The Astrophysical Journal, 712:511–515, 2010 March 20 doi:10.1088/0004-637X/712/1/511

RANDOM FORESTS FOR PHOTOMETRIC REDSHIFTS

Samuel Carliles[1], Tamás Budavári[2], Sébastien Heinis[2], Carey Priebe[3], and Alexander S. Szalay[1,2]
[1] Department of Computer Science, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA; carliles@pha.jhu.edu
[2] Department of Physics and Astronomy, Johns Hopkins University, 3701 San Martin Drive, Baltimore, MD 21218, USA
[3] Department of Applied Mathematics and Statistics, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA
Received 2009 April 15; accepted 2010 February 4; published 2010 March 3

ABSTRACT

The main challenge today in photometric redshift estimation is not in the accuracy but in understanding the uncertainties. We introduce an empirical method based on Random Forests to address these issues. The training algorithm builds a set of optimal decision trees on subsets of the available spectroscopic sample, which provide independent constraints on the redshift of each galaxy. The combined forest estimates have intriguing statistical properties, notable among which are Gaussian errors. We demonstrate the power of our approach on multi-color measurements of the Sloan Digital Sky Survey.

Key words: galaxies: distances and redshifts – methods: data analysis – methods: statistical – techniques: photometric

1. INTRODUCTION

The redshifts of extragalactic sources are accurately determined from spectroscopic measurements. Spectroscopy, however, is limited by the high wavelength resolution, which can only partly be overcome with more observing time. Recently, an increasing number of studies rely on less precise statistical estimates of the redshifts based on more efficient broadband photometry. In fact, these *photometric redshifts* are in the core of many key projects of the upcoming survey telescopes.

Various successful techniques have been developed. Some leverage training sets (e.g., Connolly et al. 1995; Wang, Bahcall & Turner 1998; Brunner et al. 1999; Collister & Lahav 2004), others utilize template spectra for comparisons (e.g., Baum 1962; Coleman, Wu & Weedman 1980; Koo 1985; Gwyn & Hartwick 1996; Sawicki et al. 1997; Benítez 2000; Fernández-Soto et al. 1999; Bruzual & Charlot 2003), and some use a combination of the two (Budavári et al. 2000; Csabai et al. 2000). Applied to the same data sets, these techniques by and large converge, reaching similar accuracy, primarily limited by the systematic errors in the data.

Most redshift estimators today fall short in providing reliable models of the uncertainties. All things being equal, a technique that offers a verifiable model of the estimation error is preferable to one without. Some work on error estimators which analyze performance post hoc has been done, e.g., by Oyaizu et al. (2008a). The method outlined in Carliles et al. (2007) promises precise redshift estimates along with Gaussian errors that reflect the true uncertainty for each source. In this paper, we focus on a new empirical technique borrowed from the arsenal of the machine learning community; a method that has intriguing statistical properties that makes it well suited for photometric redshift estimation.

The structure of the paper is as follows. In Section 2, our choice of method called Random Forest (RF) regression is introduced for the problem of redshift estimation. In Section 3, we apply this new technique to a well-studied data set from the Sloan Digital Sky Survey (SDSS; York et al. 2000) and discuss the results. Section 4 concludes our study.

2. RANDOM FORESTS

Empirical redshift estimation can be viewed as a regression problem, if one believes that the redshift is a function of the photometric observables, e.g., the apparent magnitudes in various passbands. Several parametric and non-parametric methods have been applied to this problem but most are geared toward accuracy and loose control of the uncertainties. Our approach is to focus on the error properties of the estimates, which we achieve by using a method called RF regression (Breiman 2001). The idea is to build independent regression trees on the training set, and to utilize the resulting distribution for characterizing the error and deriving an accurate estimate for each object.

2.1. Regression Trees

Regression trees (Breiman et al. 1984) are used for modeling continuous functions. The trees are built by a deterministic procedure that recursively partitions the training set into a hierarchy of clusters of similar objects. This hierarchy is represented (and stored in an implementation) as a binary tree, whose nodes contain the collections of sources. New nodes of the tree are created by splitting the nodes and their collections in an optimal way. The split at each node is done along one of the axes of the input space (e.g., the SDSS *ugriz* magnitudes), and the choice of which dimension is best to split on is done according to which dimension gives the lowest *resubstitution error* in the resulting subsets. The resubstitution error is equivalent to the standard deviation from the mean along the direction of the desired parameter, i.e., the known spectroscopic redshift $z_{\rm spec}$, summed over the two new subsets,

$$\epsilon_{\rm resubs} = \epsilon_{\rm left} + \epsilon_{\rm right}, \tag{1}$$

with components

$$\epsilon_{\rm left} = \frac{1}{N_{\rm left}} \sum_{z_i \in {\rm left}} (z_i - \bar{z}_{\rm left})^2 \tag{2}$$

$$\epsilon_{\rm right} = \frac{1}{N_{\rm right}} \sum_{z_i \in {\rm right}} (z_i - \bar{z}_{\rm right})^2, \tag{3}$$

where $N_{\rm left}$ and $N_{\rm right}$ are the numbers of objects on the two sides of the splitting point, and $\bar{z}_{\rm left}$ and $\bar{z}_{\rm right}$ are the means of those respective collections. Along each dimension,

there is an optimal split point which will minimize the above score in Equation (1). We choose the best axis according to resubstitution error and we split the node accordingly. The reason for computing the resubstitution error around the mean is that the mean is the optimal parameter estimator for the response (in our case the redshift) of the objects in a given cluster; that is, if you had to pick a scalar value to represent the redshift of all objects in the cluster, the mean would be the optimal choice according to a Euclidean distance metric. We choose to minimize the resubstitution error for the same reason; the resulting splits are optimal according to Euclidean distance. One could try a more robust estimator than the mean, but the mean works well in practice, it is easy to compute, and the behavior of regression trees constructed thusly is well understood. There is also a nice intuitive clustering analog: choosing a split point in this way can be seen as simply doing k-means clustering with $k = 2$.

Regression trees are typically grown fully, that is, until each leaf node contains only one value, and then *pruned* back to optimize performance on cross-validation data sets. If branching at a given node does not improve performance on test data, branches from that node are cut off.

2.2. Randomized Trees

RFs are ensembles of regression trees trained on *bootstrap* samples. Given a training set D of size N, a bootstrap sample is a subset of D selected by choosing N objects from D with replacement (Efron & Tibshirani 1994). The idea is that one can generate various bootstrap samples and train separate predictors (in our case, build regression trees) with those samples to produce many different estimates. One can then average these estimates into a more robust aggregate. This is called *bootstrap aggregating*, or *bagging* (Breiman 1996), and in addition to giving an accurate estimate, it also provides a distribution of the individual estimates. It has been shown that bagging predictors using bootstrap samples drawn from the population reduce variance (Hastie et al. 2001, p. 247). The assumption then is that this holds to some extent with bootstrap samples drawn from the data as is the only option in the real world. In practice, the amount of variance reduction gained per additional tree is determined empirically for each data set by increasing the forest size until the variance on a held-out test set appears to converge to a lower limit.

RF regression trees also introduce some additional randomization beyond what results from the bootstrap process. At each node, rather than splitting on the best dimension from the input space, the split is done on the best dimension from a random subspace (Ho 1998). For example, out of the u, g, r, i, and z dimensions, a node might randomly choose only three of these dimensions to consider, say u, r, and z. Each node chooses its random subspace independently of all other nodes, including parent nodes. This random subspace method helps to increase independence between trees, and it has the additional benefit of reducing computational cost. Each tree branch is grown until a user-specified minimum number of training objects (commonly five) is reached in a node and the tree does not branch any further from that node. Its value is then defined as the mean of the known redshifts associated with the objects it contains. After all trees in a RF are grown, the forest can predict responses to new input points. A query point is classified left or right starting at the root of each tree in the forest, moving to the next level until it reaches a leaf node, and the aggregate estimate $z_{\rm phot}$ for the new object is defined as the mean of these leaf node values.

3. APPLICATION TO SDSS GALAXIES

Now we turn to apply the above method to multi-color observations of galaxies in the SDSS. The SDSS is two surveys in one: it is a photometric survey that takes multi-color images of the sky on the best nights in five passbands u, g, r, i, z, and it is also a spectroscopic survey that spends most of the time measuring the spectra of close to a million objects. For our exercise, we use a subset of the Main Galaxy Sample (MGS; Strauss et al. 2002) in the Data Release 6 catalog (Adelman-McCarthy et al. 2008).

3.1. Sample Selection

To ensure high data quality we use a strict set of selection criteria. The ranges are chosen to eliminate erroneous measurements that are obvious outliers. We expect RFs to be somewhat robust to missing and erroneous data as the randomization process reduces the reliance on any particular data element. Breiman and Cutler describe several possible approaches to dealing with missing values.[4] However, our primary concern is the development of good estimates with error distributions, so we choose to take advantage of the large amount of complete and accurate data available in SDSS. We perform the final selection of the training and test sets using the SDSS Science Archive stored in an SQL Server database engine. The Catalog Archive Server is searched via the online CasJobs Web site[5] using the following SQL command:

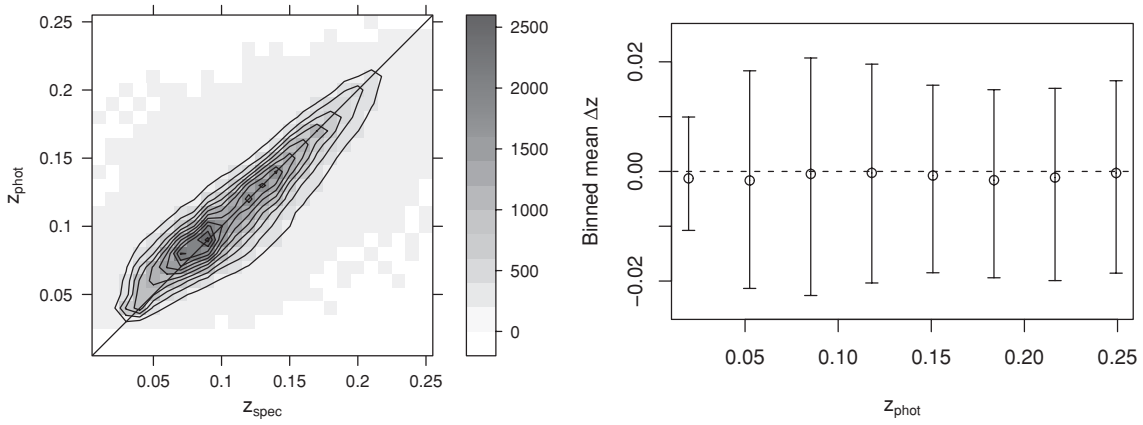
```
SELECT a.SpecObjID, a.ObjID, a.PrimTarget, a.z, ...
FROM SpecPhoto a
     JOIN UberCal b ON b.ObjID = a.ObjID
WHERE a.SpecClass = 2 - Galaxies
     AND a.SciencePrimary = 1
     AND a.ZConf > 0.9 AND a.ZErr < 0.1
     AND a.z BETWEEN 0.0001 AND 1
     AND (a.ZWarning & 0xFFFF1B10) = 0
     AND (a.PrimTarget & 448) > 0
     AND a.ModelMag_U > -9999 AND...
     AND b.ModelMag_U > -9999 AND...
GO
```

Here, we select highly confident redshifts, extinction-corrected magnitudes including ones using the so-called *über-calibration* (Padmanabhan et al. 2008) and the sets of flag bits from PrimTarget, which allows us to select the appropriate target categories. In particular, we select only galaxies, chosen both from the SDSS MGS and the luminous red galaxies (LRGs).

3.2. Results

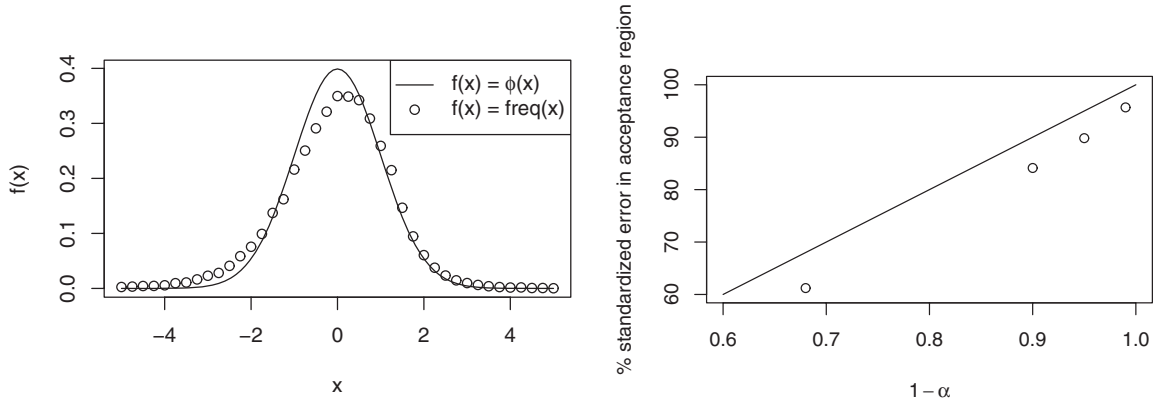
We constructed a forest of 400 trees trained on 80,000 objects to estimate redshifts for 100,000 previously held-out test objects. For a given test point, a forest with B trees provides B estimates of the redshift, $\{z_i\}$. Then for this particular input, the aggregate estimate for the redshift, $z_{\rm phot}$, is the mean of these z_i estimates. We also evaluated the trimmed mean (eliminating those z_i outside of 2σ of their mean,) and the results were virtually identical. The rms error between trimmed means and

[4] http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm
[5] http://casjobs.sdss.org/CasJobs/



(a) (b)

Figure 1. (a) Photometric vs. spectroscopic redshift for 100,000 test objects distributed into 25 bins along each axis with seven levels. (b) Mean error for 100,000 test objects in eight bins vs. photometric redshift with bars marking region containing 34% of errors on either size of the mean.



(a) (b)

Figure 2. (a) Observed standardized error ($\epsilon_{\text{phot}}/\sigma_\epsilon$) for 100,000 test objects binned (circles) and standard normal distribution (curve). (b) Percent observed standardized error within level-α critical values for 100,000 test objects vs. $1-\alpha$ (circles) and percent error expected within level-α critical values vs. $1-\alpha$ (line).

corresponding spectroscopic redshifts is 0.023. The character of our estimates over the usable range for our methodology is shown in Figure 1(a). The estimates are generally good, with some slight bias visible near the origin due to the local skewness of the underlying distribution of redshifts. The average difference between z_{phot} and z_{spec} is shown as a function of z_{phot} in Figure 1(b). This shows that over the usable range, given what we believe z_{phot} should be, we are just as likely to err low or high, meaning that what z_{spec}-dependent bias we have is dominated by variance in the same z_{spec} neighborhood.

For a given test point, each z_i estimate has an associated estimation error, $\epsilon_i \equiv z_i - z_{\text{spec}}$, and we define the aggregate estimation error as

$$\epsilon_{\text{phot}} \equiv z_{\text{phot}} - z_{\text{spec}}, \tag{4}$$

which is equivalent to the mean of the ϵ_i values. We can think of the z_i as realizations of identically distributed random variables with some physical mean. Since z_{phot} is the sample mean of these z_i, there is a central limiting behavior; z_{phot} tends toward the physical mean. Under ideal conditions ($B \to \infty$, independence among the z_i), the central limit theorem would give us the distribution from which z_{phot} is drawn. If the mapping from color to redshift space were non-degenerate, the physical mean would be equivalent to z_{spec}, and this would give us the distribution from which ϵ_{phot} is drawn, i.e., the distribution of our estimation error. Following this intuition and applying it to our SDSS galaxy sample leads us to a useful estimate of this distribution. To wit, we observe that

$$\frac{\epsilon_{\text{phot}}}{\sigma_\epsilon} \sim N(0,1) \text{ approximately}, \tag{5}$$

where σ_ϵ indicates the standard deviation of the tree error realizations ϵ_i. Figure 2(a) shows a histogram of errors standardized and plotted along with the Gaussian distribution. The agreement is striking, though there is a slight skew which is anticipated by Figure 1(a); the galaxy distribution is more concentrated at lower values of z_{spec}, where we tend to overestimate. Still, the agreement is remarkable, even if not perfect.

As an additional sanity check, we can compute the percentage of our standardized observed errors that fall within the level-α critical values for a given α to see how well this compares with $1-\alpha$, the area under the standard normal curve between those critical values. For instance, if $\alpha = 0.05$, the area under the lower tail of the standard normal curve is $\alpha/2 = 0.025$ as is the area under the upper tail; thus the area between the tails is 0.95. We therefore expect 95% of our standardized errors to fall between the boundaries delimiting the tails under the assumption that our errors are normal. We do this test for several values of α and plot the results in a quantile–quantile plot in Figure 2(b). Again, though the results are not perfect, they are very close to what we expect. Figure 2(a) anticipates that not quite as

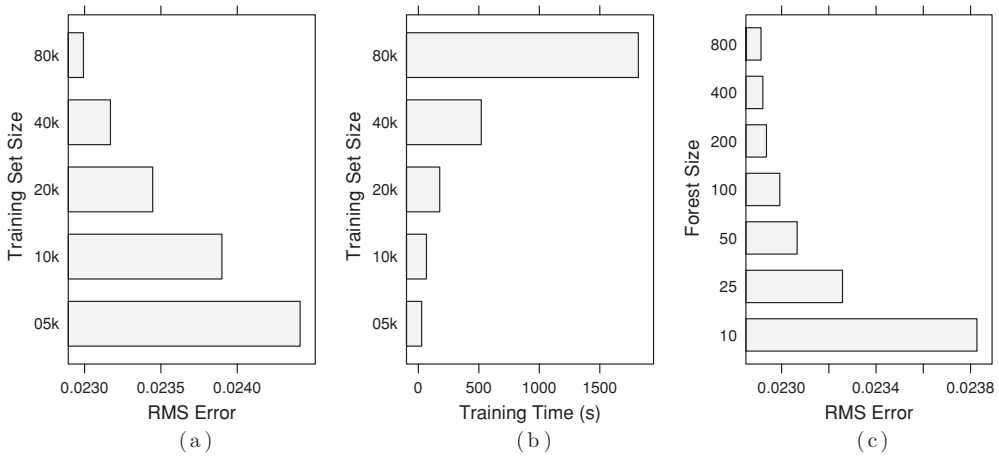


Figure 3. (a) rms error for estimates on 10,000 test objects given by forests of 100 trees trained on 5000, 10,000, 20,000, 40,000, and 80,000 objects. (b) Training time for a forest of 100 trees trained on 5000, 10,000, 20,000, 40,000, and 80,000 objects. (c) rms error for estimates on 10,000 test objects given by forests of 10–800 trees trained on 80,000 objects.

many of our standardized errors are near zero as should be, so, for instance, roughly 61% of our errors fall within the critical values for $\alpha = 0.32$, corresponding approximately to the range within one standard deviation of zero, where we expect to see 68%. Notwithstanding this imperfection, Figure 2(b) indicates that our errors fall within prescribed bounds with nearly the correct probability.

We wish to emphasize that though we standardize our errors for the purposes of analyzing their behavior in the aggregate, before standardization they are intrinsically unique *per-object* error distribution estimates. The value of σ_ϵ is unique to each new test object; it reflects something about the quality of the input data and our confidence in our estimate for this particular observation.

3.3. Practical Details of the Procedure

In separate tests from those described above, we measure how RF performance scales with data and with forest size. We generate training sets of sizes 80,000, 40,000, 20,000, 10,000, and 5000 by first uniformly sampling the full result set without replacement, and then uniformly sampling the resulting subset (again without replacement), so that smaller training sets are proper subsets of each larger set.

For each training set size, we train eight forests of 100 trees each. Since final estimates can be aggregated from any number of tree predictions, this allows us to observe how the quality of estimates scales with forest size; in our case, from 100 to 800 trees. It also has the additional benefit of allowing the computation to be done on a machine with a "modest" amount of memory. Constructing trees with different training set sizes, of course, allows us to observe how the quality of estimates scales with training set size.

We observe the accuracy one could expect to see for various training set and forest sizes. We trained RFs of 100 trees on training sets of sizes 5000, 10,000, 20,000, 40,000, and 80,000 using colors from the über-calibration. Performance is then tested on a random subset of 10,000 held-out objects. The resulting rms error over these 10,000 objects is shown in Figure 3(a). The training times for these forests are shown in Figure 3(b). The gain in accuracy bought by larger training sets is significant. Predictably, so is the gain in training time. Since on average one can train one tree as fast as another with the same amount of data, one does not need a plot to see that training timescales linearly with the number of trees. In our tests, all work was done on an Apple MacBook equipped with a 2.0 GHz Intel Core 2 Duo processor with 2GB RAM using R version 2.5.1 with the Random Forest package version 4.5–18.[6]

Next, we tested the effect of increasing the number of trees in the forest. We trained eight RFs of 100 trees each on a training set of 80,000 objects using ubercal colors. For our random subset of 10,000 held-out objects, we computed aggregate estimates using individual predictions from first one tree, then two trees, then three, and so on for effective forest sizes of 1 up to 800. The resulting rms error over these 10,000 objects is shown in Figure 3(c). For this training set with the bootstrap sample size we used, the rms error on this test set stops improving significantly beyond the first 50 trees. With each additional tree, the forest converges toward a limiting error which is intrinsic to the data. Between 50 and 800 trees the gain in rms error is only about 1%, and certainly by the time we had trained 200 trees we had reached a point of diminishing returns. One should note that the rate of convergence will depend on the forest parameters such as bootstrap sample size and training sample size.

4. DISCUSSION

The performance of RFs on the SDSS MGS is comparable to that of other machine learning methods, e.g., artificial neural networks (Oyaizu et al. 2008b). Measured over the entire test set, the RF error is nearly mean zero, i.e., RFs are essentially unbiased; however, they exhibit the same boundary biases common to other empirical methods (Figure 1(a)). We suspect that due to systematic errors in the input data RFs and other empirical methods may have come close to a lower bound on error achievable by treating photo-z as a regression problem. This issue is discussed in more detail in Budavári (2009).

Our error estimation method appears to perform comparably to the Nearest Neighbor Error method of Oyaizu et al. (2008a), yielding normally distributed errors with accurate per-object parameter estimates. Our method has the additional benefit

[6] The R statistical computing environment, as well as the Random Forest package for R, may be downloaded from http://www.r-project.org/. Sample R code along with a small subset of our DR6 data selection suitable for demonstrating the methodology is available at http://www.sdss.jhu.edu/~carliles/photoZ/.

of being applicable to arbitrary unknown data distributions with strong theoretical support and some informal empirical confirmation using highly skewed synthetic data. Theoretically speaking, our estimates are means, and thus there is a central limiting effect regardless of the distribution of the underlying errors—the underlying process need not be Gaussian. And though a rigorous theoretical explanation for why our process works as it does is much more subtle than just applying the central limit theorem, this is still the strong tendency and the reason that a good characterization of the error distribution is possible. A rigorous theoretical explanation of why the process works as it does is forthcoming.

More generally, RF regression is appealing because it overcomes several crucial weaknesses of other regression techniques. As with other non-parametric techniques, RFs impose no statistical model on the underlying data. Parametric methods all require a model which must be well suited to the underlying data distribution in order to perform well. Other non-parametric methods almost invariably side-step the issue of error estimation entirely. In contrast, RFs yield reliable error distribution estimates even on data with highly skewed noise distributions, and there is strong theoretical support to explain this behavior. Thus, RF regression offers a robust error model in addition to the flexibility of non-parametric methods. Furthermore, RFs have been shown to converge to a limiting generalization error (Breiman 2001). Lastly, in the course of developing our own forthcoming RF code, we have discovered that RFs can be implemented in a computationally much more efficient and scalable way than the widely used R version.

RF regression improves the utility of redshift estimates by giving us good measurements of the estimation error, and thus compares favorably to other methods giving comparable estimates. Care should be exercised in estimating the per-object variance of the estimation error, and it may be necessary to estimate a scaling factor of the variance estimates empirically for each new training set and RF configuration. Given the performance and the reliable per-object estimation error distributions offered by RFs, they represent an attractive alternative to other photo-z methodologies. Future work will likely include extending the RF technique to provide a redshift distribution estimate rather than a single scalar estimate, improving the quality of estimates (for instance, by weighting training object contributions according to quality measures such as magnitude errors), and attempting to extrapolate to new objects not represented by the training data.

We acknowledge support from the W. M. Keck Foundation and the Gordon and Betty Moore Foundation as part of the Institute for Data Intensive Engineering and Science (IDIES) effort at JHU. We also thank Tin Kam Ho for her input and expertise on the Random Forest regression method.

REFERENCES

Adelman-McCarthy, J. K., et al. 2008, ApJS, 175, 297
Baum, W. A. 1962, in IAU Symp. Ser.15, Problems of Extra-Galactic Research, ed. G.C. McVittie (Cambridge: Cambridge Univ. Press), 390
Benítez, N. 2000, ApJ, 536, 571
Breiman, L. 1996, Mach. Learn., 24, 123
Breiman, L. 2001, Mach. Learn., 45, 5
Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. 1984, Classification and Regression Trees (Belmont, CA: Wadsworth)
Brunner, R. J., Connolly, A. J., & Szalay, A. S. 1999, ApJ, 516, 563
Bruzual, G., & Charlot, S. 2003, MNRAS, 344, 1000
Budavári, T. 2009, ApJ, 695, 747
Budavári, T., Szalay, A., Connolly, A., Csabai, I., & Dickinson, M. 2000, AJ, 120, 1588
Carliles, S., Budavári, T., Heinis, S., Priebe, C., & Szalay, A. 2007, in ASP Conf. Ser. 394, Astronomical Data Analysis Software and Systems XVII, ed. R. W. Argyle, P. S. Bunclark, & J. R. Lewis (San Francisco, CA: ASP), 521
Coleman, G. D., Wu., C.-C., & Weedman, D. W. 1980, ApJS, 43, 393
Collister, A., & Lahav, O. 2004, PASP, 116, 345
Connolly, A. J., Csabai, I., Szalay, A. S., Koo, D. C., Kron, R. G., & Munn, J. A. 1995, AJ, 110, 2655
Csabai, I., Connolly, A. J., Szalay, A. S., & Budavári, T. 2000, AJ, 119, 69
Efron, B., & Tibshirani, R. 1994, An Introduction to the Bootstrap (New York: Chapman & Hall)
Fernández-Soto, A., Lanzetta, K. M., & Yahil, A. 1999, ApJ, 513, 34
Gwyn, S. D. J., & Hartwick, F. D. A. 1996, ApJ, 468, L77
Hastie, T., Tibshirani, R., & Friedman, J. 2001, The Elements of Statistical Learning: Data Mining, Inference, and Prediction (New York: Springer)
Ho, T. K. 1998, IEEE Trans. Pattern Anal. Mach. Intell., 20, 832
Koo, D. C. 1985, AJ, 90, 148
Oyaizu, H., Lima, M., Cunha, C. E., Lin, H., & Frieman, J. 2008a, ApJ, 689, 709
Oyaizu, H., Lima, M., Cunha, C., Lin, H., Frieman, J., & Sheldon, E. S. 2008b, ApJ, 674, 768
Padmanabhan, N., et al. 2008, ApJ, 674, 1217
Sawicki, M. J., Lin, H., & Yee, H. K. C. 1997, AJ, 113, 1
Strauss, M. A., et al. 2002, AJ, 124, 1810
Wang, Y., Bahcall, N., & Turner, E. L. 1998, AJ, 116, 2081
York, D. G., et al. 2000, AJ, 120, 1579