TERM PROJECT FINAL REPORT

TROY RAEN

Abstract. In this work I study how the errors in photoz estimates scale with the size of the training set for various machine learning algorithms. A 'photoz' is an estimate of the redshift of a particular object (usually a star or a galaxy) that uses photometric data.

1. Introduction and Related Work

It is well established that the light reaching our telescopes from distant galaxies is shifted toward the red end of the spectrum (relative to the frequency it was originally emitted at), and that the magnitude of this 'redshift' (denoted by z) increases with the galaxy's distance from us (e.g. see [6], [3], [5]). The combined measurements from many galaxies indicate that the universe itself, the space between galaxies, is expanding at a rate that increases with time. The precise calculation of this expansion rate is being pursued and it depends strongly on our ability to make accurate calculations/estimations of the amount by which the light from a distant galaxy is redshifted. (Actually, the calculations of all fundamental quantities in cosmology rely heavily on our ability to accurately calculate these redshifts.)

The calculation of the redshift from measurements of light intensity generally depends on being able to find known features in the intensity as a function of frequency. Poor frequency resolution then increases the error in the redshift.

There are two ways in which telescopes can take measurements: spectroscopy and photometry. Spectroscopy records information about the amount of light coming in over a wide range of the frequency spectrum, at high resolution. Therefore, a redshift calculated from spectroscopic measurements is very precise and can be taken as the true redshift (sometimes called a 'spec-z'). Photometry essentially divides the spectrum into a small number of bins (on the order of 5) and records only aggregated information for each bin. Thus photometry is much cheaper to do and so we have more data of this type. However, this low resolution translates into large errors on our estimates of redshift using this data. A redshift calculated in this way is called a 'photo-z'.

Various machine learning algorithms have been used to estimate photo-z's, with neural nets and random forest regressors showing the most success (see, for example, [7], [4], and [5]). **In this work, I study how the errors in photo-z estimates scale with the size of the training set for selected ML algorithms.**

1.1. Dataset.

I use the dataset `Catalog_Graham+2018_10YearPhot` ([3]) which consists of simulated telescope data (measurements in 6 photometric bins, plus errors on the measurements) for $\sim 3 \times 10^6$ galaxies. The dataset includes the true redshift for each galaxy, so this is a supervised, regression problem.

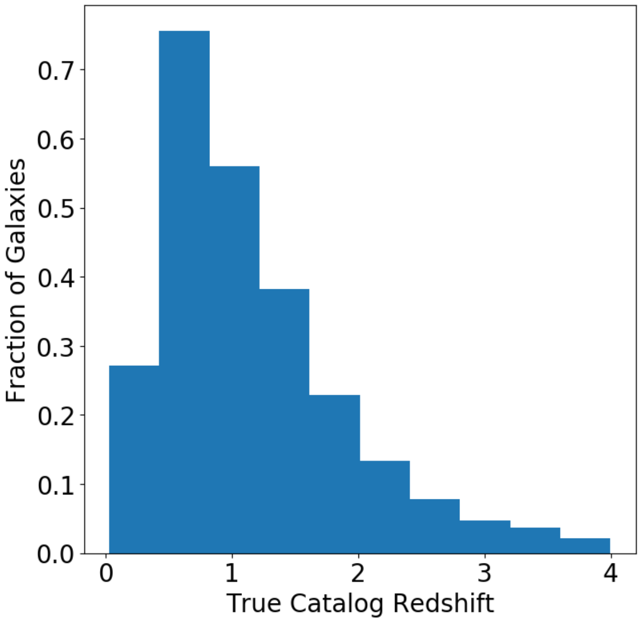
The simulated data is intended to mimic the data anticipated from the upcoming Large Synoptic Survey Telescope (LSST). LSST will collect data from large volumes of the sky and at rates several orders of magnitude above any other telescope to date. The community is making large efforts towards dealing with data at this scale, and one of these efforts is toward quick and accurate photo-z calculations.

1.1.1. *Features:*

Previous algorithms have had more success using a set of transformed features commonly called 'colors'. This transformation is motivated by physics, and is done by subtracting the measurements in adjacent bins. We ensure that no information is lost in this transformation by including the raw measurement from one bin (it doesn't matter which one). So the final feature set includes 5 colors and 1 raw measurement for a total of 6 features.

We could transform the measurement errors, via error propagation, and include them as features. The simplest way to do this is to assume the errors are uncorrelated and add them in quadrature for each color. Indeed, this is what is done in [3]. However, because of the physics at play and the way in which the measurements are taken, the errors are expected to be correlated. I have not taken the time to explore this further, but it could be pursued in future work. (Note that the GPz code requires the inclusion of errors in the feature set, see section 2.1.3 for details).

FIGURE 1. Histogram of true redshifts in the dataset. True redshifts become more difficult to obtain as the redshift increases. Algorithms may have a more difficult time making accurate predictions at higher z because of the sparsity of training data.



	Correlations with Redshift
redshift	1.0
tu	**0.4551**
tg	0.1438
tr	0.2717
ti	0.3933
tz	0.4234
ty	0.4136
u10	0.3932
uerr10	**0.518**
g10	0.1435
gerr10	0.06651
r10	0.2716
rerr10	0.1222
i10	0.3931
ierr10	0.3822
z10	0.423
zerr10	**0.4605**
y10	0.4109
yerr10	0.45

2. Methodology

I follow [6] in evaluating the performance of ML algorithms as a function of sample size. My final results are in the form of curve fits to two statistics (detailed below) calculated on the algorithm predictions. Specifically I fit (using the Python function `scipy.optimize.curve_fit()`) the parameters {a,b,c} in the function $a + b \times N^c$, where N is the training sample size. We are primarily interested in the value of the exponent, c. I show the results of [6] in figure 3 for comparison with my results.

The traditional metric for evaluating photo-z estimates is the scaled difference

$$\Delta z = \frac{photo_z - spec_z}{1 + spec_z}. \tag{1}$$

I evaluate the following two statistics on the metric:

$$\text{NMAD} = 1.48 \times \text{median}(|\Delta z|) \tag{2}$$

$$\text{OUT10} = \frac{1}{N}\sum_{n=1}^{N}\left[|\Delta z_n| > 0.1\right] \tag{3}$$

NMAD is the normalized, median absolute deviation and OUT10 is the fraction of predictions for which $|\Delta z| > 10\%$. OUT10 is an important statistic because photo-z algorithms are prone to catastrophic errors in the predictions, due to both the physics involved and the inherently low resolution of photometry.

Because I aim to evaluate and compare the performance of different algorithms (rather than any specific instance of a trained model), I train 20 models (see 2.1.3 for exception) for each algorithm and training sample size (N) and pool the results before calculating the statistics.

2.1. **Algorithms.** I evaluate and compare the performance of four ML algorithms: 1) Neural Net composed of 2 hidden layers with 10 units each (NN_2x10); 2) Neural Net composed of 3 hidden layers with 15 units each (NN_3x15); 3) Random Forest regressor (RF) composed of bagged decision trees; and 4) GPz which is a publicly available code based on Gaussian Processes and developed specifically for photo-z's.

2.1.1. *Neural Nets.* I train multi-layer neural networks using two different architectures: 2x10 with 2 hidden layers, each with 10 units; and 3x15 with 3 hidden layers, each with 15 units. Both are motivated by approaches in [5] (see sections 4.1.1 DESDM and 4.1.2 ANNZ).

I use the Matlab `fitnet()` function with backpropagation optimized using the Levenberg-Marquardt method. After running a few tests I set the parameters `epochs=500`, `max_fail=50`, `min_grad=1e-10`.

2.1.2. *Random Forest Regression.* I train random forest regression models using the Matlab `fitrensemble()` function. I did some preliminary runs with `OptimizeHyperparameters='auto'` and found the following "best" options:

Method	Bag
NumLearningCycles	495
MinLeafSize	1

Guided by these results I tested some settings and ultimately use `Method='Bag'` with `Learners='tree'`, `MaxNumSplits=Nsamples-1`, and `NumLearningCycles = BLANK`, `Crossval = BLANK`. This generates a random forest model using BLANK decision trees, each trained on a subset of data of size N generated via bootstrap resampling.

EXPLAIN RF REGRESSION

2.1.3. *Gaussian Process Regression using GPz.* GPz is a publicly available code developed specifically for photo_z estimates. The method is described in [2], and the specific code is introduced in [1] and available at `https://github.com/OxfordML/GPz`. Since we studied this type of technique only briefly in the course I will outline the approach in a little more detail than I have done for NN and RF.

A Gaussian Process (GP) is a non-parametric, non-linear, regression algorithm. It assumes output, y, is predicted by some function of the input, $\mathbf{x}$, plus Gaussian noise $\epsilon \sim N(0, \sigma^2)$:

$$y = f(\mathbf{x}) + \epsilon.$$

Then the conditional probability of y given f is Normally distributed as $p(y|f) \sim N(f, \sigma^2)$ and Bayes' Theorem can be used to write

$$p(f|y, \mathbf{X}) = \frac{p(y|f)p(f|\mathbf{X})}{p(y|\mathbf{X})} \tag{4}$$

The prior, $p(f|\mathbf{X})$, is modeled non-parametrically using kernels that model the density around each input point. GPz uses radial basis functions for these kernels. Standard GP models are computationally expensive since there is a kernel for each datapoint and the solution requires us to invert an NxN covariance matrix associated with the kernels. GPz dramatically reduces the complexity by using sparse kernels and maintains performance by optimizing hyperparameters (governing the shape and length scale) that are unique to each kernel rather than standard, global hyperparameters. This allows kernels to specialize on different regions of parameter space, and this flexibility is cited as a key reason for the success of GPz (see [2] for a detailed derivation).

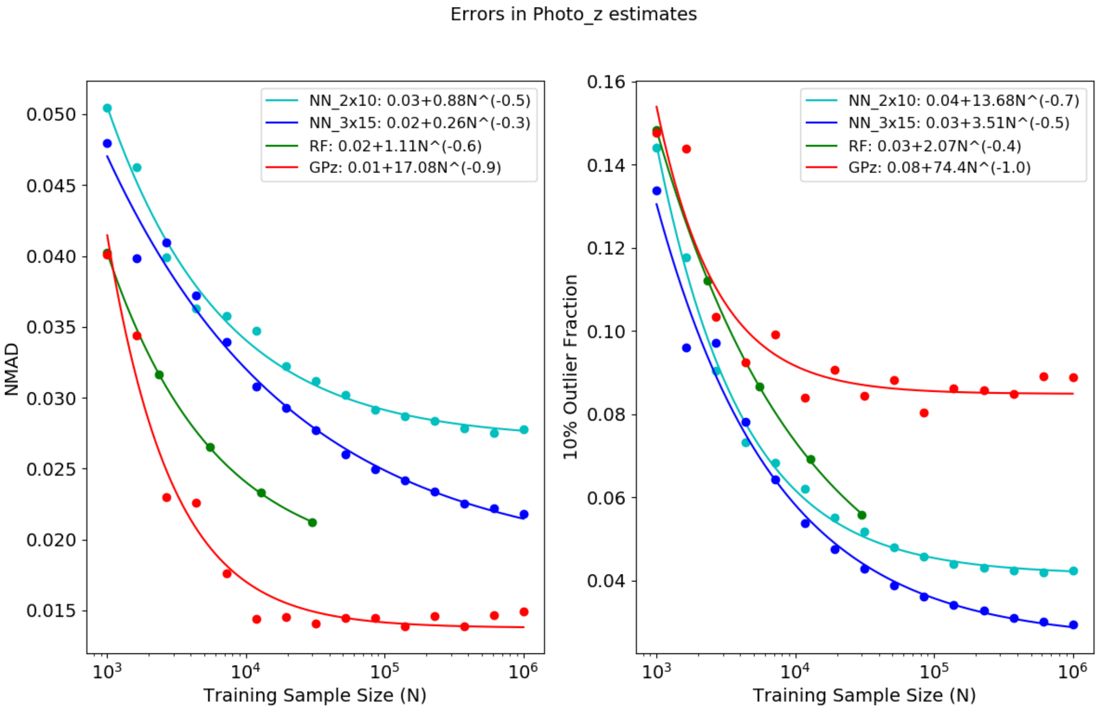
GPz also has a large component that is focused on modeling the input noise and estimating the variance in the prediction due to two factors, the input noise and uncertainty due to the density of training points in a particular region. 'True' redshifts are more difficult to obtain at higher redshifts, and so the training data is not uniformly sampled (see Figure 1) However, I only use the point estimate of (4) in this work, so I will not describe the error calculations.

GPz performs regression to optimize the hyperparameters, and in this process it explicitly minimizes the absolute value of the metric itself ($|\Delta z|$) rather than something more common like the mean squared error.

3. Experimental Results

The results of this work are shown in Figure 2. Figure 3 shows the results of [6] for comparison.

Figure 2. Errors in photo-z estimates as a function of training set size for selected ML algorithms.



4. Analysis of Results and Discussion

Requires the use of the measurement errors, so these models use more information than the NN or RF cases. GPz performs the feature transformation internally, so the feature inputs to this algorithm are the raw measurements, including the errors.

TRY VARYING m, THE NUMBER OF BASIS FUNCTIONS [2] pg 4, but plot 5 shows delta z does not vary much with training size for a given number of basis fncs =¿ this could improve the predictions but should not change shape of scaling with N? Could bring down out10?

5. Conclusion and Future Work

In [1] they use principal component analysis to further pre-process the features, speeds up optimization.

5.1. **Future Work.**

- There is a lot of room here to tune the algorithms more finely and to test other algorithms (e.g. ANNz2, TPz). I spent several days trying to get ANNz2 running, but was unable to get it to install properly.
- There is also the question of why GPz performs better on the pre-packaged dataset than on the dataset I used here. Much could be explored in terms of the differences between the datasets themselves and the models learned from them.
- Experiment with the number of basis functions used by GPz. Find a small number (of order 5) of basis functions that still produces good predictions (must quantify 'good'). Compare these fitted kernel density estimates, and where they lie in parameter space, with two things:
  1. Current physical models that predict specific types of galaxies and what their photometry measurements should look like as a function of redshift. In other words, x-type galaxies at redshift z should live in R-region of parameter space.
  2. Results of unsupervised, clustering methods which may provide insight on how many distinct types of galaxies there are as a function of redshift.
- Density estimation on the subset of the test data with out10 > 10% to see if there are localized regions of the parameter space that are not well predicted. These results could also be compared to (1) and (2) above to search for insight.

Figure 3. Reproduced from Newman et al., 2019 ([6])

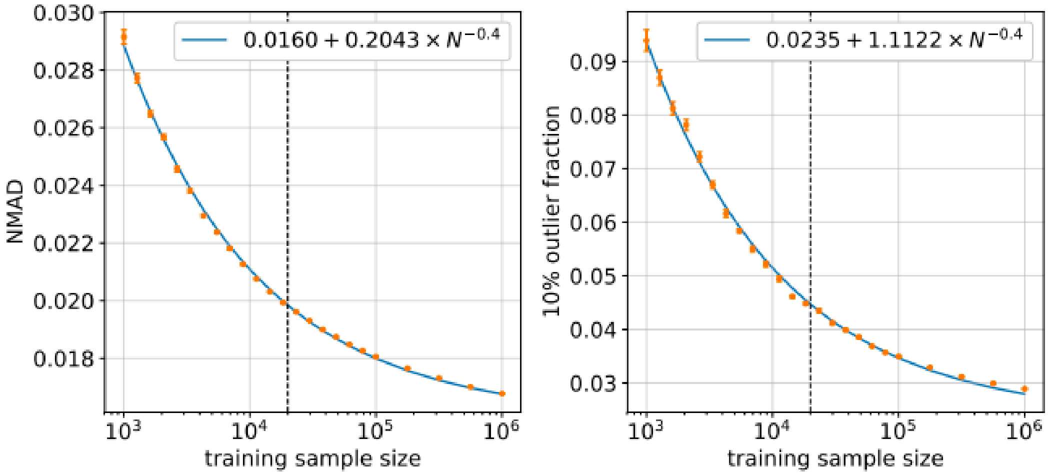


Figure 1: Orange points show photometric redshift errors and outlier rates versus the number of galaxies in the training set for galaxies with simulated LSST photometric errors. Photo-z's were calculated using a random forest regression algorithm. The left panel shows the photo-z error, quantified by the normalized median absolute deviation (NMAD) in $(z_{\text{phot}} - z_{\text{spec}})/(1+z_{\text{spec}})$, as a function of training set size; similarly, the right panel shows the fraction of 10% outliers, i.e. objects with $|z_{\text{phot}} - z_{\text{spec}}|/(1+z_{\text{spec}}) > 0.1$. A vertical dashed line shows the sample size for the baseline training survey from [5]. The blue curves represent simple fits to the measurements as a function of the training set size, N. This analysis uses a set of simulated galaxies from Ref. [11] that spans the redshift range of $0 < z < 4$, using a randomly-selected testing set of 10^5 galaxies for estimating errors and outlier rates.

References

[1] I. A. Almosallam, M. J. Jarvis, and S. J. Roberts. GPZ: Non-stationary sparse Gaussian processes for heteroscedastic uncertainty estimation in photometric redshifts. *Monthly Notices of the Royal Astronomical Society*, 462(1):726–739, 2016.

[2] I. A. Almosallam, S. N. Lindsay, M. J. Jarvis, and S. J. Roberts. A sparse Gaussian process framework for photometric redshift estimation. *Monthly Notices of the Royal Astronomical Society*, 455(3):2387–2401, 2016.

[3] M. L. Graham, A. J. Connolly, Ž. Ivezić, S. J. Schmidt, R. L. Jones, M. Jurić, S. F. Daniel, and P. Yoachim. Photometric Redshifts with the LSST: Evaluating Survey Observing Strategies. 1, 2017.

[4] M. C. Kind and R. J. Brunner. TPZ: Photometric redshift PDFs and ancillary information by using prediction trees and random forests. *Monthly Notices of the Royal Astronomical Society*, 432(2):1483–1501, 2013.

[5] C. Lidman, H. Lin, D. Burke, C. Cunha, N. Greisel, J. Zuntz, A. Fausti, K. Glazebrook, G. Bernstein, J. Gschwend, D. Gerdes, M. A. G. Maia, C. Sánchez, S. S. Allam, A. Dey, E. Sánchez, H. T. Diehl, D. Finley, M. Lima, D. Capozzi, R. Scalzo, A. Sypniewski, S. Jouvel, E. Fernández, I. Sadeh, S. Seitz, J. L. Marshall, J. de Vicente, J. Frieman, A. Walker, P. Pellegrini, I. Sevilla-Noarbe, A. Roodman, M. Carrasco Kind, M. J. Childress, B. X. Santiago, R. C. Nichol, F. J. Castander, A. Carnero, R. Miquel, P. Doel, F. Ostrovski, T. Davis, A. Kim, A. Evrard, F. Valdés, S. Desai, J. Estrada, O. Lahav, C. Bonnett, G. Tarle, F. B. Abdalla, D. L. DePoy, E. Gaztanaga, J. P. Bernstein, R. L. C. Ogando, D. L. Tucker, M. E. C. Swanson, W. Hartley, A. Amara, F. Yuan, K. Honscheid, S. A. Uddin, N. Kuropatkin, T. Abbot, D. Thomas, B. Flaugher, M. M. Rau, R. Brunner, E. Buckley-Geer, L. A. N. da Costa, M. Sako, K. Kuehn, M. Banerji, D. Atlee, P. Martí, M. Makler, and R. C. Smith. Photometric redshift analysis in the Dark Energy Survey Science Verification data. *Monthly Notices of the Royal Astronomical Society*, 445(2):1482–1506, 2014.

[6] J. A. Newman, J. Blazek, N. E. Chisari, D. Clowe, I. Dell'Antonio, E. Gawiser, R. A. Hložek, A. G. Kim, A. von der Linden, M. Lochner, R. Mandelbaum, E. Medezinski, P. Melchior, F. J. Sánchez, S. J. Schmidt, S. Singh, and R. Zhou. Deep Multi-object Spectroscopy to Enhance Dark Energy Science from LSST. 2019.

[7] I. Sadeh. ANNz2 - Photometric redshift and probability density function estimation using machine-learning. *Proceedings of the International Astronomical Union*, 10:316–318, 2015.