Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

Mon. Not. R. Astron. Soc. **414,** 1987–2004 (2011) doi:10.1111/j.1365-2966.2011.18514.x

Statistical classification techniques for photometric supernova typing

J. Newling,[1,2★] M. Varughese,[3] B. Bassett,[1,2,4] H. Campbell,[5] R. Hlozek,[6] M. Kunz,[7] H. Lampeitl,[5] B. Martin,[2,8,9] R. Nichol,[5] D. Parkinson[10] and M. Smith[1,9]

[1]*Department of Mathematics and Applied Mathematics, University of Cape Town, Rondebosch 7701, South Africa*
[2]*African Institute for Mathematical Sciences, 6-8 Melrose Road, Muizenberg 7945, South Africa*
[3]*Department of Statistical Sciences, University of Cape Town, Rondebosch 7701, South Africa*
[4]*South African Astronomical Observatory, PO Box 9, Observatory 7935, South Africa*
[5]*Institute of Cosmology and Gravitation, University of Portsmouth, Portsmouth PO1 3FX*
[6]*Department of Astrophysics, Oxford University, Oxford OX1 3RH*
[7]*Département de Physique Théorique, Université de Genève, Genève CH1211, Switzerland*
[8]*Department of Astronomy, University of Cape Town, Rondebosch 7701, South Africa*
[9]*Astrophysics, Cosmology and Gravity Centre, University of Cape Town, Rondebosch 7701, South Africa*
[10]*Astronomy Centre, University of Sussex, Brighton BN1 9QH*

Accepted 2011 February 8. Received 2011 January 28; in original form 2010 October 20

ABSTRACT

Future photometric supernova surveys will produce vastly more candidates than can be followed up spectroscopically, highlighting the need for effective classification methods based on light curves alone. Here we introduce boosting and kernel density estimation techniques which have minimal astrophysical input, and compare their performance on 20 000 simulated Dark Energy Survey light curves. We demonstrate that these methods perform very well provided a representative sample of the full population is used for training. Interestingly, we find that they do not require the redshift of the host galaxy or candidate supernova. However, training on the types of spectroscopic subsamples currently produced by supernova surveys leads to poor performance due to the resulting bias in training, and we recommend that special attention be given to the creation of representative training samples. We show that given a typical non-representative training sample, S, one can expect to pull out a representative subsample of about 10 per cent of the size of S, which is large enough to outperform the methods trained on all of S.

Key words: methods: statistical – techniques: photometric – supernovae: general.

1 INTRODUCTION

Type Ia supernovae (SNeIa) provided the first widely accepted evidence for cosmic acceleration in the late 1990s (Riess et al. 1998; Perlmutter et al. 1999). Based on small numbers of predominantly spectroscopically confirmed SNeIa, those results have been confirmed by independent analyses (Eisenstein et al. 2005; Percival et al. 2007, 2010; Fu et al. 2008; Giannantonio et al. 2008; Komatsu et al. 2011; Mantz et al. 2010) and by a series of steadily improving SNeIa surveys. These modern SNeIa surveys have acquired about an order of magnitude more SNeIa than those early offerings, now covering redshifts out to $z \sim 1.5$ (Filippenko et al. 2001; Aldering et al. 2002; Astier et al. 2006; Clocchiatti et al. 2006; Kessler et al. 2009; Folatelli et al. 2010). In addition, these surveys now have excellent light-curve coverage with rolling search strategies and multifrequency light-curve data with significantly better control of photometric errors due to the use of a single telescope to acquire the data in each major survey.

The next generation of SNeIa surveys will be integrated into major photometric surveys, such as the Dark Energy Survey (DES; Wester et al. 2005), PanSTARRS (Kaiser & Pan-STARRS Team 2005), SkyMapper (Schmidt et al. 2005) and Large Synoptic Survey Telescope (LSST; Tyson 2002). These next generation surveys promise to catalyze a new revolution in SNIa research due to the sheer number of high-quality SNIa candidates that will be discovered: tens of thousands and perhaps millions of good SNIa candidates over the decade 2013–2023. Spectroscopic follow-up will probably be limited to a very narrow subset of these candidates and so finding ways to best choose the follow-up subset to utilize the photometric data is a key challenge in SN cosmology for the coming decade.

In this paper we are interested in methods that can be used to accurately identify SNeIa from their light curves alone, i.e. their variation in brightness in different colour bands as a function of time. This is a departure from traditional studies of SNeIa where all SNe

★E-mail: james.newling@gmail.com

used in cosmological parameter estimation studies have had their type confirmed via one or more spectra. Previous endeavours to use light curves for classification include Poznanski et al. (2002) and Rodney & Tonry (2009). In addition template-based photometric typing was used in the Sloan Digital Sky Survey II (SDSS II) SN survey (Frieman et al. 2008) to select the most likely SNIa candidates for spectroscopic follow-up with high confidence.

There are two ways that one can imagine using photometric candidates. The first approach is to use all the SNe, irrespective of how likely they are to actually be a SNIa. This is the approach exemplified by the BEAMS formalism, which accounts for the contamination from non-Ia SN data using the appropriate Bayesian framework (Kunz, Bassett & Hlozek 2007). The more conservative approach is to try to classify the candidates into Ia, Ibc or II SNe, and then only use those objects that are believed to be SNeIa above some threshold of confidence.

The origin of this paper was the Supernova Photometric Classification Challenge (SNPCC) run by Kessler et al. (2010a). The SNPCC provided a simulated spectroscopic training data sample of approximately 1000 known SNe. The challenge was then to predict the types of approximately 20 000 other objects from their light curves alone. The challenge is now over, and the results from the different contributors are summarized in Kessler et al. (2010b).

In this paper we present the details of a number of approaches to this problem, and their successes and failures. In Section 3 we discuss methods we have implemented to go from multiband light curves to probabilities, while in Section 4 we discuss the performance of the methods in the SNPCC. In particular we highlight how a non-representative training sample negatively affects the performance of the different algorithms. Finally, we conclude with recommendations for the future.

2 THE LIGHT-CURVE DATA

2.1 The supernova challenge data

The data used in this paper consist of ∼20 000 simulated SN light curves with associated SN types released after the SNPCC.[1] The SNPCC data[2] are only relevant in our discussion of competition scores. Our reason for using the post-data is that it has numerous improvements and bug fixes and is a more accurate simulation. The simulation was based on a DES-like survey, consisting of five SN fields, each of 3 deg^2, such that 10 per cent of the total survey time is allocated to the SN survey. The SNPCC data set consists of a mixture of SN types (Ia, II, Ib, Ic), sampled randomly with proportions given by their expected rates as a function of redshift.

Each simulated SN consists of flux measurements in the *griz* filters (Fukugita et al. 1996) and includes information about the sky noise, point spread function and atmospheric conditions that are anticipated for the DES site. Distances were calculated assuming a standard Λ cold dark matter (ΛCDM) cosmology ($\Omega_M = 0.3$, $\Omega_\Lambda = 0.7$ and $w = -1$), with anomalous scatter around the Hubble diagram drawn from a Gaussian distribution with $\sigma_m = 0.09$ and applied coherently to each passband. The SNPCC data include two selection criteria. Each object is required to have at least one observation with a signal-to-noise ratio (S/N) above 5 in any filter, and must also have at least five observations after explosion. A complete summary of the SNPCC is given in Kessler et al. (2010a,b).

[1] These post-SNPCC light curves are available at http://sdssdp62.fnal.gov/sdsssn/SIMGEN_PUBLIC/

[2] These competition light curves are available from http://www.hep.anl.gov/SNchallenge/

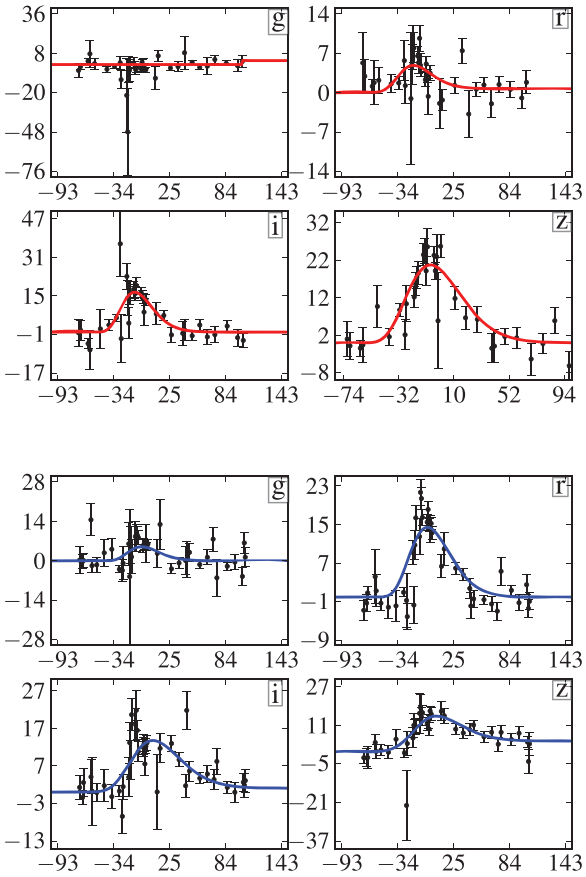

Figure 1. Above: a typical well-sampled SNIa light curve, in this case at redshift $z = 0.694$. Below: the light curve of a typical well-sampled non-Ia SN at $z = 0.663$. Overplotted are the best-fitting curves using equation (1).

We took part in two of the SNPCC challenges. In the first (+HOSTZ) challenge, participants were provided with photometric host galaxy redshift estimates, based on simulated galaxies analysed using the methods discussed in Oyaizu et al. (2008) and asked to return the type of each SN candidate. In the second (−HOSTZ) challenge, no redshift estimates for simulated SNe were provided. Both challenges are considered in this paper, but with emphasis on the +HOSTZ challenge. We did not attempt to distinguish between non-Ia subtypes (such as Type II and Type Ib/c SNe).

Fig. 1 shows the multiband light-curve data for a randomly selected Ia and non-Ia SN. To these measurements, a parametric curve has been fitted as discussed in Section 2.2.1.

2.1.1 Training samples

The aim of the SNPCC was for the participants to classify each of the simulated SNe into Ia or non-Ia (and non-Ia subclasses if they desired) with the aim of minimizing false Ia detections and maximizing correct Ia detections. To aid this, a spectroscopic training sample of ∼1000 SNe with known type was provided which is a simulation of expected spectroscopic observations on a 4-m class telescope with a limiting magnitude of $r \sim 21.5$, and an 8-m class telescope with limiting i band magnitude of 23.5. Because spectroscopy is harder than photometry the distribution of SNe in

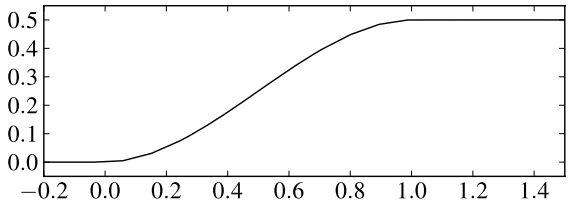

Figure 2. The tail function Ψ, which is used in fitting equation (1). Parameters (ψ, ϕ, τ) are kept fixed at $(0.5, 0, 1)$ here.

this spectroscopic sample is much brighter on average than the full photometric sample, and hence is not representative of the full sample. This is a crucial point to appreciate and as a result in this paper we refer to this sample as the *non-representative training sample*.

We will often compare with the results from a representative sample, generated by spectroscopically following up a sample of objects that is representative of the full photometric SN population. To produce an unbiased training sample, at the conclusion of the SNPCC when the types of each SNPCC object were revealed, we randomly selected ∼1000 SNe from the entire SNPCC data set, and considered the effect of using this as our training sample. This is referred to in the text as the *representative training sample*. We refer to the SNe that require classification as the *unclassified set*.

2.2 Post-processed data

2.2.1 Fitting a parametrized curve

In the provided photometric data the number, sampling times, frequency and accuracy of the sampled magnitudes varies greatly for each SN, as illustrated in Fig. 1. In order to standardize the raw data we fit, by weighted least squares, a parametrized function to the light curves in each of the four colour bands. Our parameters are $(A, \phi, \psi, k, \sigma)$ and the flux in each band is taken to be[3]

$$F(t) = A\left(\frac{t-\phi}{\sigma}\right)^k \exp\left(-\frac{t-\phi}{\sigma}\right) k^{-k}\mathrm{e}^k + \Psi(t). \quad (1)$$

The five parameters to be fit in each band have the following interpretations: $A + \psi$ is the peak flux, ϕ is the starting time of the explosion, k determines relative rise and decay times and σ is a temporal stretch term. τ, the time of peak flux, is determined by these parameters via $\tau = k\sigma + \phi$. The function Ψ is a 'tail' function such that $F(t) \to \psi$ as $t \to \infty$. The exact form (illustrated in Fig. 2) of Ψ is

$$\Psi(t) = \begin{cases} 0 & -\infty < t < \phi, \\ \text{cubic spline} & \phi < t < \tau, \\ \psi & \tau < t < \infty, \end{cases}$$

where the cubic spline is uniquely determined to have zero derivative at $t = \phi$ and $t = \tau$. The effect of each parameter is illustrated in Figs 3–6. We have also posted two files at Cosmology at AIMS (2010), each containing 200 randomly selected and fitted SNe to illustrate the range of fits possible. With five free parameters, A, ψ, ϕ, k and σ in each colour band and a host redshift (in +HOSTZ challenge), we have 21 parameters specifying each SN. We do not require that there be any correlation between the derived parameters in any band, e.g. between explosion time, time at peak or stretch. This is a natural extension to study in future work.

[3] This function has a single maximum and therefore cannot fit examples which have a double peak. However, for the data we use in this paper this turns out not to be an important limitation.

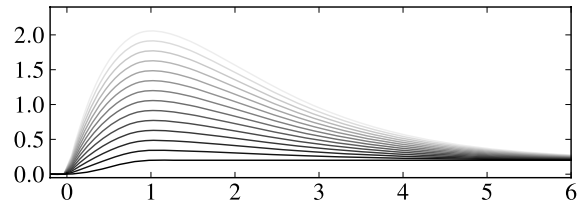

Figure 3. The effect of varying A on the function $F(t)$ from low (dark) to high (light). We keep the parameters (k, σ, ϕ, ψ) fixed at $(1, 1, 0, 0)$.

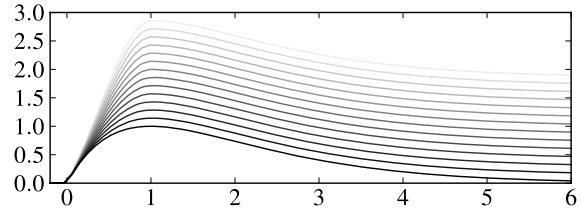

Figure 4. The effect of varying ψ on the function $F(t)$ from low (dark) to high (light). We keep the parameters (k, σ, ϕ, A) fixed at $(1, 1, 0, 1)$.

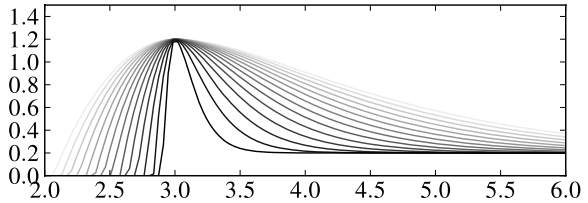

Figure 5. The effect of varying σ on the function $F(t)$ from 0.1 (dark) to 1.0 (light). Increasing σ linearly stretches the curve away from the $t = \phi$. We keep the parameters (A, ϕ, k, τ) fixed at $(1, 0, 1, 3)$.

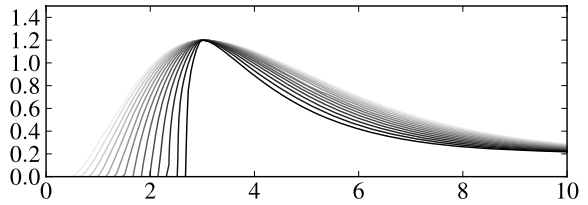

Figure 6. The effect of varying k on the function $F(t)$ from 0.2 (dark) to 1.8 (light). Increasing k decreases the ratio of rise to decay time (rapid rise relative to decay means low k). We keep the parameters (A, σ, ϕ, τ) fixed at $(1, 1.5, 0, 3)$.

2.2.2 Sparse data sets

About 5 per cent of all the SNe had fewer than eight observations in one or more of the four bands. To avoid overfitting, we did not fit these SNe with equation (1). Instead, these sparsely sampled SNe were each fit to a five-dimensional point – the maximum flux in each of the four colour bands plus the host redshift. The kernel density estimation (KDE) and boosting methods (Section 3) were applied to these SNe in the same way as was done in the 21-dimensional (21D) case (Sections 4.1.1 and 4.2.1). Unless otherwise stated, discussions and illustrations will all reference the 95 per cent of SNe which had eight or more observations in all bands and hence were fitted with 21 parameters.

2.2.3 *SALT fits*

In Section 4.4, we consider classification methods that require information on the distances to SNe to constrain their type. Distance moduli for all SNPCC SNe were derived using the publicly available light-curve fitter *SALT2* (Guy et al. 2007). Fits were carried out using the g, r and i passbands (i.e. z colour band data were not included). All available SNe were considered, which is significantly more liberal than the usual data-quality cuts applied during past SN cosmology analyses (Kessler et al. 2009). In this way, we maximized the number of SNe available for this work. We applied *SALT2* to 1256 SNe available in the non-representative training sample. Immediately, we found that 165 SNe failed to pass through *SALT2* with the reported error of the light curve either having a too low S/N or missing g-band data. We did not investigate these errors further and simply exclude these SNe. Furthermore, when the S/N is low, *SALT2* fits some SNe but returns a default upper limit magnitude of 99 and is unable to produce meaningful parameters from the light-curve fit. This affected 62 SNe in the training sample, which were also removed from the sample. For the 1029 SNe that were successfully fitted, *SALT2* returned a best-fitting value for four parameters M, X_0, X_1 and c for each event (which relate the peak magnitude and stretch/colour corrections to the light curve). The best-fitting Ia model light curve was also returned in the observer frame, which we used to calculate the χ^2 value for each SN in each passband (g, r, i) which are used in Section 4.4 to classify SNe. Distance moduli are calculated with

$$\mu = (m_B - M) + \alpha x_1 - \beta c, \tag{2}$$

where we used values of $\alpha = 0.1$, $\beta = 2.77$ and $M = 30.1$ to calculate the distance moduli, as discussed in Lampeitl et al. (2010). These values are consistent with those found in other analyses and were not expected to significantly affect our results. Fig. 7 shows the Hubble diagrams for the two training samples considered in this analysis. Also shown is the best-fitting cosmology to each Ia data set assuming a flat ΛCDM model. The non-representative training sample has a best-fitting value of $\Omega_m = 0.30$ compared to a value of $\Omega_m = 0.23$ for the representative training sample. In the non-representative training sample, non-Ia SNe are predominately found at lower redshifts than the representative training sample due to the effective magnitude cuts coming from the spectroscopic requirement of the non-representative sample.

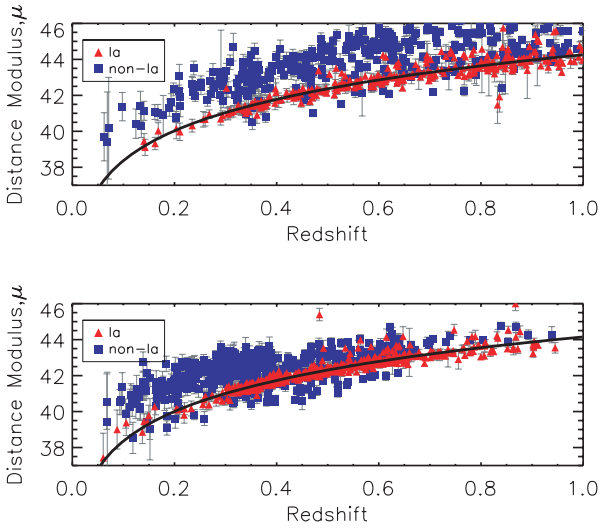

Figure 7. Hubble diagrams for the two training samples considered in this paper. SNeIa are shown as red triangles, while non-Ia SNe are plotted as blue squares. Also shown is the best-fitting cosmology to each SNIa sample. Above: the representative training sample, with $\Omega_m = 0.23$. Below: the non-representative training sample, as provided for the SNPCC, with $\Omega_m = 0.3$.

3 NEW CLASSIFICATION METHODS

We now describe in very general terms the classification algorithms we have used to facilitate application to other areas of cosmology and astrophysics. In order to classify a given object Y as either Ia or non-Ia, one would like the posterior probabilities $P(Y = \text{Ia}|x)$ and $P(Y = \text{non-Ia}|x) = 1 - P(Y = \text{Ia}|x)$. Here x are the parameters or features that characterize the SN. Knowing these posterior probabilities is equivalent to knowing the odds:

$$\text{odds}(x) = \frac{P(Y = \text{Ia}|x)}{P(Y = \text{non-Ia}|x)}.$$

Now one classifies Y as a Ia for example if $\text{odds}(x) > 1$, i.e. if $P(Y = \text{Ia}|x) > 0.5$. The two methods we discuss in this section approximate the odds in different ways.

(1) KDE estimates $P(x|Y = \text{Ia})$ and $P(x|Y = \text{non-Ia})$, the density of the features in classes Ia and non-Ia, respectively, and then uses Bayes formula to give

$$\text{odds}(x) = \frac{P(x|Y = \text{Ia})\, P(Y = \text{Ia})}{P(x|Y = \text{non-Ia})\, P(Y = \text{non-Ia})}.$$

(2) Boosting directly estimates $\text{odds}(x)$ through regression methods, as a sum of small trees built by a type of functional gradient descent.

These methods are discussed in detail below.

3.1 Kernel density estimation

KDE is a non-parametric method for estimating the probability density function (pdf) of a sequence of observables. Within this paper, the probability densities of the post-processed data described in Sections 2.2.1–2.2.3 are used for classification. Pdfs are useful as we may base a classification rule upon the relative probabilities that a candidate SN is either Type Ia or not Type Ia. Such a classification rule will require both the Ia and the non-Ia probability densities for the observed SN data. KDE enables us to derive these pdfs in a fairly model-independent manner, as we now discuss.

Suppose we have a set of d observables and that we would like to estimate the value of the pdf at a point $\boldsymbol{x}$ in this d-dimensional space. Given a training set with n observations, i.e. n points $\boldsymbol{X}_i$ in this d-dimensional space, the KDE is given by

$$\hat{f}_h(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h^d} K_i\left(\frac{\boldsymbol{x} - \boldsymbol{X}_i}{h}\right), \tag{3}$$

where $\hat{f}_h(\boldsymbol{x})$ is the KDE, $\boldsymbol{X}_i$ is the ith training observation, K_i is the kernel function for the ith training observation and h is the global kernel bandwidth. h is a tuning parameter: the kernels become more 'peaked' about the training observations as h becomes smaller. The optimal bandwidth may be obtained by cross-validation (see Appendix A). The choice of kernel is arbitrary, except that any proposed kernel should satisfy the following two conditions:

(i) $\int K(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = 1$,
(ii) $K(-\boldsymbol{x}) = K(\boldsymbol{x})$.

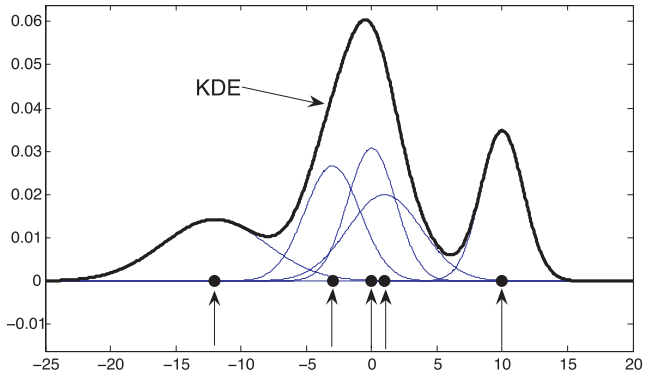

Figure 8. Schematic figure illustrating the idea of a KDE in one dimension. The training data points are shown as dark points with arrows. The Gaussian kernels are shown together with the sum of the kernels. Note that the KDE is not normalized in this figure and is thus close to what we actually use in this paper.

The first condition ensures that the KDE integrates to unity and that all observations carry equal weight, whilst the second condition ensures that the KDE is unbiased and is centred about one of the n d-dimensional training data points. The basic idea of the KDE method is illustrated in Fig. 8 in a simple 1D example. A commonly used kernel (and the kernel that we will use throughout this paper) is a multivariate Gaussian, normalized to unit volume:

$$K\left(\frac{\boldsymbol{x}-\boldsymbol{X}_i}{h},\Sigma_i\right) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_i|}}\exp\left\{-\frac{1}{2}\left(\frac{\boldsymbol{x}-\boldsymbol{X}_i}{h}\right)^{\mathrm{T}}\Sigma_i^{-1}\left(\frac{\boldsymbol{x}-\boldsymbol{X}_i}{h}\right)\right\}. \quad (4)$$

Here $\boldsymbol{x}$ and $\boldsymbol{X}_i$ are d-dimensional vectors and Σ_i is a $d \times d$ covariance matrix that changes the orientation and shape of the kernel around each training observation i; for example the covariance matrix Σ_i can be estimated from the nearest ℓ neighbours of a training data point, which is what we do, as described in Section 4.1 and as illustrated in Fig. 9. This provides the possibility of adapting the kernel to local variation. In contrast the bandwidth parameter h affects the global behaviour of the kernels. While it is more common to choose the covariances to be equal, for the SNPCC and the current application this would have been a bad choice (as described in Section 4.1).

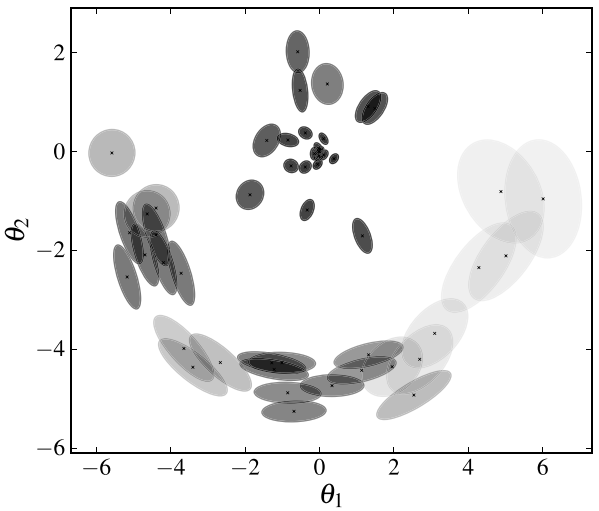

Figure 9. A realization of 50 points from an unusual distribution. Around each observed point a kernel is constructed. The axes of each kernel are the eigenvalues of the point's (2×2) Σ_i matrix (equation 4). Each Σ_i is the covariance matrix of the nearest ℓ points multiplied by the global bandwidth, h. Here $h = 0.6$ and $\ell = 10$.

3.1.1 *Integration over data errors*

In order to classify a SN with light-curve measurements $\boldsymbol{x}$, we must evaluate the KDE at $\boldsymbol{x}$. However, in our case we are not sure where $\boldsymbol{x}$ lies in parameter space as the light-curve measurements have errors and are not perfectly sampled.

Using a Gaussian kernel, we write the KDE as

$$\hat{f}(x) = \frac{1}{n}\sum_i \frac{1}{h^d}K\left(\frac{x-X_i}{h},\Sigma_i\right). \quad (5)$$

For simplicity we suppress vector notation but all quantities (other than h) are d-dimensional vectors or matrices, and the index i runs over the points in the training set.

Now assume that the location of a point in the d-dimensional space is not known exactly and is instead given by a Gaussian pdf. We take the mean to be x and the covariance matrix to be Y. The KDE value is then given by integrating the KDE over the unknown pdf of the point being classified:

$$\int \mathrm{d}z\, K(z-x,Y)\,\hat{f}(z) = \frac{1}{n}\sum_i \frac{1}{h^d}\int \mathrm{d}z\, K(z-x,Y)\,K\left(\frac{z-X_i}{h},\Sigma_i\right). \quad (6)$$

We notice that this reverts back to the original value if K is a δ function located at x. Further, the function being integrated is a product of two Gaussians, which is itself another Gaussian. The KDE value then simplifies to

$$\hat{f}(x) = \frac{1}{n}\sum_i \frac{1}{h^d}K\left(\frac{x-X_i}{h};\Sigma_i + h^{-2d}Y\right), \quad (7)$$

i.e. the KDE kernels simply have an increased variance, given by the sum of their covariance matrix and the covariance matrix of the point being evaluated, scaled by h^{-2d}. The importance of including this increased variance for uncertain observations should not be ignored, especially when the variances of the points being classified are large (as is the case in this paper). Correctly implementing equation (7) can significantly improve classification performance. In Section 4.1 we compare analyses on the SN data including and ignoring the covariance Y.

3.2 Boosting

Boosting (Freund & Schapire 1995) is a learning algorithm for classification. Until recently the most popular boosting algorithm was AdaBoost (Freund & Schapire 1997). AdaBoost works by combining weak classifiers into a committee, whose combined decision is significantly better than that of individual weak classifiers. The precise workings behind AdaBoost's success remained hazy until it was shown (Friedman, Hastie & Tibshirani 2000) that boosting produces the powerful committee by sequentially adding together weak classifiers calculated by steepest descent. The further ideas of slow learning (Friedman 2001) and bagging (Friedman 2002) were later introduced into boosting, culminating eventually in the gradient boosting machine (GBM) algorithm. The algorithm, implemented as a package in the statistical programming language

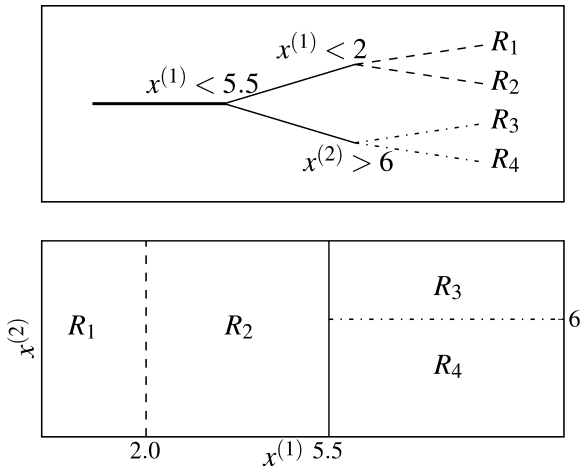

Figure 10. Above: a tree of depth 2 for classifying an object into one of 2^2 regions. Below: the tree domain containing 2^2 distinct regions as defined by the tree.

R,[4] is described in Section 3.2.3. A brief discussion of trees and loss functions is presented in Sections 3.2.1 and 3.2.2 in preparation for the presentation of the GBM algorithm.

3.2.1 Tree functions

The most widely used weak classifiers (a.k.a. basis functions) in boosting are trees. Trees are discontinuous functions which take discrete values in different regions of a domain. That is to say, a tree T has the form

$$T(\boldsymbol{x}) = \begin{cases} z_1 & \text{if } \boldsymbol{x} \in R_1, \\ \vdots & \\ z_K & \text{if } \boldsymbol{x} \in R_n, \end{cases}$$

where the K distinct regions $R_1 \cdots R_K$ together partition $\boldsymbol{x}$ space. The region boundaries can be described through the branchings of a tree, as illustrated in Fig. 10. For boosting, it is common to only use trees of a very simple form, that is only trees with branchings of the form $x^{(i)} < v$, where $x^{(i)}$ is one of the dimensions of $\boldsymbol{x}$ space and v is a real number. In the case of the SNPCC, $\boldsymbol{x}$ are the parameters fitted to the light curves in Section 2.2.1.

3.2.2 Loss function for classification

Suppose we have observed n training points, each consisting of data and type: $(\boldsymbol{X}_i, \tau_i)$, where the data $\boldsymbol{X}_i$ is a d-dimensional vector, and the type τ_i is ± 1, corresponding to the two classes. Suppose that we are required to find a function F: $R^d \rightarrow R$ which minimizes the following *loss function*:

$$L(F) = \sum_{i=1}^{n} \log(1 + \exp[-2F(\boldsymbol{X}_i)\tau_i]). \tag{8}$$

The specific form chosen for the loss function (8) can be explained by considering its partial derivatives with respect to $F(\boldsymbol{X}_i)$. Doing so (Hastie, Tibshirani & Friedman 2009), it can be shown that the form of F which minimizes (8) is given by

$$F(\boldsymbol{X}_i) = \frac{1}{2} \log \frac{\#\text{ observations: } \boldsymbol{X}_i, \tau = 1}{\#\text{ observations: } \boldsymbol{X}_i, \tau = -1}. \tag{9}$$

This is an approximation to half the log odds (the log of the odds):

$$\log \text{odds} \equiv \log \frac{P(\tau_i = 1 | \boldsymbol{x} = \boldsymbol{X}_i)}{P(\tau_i = -1 | \boldsymbol{x} = \boldsymbol{X}_i)}. \tag{10}$$

This is the key result: a function which minimizes the loss function (8) is a good approximation to half the log odds. A good approximation to the log odds is exactly what is needed for classification problems. The boosting algorithm aims to approximately minimize this loss function and in so doing arrive at an approximation of the log odds which can then be used for classification.

If you have observations at every possible data point, you can directly approximate the log odds through (9). In reality, you will not have observations at all possible data points, and so cannot do this. This trivially corresponds to not having observed all possible light curves, and so needing to make inferences from similar light curves. Boosting does this inference through constrained minimization of the loss function, as described in the following section.

3.2.3 The gradient boosting machine

The GBM (Friedman 2001) works by sequentially adding new trees to a function F, each addition reducing $L(F)$ (8) and so improving the approximation of F to half the log odds.

The trees, which have depth D, are appended to F at each of the M iterations of the GBM algorithm. Choosing larger M and D values results in a final $L(F)$ nearer to the global minimum value (9). However, our end objective is not to reach the global minimum but to construct a good approximation to the log odds, and trees of lower depth are generally better suited to this end.

Algorithm 1 (below) outlines an implementation of the GBM. A few subtleties have been omitted from it here, and we refer you to Appendix E for a fuller description. We recommend watching our demonstrative animation of the algorithm while reading Algorithm 1. The animation can be found at Cosmology at AIMS (2010), the URL in the references.

Algorithm 1 – gradient boosting machine

Input: $\boldsymbol{X}_i$, τ_i for observations $i = 1$ to n.

Initialize: $F_0(\boldsymbol{x}) \leftarrow \frac{1}{2} \log \frac{1+\bar{\tau}}{1-\bar{\tau}}$, where $\bar{\tau} = \frac{1}{n} \sum_{i=1}^{n} \tau_i$.

Initialize: $z_i \leftarrow 0$ for observations $i = 1$ to n. The z_is will measure how much of a 'misfit' each observation is.

Choose tree depth D and number of trees M.
For $m = 1$ to M:

(1) for $i = 1$ to n, update z_i:

$$z_i \leftarrow -\frac{\partial L}{\partial F_{m-1}(\boldsymbol{X}_i)} = \frac{2\tau_i}{1 + \exp[2F(\boldsymbol{X}_i)\tau_i]}.$$

(2) Fit by least squares T_m, the new tree: $z_i \sim T_m(\boldsymbol{X}_i)$ (where T_m has regions $R_{m,1} \cdots R_{m,2^D}$ fitted to minimize the in-group variance: see Appendix C for details).

(3) Choose constants $\gamma_{m,1} \cdots \gamma_{m,2^D}$ for $R_{m,1} \cdots R_{m,2^D}$ (chosen to minimize $L(F_{m-1} + T_m)$).

(4) $F_m \leftarrow F_{m-1} + T_m$.

Finally, $F \leftarrow F_M$.

[4] R and its associated packages can be downloaded from http://www.r-project.org

F is our final approximation to half the log odds, and it can now be used to classify with a simple rule of the form

IF $F(\boldsymbol{x}_i) > v \Rightarrow \tau_i = 1$; ELSE $\tau_i = -1$, where the optimal v depends on the figure of merit (FoM).

Notice that the variable z_i, updated in step 1, is positive if $\tau_i = 1$ and negative if $\tau_i = -1$. For this reason, when T_m is fit to the z_is at step 2, observations of the same type are more likely to fall into the same region of T_m. Moreover, observations with large z_is carry more weight while fitting T_m, and hence are even more likely to be placed with objects of the same type. This acts to place special attention on unusual objects, or objects whose type is not clear.

While values are fitted for each tree region in step 2 (as described in Appendix C), these values will not necessarily result in a reduced $L(F_{m-1} + T_m)$. Hence at step 3 of the algorithm, $\gamma_{m,k}$ values are explicitly chosen to minimize $L(F_{m-1} + T_m)$. In effect, only the tree *shape* is taken from step 2.

4 RESULTS

The entries in the SNPCC were evaluated using the FoM:

$$f(N_{\rm Ia}^{\checkmark}, N_{\rm non\text{-}Ia}^{\times}) = \text{efficiency} \times \text{pseudo-purity}$$
$$= \left(\frac{N_{\rm Ia}^{\checkmark}}{N_{\rm Ia}^{\rm TOT}}\right)\left(\frac{N_{\rm Ia}^{\checkmark}}{N_{\rm Ia}^{\checkmark} + 3N_{\rm non\text{-}Ia}^{\times}}\right),$$

where $N_{\rm Ia}^{\checkmark}$ is the number of correctly classified SNeIa, $N_{\rm non\text{-}Ia}^{\times}$ is the number of non-Ia SNe classified as SNeIa and $N_{\rm Ia}^{\rm TOT}$ is the total number of SNeIa. Had the coefficient of $N_{\rm non\text{-}Ia}^{\times}$ in the denominator of the pseudo-purity term been 1 and not 3 the term would have been true purity, i.e. the proportion of SNeIa in the final Ia-classified group. How relevant this FoM is to cosmology is not absolutely clear, but it is a robust measure of how well a classification algorithm penalizes both missed detections and false discoveries. For applications such as BEAMS (Kunz et al. 2007) a FoM which takes type probabilities as inputs would be more useful.

In this section we discuss the implementation and performance of each of our methods. Unless stated otherwise, the scores given in this section refer to the SNPCC, while all figures are using the post-SNPCC data described in Section 2.1. Of particular interest to us is the comparison of results obtained when the training is done with representative and non-representative samples. We also briefly mention applications that these methods have previously found in cosmology and related fields.

4.1 21D KDE

4.1.1 Application

KDEs have been used before in astronomy for estimating the pdf from a discrete or noisy data set (Fadda, Slezak & Bijaoui 1998; Bissantz et al. 2007; Ascasibar 2010), identifying groups (Balogh et al. 2004) and clusters (Valtchanov et al. 2004) in galaxy surveys and determining the timings of millisecond pulsars (Carstairs et al. 1991) and gamma-ray bursts (de Jager, Raubenheimer & Swanepoel 1986), to name a few examples.

In Section 2.2.1 we described how we fit the SN light curves in each of the four bands using the parametrized function (1), resulting in 20 light-curve parameters. With the addition of host redshift in the case of the +HOSTZ challenge, each SN is described by a 21D point. We use KDE to approximate the 21D Ia and non-Ia pdfs based on the training data.

We allowed the 21D training points to have different covariance matrices, as described in Section 3.1. As previously mentioned a single global covariance is most common for KDE, but in cases where a pdf has large regions of high and low probability, this can be problematic. In low-probability regions the kernel density will be too 'spikey', while in high-probability regions it will be too smooth. To understand this, consider what would happen if, in Fig. 9, the ellipses were constrained to all be of the same size. Chosen too small and the low-probability region would have 'bumps', too large and the high-probability region would lose features. The 21D points for the SNPCC are not uniformly distributed, as illustrated by the cumulative plots of Appendix F, and so are susceptible to this problem. Using cross-validation we chose $\ell = 10$ and $h = 0.6$ (using the notation from Section 3.1).

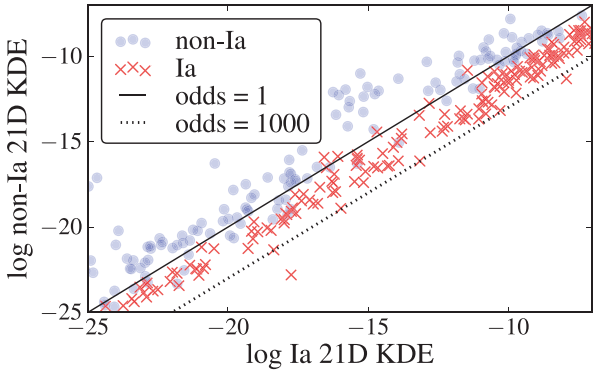

Figure 11. Ia (red crosses) and non-Ia (blue circles) in the non-representative training sample. The KDE values at calculated using 10-fold cross-validation.

Having constructed two KDEs from a training sample, each unclassified SN may be classified as follows.

(i) Fit equation (1) to each of the four light curves thus obtaining a 21D point for the candidate.

(ii) Evaluate the Ia and non-Ia kernel probabilities derived from the training sample at the 21D point, and then evaluate the odds.

(iii) If the odds (or log odds) is above some threshold, classify as Ia.

In cases where one or both of the KDEs are a poor representation of the underlying pdf, it may be preferable to modify step (iii). For example if one of the KDEs is particularly inaccurate, one may prefer to classify by using only the other KDE. For the SNPCC leaving step (iii) unchanged was advisable, as can be deduced from Fig. 11. The lines in Fig. 11 are lines of constant odds. If KDEs are accurate approximations to pdfs, a line of constant odds is optimal for discriminating between Ias (below the line) and non-Ias (above the line), irrespective of the FoM used. Furthermore, if the KDEs are accurate approximations to pdfs, there should be an equal number of Ias and non-Ias on the line odds = 1 and 1000 times more Ias as non-Ias on the line odds = 1000. This is roughly observed in Fig. 11 and so we can proceed to choose the odds line which maximizes the SNPCC FoM.

For the entry in the SNPCC, we failed to include the parameter covariance matrices when calculating KDE values (in effect, we set Y to be a matrix of zeros in equation 7). Our final score suffered as a result – the benefit of correctly implementing the calculation (7) is illustrated in Fig. 12, where we see from both the histograms and the cumulative plots an increased separation between Ias and non-Ias when equation (7) is correctly implemented. We find a 15 per cent increase in score when correctly implemented on the post-SNPCC data. The KDE method still obtained the second and third highest scores in the −HOSTZ and +HOSTZ competitions,

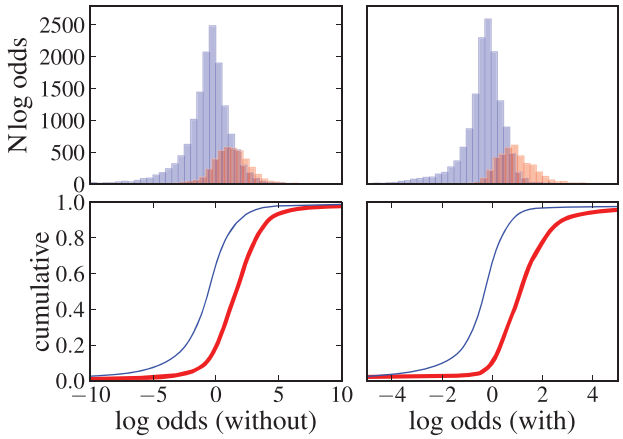

Figure 12. Histograms (above) and cumulative plots (below) of the 21D (representatively constructed) KDE log odds. Left: the parameter covariance matrix is not included in KDE evaluation as proposed in Section 3.1.1. Right: the parameter covariance matrix is included in KDE evaluation.

respectively, with scores of 0.37 and 0.39. Of interest is that the 20D KDE (−HOSTZ) is almost as good at classifying as the 21D KDE (+HOSTZ). The winning competition scores (Kessler et al. 2010b) were 0.51 (−HOSTZ) and 0.53 (+HOSTZ).

4.1.2 Non-representative versus representative

As with all of our methods, we constructed classifiers using both the non-representative sample provided and a representative sample of equal size, as described in Section 2.1.1. In each case, the remaining unclassified SNe were used as a test of the performance of the classifier.

Fig. 13 carries useful information about the performance of the non-representatively trained KDEs and representatively trained KDEs. For example, the efficiency of classifying Ias with a log odds threshold of 2 is simply the cumulative value of the unclassified Ias (solid red) at log odds = 2. For both representatively and non-representatively trained KDEs this is about 0.75, meaning that

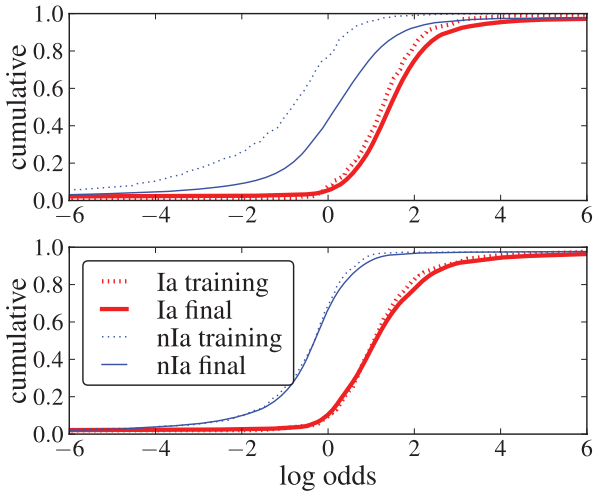

Figure 13. The cumulative frequency of log odds for non-Ia (blue) and Ia (red) SNe, for the training (dashed) and unclassified (solid) samples. The training log odds was calculated using 10-fold cross-validation. Above: using non-representative training. Below: using representative training.

about 75 per cent of SNeIa are correctly classified when a threshold of log odds = 2 is used.

To obtain high purity, the log odds threshold must be chosen such that the non-Ia cumulative frequency is low compared to the Ia cumulative frequency. To obtain high efficiency, the log odds threshold must be chosen such that the Ia cumulative frequency is high. Putting these together, to obtain both high purity and high efficiency, a log odds threshold must be found at which the non-Ia cumulative frequency is low and the Ia cumulative frequency is high.

The dashed lines in Fig. 13 are the cumulatives of the training data using 10-fold cross-validation. In the case of representative training, we see that these are accurate predictors of the true cumulatives. However, in the case of non-representative training, the non-Ia cumulatives of training and unclassified SNe are vastly different. If in the case of non-representative training one assumed that the training sample were in fact representative, one would predict a non-Ia misclassification rate of under of 10 per cent using a log odds cut-off of 1. In reality it is 30 per cent. Such dangerous predictions are impossible to make if a representative sample is used in KDE construction, as illustrated by the hugging of the solid lines to the dotted lines.

4.2 Boosting

4.2.1 Application

Boosting has been used in particle physics for example by the MiniBooNE neutrino oscillation experiment (Roe et al. 2005) and is implemented in the photometric redshift package ABORZ (Gerdes et al. 2010). In the SNPCC we applied boosting to the 20 fitted light-curve parameters for the −HOSTZ competition, and the 21 parameters for the +HOSTZ competition. Using 10-fold cross-validation we chose to use 4000 trees to maximize the FoM (11). We chose the learning rate to be 0.05 and the bagging fraction to be 0.5 (these parameters are described in Appendix E).

During the training phase of the SNPCC we expected, based on the idea that the training sample was representative, that boosting would significantly outperform the 21D KDE. In reality boosting performed more poorly than the 21D KDE, obtaining scores of 0.20 (−HOSTZ) and 0.25 (+HOSTZ; Kessler et al. 2010b) strongly suggesting that the 21D KDE method is more robust to biases in the training set than boosting.

In the case of the post-SNPCC data, the score obtained with non-representative training is even lower (0.15) (+HOSTZ) due to bugs in the original SNPCC data such as too dim non-Ias which made classification easier, as described in Kessler et al. (2010c). As a result comparison of scores in this paper with those in the competition cannot be made directly.

4.2.2 Non-representative versus representative

Our failure to correctly predict our score in the SNPCC was a result of the biases in the training sample. Boosting appears to be even more sensitive to training sample bias than the 21D KDE method. This is illustrated by the large deviation in Fig. 14 of the unclassified non-Ia curve from the training non-Ia curve with non-representative training.

While boosting is more sensitive to bias in the training sample than the 21D KDE, it is a superior classifier when a representative training sample is used. This is illustrated in Fig. 14 by the large vertical separation between non-Ia and Ia cumulative curves when

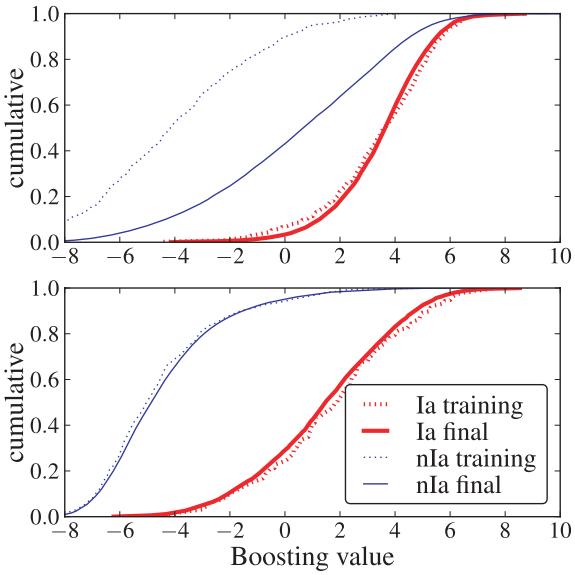

Figure 14. Boosting values obtained using (above) the non-representative training sample and (below) the representative training sample. The boosting values are approximations to (1/2)log odds.

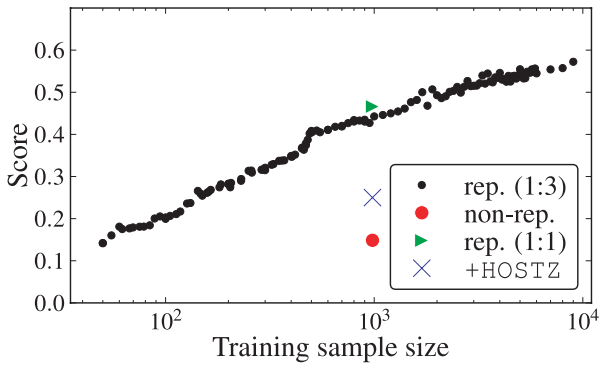

Figure 15. Small black circles: the score obtained by boosting when trained with random representative samples of varying size (100 to 6000 SNe). Large red circle: training on the given non-representative sample. Blue cross: the score obtained in the +HOSTZ competition. Green triangle: the performance when trained with a 'random' sample with non-random Ia:non-Ia ratio of 1:1 as opposed to true ratio Ia:non-Ia $\sim$ 1:3.

a representative sample is used. The vertical separation between the Ia and non-Ia curves is larger in the case of the boosting than the 21D KDE, resulting in a lower contamination rate and higher efficiency when boosting is used.

We see from Fig. 15 that training with 1000 representative SNe results in a score three times greater than training with 1000 non-representative SNe. We also see from Fig. 15 that training with a non-representative sample of size 1000 can be matched by training with only 50 representative SNe. The score obtained when 500 representative Ia and 500 representative non-Ia SNe are used for training, as opposed to the truly representative case where the Ia:non-Ia training ratio is 1:3, is only slightly higher; the advantage of extra Ias at the cost of non-Ias is marginal.

We did not include the parameter covariance matrices in any way in boosting. It is not clear how this inclusion would best be done, but the noticeable improvement to the 21D KDE score when the covariance is included suggests that it is worthwhile considering this question for future implementations. Two possibilities are (a) 'supersampling' – converting each training point into 100 training points drawn from a distribution with covariance given by the parameter covariance matrix, and (b) including the covariance matrix determinant as a 22nd boosting parameter.

We find that with boosting if a non-representative training sample is used the cumulative frequency lines of the unclassified SNe do not follow those of the training sample. On the other hand if a representative sample is used, 10-fold cross-validation provides accurate predictions for the unclassified SNe boosting values, as illustrated by the close hugging of training and unclassified cumulative lines in Fig. 14.

We see that boosting the 21D light-curve parameters with a representative sample results in a robust photometric classifier. To illustrate this point we have created on online archive of 200 randomly selected unclassified SNe, and labelled them according to boosting's output Cosmology at AIMS (2010). In some cases it is difficult to identify obvious Ia or non-Ia features, yet the algorithm classifies correctly.

4.3 Parameter importance

One advantage of the boosting algorithm is its ability to quantify the importance of parameters in classification (see Appendix D for details). In this section we look at these quantities in an effort to discover which fitted parameters are most useful for classification. We also ask which are the parameters that distinguish the non-representative training sample from the representative training sample, i.e. what makes the non-representative Ias and non-Ias a biased sample. We answer this question by performing boosting on a sample of representative and non-representative Ias, as if the SNPCC had been a competition to determine if a SN attains a spectrum or not.

Fig. 16 illustrates which parameters are most useful in distinguishing Ia from non-Ia in the representative training sample. One interesting feature illustrated in Fig. 16 is that every parameter appears to carry information.

The third most important parameter (after redshift and A in z band) is the parameter k in the i band. To interpret this piece of information, we first see in Fig. F3 that non-Ia SNe have on average

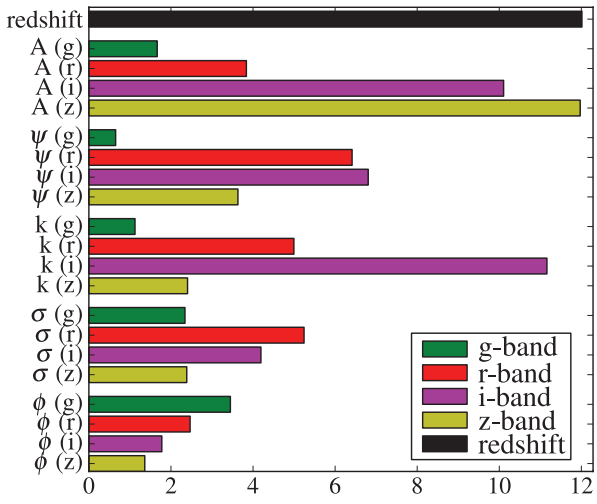

Figure 16. The importance of each of the 21 parameters in classifying SNe as Ia (or not) using boosting on the representative training sample.

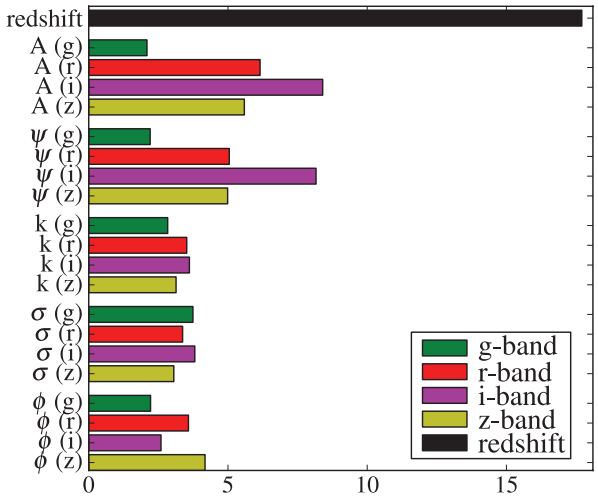

Figure 17. The importance of parameters in distinguishing representative from non-representative SNeIa using boosting.

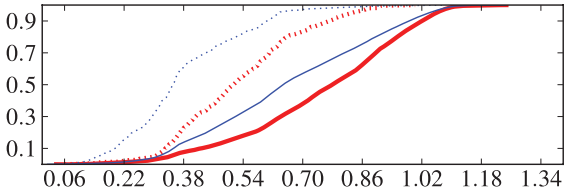

Figure 18. Cumulative plot of redshift, non-representative training (dashed) versus unclassified (solid), and Ia (red, thick) versus non-Ia (blue, thin).

lower k values than Ias. From this we then infer from Fig. 6 that Ias have a higher rise-time to decay-time ratio than non-Ia SNe.

The equivalent figure for the non-representative training (Fig. G1 in Appendix G) paints a similar picture with one noticeable difference: the information for distinguishing between Ia and non-Ia SNe in the non-representative training sample is carried almost exclusively in the r band.

We now turn to the comparison of representative and non-representative SNe. Fig. 17 suggests that the most biased parameter in the non-representative training sample is redshift. This is not surprising given that we know that the non-representative SNeIa are at lower redshift than the true Ia population (Fig. 18). Indeed, we see from Fig. 18, 70 per cent of Ia SNe in the non-representative training set are at a redshift of less than 0.6, while only 20 per cent of Ias in the unclassified set are within this redshift.

In the case of non-Ias SNe (Fig. G2 in Appendix G) boosting allocates the majority of the bias in the non-representative sample to the As. This is also unsurprising given that we are more likely to obtain a spectrum from bright objects than dim objects. It is not clear to us why boosting designates non-Ia bias to the As and Ia bias to redshift.

4.4 Hubble KDE

4.4.1 Applications

An alternative method for using the idea of KDEs is to use the *SALT2* light-curve fitter (with $\alpha = 0.1$ and $\beta = 2.77$ as in Hicken et al. 2009) to estimate distance moduli, μ_i and errors σ_i for all the objects in both the training and unclassified data, assuming that all the data are SNeIa. We can then construct two 2D KDEs for the training data: one consisting of all the known SNeIa and one from all the non-Ia data. Each kernel is normalized to have a total volume of unity and we use a slight modification of the standard KDE formalism because we do not normalize the KDE. Instead the heights of the summed KDEs are proportional to the number of SNeIa and non-Ia, respectively. In this way we include prior information related to the SN rates. A redshift range where there are many more SNIa than non-Ia will automatically tend to lead to a larger Ia KDE as a result. Of course, this does increase sensitivity of the method to biases/non-representativity in the training sample rates.

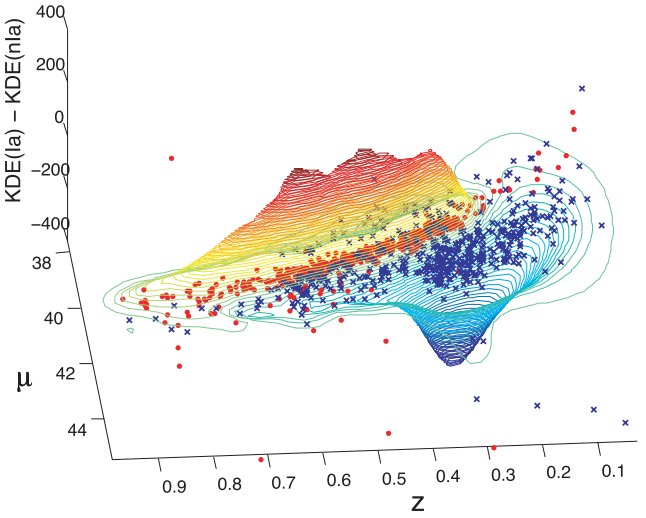

Figure 19. 3D contours of the difference between the Ia and non-Ia Hubble diagram KDEs as a function of redshift and distance modulus (μ) together with the actual non-representative training data used to produce the KDEs. The data used to construct the KDEs are also shown: Ia data as red circles and non-Ia data as blue crosses. There is a clear offset in the two KDEs reflecting the fact that in this training data the non-Ias are fainter, hence predominantly at lower redshift and with a much larger scatter than the Ias.

The 2D Gaussian kernel chosen for the Hubble KDE algorithm had a fixed bandwidth (standard deviation) in the redshift direction of 0.05 (chosen simply to avoid being too peaked but small enough that the distance modulus does not change significantly across it), while the bandwidth in the μ direction was determined by the error σ_i, on the distance modulus coming from *SALT2*, and also includes a 0.12 mag intrinsic dispersion error as usual. This means that points with large errors contribute very broad, low-amplitude humps to the final KDE, while points with small errors are much more peaked, reflecting our confidence in that point. For illustrative purposes we plot the difference $\mathrm{KDE_{Ia}} - \mathrm{KDE_{non\text{-}Ia}}$ of the two KDEs in Fig. 19. Positive values correspond to places where the Ia KDE dominates, negative values to where the non-Ia KDE dominates. In addition we plot the training data used to construct the KDEs.

Classification using these KDEs is then simple. For any candidate object, we run it through *SALT2* to give an estimated μ and σ. We can only use this approach on the data with a redshift estimate, z, unlike the 21D KDE and boosting algorithms which do not require a redshift. We then simply find the values of the two KDEs at that (μ, z) to yield probabilities of the object being a Ia or non-Ia. As

in the other KDE method, one should fold in the error σ on the candidates which, assuming Gaussianity, is simple, as described in Section 3.1.1. The result of this analysis is that each candidate has a pair of probabilities: ($P_{\rm Ia}$, $P_{\rm non\text{-}Ia}$) that can be used to classify the candidate.

4.4.2 *Non-representative training sample*

We applied this methodology to the whole sample of unknown SNe supplied. In total, we started with 17 065 SNe and lost 4578 SNe as junk because of *SALT2* failures previously mentioned (of which 2619 were complete failures and 1959 failed to return meaningful parameters from the Ia light-curve fit), leaving 12 487 SNe for further analysis.

Essentially this Hubble KDE approach simply checks whether or not an object lies close to the true cosmology curve on the Hubble diagram (defined by the Ia KDE) at that redshift. However, there are many non-Ias which lie close to the true cosmology curve. As a result one either has to be very strict with cuts (and therefore lose many true Ias) or one has to accept a large number of false positives: non-Ias that are classified as Ias.

Because there are so many non-Ias this and similar Hubble-diagram-based methods (such as the Portsmouth entry to the SNPCC) are less competitive as classifiers. In addition they also require a redshift estimate for the SNe and are hence doubly inferior compared with the 21D KDE and boosting.

4.5 Combining 21D and Hubble KDEs

In Section 4.1 we described the 21D KDE approach, and in Section 4.4 we described the Hubble KDE approach. In this section, we describe how we combined these approaches. As outlined in Appendix B, there are several ways of combining odds from different algorithms to construct a better combined classifier. For our combination entries in the SNPCC, we constrained our classifier to be of the form

$$(\text{Hubble odds})^{\alpha}\,(\text{21D odds})^{\beta} > \eta. \tag{11}$$

This corresponds to a straight line in Fig. 20. The scores for the combination entry was 0.28. Surprisingly, this was less than the score obtained using the 21D KDE alone, and so we believe that the line chosen for the SNPCC was poor. A straight line does seem to be a good choice for the distribution of values in Fig. 20, but perhaps a better choice would be of the form

$$\text{Hubble odds} > \gamma_1 \quad \text{and} \quad \text{21D odds} > \gamma_2. \tag{12}$$

A pure 21D odds classifier would rely on a vertical decision line, and a pure Hubble odds classifier would rely on a horizontal line, but it is clear from Fig. 20 that a classifier of the form (11) (dashed) or (12) (solid) should work better. Fig. 20 shows the separation of Ias and non-Ias that come from using the Hubble KDE odds and 21D KDE odds with the integration of errors presented in Section 3.1.1. The optimal lines of forms (11) and (12) result in scores of 0.24 and 0.22, respectively, in the case of non-representative training and 0.45 and 0.42, respectively, in the case of representative training. These scores are calculated using a purely 21D odds classification for the ∼8000 SNe without *SALT2* fits, and a 21D–Hubble combination for the remaining ∼12 500 SNe with *SALT2* fits. As with boosting, the 21D KDE classifier is significantly worse using the post-SNPCC data as previously discussed in Section 4.2.2, and so comparison between these post-SNPCC scores and other SNPCC scores should not be made until further analyses have been done.

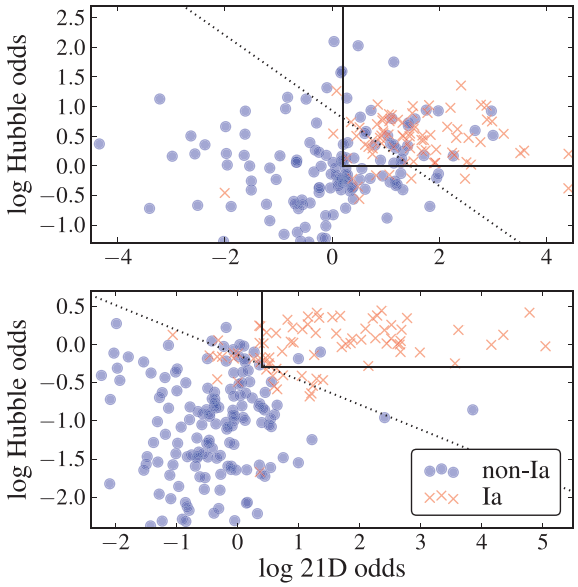

Figure 20. SNe of Type Ia (cross) and non-Ia (circle), located according to their 21D odds (x-axis) and Hubble odds (y-axis). Above: odds were calculated from KDEs constructed using the non-representative training sample. Below: a corresponding plot where KDEs were constructed with the representative training sample. We see that the separation obtained is smaller when non-representative training is used, and indeed the score obtained in the non-representative case is significantly lower. Note that the SNe in this figure are a random sample of the ∼12 500 with a meaningful *SALT2* fit.

To be in the top right-hand corner of Fig. 20, and therefore be classified as Ia, requires that a candidate must simultaneously lie close to the true cosmology distance modulus and have multiband light curves that have the right shape; a very natural approach to SNIa classification. It would be interesting to combine the Hubble odds with 21D boosting instead of 21D KDE, as boosting the 20 parameters produces better results, as seen in Section 4.2.

An obvious extension, if one wanted to combine the outputs from more than two classifiers, would be to use them as inputs to a new boosting analysis. The odds from the 21D KDE, the Hubble KDE, the 21D boosting and indeed any classifier of sufficient ability can be used as weak classifiers in boosting. A reason to exercise caution in using boosting or a neural network as a final classifier in this way is the possibility of overtraining, but this can be prevented by using 10-fold cross-validation.

5 BIAS REDUCTION

We saw in Fig. 15 that representative training samples with more than 50 objects outperform the 1000 strong non-representative training sample. In light of this astonishing fact we ask: how large a representative sample can be extracted from the non-representative sample?

Extracting a representative sample involves removing particular SNe from the non-representative training set such that what remains is representative of the unclassified set. This involves removing a large proportion of bright, low-z SNe from the training set (see Fig. 17). To decide exactly how many SNe of a given brightness or redshift need be removed, we look at the distributions of parameters A and redshift, and remove SNe from the training set such that

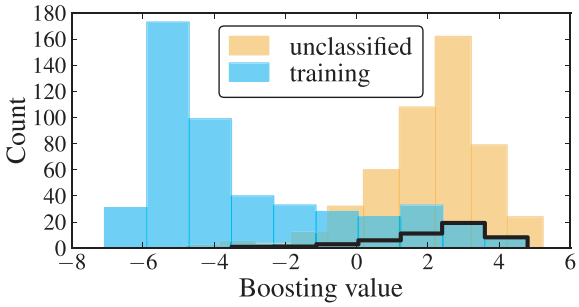

Figure 21. 500 randomly selected unclassified (orange) and training (blue) boosting values. The black line is proportional to the unclassified (orange) histogram. Therefore, the set consisting only of training (blue) SNe falling under the black line forms a representative sample of the unclassified SNe. In other words, if we discard all the training SNe with boosting values below −2, keep a certain fraction of the training SNe between −2 and 3 and keep all the training SNe with boosting values greater than 3, the resulting training set will be representative.

the distributions of the remaining training SNe coincide with the distributions of the unclassified SNe. Doing this we conservatively estimate that by appropriately removing 90 per cent of the training set we will be left with a representative sample. In other words, in the cumulative plots of Figs 18 and F1, the dashed lines will sit atop the solid lines if the appropriate 90 per cent of training SNe are removed.

A second approach required the calculation of boosting probabilities of the SNe being unclassified. Note that these new probabilities are distinct from the previously discussed Type Ia probabilities. Had the training set been representative, the distributions of these probabilities would have been the same for training and unclassified SNe. However, the training set has many unusually bright SNe which have particularly low probabilities as they do not look like typical unclassified SNe. As a result the distribution of the boosting values for the training set is skewed towards low probability. We therefore wish to remove some low-probability training SNe so that the probability distribution of the remaining SNe is proportional to that of the unclassified SNe. Doing this we again estimate that 90 per cent of the training SNe need to be removed. This is illustrated in Fig. 21, where by keeping only the 10 per cent of training SNe under the black curve, we obtain a training set whose probabilities are representative of the unclassified set.

In practice one would not discard SNe from the training set. Instead of removing 2/3 of SNe in a particular bin, one should rather give each SN in that bin a weight of 1/3. Note also that to be able to know which SNe to remove from the training set required that we knew the types of the SNe in the unclassified set. For real surveys this is of course unrealistic, but if simulations of the rates observed at different redshifts and magnitudes are accurate, these can be used to decide which SNe to remove from the actual training set.

6 DISCUSSION AND CONCLUSIONS

In this paper we have discussed the problem of classifying SNe into subclasses (Type Ia or non-Ia) based on photometric light-curve data alone, i.e. multiband fluxes as a function of time. This will be necessary for future surveys which will detect vastly more candidates than will be possible to follow up spectroscopically.

We have investigated two novel classes of classification algorithms, KDE and boosting, and applied them to simulated SNe light-curve data, finding that the methods performed impressively as long as they were trained on a representative sample. Using the KDE approach, we considered both a 21D case based on light-curve parameters from all bands and a 2D version based on fits to the Hubble diagram, using redshift information and an estimate of the distance modulus obtained using standard light-curve fitting software.

A key issue for the classification methods we used was the issue of the training data sets. We compared the results based on training on two very different data sets: the first, a non-representative set, mimicking the kind of spectroscopic sample available as part of the follow-up program of a typical current-generation SN survey. The second was a representative sample of the same size where training objects were selected at random from the full sample.

In general we found that the training on the representative sample produced exceptionally good results and that cross-validation on the training sample was able to accurately predict the purity and efficiency of the method on the full sample. On the other hand, training on the non-representative sample leads to relatively poor performance on the full data set. The importance of having an unbiased, representative sample is illustrated by the fact that for boosting, representative samples larger than about 50 objects outperformed the full non-representative sample of 1000 objects, as shown in Fig. 15.

Our primary result and recommendation therefore is that boosting and KDE are powerful methods for SN classification, with remarkably little astrophysical input. However, they require training samples that are as unbiased and representative as possible. Further, we found that a small unbiased training sample outperforms a much larger, but biased, training sample.

Our other main result is that neither boosting nor the 21D KDE method suffered particularly when the SN redshift information was unavailable. This is particularly gratifying given that accurate SN/host galaxy redshifts will not be available for most candidates in the future and that methods based on the Hubble diagram critically require redshift information to perform successfully.

While the algorithms we have presented were successful, there are modifications to our boosting implementation that should be experimented with for example different choices of light-curve parametrization. Further it would be very useful to investigate methods to reduce sensitivity to biases in the training data.

Finally, it is perhaps useful to comment on how our methods compare to the winner of the SNPCC (the methods we described in this paper finished second and third in the competition) which used a template-based method and performed very well. Our first comment is that comparison is hard because there was an overlap between the templates used by Sako and those used to generate the SNPCC, as described in Kessler et al. (2010b), so it is not clear how the method would perform on completely independent data. Secondly, it is not known how the various classification methods would perform with different FoMs. For cosmological applications (see Figs 22 and 23) one might prefer to use a FoM which looks at the bias in recovered cosmological parameters such as the w_0, w_a dark energy equation of state parameters. Investigating this important issue is left to future work. It is clear that finding the best approach to SN classification, and the best way to combine results from different classifiers, will be an active area of research in the coming decade.

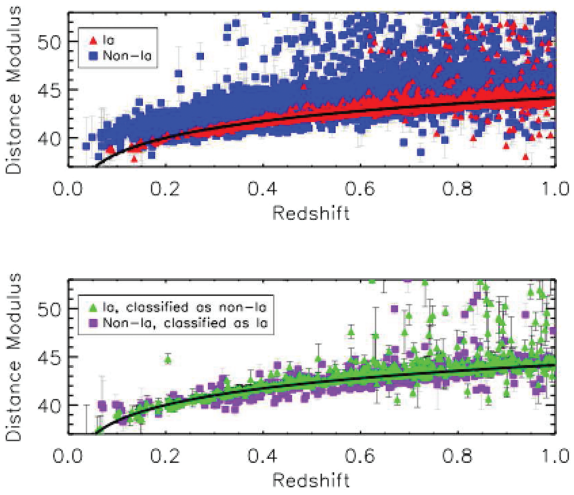

Figure 22. Hubble diagrams for the boosting results using the representative training sample. Above: objects that were correctly identified by the boosting method. SNeIa are plotted as red triangles, with non-Ia SNe shown as blue squares. Below: SNe that were incorrectly typed by boosting. SNeIa that were considered to be non-Ia SNe by boosting are shown as green triangles, with incorrectly typed non-Ia SNe shown as purple squares. Overplotted on each graph is the best-fitting cosmological model inferred from the representative training sample.

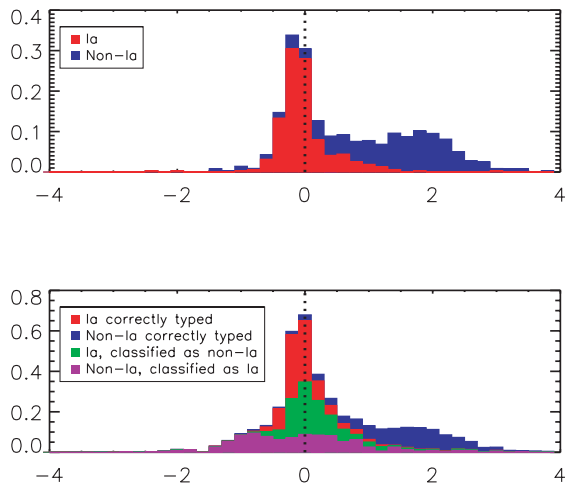

Figure 23. Cumulative histograms of the residuals from the best-fitting Hubble diagram, determined using the SNeIa in the representative training sample. Above: residuals for the representative training sample. SNeIa are plotted in blue, with non-Ia SNe shown in red. Below: residuals for the boosting results. SNeIa that were correctly typed are shown in blue, with correctly typed non-Ia SNe shown in blue. SNeIa that were considered to be non-Ia SNe by boosting are shown in green, with incorrectly typed non-Ia SNe shown in purple.

ACKNOWLEDGMENTS

We thank Trevor Hastie for suggesting the use of boosting and comments on the draft and Matt Hilton and Prina Patel for initial work on the project. We also thank Rick Kessler for comments on the draft. This work was begun during the JEDI4 workshop in Cape Town and is supported by the NRF and a Royal Society–NRF bilateral exchange grant. JN has a SKA bursary and MS is funded by a SKA fellowship. BB acknowledges funding from the NRF and Royal Society. MK acknowledges financial support by the Swiss NSF. DP was supported by the Science and Technology Facilities Council [grant number ST/F002858/1]. RH acknowledges funding from the Rhodes Trust.

REFERENCES

Aldering G. et al., 2002, in Tyson J. A., Wolff S., eds, Proc. SPIE Conf. Ser. Vol. 4836, Overview of the Nearby Supernova Factory. SPIE, Bellingham, p. 61
Ascasibar Y., 2010, Comput. Phys. Communications, 181, 1438
Astier P. et al., 2006, A&A, 447, 31
Balogh M. et al., 2004, MNRAS, 348, 1355
Bissantz N., Dömbgen L., Holzmann H., Munk A., 2007, J. R. Statistical Soc., 69, 483
Carstairs I. R. et al., 1991, Advances Space Res., 11, 95
Clocchiatti A. et al., 2006, ApJ, 642, 1
Cosmology at AIMS, 2010, Boosting for Supernova Classification. http://cosmoaims.wordpress.com/2010/09/30/boosting-for-supernova-classification/
de Jager O. C., Raubenheimer B. C., Swanepoel J. W. H., 1986, A&A, 170, 187
Eisenstein D. J. et al., 2005, ApJ, 633, 560
Fadda D., Slezak E., Bijaoui A., 1998, A&A, 127, 335
Filippenko A. V., Li W. D., Treffers R. R., Modjaz M., 2001, in Paczynski B., Chen W.-P., Lemme C., eds, IAU Colloq. 183, ASP Conf. Ser. Vol. 246, Small Telescope Astronomy on Global Scales. Astron. Soc. Pac., San Francisco, p. 121
Folatelli G. et al., 2010, AJ, 139, 120
Freund Y., Schapire R. E., 1995, in Blum E. K., ed., European Conference on Computational Learning Theory. Elsevier, p. 23
Freund Y., Schapire R. E., 1997, J. Comput. System Sci., 55, 119
Friedman J. H., 2001, Ann. Statistics, 29, 1189
Friedman J. H., 2002, Comput. Statistics Data Analysis, 38, 367
Friedman J., Hastie T., Tibshirani R., 2000, Ann. Statistics, 28, 337
Frieman J. et al., 2008, AJ, 135, 338
Fu L. et al., 2008, A&A, 479, 9
Fukugita M., Ichikawa T., Gunn J. E., Doi M., Shimasaku K., Schneider D. P., 1996, AJ, 111, 1748
Gerdes D. W., Sypniewski A. J., McKay T. A., Hao J., Weis M. R., Wechsler R. H., Busha M. T., 2010, ApJ, 715, 823
Giannantonio T., Scranton R., Crittenden R. G., Nichol R. C., Boughn S. P., Myers A. D., Richards G. T., 2008, Phys. Rev. D, 77, 123520
Guy J. et al., 2007, A&A, 466, 11
Hastie T., Tibshirani R., Friedman J., 2009, The Elements of Statistical Learning: Data Mining, Inference and Prediction, 2nd edn. Springer-Verlag, Berlin
Hicken M., Wood-Vasey W. M., Blondin S., Challis P., Jha S., Kelly P. L., Rest A., Kirshner R. P., 2009, ApJ, 700, 1097
Kaiser N., Pan-STARRS Team, 2005, BAAS, 37, 1409
Kessler R. et al., 2009, ApJS, 185, 32
Kessler R., Conley A., Jha S., Kuhlmann S., 2010a, arXiv e-prints (arXiv:1001.5210)
Kessler R. et al., 2010b, PASP, arXiv e-prints (arXiv:1008.1024)
Komatsu E. et al., 2011, ApJS, 192, 18
Kunz M., Bassett B. A., Hlozek R. A., 2007, Phys. Rev. D, 75, 103508
Lampeitl H. et al., 2010, MNRAS, 401, 2331
Mantz A., Allen S. W., Rapetti D., Ebeling H., 2010, MNRAS, 406, 1759
Oyaizu H., Lima M., Cunha C. E., Lin H., Frieman J., Sheldon E. S., 2008, ApJ, 674, 768
Percival W. J., Cole S., Eisenstein D. J., Nichol R. C., Peacock J. A., Pope A. C., Szalay A. S., 2007, MNRAS, 381, 1053
Percival W. J. et al., 2010, MNRAS, 401, 2148
Perlmutter S. et al., 1999, ApJ, 517, 565

Poznanski D. et al., 2002, http://wise-obs.tau.ac.il/ dovip/typing/
Riess A. G. et al., 1998, AJ, 116, 1009
Rodney S. A., Tonry J. L., 2009, http://wise-obs.tau.ac.il/ dovip/typing/
Roe B. P., Yang H., Zhu J., Liu Y., Stancu I., McGregor G., 2005, Nuclear Instruments Methods Phys. Res. A, 543, 577
Schmidt B. P., Keller S. C., Francis P. J., Bessell M. S., 2005, BAAS, 37, 457
Tyson J. A., 2002, in Tyson J. A., Wolff S., eds, Proc. SPIE Conf. Ser. Vol. 4836, Large Synoptic Survey Telescope: Overview. SPIE, Bellingham, p. 10
Valtchanov I. et al., 2004, A&A, 423, 75
Wester W. et al., 2005, in Wolff S. C., Lauer T. R., eds, ASP Conf. Ser. Vol. 339, Observing Dark Energy. Astron. Soc. Pac., San Francisco, p. 152

APPENDIX A: CROSS-VALIDATION

Cross-validation is a statistical technique that enables one to tune model parameters so as to optimize model prediction. Within the context of the 21D KDE, both the kernel bandwidth h, the number of nearest neighbours k and the odds threshold may be optimized for some FoM by 10-fold cross-validation. This entails partitioning the training set into 10 roughly equal parts. One may then use nine-tenths of the data to estimate the Ia and non-Ia probability densities and then use these probability densities to classify the remaining one-tenth of the training set. This may be repeated 10 times, predicting the class for each of the 10 partitions of the data using the KDEs estimated from the remaining nine partitions. Since we know the SN types of the training set, we can then find a combination of the aforementioned three parameters that maximizes the FoM. Cross-validation can be used in a similar way for boosting. Fig. E1 uses cross-validation to determine that 4000 trees will be near optimal.

APPENDIX B: PROBABILISTIC INTERPRETATION AND COMBINATION OF PROBABILITIES

By evaluating each KDE, we may obtain the probability of observing a light curve (with the light-curve data denoted as x) conditioned on the SN being a Ia or not, i.e. we get $p_1 = p(x|\text{Ia})$ and $p_2 = p(x|\text{non-Ia})$. The ratio of p_1 to p_2 is known as the *Bayes factor*, B_{12}. What interests us, however, is the relative probability of the observation x being from a Ia SN versus another type. That relative probability, called the odds ratio, odds(x) is

$$P(\text{Ia}|x) = p_1 \frac{P(\text{Ia})}{P(x)}, \tag{B1}$$

$$P(\text{non-Ia}|x) = p_2 \frac{P(\text{non-Ia})}{P(x)}, \tag{B2}$$

$$\begin{aligned}\text{odds(x)} &= \frac{P(\text{Ia}|x)}{P(\text{non-Ia}|x)} = \frac{p_1 P(\text{Ia})}{p_2 P(\text{non-Ia})} \\ &= B_{12} \frac{P(\text{Ia})}{P(\text{non-Ia})}.\end{aligned} \tag{B3}$$

The probabilities P(Ia) and P(non-Ia) are the prior probabilities to observe a Ia SN or one of another type, respectively.

To convert the relative probability back into absolute probabilities we need use the fact that there are only two possibilities (Ia or not), so that $P_2 = (1 - P_1)$. In this case we have that

$$P_1 = \text{odds}(x)/(1 + \text{odds}(x)). \tag{B4}$$

If we have two independent observations x and y then we can update the relative probability odds(x) from observation x:

$$\text{odds}(x, y) = \text{odds}(x) \frac{p(y|\text{Ia})}{p(y|\text{non-Ia})}. \tag{B5}$$

We can use this to combine for example the probability from the 21D KDEs with information from the Hubble diagram, but we have to be careful if the 21D KDEs already contain some of the Hubble information implicitly e.g. through the evolution of the overall amplitudes of the light curves as a function of redshift.

It is possible that the KDEs should not be interpreted as probabilities. This may be due to oversmoothing of too wide kernels, or shot noise from too narrow kernels. With a sufficiently large training set one can test how accurately the KDEs represent probabilities – the proportion of SNeIa in a (calculated) odds bin should equal that predicted by equation (B4). If it is not, one can consider making a mapping from the calculated odds to the true odds.

If in combining probabilities one does not want to assume independence, or does not trust the probabilities and does not want to make a mapping to true probabilities, there are several alternatives to equation (B5). Some of these include capping unreliable odds at 1, using linear combinations of odds instead of products, using p-values instead of probabilities and down-weighting particularly small/large Bayes' factors. Often an optimal method can be decided on by considering a scatter plot (like Fig. 20) of the training set. In Section 4.5 we considered two new ways of combining odds, equations (11) and (12).

APPENDIX C: BEST TREES

Suppose we have some data $\boldsymbol{X}_i \in R^2$, $z_i \in R$, and we would like to fit $z_i \sim \boldsymbol{X}$ using a tree. To be precise, we would like to find a tree which minimizes $\sum_{i=1}^{n} (T(\boldsymbol{X}_i) - z_i)^2$, where $T(\boldsymbol{X}_i) = v_k$ when $\boldsymbol{X}_i$ falls into node k of the tree. We therefore need to find two things, the tree shape and the 'leaf' values (the v_ks). Fig. C1 illustrates the idea of 'greedy' tree construction. Note that this may not be *the* best depth three tree. The greedy approach ignores several potential trees. However, it is quick and easy, and for boosting where thousands of trees are made it is not necessary to have exactly minimizing trees at each step. See also our animation of tree construction on the arXiv at Cosmology at AIMS (2010).

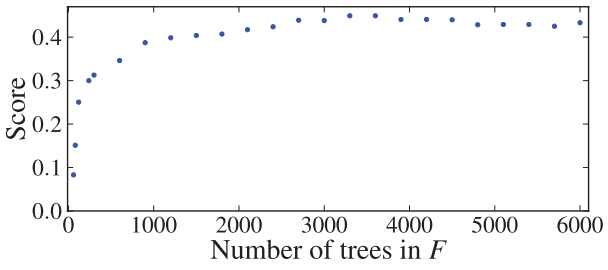

Figure C1. Above left: at several $\boldsymbol{X}_i \in R^2$ we have a value $z_i \in R$, represented by a rectangle if negative and a circle if positive, with the size of the shape being proportional to the magnitude of z_i. We want to split the set of observations by $X^{(1)}$ or $X^{(2)}$ to minimize the average in-group variance. Above right: after considering all vertical and horizontal lines, we settle on this vertical line as our first 'branching' as it minimizes in-group variance. Below left: subbranches are chosen to minimize in-group variance. Below right: a tree of depth 3.

APPENDIX D: CALCULATING PARAMETER IMPORTANCE

To measure how much information each parameter carries in the boosting classifier, we can do the following. For each branching within each tree constructed from the training data, calculate how much total in-group variance was reduced by this branching. Then for each parameter, for all the branchings which it defines add up the in-group variance reductions. This value is a good indicator of a parameter's importance in classification.

APPENDIX E: GBM IN FULL

In this appendix we complete the GBM algorithm presented in Section 3.2.3. There was no mention in Section 3.2.3 of the learning rate ν, or the bagging fraction ϕ. The learning rate $\nu \in [0, 1]$ should appear in step 4 of the main loop. Originally given as

$$F_k \leftarrow F_{k-1} + T_k,$$

step 4 should appear as

$$F_k \leftarrow F_{k-1} + \nu T_k.$$

The learning rate should be set quite low, we used 0.05. It acts to reduce the sensitivity of F to the initial tree choice.

The use of bagging has been shown to improve the efficiency of the GBM algorithm and the accuracy of the final classifier (Friedman 2002). The idea of bagging is that instead of all the training data being used for every tree construction, a fraction (ϕ) is randomly chosen to fit the tree at each step. For the SNPCC we used $\phi = 0.5$. To include bagging, the inner for loop should be modified to read

for i in {sample of size ϕN from integers 1 to N}.

The last modification that needs to be made to complete the GBM algorithm is at step 3 of the main loop. Full line searches for optimal $\gamma_{k,j}$s are not used, instead to speed up the algorithm only the initial step of Newton's method is used:

$$\gamma_{k,j} = \frac{\sum_{X_i \in R_{k,j}} z_i}{\sum_{X_i \in R_{k,j}} |z_i|\,(2 - |z_i|)}. \tag{E1}$$

APPENDIX F: PARAMETER DISTRIBUTIONS

In this appendix we look at how the five light-curve fitting parameters and redshift differ between Ias and non-Ias, and between training and unclassified SNe. Ia SNe cumulative frequency lines are red and thick, while non-Ia SNe are blue and thin. The cumulative frequency lines for training SNe are dotted, while the cumulative frequency lines for the unspecified SNe are solid. This appendix comprises Figs F1–F5.

APPENDIX G: ADDITIONAL FIGURES

This appendix contains two more boosting parameter importance figures: Figs G1 and G2.

APPENDIX H: RANDOM SNe

This appendix contains a random selection of unclassified Ia and non-Ia SNe and their boosting values from representative training.

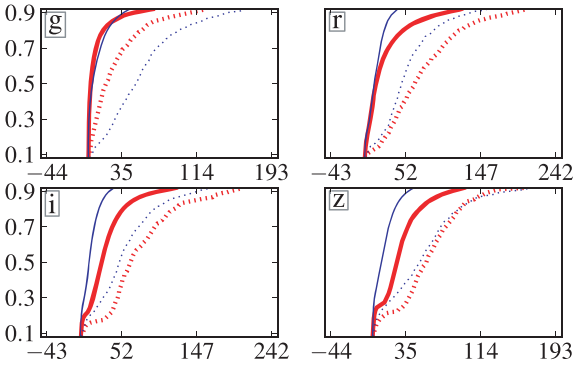

Figure F1. Cumulative plots of parameter A in bands g, r, i, z. Non-representative training (dashed) versus unclassified (solid), and Ia (red, thick) versus non-Ia (blue, thin). In all bands, the magnitude A of SNe is far larger in the non-representative training set than in the unclassified set.

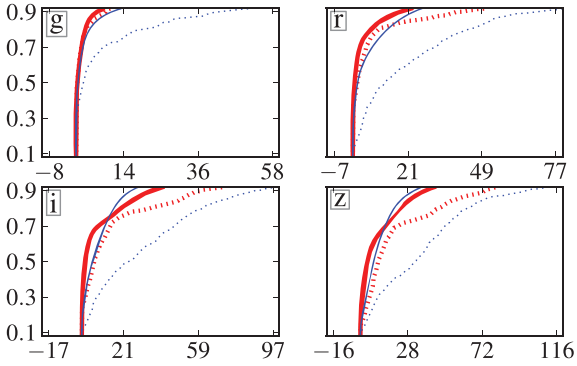

Figure F2. Cumulative plot of parameter tail in bands g, r, i, z. Non-representative training (dashed) versus unclassified (solid), and Ia (red, thick) versus non-Ia (blue, thin).

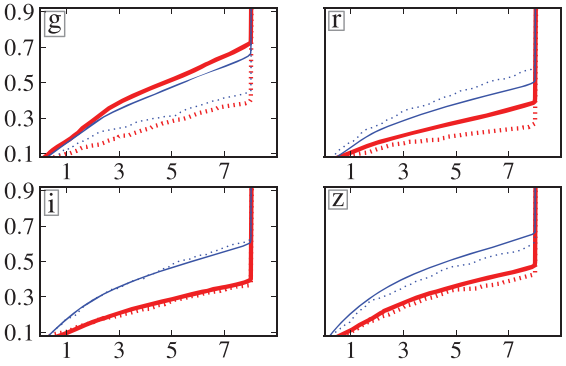

Figure F3. Cumulative plot of parameter k in bands g, r, i, z. Non-representative training (dashed) versus unclassified (solid), and Ia (red, thick) versus non-Ia (blue, thin).

They are Figs H1–H10. Also, had a threshold of zero been used on the boosting value, would the classification have been correct (✓) or incorrect (✗). An extension of this appendix (200 SNe) can be found online at Cosmology at AIMS (2010).

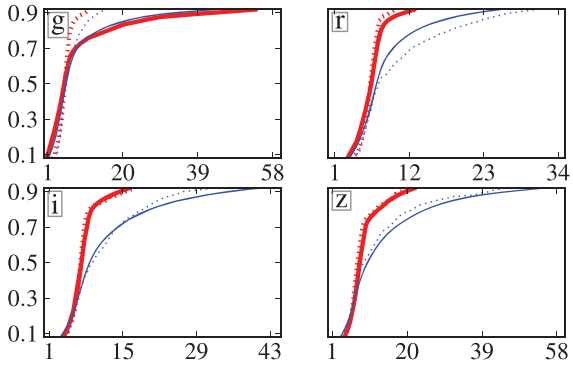

Figure F4. Cumulative plot of parameter σ in bands *g*, *r*, *i*, *z*. Non-representative training (dashed) versus unclassified (solid), and Ia (red, thick) versus non-Ia (blue, thin).

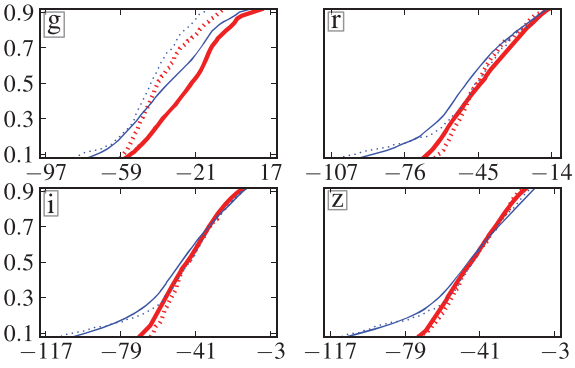

Figure F5. Cumulative plot of parameter ϕ in bands *g*, *r*, *i*, *z*. Non-representative training (dashed) versus unclassified (solid), and Ia (red, thick) versus non-Ia (blue, thin).

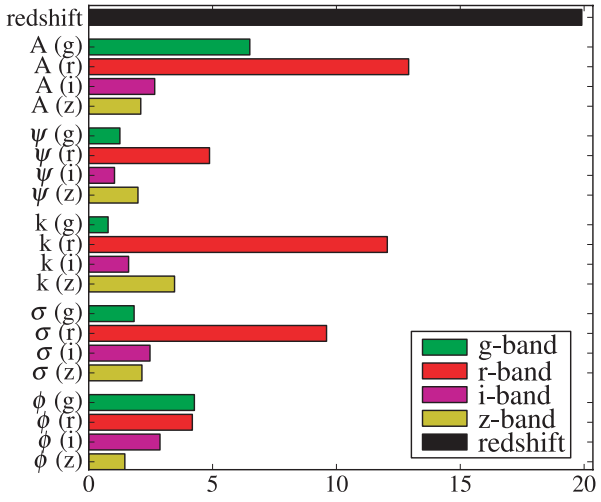

Figure G1. The importance of parameters in distinguishing Ia from non-Ia in the non-representative training sample.

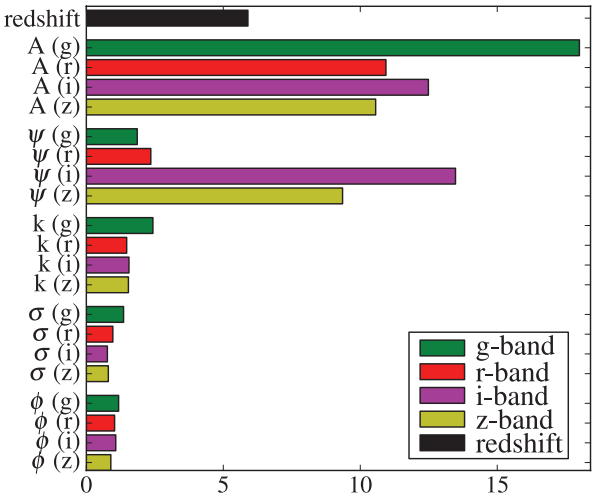

Figure G2. The importance of parameters in distinguishing non-representative training (with spectrum) from unclassified (without spectrum) non-Ia SNe using boosting.

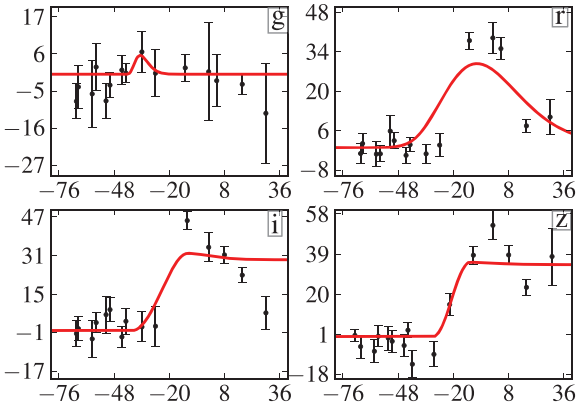

Figure H1. Ia SN at $z = 0.64$. Boosting value of 1.78. ✓

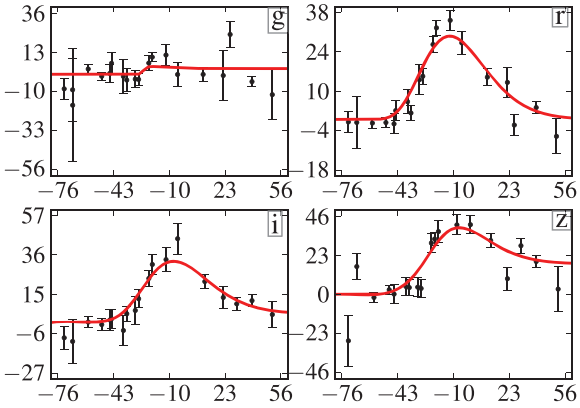

Figure H2. Ia SN at $z = 1.08$. Boosting value of 4.29. ✓

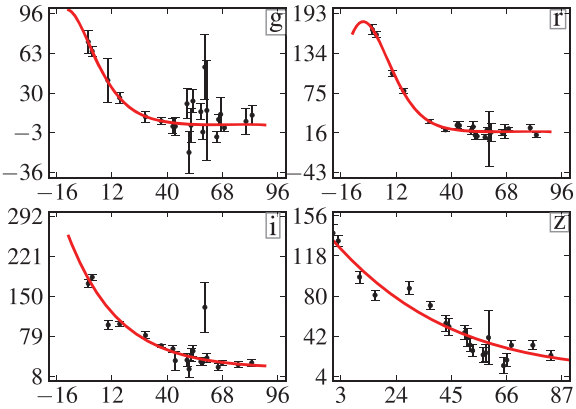

Figure H3. Ia SN at $z = 0.439$. Boosting value of 1.15. ✓

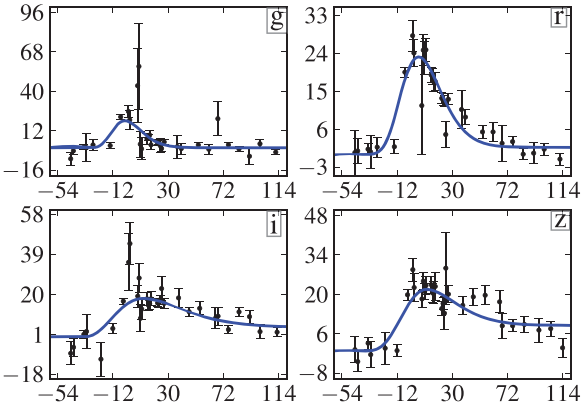

Figure H6. Non-Ia SN at $z = 0.578$. Boosting value of −5.79. ✓

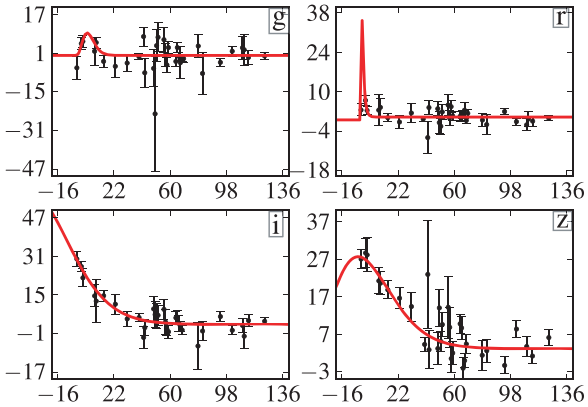

Figure H4. Ia SN at $z = 1.01$. Boosting value of 5.94. ✓

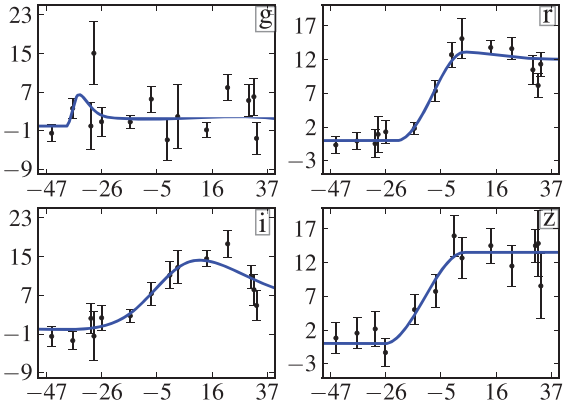

Figure H7. Non-Ia SN at $z = 0.674$. Boosting value of −3.17. ✓

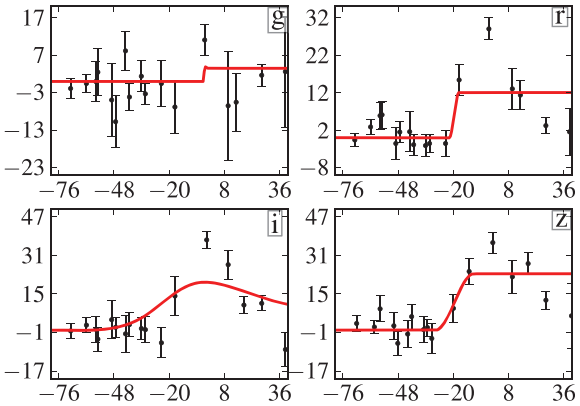

Figure H5. Ia SN at $z = 0.692$. Boosting value of −0.869. ✗

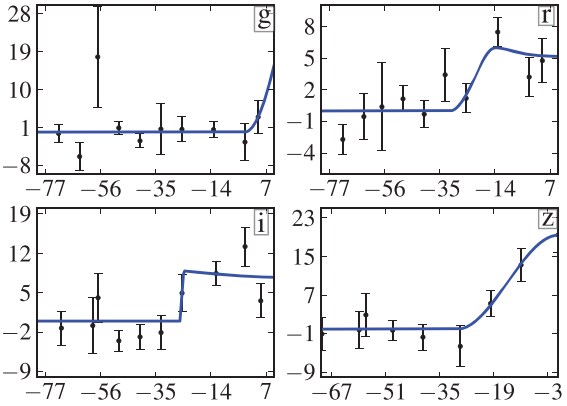

Figure H8. Non-Ia SN at $z = 1.04$. Boosting value of −1.13. ✓

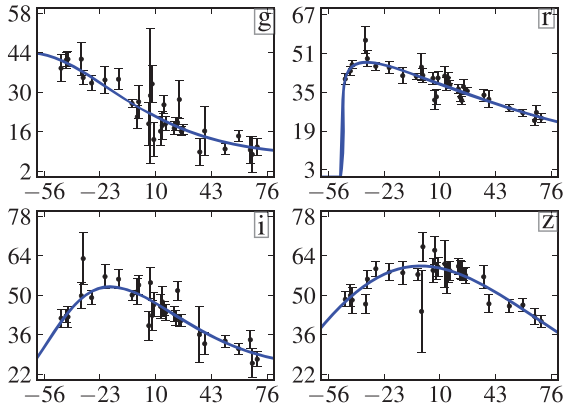

Figure H9. Non-Ia SN at $z = 0.722$. Boosting value of -3.33. ✓

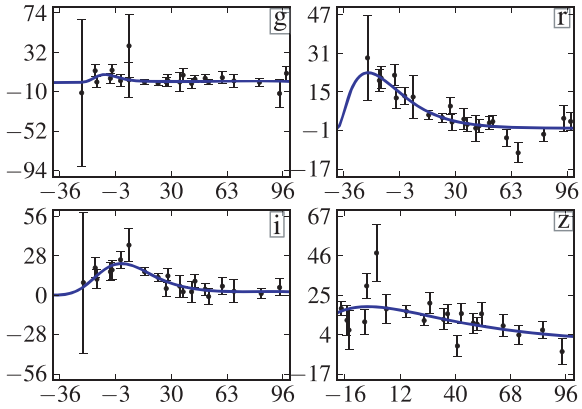

Figure H10. Non-Ia SN at $z = 0.688$. Boosting value of 0.52. ✗

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