A New Method to Classify Type IIP/IIL Supernovae Based on Their Spectra

Shihao Kou, Xingzhuo Chen, and Xuewen Liu

College of Physics, Sichuan University, Chengdu, 610064, People’s Republic Of China: liuxuew@scu.edu.cn, chenxingzhuo@tamu.edu

Received 2018 September 5; revised 2019 December 27; accepted 2019 December 28; published 2020 February 26

Abstract

Type IIP and Type IIL supernovae are defined on the basis of their light curves, but the spectral criteria for distinguishing these two types of supernovae (SNe) remain unclear. We propose a spectral classification method. First, we subtract the principal components of different wavelength bands in the spectra based on the functional principal components analysis method. Then, we use support vector machine and artificial neural network to classify these two types of SNe. The best F1_Score of our classifier is 0.871 for SNe IIL, and 0.974 for SNe IIP. We found that by only using the Hα line at 6150–6800 Å for classification, the F1_Score up to 0.961 for Type IIP, and 0.818 for Type IIL SNe can be obtained. These results indicate that the profile of the Hα spectral line is the key to distinguishing the two types of SNe.

Unified Astronomy Thesaurus concepts: Astronomy data analysis (1858); Supernovae (1668)

1. Introduction

Originating from massive stars (8 ~ 20M☉), hydrogen-rich, core-collapse supernovae, also known as Type II Supernovae (SNe II), are one kind of supernova (SN) with a conspicuous hydrogen spectral line at 6335 Å (Gal-Yam 2017). Some SNe II are generally divided into four subtypes, IIP, IIL, IIb, and InI, while some SNe with peculiar spectroscopic or light-curve features are divided into SNe II-pec, such as SN1987A (McCray 2017). Helium spectral lines at the wavelengths of 5876, 6678, and 7065 Å are characteristic lines observed in SNe IIb, while SNe IIn usually have a narrow Hα line at 6563 Å. Because of similar spectral features, SNe IIP/IIL are defined by their light curves.

The light curves of SNe are generally produced by four different mechanisms (Arcavi 2017): shock breakout; shock cooling and ejecta recombination; radioactive decay; and circumstellar material interaction. The main feature of SNe IIP are their plateau-like light curves, which quickly rise to the peak (~15 days) after explosion and follow a plateau phase (~90 days) that is powered from the cooling of ejecta. After the plateau phase, the luminosity declines linearly (>1.4 Mag/100 day) because of the nickel decay chain 56Ni → 56Co → 56Fe. In contrast, the cooling phase of SNe IIL (typically less than 10 days) is much shorter than that of SNe IIP, and its luminosity linearly decreases as ~0.3Mag/15 day after the peak (Gal-Yam 2017). Unlike the simple shape of SNe IIP/IIL, SNe IIb have versatile light curves, some of which even show a double-peak light curve. The first peak is originated from the fast cooling effect after the explosion, and the second one links to the high-energy photon escaping the relatively small-mass envelope (Richmond et al. 1994; Li et al. 2011; Arcavi 2017).

Many studies indicates that the progenitors of SNe IIP are red super giants (RSG) with a mass range of 8.5–16.5 M☉ (Smartt 2015), whereas SNe IIb are believed originated from yellow super giants (YSG; Anderson et al. 2012). Some studies suggest SNe IIn progenitors are luminous blue variables (Schlegel 1990). The progenitor of SNe IIL is still on debate. Only one observational evidence of SN2009kr suggests that an RSG or YSG could be its progenitor (Fraser et al. 2010; Elias-Rosa et al. 2011).

Previous studies of SNe IIP/IIL showed that the absorption–emission ratio of a Hα P-Cygni profile in SNe IIL is relatively smaller than that of SNe IIP after the luminosity peak, which is possibly because of their different envelope mass and density (Patat et al. 1994; Gutiérrez et al. 2014). Moreover, a study of Hα and O(1)7774 equivalent width (EW) indicates that SNe IIP have a smaller spectral ratio EWα7774/EWα4686 (Faran et al. 2014). In addition, the absolute peak luminosities of SNe IIL/ IIP are −17.44 ± 0.22 and −15.66 ± 0.16, respectively (Li et al. 2011). However, unlike the classification of SNe I, no adequate spectroscopic criteria are presented to distinguish SNe IIP from SNe IIL (Sun & Gal-Yam 2017).

The current classification scheme of SNe, including numerous types and subtypes according to their spectra and photometric properties, is universally recognized, even though there are controversies regarding the definitions and specific progenitor stars (Gal-Yam 2017). In the coming era, existing and upcoming wide-field optical surveys and facilities, e.g., the Large Synoptic Survey Telescope (Ivezic et al. 2019), the Dark Energy Survey (Abbott et al. 2016), the All-Sky Automated Survey for Supernovae (Kochanek et al. 2017), the Panoramic Survey Telescope & Rapid Response System (Pan-STARRS; Chambers & Pan-STARRS Team 2017), and the Zwicky Transient Facilities (Wozniak et al. 2014) will quickly expand the SNe number from a few thousand to millions of events. It is becoming possible to find the direct observational evidence to clarify the controversies in classification by analyzing the large amount of forthcoming data, which needs new effective automatic classification methods. Since the launch of the Supernova Photometric Classification Challenge (SPCC) in 2010 (Kessler et al. 2010), several machine-learning algorithms were proposed for SN classification by using the features of light curves extracted from the SPCC data set, including either parametric fits to the light curves, template fitting, or model-independent wavelet decomposition of the data. Lochner et al. (2016) found that the boost decision trees performed most effectively. Besides photometric features, spectral information is also applied to SN classification. Sasdelli et al. (2016) combined the principal component analysis method with K-means algorithms to use spectral features to classify subtypes of SNe Ia. Muthukrishna et al. (2019) has constructed a 4-layered neural network to automatically estimate the...
The Number of every type II (subtype) Supernovae' Object

![Pie chart showing the distribution of supernovae subtypes.]

The Astrophysical Journal, 890:177 (11pp), 2020 February 20

Kou, Chen, & Liu

redshift and classify the type of SNe based on spectra, which perform quite well on general-type classification (SNe Ia, Ib and Ic, II) but failed in the classification on the subtypes.

Based on spectra of SNe IIP/IIL, for the first time, in this paper, we apply the Functional Principal Component Analysis (FPCA) to extract a series of principle components from SNe IIP/IIL spectra, then feed into the machine-learning algorithm such as the support vector machine (SVM) and artificial neural network (ANN) to classify these two type of SNe. In Section 2, we will introduce the data source and the algorithms (FPCA, SVM, and ANN) used in the paper. In Section 3, we will discuss the performance of our classifiers. The prospective and the summary will be presented in Section 4. All the codes used in the analysis are uploaded onto https://github.com/GeronimoChen/IIL-IIP-SNe.

2. Methods

2.1. Data Overview

The data set includes all spectra of SNe II downloaded from the Weizmann Interactive Supernova Data Repository (WISeREP), which served as an archive of SN spectra and photometry (Yaron & Gal-Yam 2012). To expand the data set, we use the spectra of another six SNe III from the literature as shown in Table 1. The number of different Type of SNe and their spectra are shown in Figure 1.

We discuss in detail the preprocessing of the spectra in Appendix A, which entails removing redshift, Savitz–Golay filtering, and normalizing the spectral flux (Savitzky & Golay 1964). The resulting spectra are divided into nine small wavelength windows as shown in Figure 2. The wavelength windows include the major spectral lines in SN identification as shown in Table 2. (Marcaide et al. 2002; Maguire et al. 2010; Smith et al. 2010; Taddia et al. 2016; Anderson et al. 2018; de Jaeger et al. 2018; Gutiérrez et al. 2018; Singh et al. 2018). We carefully select the H\(\alpha\) and H\(\beta\) windows to include all possible absorption and emission lines, because hydrogen lines are the key lines in defining SNe II. Except for the wave window of “Gap,” all other wave windows are denoted by the dominating spectral lines, such as FeMg, H\(\alpha\), FeOMgSi, Na, H\(\alpha\), NaMg, and Ca. As the “Gap” window may be contaminated by the telluric H\(2\)O spectral line at 7165 Å. However, some SNe II may embody calcium spectral lines (Ca(II)7291,7323) in this region (Maguire et al. 2010; Faran et al. 2014; de Jaeger et al. 2018). But we exclude the wave window of 6800–7000 Å and 7400–7700 Å due to the telluric O\(2\)6867 and O\(2\)7620 contamination. The numbers of the available spectra in each of the wave windows is listed in Table 2. Every individual spectra in different wave windows are normalized with its average value being equal to zero.

2.2. FPCA Algorithm

Now, we can use FPCA algorithm to extract a series of principal components from every single spectrum. The principal components are a set of orthogonal functions that could recover the spectra via their linear combination. The numbers of the available spectra in each of the wave windows is listed in Table 2. Every individual spectra in different wave windows are normalized with its average value being equal to zero.

The Number of every type II (subtype) Supernovae' Spectra

![Spectral analysis diagram showing the distribution of SNe spectra.]

Table 1

| Supernovae Name | Number of Spectra | Type in WISeREP | Reference                  |
|-----------------|-------------------|-----------------|----------------------------|
| SN1979C         | 4                 | II              | (Ray et al. 2001)           |
| SN2013bj        | 5                 | II              | (Bose et al. 2016)          |
| SN2013by        | 18                | II              | (Valenti et al. 2015)       |
| SN2013ej        | 112               | II              | (Yuan et al. 2016)          |
| SN2009kr        | 2                 | II              | (Anderson et al. 2012)      |
| SN1980K         | 6                 | II              | (Eldridge & Tout 2004)      |
| SN1979C         | 4                 | II              | (Ray et al. 2001)           |
| SN2013bj        | 5                 | II              | (Bose et al. 2016)          |
| SN2013by        | 18                | II              | (Valenti et al. 2015)       |
| SN2013ej        | 112               | II              | (Yuan et al. 2016)          |
| SN2009kr        | 2                 | II              | (Anderson et al. 2012)      |

Figure 1. Number of different SNe and their spectra in our analysis.
The convolution form of the $m$-th Score is

$$\beta_{m,n} = \int_{\lambda_{\text{min}}}^{\lambda_{\text{max}}} \phi_m(\lambda)(X_n(\lambda) - \mu(\lambda))d\lambda,$$

(3)

which is sorted by its variance among all spectra:

$$\forall m: \text{Var}_n(\beta_{m,n}) > \text{Var}_n(\beta_{m+1,n}).$$

(4)

The basis function is selected by the following equation:

$$\phi_m(\lambda) = \arg \max_{\phi_n(\lambda)} \{\text{Var}_n(\beta_{m,n})\}$$

$$= \left\{ \phi_m^{(x)} | \text{Var}_n \left( \int_{\lambda_{\text{min}}}^{\lambda_{\text{max}}} (X(\lambda) - \mu(\lambda))\phi_n^{(x)}(\lambda)d\lambda \right) \right\},$$

(5)

which ensures that the most information is extracted from the spectrum $X(\lambda)$.

In this paper, we use the R language package fpca to get the best set of basis functions (Peng & Paul 2007). Because of the limitation of our computational resource, we extract only 30 basis functions from each wave window of every single spectra. Then, we use these basis functions to reconstruct the spectra of SN1990E (SNe IIP, WISEREP ID: 1194) and SN2014G (SNe III, WISEREP ID: 30101), the shape of which match their normalized spectra quite well, as shown in Figure 3. The variance distribution and the fidelity of the basis functions are discussed in Appendix B.

### 2.3. Classification

Then, a series of FPCA Scores extracted from spectra of SNe II are ready for classification. In Figure 4, we show the correlations of the first and second Scores in the H$_\alpha$ wave window of different type of SNe IIP/III/Ib/IIn. Although different patterns appear, it is still impossible to tell the boundaries between different SNe by using only two Scores. To use all of the Scores information, we apply SVM (Chen et al. 2005; Cristianini & Ricci 2008; Chang & Lin 2011) and ANN (LeCun et al. 2015) for the classification of SNe IIP/IIL.

#### 2.3.1. Support Vector Machine

SVM is a popular machine-learning method for classification that aims to find the optimal hyperplane to separate two sets of dots with largest margin distance, as shown in Figure 5. This method was first proposed by Boser et al. (1992).
Consider two kinds of nodes in a hyperspace; the target of SVM is to find a hyperplane that could separate the data, which is written in the form

\[ 0 = X \cdot \omega + b, \]

where \( X \) is the coordination on the hyperplane, \( \omega \) is the weight of each dimension, and \( b \) is the bias. The dots closest to the hyperplane are denoted as support vectors; the distance between the support vector and the hyperplane is

\[ \min_{\omega} \left( \frac{\omega \cdot X_i + b}{||\omega||} \right), \]

where \( X_i \) is the coordination of the nodes. The minimal is reached if the \( X_i \) is the support vector. To construct an optimized hyperplane, the distances between the support vectors and the hyperplane should be as large as possible; the optimization target is

\[ \max_{\omega, b} \min_{\omega} \left( \frac{\omega \cdot X_i + b}{||\omega||} \right). \]

If the distance between the support vector and the hyperplane is denoted to 1, then the optimization target is

\[ \max \left( \frac{1}{||\omega||} \right), \text{ subject to: } y_i (\omega \cdot X_i + b > 1), \]

where \( y_i = \pm 1 \) is the sign of the nodes, which marks their tags (IIP or IIL). Because the nodes may not been intrinsically
The cost function for the optimization is written as
\[
L(y_t, y_p) = -\sum_i [y_t \ln(y_p) + (1 - y_t)\ln(1 - y_p)] ,
\]
where \(y_t\) is the true probability of 0 or 1, and \(y_p\) is the probability given by the neural network (Shannon 2001). The training target of ANN is to modify the weights and bias to minimize the loss function, which make the prediction of “positive” data close to 1 and “negative” data close to 0. In the basic stochastic gradient descent algorithm, the update of weights and bias are following the equation
\[
\theta_{new} = \theta_{old} - \alpha g,\]
where \(\alpha\) is the learning rate, and \(\theta\) is the weights or bias. In each training epoch, the update of weights and biases are uploaded upon the gradient of the loss functions: \(g_w = \frac{\partial L}{\partial w}\) and \(g_b = \frac{\partial L}{\partial b}\). In this paper, we use the Adam algorithm (Kingma & Ba 2014) for optimization, in which the weights and biases are updated with
\[
m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \]
\[
\hat{m}_t = m_t / (1 + \beta_1^t),\]
\[
\hat{v}_t = v_t / (1 + \beta_2^t),\]
\[
\theta_{new} = \theta_{old} - \alpha \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon),\]
where \(m_t\) and \(v_t\) are the \(t\)-th epoch first and second momentum, \(\epsilon = 10^{-8}\) is set to avoid overflow. In this paper, we choose the decay parameters \(\beta_1 = 0.9, \beta_2 = 0.999\) and the learning rate \(\alpha = 0.001\), following the original paper of Adam (Kingma & Ba 2014). To avoid the over fitting problem, we add \(L1\) and \(L2\) penalties into the loss function which is written as
\[
L(y_t, y_p) = -\sum_i [y_t \ln(y_p) + (1 - y_t)\ln(1 - y_p)] + l_1(\Sigma \theta) + l_2(\Sigma \theta^2)^{0.5}.
\]

2.4. Artificial Neural Network

A simple ANN is composed of three parts: the input layer, hidden layers, and the output layer, as shown in Figure 6 (LeCun et al. 2015). Nodes in adjacent layers are linked with different weights. In the ANN, data are propagated from the input layer to the output layer with the equations
\[
z_i = W_i \cdot y_i + b_i ,
\]
where \(W_i\) is the weight matrix, \(b_i\) is the bias, and \(f(z)\) is the activation function within the \(i\)-th layer. The output layer will give the probabilistic classification upon a certain input data. In this paper, we use the rectified linear unit (ReLU) function \(f(z) = \max(0, z)\) as the hidden layer’s activation function, as it constrains the output in the region (0, 1) and allows us to predict the positive and negative. Unlike SVM, we set 1 and 0 for SNe IIP and IIL separately for the application of ANN.

The weight and bias parameters are trained using the back propagation procedure. In this paper, we choose binary cross entropy as the loss function, which is written as
\[
L(y_t, y_p) = -\sum_i [y_t \ln(y_p) + (1 - y_t)\ln(1 - y_p)] ,
\]
where \(y_t\) is the true probability of 0 or 1, and \(y_p\) is the probability given by the neural network (Shannon 2001). The training target of ANN is to modify the weights and bias to minimize the loss function, which make the prediction of “positive” data close to 1 and “negative” data close to 0. In the basic stochastic gradient descent algorithm, the update of weights and bias are following the equation
\[
\theta_{new} = \theta_{old} - \alpha g,\]
where \(\alpha\) is the learning rate, and \(\theta\) is the weights or bias. In each training epoch, the update of weights and biases are uploaded upon the gradient of the loss functions: \(g_w = \frac{\partial L}{\partial w}\) and \(g_b = \frac{\partial L}{\partial b}\). In this paper, we use the Adam algorithm (Kingma & Ba 2014) for optimization, in which the weights and biases are updated with
\[
m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \]
\[
\hat{m}_t = m_t / (1 + \beta_1^t),\]
\[
\hat{v}_t = v_t / (1 + \beta_2^t),\]
\[
\theta_{new} = \theta_{old} - \alpha \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon),\]
where \(m_t\) and \(v_t\) are the \(t\)-th epoch first and second momentum, \(\epsilon = 10^{-8}\) is set to avoid overflow. In this paper, we choose the decay parameters \(\beta_1 = 0.9, \beta_2 = 0.999\) and the learning rate \(\alpha = 0.001\), following the original paper of Adam (Kingma & Ba 2014). To avoid the over fitting problem, we add \(L1\) and \(L2\) penalties into the loss function which is written as
\[
L(y_t, y_p) = -\sum_i [y_t \ln(y_p) + (1 - y_t)\ln(1 - y_p)] + l_1(\Sigma \theta) + l_2(\Sigma \theta^2)^{0.5}.
\]

2.5. Performance Evaluations

Because the number of the available spectra of SNe III/IIP are small, as is shown in Table 2 and Figure 1, we adopt a cross validation method to evaluate the performance. The whole data set is divided into a training set (80%) and a testing set (20%). We use the training set to train the classifier. We evaluate the performances of our classifiers using the F1 Score, which is defined as
\[
\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}},
\]
\[
\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}.
\]
F1\_Score = \frac{2}{\frac{1}{\text{Precision}}} + \frac{1}{\text{Recall}}, \quad (26)

where True Positive is the number of “positive” spectra retrieved by the classifier; True Negative is the number of “negative” spectra that did not retrieved by the classifier; False Positive is the number of “negative” spectra wrongly retrieved by the classifier; and False Negative is the number of “positive” spectra did not retrieved by the classifier (Olson & Delen 2008).

The training and testing processes will be repeated 200 times for SVM algorithm and 20 times for ANN algorithm, and the data set is separated randomly in each iteration. When the training and testing processes are finished, we calculated the average F1\_Score to evaluate the classifier. In addition, the standard errors of F1\_Score are also calculated to evaluate the stability of our classifiers.

### 3. Results and Discussion

First, we only use one wave-window information for the classification including 30 FPCA Scores and two normalization factors. We use keras to construct a three-layered ANN, and apply Adam algorithm as the optimizer. We set the number of nodes in the middle layer to be 40, 100, 400, 600, 800, 1000, 2000, 4000, 6000, and 8000, respectively. Initially, we tried the model with 40, 400, and 4000 hidden nodes, then chose the number of nodes around the best-performance model. When the best number of middle layer nodes that produces the most optimal F1\_Score is determined, we then add the L1 and L2 penalty into the models. The parameters are shown in Table 3.

In the SVM, the parameter \gamma is automatically generated by the program. We set the penalty parameter C to be 300, 1000, 3000, 10,000, 30,000, 100,000, 300,000, 1,000,000, and 3,000,000, respectively. Initially, we calculated the F1\_Score when C is 3000, 30,000, and 300,000, then choose the C around the best-performed model to further increase the F1\_Score. The highest F1\_Scores and their standard errors are plotted in Figure 7. The parameters of the best-performed

**Table 3**

| ANN Parameters for One-window Classification |
|---------------------------------------------|
| Items                | Keys       |
| Input layer          | 32         |
| Hidden layer         | See Table 5|
| Activation function  | ReLU       |
| Output layer         | 1          |
| Activation function  | Sigmoid    |
| Optimization method  | Adam       |
| Regularization       | None       |

Figure 7. Upper panel: the F1\_Score of SVM and ANN for SNe III. Lower panel: the F1\_Score of SVM and ANN for SNe IIP. Red bars are the F1\_Score from ANN and blue bars are the F1\_Score from SVM. The standard error are plotted with black lines on top of the bar.
model are shown in Tables 4 and 5. We notice that the H\textsubscript{\alpha} wave window has the highest F1_Score in all of the wave windows, which reaches 0.961 (IIP) and 0.818 (IIL) when using ANN, while reaching 0.930 (IIP) and 0.724 (IIL) when using SVM. Moreover, ANN performs better than SVM in every wave window. It is also found that the F1_Score of SNe IIP is higher than that of SNe IIL by 0.1 ∼ 0.2, which is probably due to the data size of SNe IIP is larger than SNe IIL.

Because the ANN performs better in the H\textsubscript{\alpha} wave window than in other wave windows, we consider adding the penalty to further increase the F1_Scores in this H\textsubscript{\alpha} window. We try different combinations of L1 and L2: (0.1, 0), (0, 0.1), (0.01, 0), (0, 0.01), (0.001, 0), (0, 0.001), (0.1, 0.1), (0.01, 0.01), and (0.001, 0.001), but none of them performs better than the model without regularization as shown in Table 6.

Now, to maximize the performance, we want to use the overall data with 288 dimensions: 30 FPCA Scores and two normalization factors in each wave window. We applied the same strategies to increase the model performance by modifying the hyperparameters (for example, C in SVM, L1 penalty, L2 penalty, and number of middle layer nodes in ANN). In Table 7, we show the F1_Score of the best-performance SVM and ANN with different parameters. In contrast to the trials in the H\textsubscript{\alpha} wave window, adding L2 penalty in all of the nine wave windows increases the performance of the ANN significantly. We obtain the best performance to our knowledge with the F1_Score of 0.871 (IIL) and 0.974 (IIP).

Whether more FPCA Scores in the ANN will increase the accuracy remains a question. We calculate F1_Scores for different number of FPCA Scores with 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 17, 20, and 25 as shown in Table 8. We adopt 800 nodes in the middle layer for the H\textsubscript{\alpha} window, 2000 nodes for the nine wave windows, and L\textsubscript{2} = 0.01 for penalty. As shown in Figure 8, the F1_Score increases quickly when the dimension of FPCA Scores is less than 10 for H\textsubscript{\alpha} window, while it does not change too much for the overall nine wave windows.

### Table 4
SVM Performance in a Single Wave Window

| Wave Window | C  | IIL F1_Score | IIP F1_Score | IIL F1_Score 1σ | IIP F1_Score 1σ |
|-------------|----|--------------|--------------|-----------------|----------------|
| FeMg        | 3000 | 0.678        | 0.904        | 0.057           | 0.017          |
| H\textsubscript{\alpha} | 3,000,000 | 0.691        | 0.917        | 0.055           | 0.015          |
| FeOMgSi     | 100,000 | 0.638        | 0.899        | 0.064           | 0.018          |
| S           | 3,000,000 | 0.661        | 0.918        | 0.068           | 0.017          |
| Na          | 300,000 | 0.681        | 0.919        | 0.054           | 0.018          |
| H\textsubscript{\alpha} | 300,000 | 0.724        | 0.930        | 0.049           | 0.013          |
| Gap         | 10,000 | 0.669        | 0.912        | 0.047           | 0.017          |
| NaMg        | 300,000 | 0.585        | 0.898        | 0.068           | 0.018          |
| Ca          | 3000   | 0.698        | 0.918        | 0.078           | 0.022          |

### Table 5
ANN Performance in a Single Wave Window

| Wave Window | Middle Layer Units | IIL F1_Score | IIP F1_Score | IIL F1_Score 1σ | IIP F1_Score 1σ |
|-------------|--------------------|--------------|--------------|-----------------|----------------|
| FeMg        | 1000               | 0.732        | 0.936        | 0.063           | 0.013          |
| H\textsubscript{\alpha} | 400 | 0.756        | 0.942        | 0.043           | 0.011          |
| FeOMgSi     | 2000               | 0.696        | 0.930        | 0.034           | 0.011          |
| S           | 4000               | 0.730        | 0.944        | 0.052           | 0.012          |
| Na          | 400                | 0.728        | 0.941        | 0.083           | 0.018          |
| H\textsubscript{\alpha} | 800 | 0.818        | 0.961        | 0.059           | 0.016          |
| Gap         | 4000               | 0.703        | 0.934        | 0.064           | 0.013          |
| NaMg        | 2000               | 0.663        | 0.927        | 0.058           | 0.012          |
| Ca          | 4000               | 0.753        | 0.944        | 0.061           | 0.017          |

### Table 6
ANN Performance in the H\textsubscript{\alpha} Wave Window with Different L1 and L2 Penalties

| L1 Penalty | L2 Penalty | IIL F1_Score | IIP F1_Score | IIL F1_Score 1σ | IIP F1_Score 1σ |
|------------|------------|--------------|--------------|-----------------|----------------|
| 0          | 0          | 0.818        | 0.961        | 0.059           | 0.016          |
| 0.1        | 0          | 0.500        | 0.920        | 0.092           | 0.016          |
| 0.01       | 0          | 0.696        | 0.930        | 0.034           | 0.011          |
| 0.001      | 0          | 0.787        | 0.955        | 0.038           | 0.009          |
| 0          | 0.1        | 0.687        | 0.939        | 0.050           | 0.013          |
| 0          | 0.01       | 0.818        | 0.961        | 0.059           | 0.016          |
| 0          | 0.001      | 0.779        | 0.950        | 0.041           | 0.010          |
| 0.1        | 0.1        | 0.745        | 0.941        | 0.069           | 0.015          |
| 0.01       | 0.01       | 0.766        | 0.949        | 0.055           | 0.011          |
| 0.001      | 0.001      | 0.793        | 0.953        | 0.066           | 0.015          |

Note. The bold parameters obtain the best performance.
Therefore, 30 FPCA Scores is enough for our analysis.

4. Summary

We decompose SNe IIP/IIL spectra into different basis functions via FPCA algorithm. Based on the FPCA scores, we have trained SVM and ANN to classify SNe IIP and SNe IIL, two types of supernovae that are believed to have no adequate spectroscopic discrepancies in previous research. The three-layered ANN has a better performance for the classification. The spectra are divided into nine small-size wave windows for FPCA analysis; the best F1_Score we got is 0.871 for SNe IIL.

Figure 8. Relation between the F1_Score and the number of FPCA Scores. Left panel: SNe IIL. Right panel: SNe IIP.

| Classifier | Wave Window | C (SVM) | Middle Layer | L1 (ANN) | L2 (ANN) | IIL F1 | IIP F1 | IIL F1 1σ | IIP F1 1σ |
|------------|-------------|---------|--------------|----------|----------|--------|--------|------------|-----------|
| SVM        | Hα          | 300,000 | /            | /        | /        | 0.740  | 0.930  | 0.049      | 0.013     |
| SVM        | 9           | 3000,000| /            | /        | /        | 0.810  | 0.962  | 0.073      | 0.015     |
| ANN        | Hα          | /       | 800          | 0        | 0        | 0.818  | 0.961  | 0.059      | 0.012     |
| ANN        | 9           | /       | 2000         | 0        | 0        | 0.836  | 0.968  | 0.090      | 0.015     |
| ANN        | 9           | /       | 2000         | 0        | 0.01     | 0.871  | 0.974  | 0.066      | 0.013     |

Note. The bold parameters obtain the best performance.

Table 8

Performance in Different Dimension of FPCA

| Basis Functions | IIL F1 | IIP F1 | IIL F1 Nine Wave Windows | IIP F1 Nine Wave Windows |
|-----------------|--------|--------|--------------------------|--------------------------|
| 30              | 0.818  | 0.961  | 0.871                    | 0.974                    |
| 25              | 0.806  | 0.958  | 0.831                    | 0.968                    |
| 20              | 0.809  | 0.956  | 0.835                    | 0.971                    |
| 17              | 0.810  | 0.959  | 0.837                    | 0.964                    |
| 14              | 0.786  | 0.952  | 0.814                    | 0.964                    |
| 11              | 0.782  | 0.953  | 0.864                    | 0.970                    |
| 10              | 0.777  | 0.954  | 0.851                    | 0.973                    |
| 9               | 0.740  | 0.944  | 0.844                    | 0.970                    |
| 8               | 0.714  | 0.941  | 0.814                    | 0.967                    |
| 7               | 0.686  | 0.942  | 0.831                    | 0.967                    |
| 6               | 0.680  | 0.939  | 0.836                    | 0.968                    |
| 5               | 0.653  | 0.934  | 0.835                    | 0.968                    |
| 4               | 0.614  | 0.930  | 0.818                    | 0.964                    |
| 3               | 0.511  | 0.920  | 0.842                    | 0.967                    |
| 2               | 0.332  | 0.910  | 0.810                    | 0.964                    |
| 1               | 0.274  | 0.910  | 0.790                    | 0.958                    |

Note. For the Hα wave window, the input dimension is the number of basis function plus two normalization factors. For the nine wave-window situation, the input dimension is the number of basis function multiplied by nine and plus two normalization factors for each nine wave windows. Columns from left to right: the number of basis functions in each wave window; the F1_Score of Type IIL when using Hα for classification; the F1_score of Type IIP when using Hα for classification; the F1_Score of Type IIL when using nine wave windows for classification; the F1_Score of Type IIP when using nine wave windows for classification.
and 0.974 for SNe IIP. It is the first quantitative classification of SNe IIP/IIL through spectral data. We show that compared with other eight wave windows, a better score is obtained in the narrow wave window at 6150–6800 Å, which covers the P-Cygni profile of Hα line. Thus we suggest that FPCA analysis directly on spectral line of a specific element could be a new approach in SNe classification.

In a future study, we will investigate the relationship between the principal components in spectra and the explosion profile of SNe, and expand the classification method to include other Type of SNe.

X.C. would like to thank Prof. Avishay Gal-Yam (Weizmann Institute of Science) and Prof. Lifan Wang (Purple Mountain Observatory) for supportive discussions. We thank Lei Hu (Purple Mountain Observatory) and Tianrui Sun (Purple Mountain Observatory) for support on calculations. We thank Weizmann Interactive Supernova data REPository (WISEREP) https://wiserep.weizmann.ac.il/ and Transient Name Server (TNS) https://wis-tns.weizmann.ac.il/ for the data.

Software: python, keras, scikit-learn, R-fpca (Peng & Paul 2007).

Appendix A
Preprocessing

We first remove the redshift of the raw spectrum by using the equation $\lambda_{\text{RF}} = \lambda_{\text{obs}}/(1+z)$ to transform the wavelength $\lambda_{\text{obs}}$ in the observer frame to the rest frame of $\lambda_{\text{RF}}$. Then the spectra in the rest frame are normalized by dividing the average flux, as the absolute flux of a spectrum in each pixel strongly depends on the telescope and the apparent magnitude. Finally, we use a Savitzky–Golay filter that has been widely used to reduce the noise in SNe spectra (Sun & Gal-Yam 2017). Tucker et al. (2019) smoothed the spectra of SNe Ia by using a two-order Savitzky–Golay filter with a smoothing width of 3000 km s$^{-1}$, which is broader than the noise. By considering the narrow-line feature in SNe II, we adopt a smaller width of 1000 km s$^{-1}$ for our two-order Savitzky–Golay filter. In the next step, each spectrum in the selected wave window is subtracted with its average flux, and divided by the standard deviation. Such a process removes the blackbody component of the spectra. Finally, we resample the spectra to the resolution of 1 Å using linear interpolation method. The whole process is illustrated in Figure 9.

**Figure 9.** Illustrative flow chart of the data-processing steps.
Appendix B

FPCA Fidelity Discussion

We can use the function principal component, average flux, and standard derivative to reconstruct the spectra by the equation

\[ F_n(\lambda) = X_n(\lambda) \times S_n + A_n \]

\[ \approx \left( \mu(\lambda) + \sum_{m=1}^{n_{\text{max}}} \beta_{m,n} \phi_m(\lambda) \right) \times S_n + A_n, \tag{27} \]

where \( A_n \) is the average flux, and \( S_n \) is the standard derivation. We choose all SNe IIP/IIL spectra for FPCA calculation in the H\(_\alpha\), Ca, S, Na, NaMg, and “Gap” wave windows, but only 20% spectra in the H\(_\beta\), FeOMgSi, and FeMg wave windows, due to our computational capability. To evaluate the information loss, we calculate the variance of every FPCA basis function by using the equation

\[ V_m = \frac{\sum_n (\beta_{m,n} - \beta_m)^2}{\sum_n \left( \int_{\lambda_{\text{min}}}^{\lambda_{\text{max}}} X_n(\lambda)^2 d\lambda \right)} \tag{28} \]

where \( V_m \) is the variance of the \( m \)-th order basis function, \( \beta_{m,n} \) is the \( m \)-th Score of the \( n \)-th spectra, \( \beta_m \) is the average of the \( m \)-th order Score of all spectra. We show the contribution of different wave windows to the variance in Figure 10. The first 10-order FPCA contribute more than 99% to the variance.

Appendix C

Possible Data Leak Hazard

Our data set includes 73 spectra of SNe IIL and 337 spectra of SNe IIP, in which 39 are from SN2013ej and 36 are from SN2004et. Because some SNe have more spectra, while other SNe have fewer, we split the total 73 or 337 spectra into the training set (80%) and the testing set (20%). This strategy will lead to the situation where some spectra of a specific SN are in the training set, while the others are in the testing set. Because the spectra from the same SN share similar features, it may lead the deep-learning model to favor the Type of SNe with more spectra. This effect is called “data leak” in the deep-learning field. To investigate this effect our result, we divide all SNe into two groups with roughly same number of spectra: the frequently observed group and the less-observed group. For SN IIP, the frequently observed group contain 165 spectra from nine SNe that have at least 12 spectra in 4200–8900 Å: SN2004et, SN2009N, SN2013am, SN2013K, iPTF13dqy, SN2012aw, SN2006bp, and SN1999em. The rest of the 172 type IIP spectra are in the less-observed group. For SN IIL, 39 spectra of SN2013ej constitute the frequently observed group, and rest IIL SNe are in the less-observed group. We then apply the neural network with the best performance to train and test the ANN for 200 times; the accuracy of each group are shown in Table 9. This result indicates that more data will improve this effect.

![Proportion of variance of each FPCA Score in different wave windows.](image)

Table 9

| SN Type   | Observation Frequency | Accuracy |
|-----------|-----------------------|----------|
| SN IIP    | Frequently observed   | 0.991    |
| SN IIP    | Less observed         | 0.983    |
| SN IIP    | All                   | 0.987    |
| SN IIL    | Frequently observed   | 0.888    |
| SN IIL    | Less observed         | 0.669    |
| SN IIL    | All                   | 0.785    |

References

Abbott, T., Abdalla, F. B., Aleksic, J., et al. 2016, MNRAS, 460, 1270
Anderson, J. P., Dessart, L., Gutiérrez, C. P., et al. 2018, NatAs, 2, 574
Anderson, J. P., Habergham, S. M., James, P. A., & Hamuy, M. 2012, MNRAS, 424, 1372
Arcavi, I. 2017, Hydrogen-Rich Core-Collapse Supernovae (Cham: Springer)
Bose, S., Kumar, B., Misra, K., et al. 2016, MNRAS, 455, 2712
Boser, B. E., Guyon, I. M., & Vapnik, V. N. 1992, in Proc. the Fifth Annual Workshop on Computational Learning Theory. COLT ’92, A Training Algorithm for Optimal Margin Classifiers (New York: ACM)
Chambers, K. C. & Pan-STARRS Team 2017, AAS Meeting Abstracts, 229, 223.03
Chang, C.-C., & Lin, Chih-Jen. 2011, ACM Trans. Intell. Syst. Technol., 2, 27:1
