Multistage Photometric Redshift Estimation

Todd Taomae
University of Hawaii Manoa
ttaomae@hawaii.edu

ABSTRACT

The redshift of a galaxy or other astronomical object can be used to determine its distance from Earth. This makes redshift measurements a valuable asset for learning about the large-scale structure of the universe. However, obtaining accurate redshift measurement requires costly spectroscopic data. Photometric redshift estimations provide a cheaper, but less accurate alternative.

Many attempts have been made to provide accurate photometric redshift estimation. In this paper I propose a new method for estimating photometric redshift. The proposed method uses a multistage classification to allow for more accurate redshift estimation. In the first stage, a multiclass classifier is used to label each object based on which range of redshifts the object belongs in. In the second stage, separate continuous classifiers are used for the objects in each range.

Categories and Subject Descriptors

J.2 [**Physical Sciences and Engineering**]: Astronomy

Keywords

redshift, artificial neural networks

1. INTRODUCTION

Redshift is the effect of light from an object being shifted toward the red end of the visible spectrum. This effect occurs when an object is moving away from the observer. Due to the Doppler effect, the wavelength of the light is increased and the object appears redder. Redshift is quantified as the relative distance between the observed and emitted wavelength of an object and the value is called z.

Due to the expansion of the universe, cosmological objects that are sufficiently far from Earth are redshifted. The amount of redshift is related to the object's distance from Earth. Due to this relationship, cosmological redshift is a valuable tool for learning about the large-scale structure of the universe.

One method for determining the redshift of an object is through spectroscopy. Spectroscopy is the study of light dispersed into its components, such as through the use of a prism. When light from an astronomical object is dispersed in this way, there will be dark lines in its spectrum. These dark lines are called absorption lines and they are a result of certain wavelengths of light being absorbed by the chemicals that make up that object. Objects that are redshifted will have these absorption lines at different frequencies. By comparing the expected and observed wavelengths of the absorption lines, you can determine how much an object is redshifted. Spectroscopy provides an accurate measure of redshift, but is also costly and time consuming.

An alternative to spectroscopy for determining redshift is the use of photometry. Photometry simply measures the amount of light from an object that passes through a variety of filters. Using photometry to determine the redshift of an object is cheaper and quicker, but also less accurate.

Photometric data cannot be directly used to determine the redshift of an object. However, many attempts have been made to estimate photometric redshift. A few examples of attempted methods are Bayesian classifiers [1], decision trees [4], artificial neural networks [3], and regression trees [2]. All existing methods which I have encountered have used a single stage, continuous classification.

In this paper I will attempt to provide more accurate photometric redshift estimations by performing a multistage classification. In the first stage I will perform a discrete classification to label each object as being in one of several ranges. In the second stage separate continuous classifiers are used for each range. This method allows for more fine-tuned estimation for each range.

In the first stage, a number of different classification techniques can be used. The system currently supports naive Bayes classification, k-nearest neighbor classification, support vector machines, and decision trees, however, any multiclass classifier could be used. Similarly, for the second stage, any existing system for photometric redshift estimation could be used. However, the system currently only uses ANNz [3] which is an artificial neural network for estimating photometric redshift.

The system will be tested on two datasets. The first dataset is taken from the Sloan Digital Sky Survey (SDSS) Data Release 1. It contains five bands of photometry data, with errors, and spectroscopic redshift for 12,000 different galaxies. This data is the test data provided with ANNz.

The second dataset consists of 1,000,000 galaxies taken from SDSS Data Release 9. It again has of five bands of photometry data, with errors, and spectroscopic redshift.

The remainder of the paper is structured as follows. Section 2 discusses related work to show what the current state of photometric redshift estimation is. In Section 3, I describe the discrete classification process in more detail. In Section 4, I describe the second step, which is the actual estimation of the photometric redshift. Section 5 will discuss experimental evaluation of my estimation method. Lastly, Section 6 will provide the conclusion.

2. RELATED WORK

Many different techniques have been used to attempt to accurately estimate photometric redshift. A program called PHoto-z Accuracy Testing (PHAT) [5] surveyed 16 different systems which use a variety techniques. These techniques can be broadly divided into two different groups. One group uses an empirical approach which use only the magnitudes of the filter data. The second group tries to fit spectral energy distribution (SED) templates to the photometric data. A spectral energy distribution is a graph of energy emitted at different wavelengths.

There are a number of different template fitting approaches, however, my focus is on the empirical approaches. Among those, some of the techniques used include decision trees, nearest neighbor fitting, regression trees, and artificial neural networks.

One of the empirical systems surveyed by PHAT was an artificial neural network named ANNz [3] which I have incorporated into my own system. Although ANNz did not perform as well as several of the other systems tested by PHAT, the package is easily accessible and contains easy to follow instructions, which made it a prime candidate for my use.

To the best of my knowledge, all of the existing methods use a single, continuous classifier to estimate the photometric redshift. None of the systems that I have encountered have used an discrete approach and none have used a multistage approach.

3. MULTISTAGE PHOTOMETRIC REDSHIFT ESTIMATION

In this system, I take an empirical approach to estimating photometric redshift. That is, I only use the magnitudes from photometric data, rather than a SED template fitting approach. My method uses a multistage classification which will be described in detain in the following subsections. In the first stage, a multiclass classifier attempts to label each object based on which range or redshifts it belongs in. The second stage, a continuous classifier is used for each class from stage one. The intuition behind this approach is that if we broadly classify each object first, then we will be able to get a better estimate of the object's actual redshift by using a classifier that is better trained for a certain range.

In the remainder of this section, I will use the following notation. Each entry in the training data D will have the following form

$$\langle f_0, f_1, ...f_{k-1}, \Delta f_0, \Delta f_1, ...\Delta f_{k-1}, z\rangle$$

where k is the number of filters, f_i is the data for a single filter and z is the spectroscopic redshift. Each entry in the test data T will have the same form, except the last element is optional.

3.1 Discrete Classification

The goal of the discrete classification stage is to place each object into a certain range before actually finding its photometric redshift. For example, we may want to determine if an object has a high, medium, or low redshift.

Dividing Test Data The user specifies the number of classes n, that he wishes to divide the training data into. The training data is then sorted and divided evenly into n groups. Each object is labeled based on which group it is in. This data can then be used by any supervised learning algorithm capable of multiclass classification. Currently, the system supports several different classification methods. They are naive Bayes classification, k-nearest neighbor classification, support vector machines, and decision trees.

There are other possible approaches to dividing the training data. For example, rather than evenly dividing the data, you may want to divide it based on specific values. For example, the user may specify that the data should be divided into groups where z is either less than or greater than 0.3. This method would require that the user have a better understanding of the test data, in order to create sensible labels. However, it has the potential to create more meaningful labels rather than randomly created ones. I have not explored any alternative methods for dividing the training data, but it is a possible topic of future research.

Training and Testing During the training process, the training data D will be divided into n separate sets of training data D_i, each with the appropriately labeled data. Once the classifier has been trained, it can label the test data. Given a set of test data T, it will output n new sets of test data T_i, each containing only the data that has been determined to be within a certain range. If a z value is provided with the test data, it will output the number of mislabeled points.

3.2 Photometric Redshift Estimation

During the second stage, the system will output an actual estimated photometric redshift value for each entry in T. In this paper, I use ANNz to perform this estimation.

ANNz. ANNz is an artificial neural network for estimating photometric redshift. For both testing and training it requires data in the same format as described in Section 3. After being train and tested, it will output the estimated photometric redshift with errors as well as the spectroscopic redshift, if it was provided with the test data.

ANNz allows you to specify the number of hidden layers that the network should have as well as the number of nodes per hidden layer. You can train multiple networks with different random seeds, which can then be used as a committee for estimating the photometric redshift.

ANNz requires both training and validation data, which helps to avoid over-fitting the network. To obtain validation data, I randomly select a portion of D_i to form V_i. The training data then becomes the set difference $E_i = D_i \setminus V_i$.

Training and Testing. Each set of training data D_i is used to train m networks N_{ij}, where $0 \leq j < m$. These networks form a committee C_i which will be used to estimate the redshifts for T_i.

4. EVALUATION

The discrete classification for the first stage was implemented using an open source Python machine learning library called scikit-learn [6]. The second stage uses ANNz with the same settings used in [3]. That is, it has two hidden layers, each with 10 nodes. A committee of five such networks were trained, with approximately one-sixth or one-fifth of the training data being used as validation.

Datasets. Two different datasets are used for testing. Both datasets contain five bands of photometry data, with errors, and spectroscopic redshift, which is considered to be the ground truth. The list of filters used can be seen in Table 1. The first dataset contains 12,000 galaxies, taken from the Sloan Digital Sky Survey (SDSS) Data Release (DR) 1. This data is provided as an example by the ANNz package. The second dataset contains 1,000,000 galaxies taken from SDSS DR9. Both datasets contain redshifts ranging from between 0 and ≈0.6.

Name	Central Wavelength (Å)
u	3551
g	4686
r	6166
i	7480
z	8932

Table 1: List of filters used by SDSS.

4.1 Discrete Classification

First I evaluate different methods of performing discrete classification. For this test, I use the DR9 dataset. I randomly selected 100,000 galaxies for training and another 100,000 galaxies for testing. The methods tested are naive Bayes (NB), k-nearest neighbor (KNN), support vector machine (SVM), and decision trees (DT). I report the number of mislabeled galaxies when dividing the training data into n different classes, where $2 \leq n \leq 6$. The results are shown in Figure 1.

From these results, we can see that the accuracy of all methods is diminished as the number of classes is increased. We can also see that naive Bayes consistently performs poorly for this task. The number of mislabels produced by naive Bayes is approximately twice as many as those produced by the support vector machine method for all cases. In the best case, support vector machines mislabel approximately 5 percent of the test data.

4.2 Overall Performance

Based on the results from Section 4.1, I only use support vector machines to perform the discrete classification in the following experiments. I first compare my multistage estimation technique using two or three classes against the standard ANNz, using the DR1 dataset. My last experiment uses the entire 1,000,000 galaxy dataset to compare a multistage estimation using two classes, versus the standard ANNz.

SDSS DR1. The DR1 dataset provided with ANNz is divided so that 5,000 galaxies are used for training, 1,000 for validation, and 6,000 for testing. To train and test the standard ANNz, I simply used the data as given to train and test the network. For the multistage estimation, I combined the training and validation data to train the discrete classifier. Once the training data has been divided, I randomly

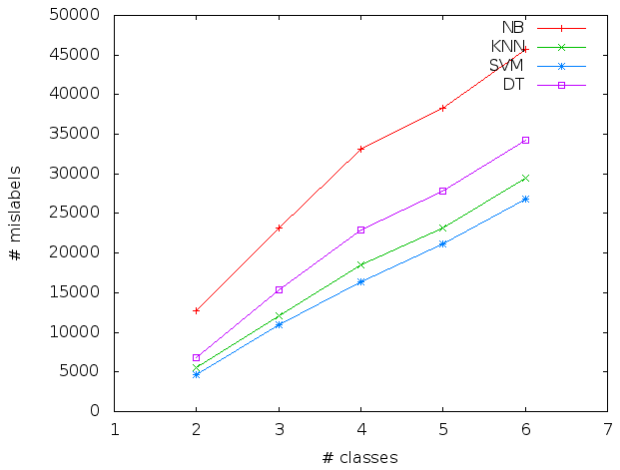


Figure 1: Number of mislabeled galaxies (out of 100,000) versus the number of classes, for four different classification methods.

select one-sixth of the data for validation. For example, when training the classifier for two classes, the 6,000 items in the training data, are divided into two groups of 3,000. From each set of 3,000 training items, I randomly select 500 to use as validation, and the remaining 2,500 are used for training.

The number of galaxies from the test data that were mislabeled are listed in Table 2. As we expect from the results from Section 4.1, using three classes produced more mislabels than only using two classes. However, a significantly higher percentage of objects are mislabeled. Approximately 16 % of objects are mislabeled when using only two classes, in contrast to the 5 % from the previous experiments. This difference is likely due to the significantly smaller size of the dataset.

Figures 2 and 3 shows the from results from this experiment. Figure 2 shows the spectroscopic redshift (ground truth) versus the estimated photometric redshift. Figure 3 shows the same data, except it shows the difference between the two numbers.

When performing a multistage classification, you can clearly see where the different classes are. These are the empty horizontal bands than can seen in Figure 2 and the diagonal bands in Figure 3. This is likely a result of the training data having specific limits for the possible redshift values. This means that the neural network will not be trained to classify any value that is outside of its range of redshift. If the discrete classifier misclassifies a point, the neural network will be completely unable to accurately estimate its redshift. For example, in Figure 2b, it seems that one class contains all points with a redshift less than ≈0.1. Any object that was incorrectly put into that class (i.e. had a spectroscopic-z greater than 0.1) had an estimated photometric-z that was close to 0.1.

One way to combat this problem might be to have some overlap between the training data sets. For example, the class containing objects with lower redshift could use 10% of the training data from the class with the higher redshift objects. This might allow the neural network to better handle objects that were mislabeled. However, it is also possible

that it will cause it to perform worse for objects that are correctly labeled.

I also calculated the average percent error between the spectroscopic-z and photometric-z given by

$$\%error = \frac{|z_{spec} - z_{phot}|}{|z_{spec}|}$$

Before calculating the average error, I removed any outliers, which I considered to be any object with an error greater than 100%. This information is presented in Table 3. Both multistage estimations perform similarly to each other, however, they both do worse than the standard ANNz.

Dataset, classes	Mislabels
DR1, 2	951
DR1, 3	1647
DR9, 2	24033

Table 2: Number of mislabeled galaxies for each experiment.

Dataset, classes	Outliers	Average error
DR1, 1	146	16.170%
DR1, 2	185	18.425%
DR1, 3	152	18.787%
DR9, 1	11776	13.368%
DR9, 2	9823	13.365%

Table 3: Number of outliers and average error for each experiment. One class refers to the standard ANNz.

SDSS DR9. I ran a similar experiment using the DR9 dataset. However, I only performed one multistage estimation using two classes. The dataset was randomly divided into 500,000 galaxies for training and the remaining 500,00 for testing. Once the training data has been divided, randomly select 50,000 out of the 250,000 (one-fifth) to use as validation data and the remaining data is used to train the network.

The number of mislabeled galaxies is presented in Table 2. Approximately 5% of the galaxies are mislabeled, which is comparable to the results from Section 4.1. This is likely due to the fact that the dataset was much larger than the DR1 data which allowed the classifier to be trained better.

The plots of spectroscopic-z versus photometric-z are given in Figures 4 and 5. These results show similar patterns to the DR1 experiment. You can clearly see what the range of the different classes are.

Another thing to note is that neither the standard ANNz or the multistage estimation give photometric redshift values near 0.6 even though there is training data for that range, as demonstrated by the fact that the spectroscopic-z values go all the way to the end of the plot. Although it may not be obvious from the plot, there is significantly less data in this range which means that the network is not well trained for objects in that range.

Table 3 shows the average error for the standard ANNz and the multistage estimation. Again, outliers are assumed to be anything with an error greater than 100%. This time, the multistage estimation performs slightly better ANNz. The average error is almost exactly the same, but the multistage estimation has fewer outliers.

5. CONCLUSION

This paper presents a novel method for estimating photometric redshifts. By first performing a discrete multiclass classification, we can filter the data and pass it to an appropriate classifier that is well trained for a specific range or redshifts. One of the strengths of this system is its modularity. It is fairly simple to adapt it do use different discrete classification method or different photometric redshift estimation methods. Experiments have shown that a naive version of this system can perform comparably to one existing photometric redshift estimation tool.

There are still many ways to improve this system. One direction for improvement that could significantly improve performance is to apply different methods to the second stage. ANNz was far from the best system surveyed by PHAT. There are a variety of alternatives that have a lot of potential to improve the accuracy of the system.

The other direction for further research is to improve the discrete classification. Currently, the system uses the very naive approach of just dividing the data evenly. An alternative would be for the user to specify certain ranges. This could allow a more advanced user with a better understanding of the data, to select ranges that might be better suited to train the second stage classifiers. Another approach would be to use unsupervised learning techniques which might be able to find patterns that are less obvious than just high versus low redshift.

The last area for improvement is to improve the transition from discrete to continuous classification. One possible solution that I mentioned earlier was to mix some of the training data when training the continuous classifiers so that it may be able to correctly estimate objects that were mislabeled in the first stage.

6. REFERENCES

[1] N. Benitez. Bayesian photometric redshift estimation. *The Astrophysical Journal*, 2000.

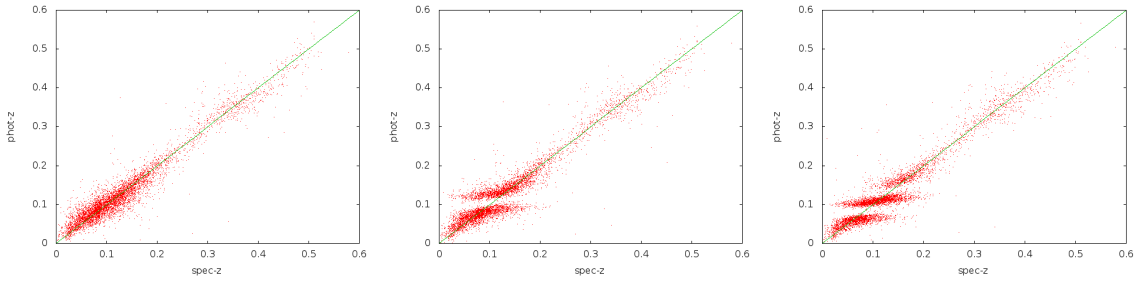
[2] S. Carliles. Rfphotoz, July 2009.

[3] A. A. Collister and O. Lahav. Annz: Estimating photometric redshift using artificial neural networks. *Publications of the Astronomical Society of the Pacific*, 2011.

[4] T. A. M. J. H. M. R. W. R. H. W. M. T. B. David W. Gerdes, Adam J. Sypniewski. Arborz: Photometric redshifts using boosted decision trees. *The Astrophysical Journal*.

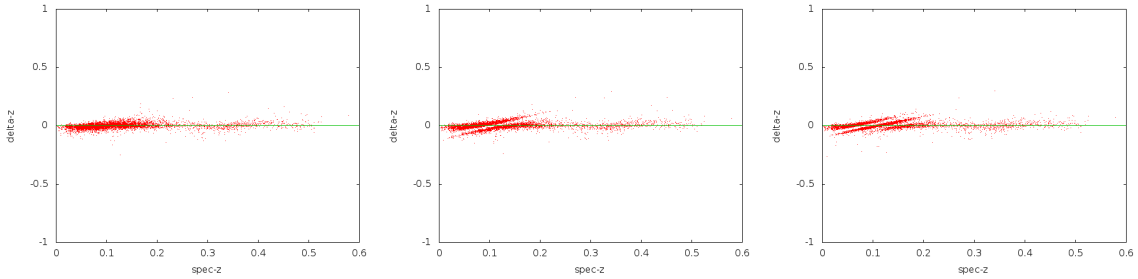
[5] P. C. L. A. M. C. W. F. B. A. R. J. A. M. B. N. B. G. B. B. T. B. S. C. D. C. T. D. R. F. D. G. B. G. O. I. R. K. O. L. I. H. L. J.-M. M. N. P. S. S. J. S. H. Hildebrandt, S. Arnouts. Phat: Photo-z accuracy testing. *Astronomy & Astrophysics*, 2010.

[6] scikit-learn: machine learning in Python `scikit-learn.org/`.



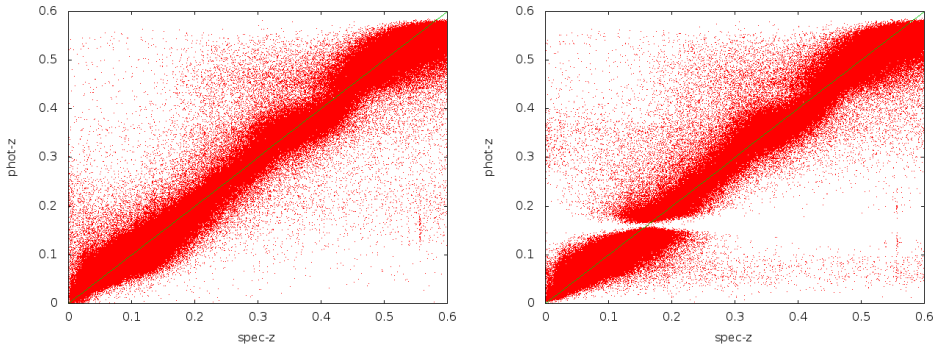
(a) Standard ANNz. (b) Multistage estimation with two classes. (c) Multistage estimation with three classes.

Figure 2: Spectroscopic-z versus photometric-z for SDSS DR1.



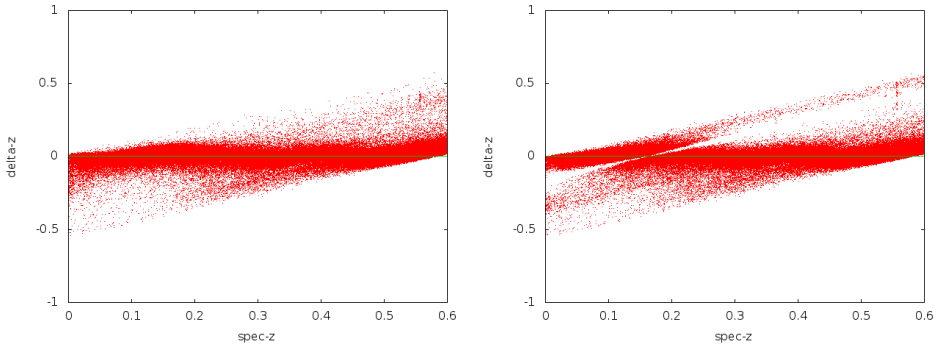
(a) Standard ANNz. (b) Multistage estimation with two classes. (c) Multistage estimation with three classes.

Figure 3: Difference between spectroscopic-z and photometric-z for SDSS DR1.



(a) Standard ANNz. (b) Multistage estimation with two classes.

Figure 4: Spectroscopic-z versus photometric-z for SDSS DR9.



(a) Standard ANNz. (b) Multistage estimation with two classes.

Figure 5: Spectroscopic-z versus photometric-z for SDSS DR9.