or host uncertainty in the data. The red solid contours show the biased posterior distribution, i.e. when there is both type and host uncertainty, which is not accounted for in the likelihood. The blue solid contours show the posterior distribution when there is type can host uncertainty in the data, which is accounted for with the zBEAMS likelihood. As can be seen, the zBEAMS likelihood is able to handle both forms of contamination with little increase in computational complexity or ellipse area. Top and right panels show the marginalised 1D histograms for Ω_m and w respectively.

redshift error into account) and with the zBEAMS likelihood. The latter makes use of the zBEAMS posterior, shown in equation 6.32, to fully marginalise over both type and redshift uncertainties and thus produce unbiased cosmological estimates. In this analysis, the cosmological parameters, Ω_m, H_0 and w are solved for, while the parameters of the populations (such as the magnitude offset and standard deviation of the non-Ia population) are assumed to be known. However, it would be conceptually simple to solve for these simultaneously as done in earlier BEAMS papers [61; 79]. Here, this step is not included owing to time constraints and because it

is not central to the analysis, though these population parameters would need to be solved for in a realistic setting (i.e. one where the population parameters are not known). The marginalised posterior distribution is inferred for w and Ω_m for each of these three instances, and their respective contours are shown in figure 6.3. The three cases are: (i) the standard likelihood used on the unbiased dataset, (ii) the standard likelihood used on the biased dataset and (iii) the zBEAMS likelihood used on the biased dataset.

6.4.2 The Photometric Case

This section deals with the case where the redshift of the SN or host galaxy is obtained photometrically. In order to use zBEAMS in this case, the marginalisation over redshift must be performed numerically, as the integral in equation 6.18 has no analytic solution. We assume for this experiment that the redshift uncertainties are Gaussian distributed with a standard deviation of $0.04(1+z)$, though any more realistic distribution can be assumed with little change in complexity. For the photometric case, 1000 SNe are simulated from a redshift distribution given by $P(z) \sim ze^{-\beta z}$ over a redshift range of $z \in [0.015, 1.4]$, with $\beta = 3$. This distribution is the same as the prior distribution used to solve for the SNe redshifts. For completeness, one would need to solve for the value of β with the other parameters simultaneously. Here, since the value of β is known exactly, this extra degree of freedom is ignored. While this is not addressed here, this distribution could be extended to include modelling of instrumental selection effects, in addition to intrinsic supernova rate information. As before, Gaussian errors with dispersion 0.2 mag in the distance modulus are assumed. One can see the magnitude residuals for the observational redshift in figure 6.4 (main figure) and for the redshifts recovered by the zBEAMS analysis (inset figure).

As in the previous experiment, 3 cases are considered: (i) the standard MCMC analysis is done on the photometric dataset, (ii) the standard MCMC analysis is done when the redshifts are known exactly and (iii) the photometric redshifts are fit for using MCMC with block Metropolis-Hastings sampling. Figure 6.7 shows the resulting posterior distributions for these 3 cases. Note that result-(i) is clearly biased with respect to the solid black contours which are obtained using the true SNe redshifts (case-(ii)). The blue contours in figure 6.7 show the result when applying zBEAMS to the biased dataset (case-(iii)). Block Metropolis-Hastings was

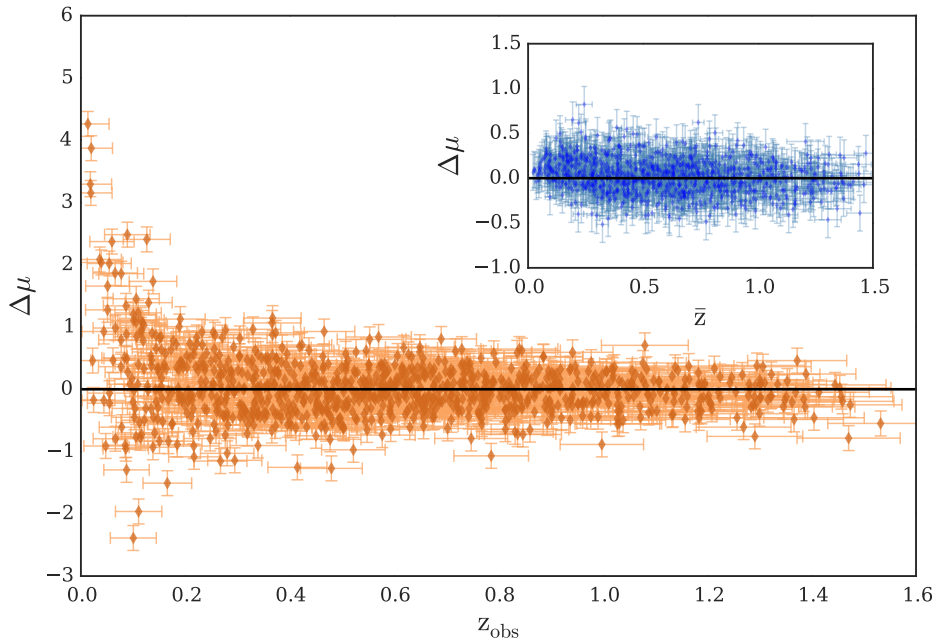

Figure 6.4: Photometric Hubble residuals for the 1000 SNe considered in the photometric case. They are drawn from the redshift distribution $P(z) \sim ze^{-3z}$, and have photometric redshift errors drawn from a Gaussian with mean 0 and $\sigma_z = 0.04(1+z)$. The main plot (gold) shows the residuals plotted against the observed redshift, $z_{\rm obs}$. The inset plot (blue) show the residuals plotted against the redshifts recovered from the MCMC chains, $\bar{z}$. The redshift uncertainties cause a large fraction of data, particularly at low redshift, to be more than 3σ away from the fiducial model (as can be seen in the main figure). The photometric redshifts are fit for simultaneously with the cosmological parameters, allowing zBEAMS to effectively put the SNe at the 'correct' redshifts (as shown in the inset figure, where the residuals are scattered about the fiducial model as would be expected given the Gaussian uncertainty on distance modulus).

used to fit for 1003 parameters simultaneously (3 cosmological parameters and 1000 redshifts), i.e., numerically computing the posterior given by equation 6.18. The block Metropolis-Hastings algorithm proceeds identically to the usual Metropolis-Hastings sampling algorithm, except that parameters are updated in blocks instead of updating all parameters every step. Here, the blocks (of variables to fit for at each step of the MCMC algorithm) used were the 3 cosmological parameters and a single redshift. Block sizes of 1-10 redshifts are however still suitable in terms of achieving convergence.

Figure 6.6 illustrates the same data returned by the MCMC analysis on the re-

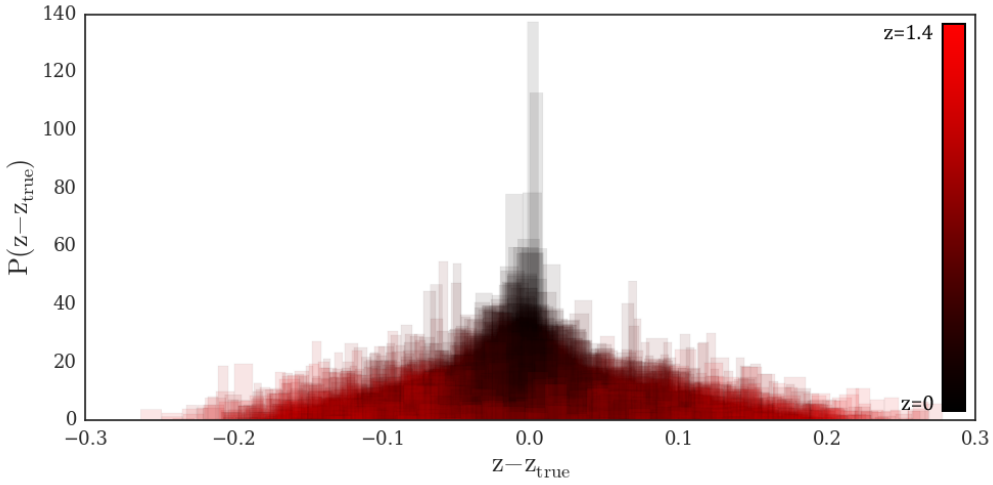

Figure 6.5: Stacked one-dimensional histograms for all 1000 redshifts from the zBEAMS analysis of the data in figure 6.4. For each supernova the histogram relative to the true redshift is shown, demonstrating that zBEAMS recovers, on average, the true redshift for each supernova. Each histogram is coloured by its redshift: black corresponding to low redshifts and red corresponding to high redshifts, showing that the recovered redshifts are less precise for increasing redshift, as expected due to the $(1 + z)$ scaling of the photometric redshift error and the flattening of the Hubble diagram.

covered redshift posteriors as are shown in figure 6.5. Figure 6.6, however, better illustrates the role that the Hubble diagram plays in placing constraints on the 'true' photometric redshifts. The top portion of the figure shows the redshift posteriors of the low-redshift SNe, and the lower portion shows the redshift posteriors of the higher-redshift SNe. The low-redshift SNe are are more easily constrained since they lie on the steeper part of the Hubble diagram, where a small change in redshift corresponds with a very poor model fit. The high-redshift SNe, by contrast, lie on the flatter part of the Hubble curve, where even fairly large changes in the modelled 'true' redshift do not impact severely on model fit.

The block size is found to have little impact on convergence of cosmological parameters in this case, as long as the blocks are small enough as not to significantly reduce the acceptance ratio. The block size does however impact on algorithm speed. It is assumed that each supernova redshift has a prior coming from the host galaxy (or from the supernova lightcurve itself) which is taken to be Gaussian centred on the observed redshift with standard deviation of $0.04(1 + z_{\rm obs})$. The prior on the overall SNIa redshift distribution was taken to be $P(z) = ze^{-\beta z}$, where the value of β is

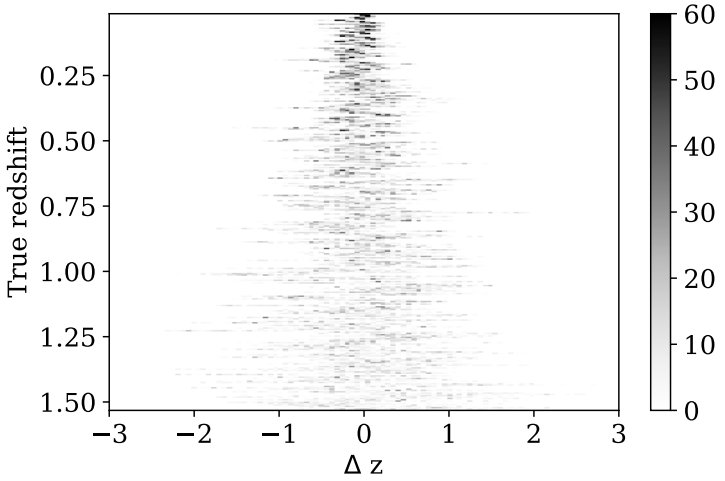

Figure 6.6: Photometric redshift posteriors for the 'true' redshift model parameters for the 1000 SNe considered in this case. Here, each row of pixels corresponds to a particular SN, with the highest probability density being represented in black, and a zero probability density in white. The x-axis (Δz) shows the difference between the 'true' redshift for each SN, and the modelled redshift fit for in the MCMC analysis. This figure illustrates that there appears to be no systematic bias in the recovered redshift posteriors, since they are by and large symmetrically distributed about a $\Delta z = 0$. The low-redshift SNe (shown towards the top of the figure) are more tightly constrained since they lie on the steeper part of the Hubble curve. The high-redshift SNe are more poorly constrained owing to the flattening of the Hubble diagram. Figure 6.7 illustrates that this does not have a significant impact on our ability to constrain cosmological parameters.

fixed at 3. In a case with real data, one would need to fit for these hyperparameters as well.

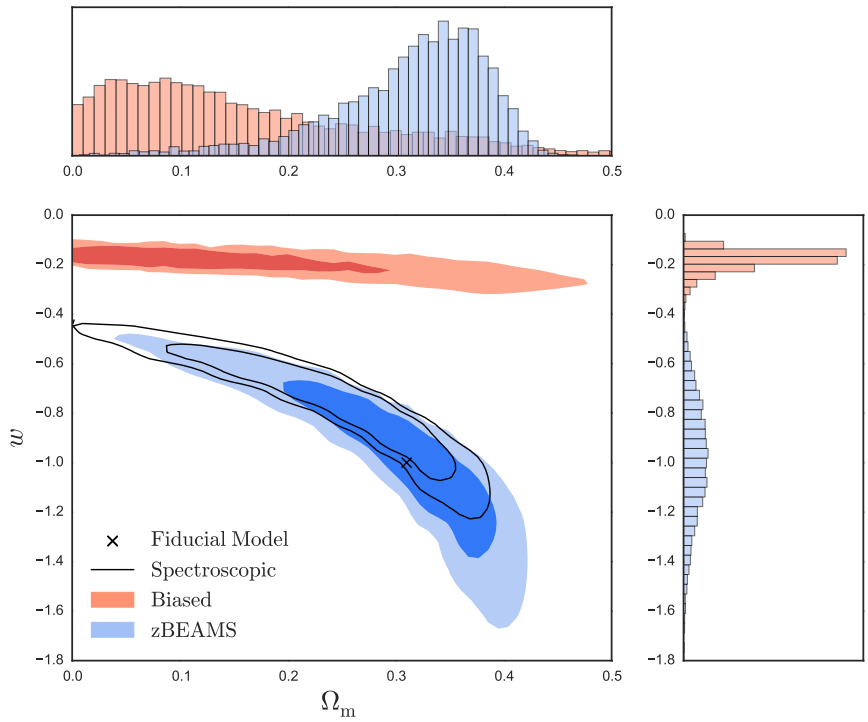

Figure 6.7: Contour plots for w and Ω_m showing the 68% and 95% credible intervals for the photometric case. The black cross shows the fiducial model from which the data were generated. The black contours show the posterior distribution if the host galaxy redshifts were spectroscopically confirmed. The red solid contours show the biased posterior distribution, i.e. when the photometric host galaxy redshifts are used, but not solved for. The blue solid contours show the posterior distribution found using zBEAMS (the 'true' host galaxy redshifts are solved for numerically). Top and right panels show the marginalised 1D histograms for Ω_m and w respectively.

Block Metropolis-Hastings recovers the true redshifts of the low-z supernovae ($\sigma_z = 0.02$ for $z < 0.25$) well, with worsening performance as the redshift increases. This is due to two effects: first the photometric redshift error scales with $(1 + z)$ and secondly the Hubble diagram progressively flattens out at $z > 0.25$ removing the signal that allows MCMC to constrain the redshift. This can be clearly seen in figure 6.5 where the stacked 1D histograms $z_{i,\text{chain}} - z_{i,\text{true}}$ for the 1000 SNIa are shown. It can be seen that while the error increases with redshift, the redshift estimates show no systematic bias. The marginalised posterior distributions for w

and Ω_m for each of these instances are represented in the contour plots shown in figure 6.7, which shows that zBEAMS recovers the correct cosmology, and contours, as desired.

Figure 6.4 illustrates the origin of the bias when using the standard likelihood. A large number of the data points are more than 3σ away from the model and some are over 20σ away. This is an artefact of using the wrong redshifts. The inset shows the same residual Hubble diagram when the data is instead plotted using the mean redshifts recovered from the MCMC chain for each redshift. Very few datapoints are now more than 2σ from the fiducial model, even at low redshifts where the excursions were the strongest.

This allows zBEAMS to produce unbiased cosmology contours that almost match the contour sizes of the perfect, spectroscopic case. Another approach to this same problem might be to significantly increase all the μ-error bars of the points to account for the redshift uncertainties. Unfortunately, doing so yields biased results for reasons discussed in section 6.3.1.2. Increasing the error bars further may reduce bias in the estimated posterior but only at the expense of significantly inflating the associated contours.

It should be noted that accurate sampling in realistic scenarios will not be trivial due to the high-dimensionality of the posterior (more unknown parameters than data points). Here, block Metropolis-Hastings was used to address this. Hamiltonian Monte Carlo [106] and Diffusive Nested Sampling [15] may also be viable solutions, and be well-suited to the high-dimensionality of this problem.

6.5 Concluding Remarks

This chapter has presented an introduction to some current challenges in the field of modern supernova cosmology. These challenges could be split into challenges with spectroscopic observations, and challenges with photometric observations, solutions to which are presented in sections 6.3 and 6.4.2 respectively. For both of these cases, zBEAMS gets contours that are close to the spectroscopic ideal, with little or no loss in the constraining power of the experiment, while naïve approaches give results that are biased at very significant levels.

These results indicate that the additional redshift uncertainties introduced in both

the spectroscopic and photometric cases, can be accounted for within the statistical formalism presented. For the photometric case, this statistical power does however come at the cost of computational complexity, which will be an issue for very large sets of photometrically observed SNe. This study finds that these redshift uncertainties do not prohibit our ability to do precision cosmology. Ignoring the biases that come from redshift uncertainties leads to catastrophic errors, but this can be dealt with in a rigorous way without loss of constraining power. A corollary of this is that large photometric surveys, such as those undertaken by the Vera C. Rubin telescope, will still offer the requisite data to contribute to precision cosmology, despite lack of spectra.

6.6 Future Work

This section lists a few natural extensions to the work presented in this chapter.

- The analysis presented in this chapter assumes that the probability of belonging to a given host galaxy γ, encoded in $P(\gamma)$, and the probability of being a given type of supernova, τ, encoded in $P(\tau|z)$, are known *a priori*. The formalism presented in this chapter can be extended to allow these to be partially known nuisance parameters that are estimated by available data.

- The zBEAMS formalism could be extended to include correlations with host galaxy information, such as host influence on Hubble residuals via stellar mass etc. [137; 85].

- The redshift distribution plays an important role for the photometric case. Specifically, if the individual true SN redshifts are being solved for, one needs to also fit for the redshift distribution. A much more complex model could be used than the one we assumed which could include some systematic effects and allow one to learn something about supernova rates.

- While the zBEAMS formalism was presented emphasising its generic nature for any data D, in the examples, D was taken to be the measured distance moduli. It would be useful to develop zBEAMS specifically for the case in which D is the set of lightcurve flux measurements in different bands; i.e. one step further back in the analysis chain.

- Realistic supernova surveys censor the true SN population because of the magnitude limits of the telescope and cuts performed during the analysis, resulting in Malmquist bias. Selection effects within a Bayesian framework have already been extensively covered in e.g. [127] and could be incorporated into the zBEAMS likelihood.

Bibliography

[1] Akhtar, N. and Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430.

[2] Astier, P., Guy, J., Regnault, N., Pain, R., Aubourg, E., Balam, D., Basa, S., Carlberg, R. G., Fabbro, S., Fouchez, D., Hook, I. M., Howell, D. A., Lafoux, H., Neill, J. D., Palanque-Delabrouille, N., Perrett, K., Pritchet, C. J., Rich, J., Sullivan, M., Taillet, R., Aldering, G., Antilogus, P., Arsenijevic, V., Balland, C., Baumont, S., Bronder, J., Courtois, H., Ellis, R. S., Filiol, M., Gonçalves, A. C., Goobar, A., Guide, D., Hardin, D., Lusset, V., Lidman, C., McMahon, R., Mouchet, M., Mourao, A., Perlmutter, S., Ripoche, P., Tao, C., and Walton, N. (2006). The Supernova Legacy Survey: measurement of Ω_M, Ω_Λ and w from the first year data set. *Astronomy and Astrophysics*, 447(1):31–48.

[3] Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press, USA.

[4] Barlow, R. E., Bartholomew, D. J., Bremner, J. M., and Brunk, H. D. (1972). Statistical Inference Under Order Restrictions: The Theory and Application of Isotonic Regression.

[5] Baron, D. and Poznanski, D. (2017). The weirdest SDSS galaxies: results from an outlier detection algorithm. *Monthly Notices of the RAS*, 465(4):4530–4555.

[6] Bassett, B. A. and Hlozek, R. (2010). *Baryon acoustic oscillations*, page 246.

[7] Bellhouse, D. R. (2004). The reverend thomas bayes, frs: A biography to celebrate the tercentenary of his birth. *Statist. Sci.*, 19(1):3–43.

[8] Betoule, M., Kessler, R., Guy, J., Mosher, J., Hardin, D., Biswas, R., Astier, P., El-Hage, P., Konig, M., Kuhlmann, S., Marriner, J., Pain, R., Regnault, N., Balland, C., Bassett, B. A., Brown, P. J., Campbell, H., Carlberg, R. G., Cellier-Holzem, F., Cinabro, D., Conley, A., D'Andrea, C. B., DePoy, D. L., Doi, M., Ellis, R. S., Fabbro, S., Filippenko, A. V., Foley, R. J., Frieman, J. A., Fouchez, D., Galbany, L., Goobar,

A., Gupta, R. R., Hill, G. J., Hlozek, R., Hogan, C. J., Hook, I. M., Howell, D. A., Jha, S. W., Le Guillou, L., Leloudas, G., Lidman, C., Marshall, J. L., Möller, A., Mourão, A. M., Neveu, J., Nichol, R., Olmstead, M. D., Palanque-Delabrouille, N., Perlmutter, S., Prieto, J. L., Pritchet, C. J., Richmond, M., Riess, A. G., Ruhlmann-Kleider, V., Sako, M., Schahmaneche, K., Schneider, D. P., Smith, M., Sollerman, J., Sullivan, M., Walton, N. A., and Wheeler, C. J. (2014). Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples. *Astronomy and Astrophysics*, 568:A22.

[9] Boone, K. (2019). Avocado: Photometric Classification of Astronomical Transients with Gaussian Process Augmentation. *Astronomical Journal*, 158(6):257.

[10] Botchkarev, A. (2018). Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology. *arXiv e-prints*, page arXiv:1809.03006.

[11] Branch, D. and Tammann, G. A. (1992). Type ia supernovae as standard candles. *Annual Review of Astronomy and Astrophysics*, 30(1):359–389.

[12] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

[13] Breiman, L. and Schapire, E. (2001). Random forests. In *Machine Learning*, pages 5–32.

[14] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104.

[15] Brewer, B. J., Pártay, L. B., and Csányi, G. (2011). Diffusive Nested Sampling. *G. Stat Comput*, 21:649.

[16] Brooks, S. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *J. Comput. Graphi. Stat.*, 7:434–455.

[17] Brout, D., Sako, M., Scolnic, D., Kessler, R., D'Andrea, C. B., Davis, T. M., Hinton, S. R., Kim, A. G., Lasker, J., Macaulay, E., Möller, A., Nichol, R. C., Smith, M., Sullivan, M., Wolf, R. C., Allam, S., Bassett, B. A., Brown, P., Castander, F. J., Childress, M., Foley, R. J., Galbany, L., Herner, K., Kasai, E., March, M., Morganson, E., Nugent, P., Pan, Y.-C., Thomas, R. C., Tucker, B. E., Wester, W., Abbott, T. M. C., Annis, J., Avila, S., Bertin, E., Brooks, D., Burke, D. L., Rosell, A. C., Kind, M. C., Carretero, J., Crocce, M., Cunha, C. E., da Costa, L. N., Davis, C., Vicente, J. D., Desai, S., Diehl, H. T., Doel, P., Eifler, T. F., Flaugher, B., Fosalba, P., Frieman, J.,

García-Bellido, J., Gaztanaga, E., Gerdes, D. W., Goldstein, D. A., Gruen, D., Gruendl, R. A., Gschwend, J., Gutierrez, G., Hartley, W. G., Hollowood, D. L., Honscheid, K., James, D. J., Kuehn, K., Kuropatkin, N., Lahav, O., Li, T. S., Lima, M., Marshall, J. L., Martini, P., Miquel, R., Nord, B., Plazas, A. A., Roodman, A., Rykoff, E. S., Sanchez, E., Scarpine, V., Schindler, R., Schubnell, M., Serrano, S., Sevilla-Noarbe, I., Soares-Santos, M., Sobreira, F., Suchyta, E., Swanson, M. E. C., Tarle, G., Thomas, D., Tucker, D. L., Walker, A. R., Yanny, B., and and, Y. Z. (2019). First cosmology results using type ia supernovae from the dark energy survey: Photometric pipeline and light-curve data release. *The Astrophysical Journal*, 874(1):106.

[18] Campbell, H., D'Andrea, C. B., Nichol, R. C., Sako, M., Smith, M., Lampeitl, H., Olmstead, M. D., Bassett, B., Biswas, R., Brown, P., Cinabro, D., Dawson, K. S., Dilday, B., Foley, R. J., Frieman, J. A., Garnavich, P., Hlozek, R., Jha, S. W., Kuhlmann, S., Kunz, M., Marriner, J., Miquel, R., Richmond, M., Riess, A., Schneider, D. P., Sollerman, J., Taylor, M., and Zhao, G.-B. (2013). Cosmology with Photometrically Classified Type Ia Supernovae from the SDSS-II Supernova Survey. *Astrophysical Journal*, 763:88.

[19] Casella, G. and George, E. I. (1992). Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174.

[20] Chambers, K. C., Magnier, E. A., Metcalfe, N., Flewelling, H. A., Huber, M. E., Waters, C. Z., Denneau, L., Draper, P. W., Farrow, D., Finkbeiner, D. P., Holmberg, C., Koppenhoefer, J., Price, P. A., Saglia, R. P., Schlafly, E. F., Smartt, S. J., Sweeney, W., Wainscoat, R. J., Burgett, W. S., Grav, T., Heasley, J. N., Hodapp, K. W., Jedicke, R., Kaiser, N., Kudritzki, R.-P., Luppino, G. A., Lupton, R. H., Monet, D. G., Morgan, J. S., Onaka, P. M., Stubbs, C. W., Tonry, J. L., Banados, E., Bell, E. F., Bender, R., Bernard, E. J., Botticella, M. T., Casertano, S., Chastel, S., Chen, W.-P., Chen, X., Cole, S., Deacon, N., Frenk, C., Fitzsimmons, A., Gezari, S., Goessl, C., Goggia, T., Goldman, B., Grebel, E. K., Hambly, N. C., Hasinger, G., Heavens, A. F., Heckman, T. M., Henderson, R., Henning, T., Holman, M., Hopp, U., Ip, W.-H., Isani, S., Keyes, C. D., Koekemoer, A., Kotak, R., Long, K. S., Lucey, J. R., Liu, M., Martin, N. F., McLean, B., Morganson, E., Murphy, D. N. A., Nieto-Santisteban, M. A., Norberg, P., Peacock, J. A., Pier, E. A., Postman, M., Primak, N., Rae, C., Rest, A., Riess, A., Riffeser, A., Rix, H. W., Roser, S., Schilbach, E., Schultz, A. S. B., Scolnic, D., Szalay, A., Seitz, S., Shiao, B., Small, E., Smith, K. W., Soderblom, D., Taylor, A. N., Thakar, A. R., Thiel, J., Thilker, D., Urata, Y., Valenti, J., Walter, F., Watters, S. P., Werner, S., White, R., Wood-Vasey, W. M., and Wyse, R. (2016). The Pan-STARRS1 Surveys. *ArXiv e-prints*.

[21] Chan, J. and Jeliazkov, I. (2009). Mcmc estimation of restricted covariance matrices. *Comput. Graph. Statist.*, 18.

[22] Chand, D., Gupta, S., and Kavati, I. (2020). Computer Vision based Accident Detection for Autonomous Vehicles. *arXiv e-prints*, page arXiv:2012.10870.

[23] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3).

[24] Cheeseman, P. and Stutz, J. (1996). Advances in knowledge discovery and data mining. chapter Bayesian Classification (AutoClass): Theory and Results, pages 153–180. American Association for Artificial Intelligence, Menlo Park, CA, USA.

[25] Chib, S. and Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335.

[26] Connolly, N. and Connolly, B. (2009). A Bayesian Approach to Classifying Supernovae With Color. *ArXiv e-prints*.

[27] Crawford, E., Norris, R. P., and Polsterer, K. (2017). WTF? Discovering the Unexpected in Next-Generation Radio Continuum Surveys. In Lorente, N. P. F., Shortridge, K., and Wayth, R., editors, *Astronomical Data Analysis Software and Systems XXV*, volume 512 of *Astronomical Society of the Pacific Conference Series*, page 109.

[28] Davenport, T., Guha, A., Grewal, D., and Bressgott, T. (2020). How artificial intelligence will change the future of marketing. *Journal of the Academy of Marketing Science*, 48(1):24–42.

[29] Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 233–240, New York, NY, USA. ACM.

[30] Delvenne, J.-C., Yaliraki, S. N., and Barahona, M. (2010). Stability of graph communities across time scales. *Proceedings of the National Academy of Sciences*, 107(29):12755–12760.

[31] Denison, D. G., Holmes, C. C., Mallick, B. K., and Smith, A. F. (2002). *Bayesian methods for nonlinear classification and regression*, volume 386. John Wiley & Sons.

[32] Djorgovski, S. G., Mahabal, A. A., Donalek, C., Graham, M. J., Drake, A. J., Turmon, M., and Fuchs, T. (2014). Automated Real-Time Classification and Decision Making in Massive Data Streams from Synoptic Sky Surveys. *arXiv e-prints*, page arXiv:1407.3502.

[33] Domingos, P. and Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2-3):103–130.

[34] Donalek, C., Arun Kumar, A., Djorgovski, S. G., Mahabal, A. A., Graham, M. J., Fuchs, T. J., Turmon, M. J., Sajeeth Philip, N., Yang, M. T.-C., and Longo, G. (2013). Feature Selection Strategies for Classifying High Dimensional Astronomical Data Sets. *arXiv e-prints*, page arXiv:1310.1976.

[35] Fawzi, A., Moosavi-Dezfooli, S.-M., Frossard, P., and Soatto, S. (2017). Classification regions of deep neural networks. *ArXiv e-prints*.

[36] Fridman, L., Brown, D. E., Glazer, M., Angell, W., Dodd, S., Jenik, B., Terwilliger, J., Patsekin, A., Kindelsberger, J., Ding, L., Seaman, S., Mehler, A., Sipperley, A., Pettinato, A., Seppelt, B., Angell, L., Mehler, B., and Reimer, B. (2017). MIT Advanced Vehicle Technology Study: Large-Scale Naturalistic Driving Study of Driver Behavior and Interaction with Automation. *arXiv e-prints*, page arXiv:1711.06976.

[37] Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

[38] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd ed. edition.

[39] Gelman, A. and Hill, J. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press.

[40] Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statist. Sci.*, 7(4):457–472.

[41] Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., and Modrák, M. (2020). Bayesian Workflow. *arXiv e-prints*, page arXiv:2011.01808.

[42] Ghosh, M. (2011). Objective Priors: An Introduction for Frequentists. *arXiv e-prints*, page arXiv:1108.2120.

[43] Goldhaber, G. (2009). The Acceleration of the Expansion of the Universe: A Brief Early History of the Supernova Cosmology Project (SCP). In Cline, D. B., editor, *Sources and Detection of Dark Matter and Dark Energy in the Universe*, volume 1166 of *American Institute of Physics Conference Series*, pages 53–72.

[44] Grace, K., Salvatier, J., Dafoe, A., Zhang, B., and Evans, O. (2018). Viewpoint: When Will AI Exceed Human Performance? Evidence from AI Experts. *Journal of Artificial Intelligence Research*, 62:729–754.

[45] Greene, D., Cunningham, P., and Mayer, R. (2008). *Unsupervised Learning and Clustering*, pages 51–90. Springer Berlin Heidelberg, Berlin, Heidelberg.

[46] Gull, S. F. (1989). *Bayesian Data Analysis: Straight-line fitting*, pages 511–518. Springer Netherlands, Dordrecht.

[47] Gupta, R. R., Kuhlmann, S., Kovacs, E., Spinka, H., Kessler, R., Goldstein, D. A., Liotine, C., Pomian, K., D'Andrea, C. B., Sullivan, M., Carretero, J., Castander, F. J., Nichol, R. C., Finley, D. A., Fischer, J. A., Foley, R. J., Kim, A. G., Papadopoulos, A., Sako, M., Scolnic, D. M., Smith, M., Tucker, B. E., Uddin, S., Wolf, R. C., Yuan, F., Abbott, T. M. C., Abdalla, F. B., Benoit-Lévy, A., Bertin, E., Brooks, D., Carnero Rosell, A., Carrasco Kind, M., Cunha, C. E., da Costa, L. N., Desai, S., Doel, P., Eifler, T. F., Evrard, A. E., Flaugher, B., Fosalba, P., Gaztañaga, E., Gruen, D., Gruendl, R., James, D. J., Kuehn, K., Kuropatkin, N., Maia, M. A. G., Marshall, J. L., Miquel, R., Plazas, A. A., Romer, A. K., Sánchez, E., Schubnell, M., Sevilla-Noarbe, I., Sobreira, F., Suchyta, E., Swanson, M. E. C., Tarle, G., Walker, A. R., and Wester, W. (2016). Host Galaxy Identification for Supernova Surveys. *Astronomical Journal*, 152(6):154.

[48] Guy, J., Astier, P., Baumont, S., Hardin, D., Pain, R., Regnault, N., Basa, S., Carlberg, R. G., Conley, A., Fabbro, S., Fouchez, D., Hook, I. M., Howell, D. A., Perrett, K., Pritchet, C. J., Rich, J., Sullivan, M., Antilogus, P., Aubourg, E., Bazin, G., Bronder, J., Filiol, M., Palanque-Delabrouille, N., Ripoche, P., and Ruhlmann-Kleider, V. (2007). SALT2: using distant supernovae to improve the use of type Ia supernovae as distance indicators. *Astronomy and Astrophysics*, 466(1):11–21.

[49] Haario, H., Laine, M., Mira, A., and Saksman, E. (2006). Dram: Efficient adaptive mcmc. *Statistics and Computing*, 16(4):339–354.

[50] Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive metropolis algorithm. *Bernoulli*, 7(2):223–242.

[51] Haario, H., Saksman, E., and Tamminen, J. (2005). Componentwise adaptation for high dimensional mcmc. *Computational Statistics*, 20(2):265–273.

[52] Halaweh, M. (2018). Viewpoint: Artificial Intelligence Government (Gov. 3.0): The UAE Leading Model. *Journal of Artificial Intelligence Research*, 62:269–272.

[53] Hand, D. J. and Till, R. J. (2001). A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2):171–186.

[54] Hansen, N. and Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 312–317.

[55] Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2):159–195.

[56] Hariri, S., Carrasco Kind, M., and Brunner, R. J. (2019). Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

[57] Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2004). The elements of statistical learning: Data mining, inference, and prediction. *Math. Intell.*, 27:83–85.

[58] Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

[59] Heavens, A. F., Seikel, M., Nord, B. D., Aich, M., Bouffanais, Y., Bassett, B. A., and Hobson, M. P. (2014). Generalized Fisher matrices. *Mon. Not. Roy. Astron. Soc.*, 445(2):1687–1693.

[60] Hložek, R., Ponder, K. A., Malz, A. I., Dai, M., Narayan, G., Ishida, E. E. O., Allam, T., J., Bahmanyar, A., Biswas, R., Galbany, L., Jha, S. W., Jones, D. O., Kessler, R., Lochner, M., Mahabal, A. A., Mandel, K. S., Martínez-Galarza, J. R., McEwen, J. D., Muthukrishna, D., Peiris, H. V., Peters, C. M., and Setzer, C. N. (2020). Results of the Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC). *arXiv e-prints*, page arXiv:2012.12392.

[61] Hlozek, R., Kunz, M., Bassett, B., Smith, M., Newling, J., Varughese, M., Kessler, R., Bernstein, J. P., Campbell, H., Dilday, B., Falck, B., Frieman, J., Kuhlmann, S., Lampeitl, H., Marriner, J., Nichol, R. C., Riess, A. G., Sako, M., and Schneider, D. P. (2012). Photometric Supernova Cosmology with BEAMS and SDSS-II. *Astrophysical Journal*, 752(2):79.

[62] Ho, T. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, page 278, Los Alamitos, CA, USA. IEEE Computer Society.

[63] Hocking, A., Geach, J. E., Davey, N., and Sun, Y. (2015). Teaching a machine to see: unsupervised image segmentation and categorisation using growing neural gas and hierarchical clustering. *arXiv e-prints*, page arXiv:1507.01589.

[64] Hsiao, E. Y., Conley, A., Howell, D. A., Sullivan, M., Pritchet, C. J., Carlberg, R. G., Nugent, P. E., and Phillips, M. M. (2007). K-Corrections and Spectral Templates of Type Ia Supernovae. *Astrophysical Journal*, 663(2):1187–1200.

[65] Ishida, E. E. O., Kornilov, M. V., Malanchev, K. L., Pruzhinskaya, M. V., Volnova, A. A., Korolev, V. S., Mondon, F., Sreejith, S., Malancheva, A., and Das, S. (2019). Active Anomaly Detection for time-domain discoveries. *arXiv e-prints*, page arXiv:1909.13260.

[66] Ivezić, Ž., Kahn, S. M., Tyson, J. A., Abel, B., Acosta, E., Allsman, R., Alonso, D., AlSayyad, Y., Anderson, S. F., Andrew, J., Angel, J. R. P., Angeli, G. Z., Ansari, R., Antilogus, P., Araujo, C., Armstrong, R., Arndt, K. T., Astier, P., Aubourg, É., Auza, N., Axelrod, T. S., Bard, D. J., Barr, J. D., Barrau, A., Bartlett, J. G., Bauer, A. E., Bauman, B. J., Baumont, S., Bechtol, E., Bechtol, K., Becker, A. C., Becla, J., Beldica, C., Bellavia, S., Bianco, F. B., Biswas, R., Blanc, G., Blazek, J., Blandford, R. D., Bloom, J. S., Bogart, J., Bond, T. W., Booth, M. T., Borgland, A. W., Borne, K., Bosch, J. F., Boutigny, D., Brackett, C. A., Bradshaw, A., Brandt, W. N., Brown, M. E., Bullock, J. S., Burchat, P., Burke, D. L., Cagnoli, G., Calabrese, D., Callahan, S., Callen, A. L., Carlin, J. L., Carlson, E. L., Chandrasekharan, S., Charles-Emerson, G., Chesley, S., Cheu, E. C., Chiang, H.-F., Chiang, J., Chirino, C., Chow, D., Ciardi, D. R., Claver, C. F., Cohen-Tanugi, J., Cockrum, J. J., Coles, R., Connolly, A. J., Cook, K. H., Cooray, A., Covey, K. R., Cribbs, C., Cui, W., Cutri, R., Daly, P. N., Daniel, S. F., Daruich, F., Daubard, G., Daues, G., Dawson, W., Delgado, F., Dellapenna, A., de Peyster, R., de Val-Borro, M., Digel, S. W., Doherty, P., Dubois, R., Dubois-Felsmann, G. P., Durech, J., Economou, F., Eifler, T., Eracleous, M., Emmons, B. L., Neto, A. F., Ferguson, H., Figueroa, E., Fisher-Levine, M., Focke, W., Foss, M. D., Frank, J., Freemon, M. D., Gangler, E., Gawiser, E., Geary, J. C., Gee, P., Geha, M., Gessner, C. J. B., Gibson, R. R., Gilmore, D. K., Glanzman, T., Glick, W., Goldina, T., Goldstein, D. A., Goodenow, I., Graham, M. L., Gressler, W. J., Gris, P., Guy, L. P., Guyonnet, A., Haller, G., Harris, R., Hascall, P. A., Haupt, J., Hernandez, F., Herrmann, S., Hileman, E., Hoblitt, J., Hodgson, J. A., Hogan, C., Howard, J. D., Huang, D., Huffer, M. E., Ingraham, P., Innes, W. R., Jacoby, S. H., Jain, B., Jammes, F., Jee, J., Jenness, T., Jernigan, G., Jevremović, D., Johns, K., Johnson, A. S., Johnson, M. W. G., Jones, R. L., Juramy-Gilles, C., Jurić, M., Kalirai, J. S., Kallivayalil, N. J., Kalmbach, B., Kantor, J. P., Karst, P., Kasliwal, M. M., Kelly, H., Kessler, R., Kinnison, V., Kirkby, D., Knox, L., Kotov, I. V., Krabbendam, V. L., Krughoff, K. S., Kubánek, P., Kuczewski, J., Kulkarni, S., Ku, J., Kurita, N. R., Lage, C. S., Lambert, R., Lange, T., Langton, J. B., Guillou, L. L., Levine, D., Liang, M., Lim, K.-T., Lintott, C. J., Long, K. E., Lopez, M., Lotz, P. J., Lupton, R. H., Lust, N. B., MacArthur, L. A., Mahabal,

A., Mandelbaum, R., Markiewicz, T. W., Marsh, D. S., Marshall, P. J., Marshall, S., May, M., McKercher, R., McQueen, M., Meyers, J., Migliore, M., Miller, M., Mills, D. J., Miraval, C., Moeyens, J., Moolekamp, F. E., Monet, D. G., Moniez, M., Monkewitz, S., Montgomery, C., Morrison, C. B., Mueller, F., Muller, G. P., Arancibia, F. M., Neill, D. R., Newbry, S. P., Nief, J.-Y., Nomerotski, A., Nordby, M., O'Connor, P., Oliver, J., Olivier, S. S., Olsen, K., O'Mullane, W., Ortiz, S., Osier, S., Owen, R. E., Pain, R., Palecek, P. E., Parejko, J. K., Parsons, J. B., Pease, N. M., Peterson, J. M., Peterson, J. R., Petravick, D. L., Petrick, M. E. L., Petry, C. E., Pierfederici, F., Pietrowicz, S., Pike, R., Pinto, P. A., Plante, R., Plate, S., Plutchak, J. P., Price, P. A., Prouza, M., Radeka, V., Rajagopal, J., Rasmussen, A. P., Regnault, N., Reil, K. A., Reiss, D. J., Reuter, M. A., Ridgway, S. T., Riot, V. J., Ritz, S., Robinson, S., Roby, W., Roodman, A., Rosing, W., Roucelle, C., Rumore, M. R., Russo, S., Saha, A., Sassolas, B., Schalk, T. L., Schellart, P., Schindler, R. H., Schmidt, S., Schneider, D. P., Schneider, M. D., Schoening, W., Schumacher, G., Schwamb, M. E., Sebag, J., Selvy, B., Sembroski, G. H., Seppala, L. G., Serio, A., Serrano, E., Shaw, R. A., Shipsey, I., Sick, J., Silvestri, N., Slater, C. T., Smith, J. A., Smith, R. C., Sobhani, S., Soldahl, C., Storrie-Lombardi, L., Stover, E., Strauss, M. A., Street, R. A., Stubbs, C. W., Sullivan, I. S., Sweeney, D., Swinbank, J. D., Szalay, A., Takacs, P., Tether, S. A., Thaler, J. J., Thayer, J. G., Thomas, S., Thornton, A. J., Thukral, V., Tice, J., Trilling, D. E., Turri, M., Berg, R. V., Berk, D. V., Vetter, K., Virieux, F., Vucina, T., Wahl, W., Walkowicz, L., Walsh, B., Walter, C. W., Wang, D. L., Wang, S.-Y., Warner, M., Wiecha, O., Willman, B., Winters, S. E., Wittman, D., Wolff, S. C., Wood-Vasey, W. M., Wu, X., Xin, B., Yoachim, P., and Zhan, H. (2019). LSST: From science drivers to reference design and anticipated data products. *The Astrophysical Journal*, 873(2):111.

[67] Jaynes, E. T. (2003). *Probability Theory: The Logic of Science*. Cambridge University Press.

[68] Jennings, E., Wolf, R., and Sako, M. (2016). A new approach for obtaining cosmological constraints from Type Ia Supernovae using Approximate Bayesian Computation. *ArXiv e-prints*.

[69] Jones, D. O., Scolnic, D. M., Foley, R. J., Rest, A., Kessler, R., Challis, P. M., Chambers, K. C., Coulter, D. A., Dettman, K. G., Foley, M. M., Huber, M. E., Jha, S. W., Johnson, E., Kilpatrick, C. D., Kirshner, R. P., Manuel, J., Narayan, G., Pan, Y. C., Riess, A. G., Schultz, A. S. B., Siebert, M. R., Berger, E., Chornock, R., Flewelling, H., Magnier, E. A., Smartt, S. J., Smith, K. W., Wainscoat, R. J., Waters, C., and Willman, M. (2019). The Foundation Supernova Survey: Measuring Cosmological Parameters with Supernovae from a Single Telescope. *Astrophysical Journal*, 881(1):19.

[70] Kendall, A. and Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *arXiv e-prints*, page arXiv:1703.04977.

[71] Kessler, R., Becker, A. C., Cinabro, D., Vanderplas, J., Frieman, J. A., Marriner, J., Davis, T. M., Dilday, B., Holtzman, J., Jha, S. W., Lampeitl, H., Sako, M., Smith, M., Zheng, C., Nichol, R. C., Bassett, B., Bender, R., Depoy, D. L., Doi, M., Elson, E., Filippenko, A. V., Foley, R. J., Garnavich, P. M., Hopp, U., Ihara, Y., Ketzeback, W., Kollatschny, W., Konishi, K., Marshall, J. L., McMillan, R. J., Miknaitis, G., Morokuma, T., Mörtsell, E., Pan, K., Prieto, J. L., Richmond, M. W., Riess, A. G., Romani, R., Schneider, D. P., Sollerman, J., Takanashi, N., Tokita, K., van der Heyden, K., Wheeler, J. C., Yasuda, N., and York, D. (2009). First-Year Sloan Digital Sky Survey-II Supernova Results: Hubble Diagram and Cosmological Parameters. *Astrophysical Journal, Supplement*, 185(1):32–84.

[72] Kessler, R., Cinabro, D., Bassett, B., Dilday, B., Frieman, J. A., Garnavich, P. M., Jha, S., Marriner, J., Nichol, R. C., Sako, M., Smith, M., Bernstein, J. P., Bizyaev, D., Goobar, A., Kuhlmann, S., Schneider, D. P., and Stritzinger, M. (2010). Photometric Estimates of Redshifts and Distance Moduli for Type Ia Supernovae. *Astrophysical Journal*, 717:40–57.

[73] Kessler, R., Narayan, G., Avelino, A., Bachelet, E., Biswas, R., Brown, P. J., Chernoff, D. F., Connolly, A. J., Dai, M., Daniel, S., Di Stefano, R., Drout, M. R., Galbany, L., González-Gaitán, S., Graham, M. L., Hložek, R., Ishida, E. E. O., Guillochon, J., Jha, S. W., Jones, D. O., Mandel, K. S., Muthukrishna, D., O'Grady, A., Peters, C. M., Pierel, J. R., Ponder, K. A., Prša, A., Rodney, S., Villar, V. A., LSST Dark Energy Science Collaboration, and Transient and Variable Stars Science Collaboration (2019). Models and Simulations for the Photometric LSST Astronomical Time Series Classification Challenge (PLAsTiCC). *Publications of the ASP*, 131(1003):094501.

[74] Kessler, R. and Scolnic, D. (2017). Correcting Type Ia Supernova Distances for Selection Biases and Contamination in Photometrically Identified Samples. *Astrophysical Journal*, 836:56.

[75] Kim, A. and Linder, E. (2011). Correlated Supernova Systematics and Ground Based Surveys. *JCAP*, 06(020).

[76] Kim, A. G., Linder, E. V., Miquel, R., and Mostek, N. (2004). Effects of systematic uncertainties on the supernova determination of cosmological parameters. *Monthly Notices of the Royal Astronomical Society*, 347(3):909–920.

[77] Kjaerulff, U. B. and Madsen, A. L. (2010). *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*. Springer Publishing Company, Incorporated, 1st edition.

[78] Kjellström, G. (1991). On the efficiency of gaussian adaptation. *Journal of Optimization Theory and Applications*, 71(3):589–597.

[79] Knights, M., Bassett, B. A., Varughese, M., Hlozek, R., Kunz, M., Smith, M., and Newling, J. (2013). Extending BEAMS to incorporate correlated systematic uncertainties. *Journal of Cosmology and Astroparticle Physics*, 1:039.

[80] Koch, K.-R. (2007). *Introduction to Bayesian Statistics*. Springer Publishing Company, Incorporated, 2nd edition.

[81] Krakowski, T., Małek, K., Bilicki, M., Pollo, A., Kurcz, A., and Krupa, M. (2016). Machine-learning identification of galaxies in the WISE × SuperCOSMOS all-sky catalogue. *Astronomy and Astrophysics*, 596:A39.

[82] Kuhn, M. and Johnson, K. (2013). *Applied predictive modeling*. Springer, New York, NY.

[83] Kunz, M., Bassett, B. A., and Hlozek, R. A. (2007). Bayesian estimation applied to multiple species. *Phys. Rev. D*, 75(10):103508.

[84] Kyriakidis, I., Kukkonen, J., Karatzas, K., Papadourakis, G., and Ware, J. (2015). New statistical indices for evaluating model forecasting performance.

[85] Lampeitl, H., Smith, M., Nichol, R. C., Bassett, B., Cinabro, D., Dilday, B., Foley, R. J., Frieman, J. A., Garnavich, P. M., Goobar, A., Im, M., Jha, S. W., Marriner, J., Miquel, R., Nordin, J., Östman, L., Riess, A. G., Sako, M., Schneider, D. P., Sollerman, J., and Stritzinger, M. (2010). The Effect of Host Galaxies on Type Ia Supernovae in the SDSS-II Supernova Survey. *Astrophysical Journal*, 722:566–576.

[86] Laplace, P.-S. (2009). *Essai philosophique sur les probabilités*. Cambridge Library Collection - Mathematics. Cambridge University Press, 5 edition.

[87] Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 413–422, Washington, DC, USA. IEEE Computer Society.

[88] Lochner, M. and Bassett, B. A. (2020). Astronomaly: Personalised Active Anomaly Detection in Astronomical Data. *arXiv e-prints*, page arXiv:2010.11202.

[89] Lochner, M., Scolnic, D. M., Awan, H., Regnault, N., Gris, P., Mandelbaum, R., Gawiser, E., Almoubayyed, H., Setzer, C. N., Huber, S., Graham, M. L., Hložek, R., Biswas, R., Eifler, T., Rothchild, D., Allam, Tarek, J., Blazek, J., Chang, C., Collett, T., Goobar, A., Hook, I. M., Jarvis, M., Jha, S. W., Kim, A. G., Marshall, P., McEwen, J. D., Moniez, M., Newman, J. A., Peiris, H. V., Petrushevska, T., Rhodes, J., Sevilla-Noarbe, I., Slosar, A., Suyu, S. H., Tyson, J. A., and Yoachim, P. (2018). Optimizing the LSST Observing Strategy for Dark Energy Science: DESC Recommendations for the Wide-Fast-Deep Survey. *arXiv e-prints*, page arXiv:1812.00515.

[90] Louppe, G. (2014). Understanding Random Forests: From Theory to Practice. *arXiv e-prints*, page arXiv:1407.7502.

[91] LSST Science Collaboration, Abell, P. A., Allison, J., Anderson, S. F., Andrew, J. R., Angel, J. R. P., Armus, L., Arnett, D., Asztalos, S. J., Axelrod, T. S., and et al. (2009). LSST Science Book, Version 2.0. *ArXiv e-prints*.

[92] LSST Science Collaboration, Marshall, P., Anguita, T., Bianco, F. B., Bellm, E. C., Brandt, N., Clarkson, W., Connolly, A., Gawiser, E., Ivezic, Z., Jones, L., Lochner, M., Lund, M. B., Mahabal, A., Nidever, D., Olsen, K., Ridgway, S., Rhodes, J., Shemmer, O., Trilling, D., Vivas, K., Walkowicz, L., Willman, B., Yoachim, P., Anderson, S., Antilogus, P., Angus, R., Arcavi, I., Awan, H., Biswas, R., Bell, K. J., Bennett, D., Britt, C., Buzasi, D., Casetti-Dinescu, D. I., Chomiuk, L., Claver, C., Cook, K., Davenport, J., Debattista, V., Digel, S., Doctor, Z., Firth, R. E., Foley, R., Fong, W.-f., Galbany, L., Giampapa, M., Gizis, J. E., Graham, M. L., Grillmair, C., Gris, P., Haiman, Z., Hartigan, P., Hawley, S., Hlozek, R., Jha, S. W., Johns-Krull, C., Kanbur, S., Kalogera, V., Kashyap, V., Kasliwal, V., Kessler, R., Kim, A., Kurczynski, P., Lahav, O., Liu, M. C., Malz, A., Margutti, R., Matheson, T., McEwen, J. D., McGehee, P., Meibom, S., Meyers, J., Monet, D., Neilsen, E., Newman, J., O'Dowd, M., Peiris, H. V., Penny, M. T., Peters, C., Poleski, R., Ponder, K., Richards, G., Rho, J., Rubin, D., Schmidt, S., Schuhmann, R. L., Shporer, A., Slater, C., Smith, N., Soares-Santos, M., Stassun, K., Strader, J., Strauss, M., Street, R., Stubbs, C., Sullivan, M., Szkody, P., Trimble, V., Tyson, T., de Val-Borro, M., Valenti, S., Wagoner, R., Wood-Vasey, W. M., and Zauderer, B. A. (2017). Science-Driven Optimization of the LSST Observing Strategy. *arXiv e-prints*, page arXiv:1708.04058.

[93] Ma, C., Corasaniti, P.-S., and Bassett, B. A. (2016). Application of Bayesian graphs to SN Ia data analysis and compression. *Monthly Notices of the RAS*, 463:1651–1665.

[94] MacKay, D. J. C. (2002). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, USA.

[95] Malz, A. I., Hložek, R., Allam, T., J., Bahmanyar, A., Biswas, R., Dai, M., Galbany, L., Ishida, E. E. O., Jha, S. W., Jones, D. O., Kessler, R., Lochner, M., Mahabal, A. A., Mandel, K. S., Martínez-Galarza, J. R., McEwen, J. D., Muthukrishna, D., Narayan, G., Peiris, H., Peters, C. M., Ponder, K., Setzer, C. N., (the LSST Dark Energy Science Collaboration, LSST Transients, t., and Variable Stars Science Collaboration (2019). The Photometric LSST Astronomical Time-series Classification Challenge PLAsTiCC: Selection of a Performance Metric for Classification Probabilities Balancing Diverse Science Goals. *Astronomical Journal*, 158(5):171.

[96] Manco, G., Ritacco, E., Rullo, P., Gallucci, L., Astill, W., Kimber, D., and Antonelli, M. (2017). Fault detection and explanation through big data analysis on sensor streams. *Expert Syst. Appl.*, 87(C):141–156.

[97] Mandel, K. S., Wood-Vasey, W. M., Friedman, A. S., and Kirshner, R. P. (2009). Type Ia Supernova Light-Curve Inference: Hierarchical Bayesian Analysis in the Near-Infrared. *Astrophysical Journal*, 704:629–651.

[98] Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.

[99] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

[100] Meyn, S. and Tweedie, R. L. (2009). *Markov Chains and Stochastic Stability*. Cambridge University Press, USA, 2nd edition.

[101] Möller, A. and de Boissière, T. (2020). SuperNNova: an open-source framework for Bayesian, neural network-based supernova classification. *Monthly Notices of the RAS*, 491(3):4277–4293.

[102] Möller, A., Ruhlmann-Kleider, V., Leloup, C., Neveu, J., Palanque-Delabrouille, N., Rich, J., Carlberg, R., Lidman, C., and Pritchet, C. (2016). Photometric classification of type Ia supernovae in the SuperNova Legacy Survey with supervised learning. *Journal of Cosmology and Astroparticle Physics*, 12:008.

[103] Mueller, C. L. (2010). Exploring the common concepts of adaptive mcmc and covariance matrix adaptation schemes. In Auger, A., Shapiro, J. L., Whitley, L. D., and Witt, C., editors, *Theory of Evolutionary Algorithms*, number 10361 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[104] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

[105] Muthukrishna, D., Narayan, G., Mandel, K. S., Biswas, R., and Hložek, R. (2019). RAPID: Early Classification of Explosive Transients Using Deep Learning. *Publications of the ASP*, 131(1005):118002.

[106] Neal, R. M. (2012). MCMC using Hamiltonian dynamics. *ArXiv e-prints*.

[107] Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 625–632, New York, NY, USA. ACM.

[108] Olmstead, M. D., Brown, P. J., Sako, M., Bassett, B., Bizyaev, D., Brinkmann, J., Brownstein, J. R., Brewington, H., Campbell, H., D'Andrea, C. B., Dawson, K. S., Ebelke, G. L., Frieman, J. A., Galbany, L., Garnavich, P., Gupta, R. R., Hlozek, R., Jha, S. W., Kunz, M., Lampeitl, H., Malanushenko, E., Malanushenko, V., Marriner, J., Miquel, R., Montero-Dorta, A. D., Nichol, R. C., Oravetz, D. J., Pan, K., Schneider, D. P., Simmons, A. E., Smith, M., and Snedden, S. A. (2014). Host Galaxy Spectra and Consequences for Supernova Typing from the SDSS SN Survey. *Astronomical Journal*, 147:75.

[109] Olsen, W., Bera, M., Dubey, A., Kim, J., Wiśniowski, A., and Yadav, P. (2020). Hierarchical modelling of covid-19 death risk in india in the early phase of the pandemic. *The European Journal of Development Research*, 32(5):1476–1503.

[110] Pang, G., Shen, C., Cao, L., and van den Hengel, A. (2020). Deep Learning for Anomaly Detection: A Review. *arXiv e-prints*, page arXiv:2007.02500.

[111] Pasquet, J., Pasquet, J., Chaumont, M., and Fouchez, D. (2019). PELICAN: deeP architecturE for the LIght Curve ANalysis. *Astronomy and Astrophysics*, 627:A21.

[112] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[113] Perlmutter, S., Aldering, G., Goldhaber, G., Knop, R. A., Nugent, P., Castro, P. G., Deustua, S., Fabbro, S., Goobar, A., Groom, D. E., Hook, I. M., Kim, A. G., Kim, M. Y., Lee, J. C., Nunes, N. J., Pain, R., Pennypacker, C. R., Quimby, R., Lidman, C., Ellis, R. S., Irwin, M., McMahon, R. G., Ruiz-Lapuente, P., Walton, N., Schaefer, B., Boyle, B. J., Filippenko, A. V., Matheson, T., Fruchter, A. S., Panagia, N., Newberg,

H. J. M., Couch, W. J., and Project, T. S. C. (1999). Measurements of Ω and Λ from 42 High-Redshift Supernovae. *Astrophysical Journal*, 517(2):565–586.

[114] Philip, N. S., Mahabal, A., Abraham, S., Williams, R., Djorgovski, S. G., Drake, A., Donalek, C., and Graham, M. (2012). Classification by boosting differences in input vectors: an application to datasets from astronomy. In Prugniel, P. and Singh, H. P., editors, *Astronomical Society of India Conference Series*, volume 6 of *Astronomical Society of India Conference Series*, page 151.

[115] Planck Collaboration, Ade, P. A. R., Aghanim, N., Arnaud, M., Ashdown, M., Aumont, J., Baccigalupi, C., Banday, A. J., Barreiro, R. B., Bartlett, J. G., and et al. (2016). Planck 2015 results. XIII. Cosmological parameters. *Astronomy and Astrophysics*, 594:A13.

[116] Platt, J. (2000). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.*, 10.

[117] Prasath, S., Abu Alfeilat, H., Lasassmeh, O., Hassanat, A., and Tarawneh, A. (2017). Distance and similarity measures effect on the performance of k-nearest neighbor classifier - a review.

[118] Pruzhinskaya, M. V., Malanchev, K. L., Kornilov, M. V., Ishida, E. E. O., Mondon, F., Volnova, A. A., and Korolev, V. S. (2019). Anomaly detection in the Open Supernova Catalog. *Monthly Notices of the RAS*, 489(3):3591–3608.

[119] Rao, C. R. (1948). The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203.

[120] Reinhold, J. C., He, Y., Han, S., Chen, Y., Gao, D., Lee, J., Prince, J. L., and Carass, A. (2020). Finding novelty with uncertainty. *arXiv e-prints*, page arXiv:2002.04626.

[121] Riabiz, M., Chen, W., Cockayne, J., Swietach, P., Niederer, S. A., Mackey, L., and Oates, C. J. (2020). Optimal Thinning of MCMC Output. *arXiv e-prints*, page arXiv:2005.03952.

[122] Riess, A. G., Filippenko, A. V., Challis, P., Clocchiatti, A., Diercks, A., Garnavich, P. M., Gilliland, R. L., Hogan, C. J., Jha, S., Kirshner, R. P., Leibundgut, B., Phillips, M. M., Reiss, D., Schmidt, B. P., Schommer, R. A., Smith, R. C., Spyromilio, J., Stubbs, C., Suntzeff, N. B., and Tonry, J. (1998). Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant. *Astronomical Journal*, 116(3):1009–1038.

[123] Riggi, S., Ingallinera, A., Leto, P., Cavallaro, F., Bufano, F., Schillirò, F., Trigilio, C., Umana, G., Buemi, C. S., and Norris, R. P. (2016). Automated detection of extended sources in radio maps: progress from the SCORPIO survey. *Monthly Notices of the RAS*, 460(2):1486–1499.

[124] Roberts, E., Lochner, M., Fonseca, J., Bassett, B. A., Lablanche, P.-Y., and Agarwal, S. (2017). zBEAMS: a unified solution for supernova cosmology with redshift uncertainties. *Journal of Cosmology and Astroparticle Physics*, 2017(10):036.

[125] Roberts, G. and Smith, A. (1994). Simple conditions for the convergence of the gibbs sampler and metropolis-hastings algorithms. *Stochastic Processes and their Applications*, 49(2):207–216.

[126] Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351 – 367.

[127] Rubin, D., Aldering, G., Barbary, K., Boone, K., Chappell, G., Currie, M., Deustua, S., Fagrelius, P., Fruchter, A., Hayden, B., Lidman, C., Nordin, J., Perlmutter, S., Saunders, C., Sofiatti, C., and Supernova Cosmology Project, T. (2015). UNITY: Confronting Supernova Cosmology's Statistical and Systematic Uncertainties in a Unified Bayesian Framework. *Astrophysical Journal*, 813:137.

[128] Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., and Müller, K.-R. (2020). A Unifying Review of Deep and Shallow Anomaly Detection. *arXiv e-prints*, page arXiv:2009.11732.

[129] Sako, M., Bassett, B., Connolly, B., Dilday, B., Cambell, H., Frieman, J. A., Gladney, L., Kessler, R., Lampeitl, H., Marriner, J., Miquel, R., Nichol, R. C., Schneider, D. P., Smith, M., and Sollerman, J. (2011). Photometric Type Ia Supernova Candidates from the Three-year SDSS-II SN Survey Data. *The Astrophysical Journal*, 738:162.

[130] Santana, E. and Hotz, G. (2016). Learning a Driving Simulator. *arXiv e-prints*, page arXiv:1608.01230.

[131] Schmidt, J., Marques, M. R. G., Botti, S., and Marques, M. A. L. (2019). Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):83.

[132] Shariff, H., Jiao, X., Trotta, R., and van Dyk, D. A. (2016). BAHAMAS: New Analysis of Type Ia Supernovae Reveals Inconsistencies with Standard Cosmology. *Astrophysical Journal*, 827:1.

[133] Shchigolev, V. K. (2015). Calculating Luminosity Distance versus Redshift in FLRW Cosmology via Homotopy Perturbation Method. *arXiv e-prints*, page arXiv:1511.07459.

[134] Simpson, J. A., Weiner, E. S. C., and Oxford University Press (1989). *The Oxford English Dictionary*. Clarendon Press.

[135] Sodemann, A., Ross, M., and Borghetti, B. (2011). A review of anomaly detection in automated surveillance. *IEEE transactions on systems, man, and cybernetics, part C*, 42.

[136] Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

[137] Sullivan, M., Conley, A., Howell, D. A., Neill, J. D., Astier, P., Balland, C., Basa, S., Carlberg, R. G., Fouchez, D., Guy, J., Hardin, D., Hook, I. M., Pain, R., Palanque-Delabrouille, N., Perrett, K. M., Pritchet, C. J., Regnault, N., Rich, J., Ruhlmann-Kleider, V., Baumont, S., Hsiao, E., Kronborg, T., Lidman, C., Perlmutter, S., and Walker, E. S. (2010). The dependence of Type Ia Supernovae luminosities on their host galaxies. *Monthly Notices of the RAS*, 406:782–802.

[138] Tarik Altuncu, M., Yaliraki, S. N., and Barahona, M. (2018). Content-driven, unsupervised clustering of news articles through multiscale graph partitioning. *arXiv e-prints*, page arXiv:1808.01175.

[139] Teimoorinia, H., Ellison, S. L., and Patton, D. R. (2017). Pattern recognition in the ALFALFA.70 and Sloan Digital Sky Surveys: a catalogue of ∼500 000 H I gas fraction estimates based on artificial neural networks. *Monthly Notices of the RAS*, 464(4):3796–3811.

[140] Tsvetkov, D., Hristov, L., and Angelova-Slavova, R. (2013). On the convergence of the Metropolis-Hastings Markov chains. *arXiv e-prints*, page arXiv:1302.0654.

[141] Turek, D., de Valpine, P., Paciorek, C. J., and Anderson-Bergman, C. (2017). Automated parameter blocking for efficient markov chain monte carlo sampling. *Bayesian Anal.*, 12(2):465–490.

[142] Varughese, M. M., von Sachs, R., Stephanou, M., and Bassett, B. A. (2015). Non-parametric transient classification using adaptive wavelets. *Monthly Notices of the RAS*, 453(3):2848–2861.

[143] Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. (2020). Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of mcmc. *Bayesian Anal.* Advance publication.

[144] Wang, Y., Gjergo, E., and Kuhlmann, S. (2015). Analytic photometric redshift estimator for Type Ia supernovae from the Large Synoptic Survey Telescope. *Monthly Notices of the RAS*, 451:1955–1963.

[145] Webb, G. I. (2010). *Naïve Bayes*, pages 713–714. Springer US, Boston, MA.

[146] Webb, S., Lochner, M., Muthukrishna, D., Cooke, J., Flynn, C., Mahabal, A., Goode, S., Andreoni, I., Pritchard, T., and Abbott, T. M. C. (2020). Unsupervised machine learning for transient discovery in deeper, wider, faster light curves. *Monthly Notices of the RAS*, 498(3):3077–3094.

[147] Wei, Y., Sheth, R., and Khardon, R. (2020). Direct loss minimization for sparse Gaussian processes. *arXiv e-prints*, page arXiv:2004.03083.

[148] Weinberg, D. H., Mortonson, M. J., Eisenstein, D. J., Hirata, C., Riess, A. G., and Rozo, E. (2013). Observational probes of cosmic acceleration. *Physics Reports*, 530(2):87–255.

[149] Williams, C. K. and Barber, D. (1998). Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351.

[150] Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390.

[151] Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

[152] Xu, X., Liu, H., and Yao, M. (2019). Recent progress of anomaly detection. *Complexity*, 2019:1–11.

[153] Yuan, F., Lidman, C., Davis, T. M., Childress, M., Abdalla, F. B., Banerji, M., Buckley-Geer, E., Carnero Rosell, A., Carollo, D., Castander, F. J., D'Andrea, C. B., Diehl, H. T., Cunha, C. E., Foley, R. J., Frieman, J., Glazebrook, K., Gschwend, J., Hinton, S., Jouvel, S., Kessler, R., Kim, A. G., King, A. L., Kuehn, K., Kuhlmann, S., Lewis, G. F., Lin, H., Martini, P., McMahon, R. G., Mould, J., Nichol, R. C., Norris, R. P., O'Neill, C. R., Ostrovski, F., Papadopoulos, A., Parkinson, D., Reed, S., Romer, A. K., Rooney, P. J., Rozo, E., Rykoff, E. S., Sako, M., Scalzo, R., Schmidt, B. P., Scolnic, D., Seymour, N., Sharp, R., Sobreira, F., Sullivan, M., Thomas, R. C., Tucker, D., Uddin, S. A., Wechsler, R. H., Wester, W., Wilcox, H., Zhang, B., Abbott, T., Allam, S., Bauer, A. H., Benoit-Lévy, A., Bertin, E., Brooks, D., Burke, D. L., Carrasco Kind, M., Covarrubias, R., Crocce, M., da Costa, L. N., DePoy, D. L., Desai, S., Doel,

P., Eifler, T. F., Evrard, A. E., Fausti Neto, A., Flaugher, B., Fosalba, P., Gaztanaga, E., Gerdes, D., Gruen, D., Gruendl, R. A., Honscheid, K., James, D., Kuropatkin, N., Lahav, O., Li, T. S., Maia, M. A. G., Makler, M., Marshall, J., Miller, C. J., Miquel, R., Ogando, R., Plazas, A. A., Roodman, A., Sanchez, E., Scarpine, V., Schubnell, M., Sevilla-Noarbe, I., Smith, R. C., Soares-Santos, M., Suchyta, E., Swanson, M. E. C., Tarle, G., Thaler, J., and Walker, A. R. (2015). OzDES multifibre spectroscopy for the Dark Energy Survey: first-year operation and results. *Monthly Notices of the RAS*, 452:3047–3063.

[154] Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 694–699, New York, NY, USA. Association for Computing Machinery.

[155] Zhang, H. (2004). The optimality of naive bayes. volume 2.