Predicting Galaxy Ellipticity to Reduce Shape Noise in Weak Lensing Research

Thomas Schuster
University of California, Berkeley

Abstract

We predict the ellipticity of galaxies from a variety of variables available through data from the Canada-France-Hawaii Lensing Survey. A survey of regression methods, including both nonparametric and tree-based methods, are applied to the prediction problem and results are discussed and compared. We present graphical methods for finding interactions between predictors. Leave-one-out cross-validation, K-fold cross-validation, and the Akaike information criterion are used to choose parameters of the models. We find that projection pursuit regression performs the best on a seperate test set of data. Tree-based models performed poorly due to strong linear relationships in the data.

I. Introduction

A. Motivation

Gravitational lensing, the bending and distortion of light by a gravitational field, provides an important tool to probe the distribution of matter in the universe. For example, the masses of galaxies and galaxy clusters can be estimated by their distortion of the light of distant galaxies lying past them. The primary problem encountered in research in gravitational lensing is our lack of knowledge of the intrinsic, pre-lensing, shapes of galaxies. This is compounded by the fact that many applications of gravitational lensing, e.g. in detection of dark matter, are to cases where the lensing is very weak.

The above reasons make weak gravitational lensing a statistical problem. While we cannot reliably determine the lensing of a single galaxy due to lack of knowledge of the galaxy's intrinsic shape, we can search for correlations in the ellipticity of galaxies in a small region of space. That is, given a galaxy sample and an predicted value of the ellipticities of these galaxies, we can detect gravitational lensing if the observed ellipticities of these galaxies differs from the predicted ellipticities in a systematic manner. The error introduced by predicting the ellipticity as the median of all galaxy ellipticies is called *shape noise*. A major struggle in weak gravitational lensing research is to reduce shape noise. Our study seeks to reduce shape noise by predicting the ellipiticity of a galaxy using regression analysis.

B. The Prediction Problem

Prediction problems are common in many areas of research and industry. In a prediction problem, one seeks to predict a *response* y from some collection of *predictors* x. We denote our prediction, or *fitted value*, as $\hat{y}$. The error in our prediction is called the *residual*, defined as $\epsilon = y - \hat{y}$. *Weights* can be assigned to each data point to determine how important that point will be when fitting our model.

In our study we focus on *shearing* from gravitational lensing, that is, its effect on a galaxy's ellipticity. Ellipticity is defined as $e = \frac{A-B}{A+B}$, with A and B being the semimajor and semiminor axes in the CFHTLenS catalog, respectively.

We use data provided by the Canada-France-Hawaii Lensing Survey (CFHTLenS). Details of the collection and processing of the data present in the CFHTLenS catalog can be found in Erben et al. (2012) and Heymans et al. (2012). We used the first 100,000 observations of the survey in our study. Of these, we excluded all which were not galaxies, were marked by the survey as a bad fit, or which were missing measurements for any of the predictors used. After these eliminations 32,120 observations remained. These were divided into a training set of size 24,022 and a test set of size 8,098. All models were fit using solely data from the training set. Weights were provided by the survey as calculated in Miller et al. (2012).

The observations in the test set were used to compare the predictive power of different models. We define the test sum of squares to be the weighted sum of the squared residuals on the test set, $\text{TSS} \equiv \frac{\sum_{i=1}^{n_{test}} w_i (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n_{test}} w_i}$. A low TSS is desired as it indicates the model predicts well on an independent set of data.

The logarithm of the ellipticity was taken as our response throughout the study. This was chosen after viewing plots of residuals vs. fitted for a variety of models using both the ellipticity and its logarithm as the response. A common feature of those models with the ellipticity as the response was residuals whose spread increased with the fitted value. Plots of residuals vs. fitted with the logarithm as a response had spread independent of fitted value (Fig. 1).

Predictors in our models were chosen at the beginning of the study. For nonparametric models they were the scale length (`scalelength`), half-light radius (`FLUX_RADIUS`), full width half maximum (`FWHM_WORLD`), isophotal area (log(`ISOAREA_WORLD`)), signal-to-noise ratio (log(`SNratio`)), bulge fraction (`bulge_fraction`), magnitude in the infrared band (exp(`MAG_i`) or `e_MAG_i_scaled`), the maximum brightness (`MU_MAX`), total flux (log(`model_flux`)), and the ascension-declination tree (`pos_factor`, see Sec. V.B.). For tree-based methods the ascension-declination tree was replaced by including the ascension and declination as predictors. The redshift was also

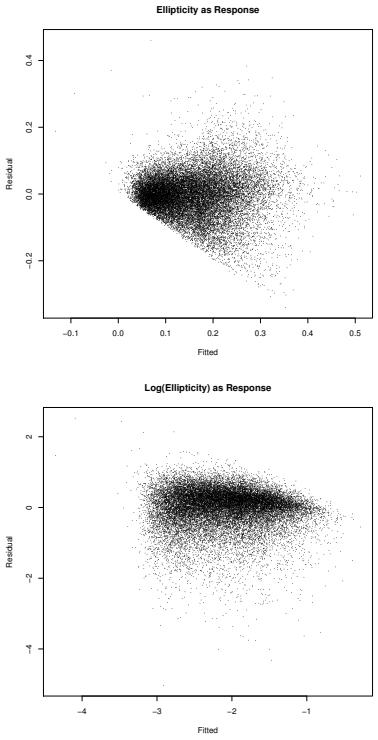

Figure 1. Plots of residuals vs. fitted for the GAM with the ellipticity as the response (top) and the logarithm of the ellipticity as the response (bottom).

included in tree-based models. Logarithmic and exponential transforms of some predictors were made to more evenly distribute the data. The definitions and calculations of most predictors can be found in Erben et al. (2012). Information on the calculation of galaxy redshifts can be found in Hildebrandt et al. (2012).

II. Nonparametric Regression: Methods

Nonparametric regression encompasses a wide variety of regression methods. A common feature of all nonparametric methods is an ability to adapt to the complexity of the relationship between the response and the predictors.

A. Types of Nonparametric Regression

Two types of nonparametric regression were used in our study, smoothing splines (abbreviated as *spline*) and super smoother (*supsmu*). In the smoothing splines method, we model $\hat{y} = f(x)$, with $f(x)$ chosen to minimize $\mathrm{RSS} + \lambda \int |f''(x)|^2 dx$. λ is a smoothing parameter and is chosen through leave-one-out cross-validation (Sec. II.D.). It is also possible to fix the effective degrees of freedom of the model, in which case only functions $f(x)$ with the specified effective degrees of freedom will be considered. If used, the effective degrees of freedom is also a smoothing parameter.

Friedman's super smoother was also used for nonparametric regression, as described in Friedman J. H. (1984). The super smoother contains a parameter span which is analogous to the span of local linear regression, and is a smoothing parameter of the method.

B. The Generalized Additive Model

Nonparametric regression can be expanded to include multiple predictors using the generalized additive model (GAM). Here we model the response as a linear combination of 1-variable nonparametric functions of the predictors: $\hat{y} = \beta_0 + \sum_{i=1}^{p} \beta_i f_i(x_i)$. In the GAM's implementation in R, the nonparametric functions are fit using the smoothing spline method.

It is possible to expand nonparametric regression to multiple predictors by simply modeling y as a single nonparametric function of all predictors, $\hat{y} = f(x_1, x_2, ..., x_p)$. There are two issues with this expansion of nonparametric regression. The first, referred to as the *curse of dimensionality*, is that points in our predictor space become increasingly spread out in predictor space as we include more predictors in our model. This leads to issues of overfitting. The second is that p-variable nonparametric functions are much more complex than the GAM with p predictors. For both of these reasons the GAM is preferred.

C. Projection Pursuit Regression

Projection pursuit regression (PPR) is a generalization of the GAM. Here we model the response as a linear combination of 1-variable *ridge functions* of a linear combination of predictors, $\hat{y} = \beta_0 + \sum_{k=1}^{M} \beta_k f_k(\boldsymbol{\alpha_k}^T \mathbf{x})$. The ridge functions are determined using some method of nonparametric regression, and the number of of ridge functions M is a smoothing parameter of PPR. The two methods of nonparametric regression in PPR considered in this study are Freidman's super smoother and smoothing splines.

D. Cross-Validation

Cross-validation (CV) is a set of techniques used to anticipate the performance of a model on an independent set of data. Thus we can use cross-validation to choose parameters which control the complexity of our models.

We focus primarily on K-*fold* cross-validation. We randomly partition our data into K equal sized subsets, or *folds*. Our model is fit using data from all folds but one, and is tested on the excluded fold. The (weighted) sum of squared errors is computed. This is repeated for all folds, and the results are in turn summed. We seek to minimize this final sum.

A specific example of K-fold cross-validation is *leave-one-out* cross-validation, where we take $K = n$.

R commonly uses a variant on cross-validation called *generalized cross-validation* (GCV).

E. Akaike Information Criterion

The Akaike information criterion (AIC) is an alternative to cross-validation. Here we seek to minimize the AIC, defined as $\mathrm{AIC} = 2k + n \log(\frac{\mathrm{RSS}}{n})$. AIC is typically preferred on much smaller data sets.

F. Graphical Methods for Finding Interactions

It is also possible to create GAMs containing nonparametric functions of multiple predictors. Due to the drawbacks of multiple variable nonparametric functions discussed in Sec. II.B. this is not desirable unless it significantly improves our model. We define the GAM to contain an interaction between two predictors if a nonparametric function of both predictors is included in the model. Two graphical methods of searching for a need for interactions between predictors were developed. In both, we begin with a model containing no interactions.

Our first method is the *highlighted residuals* method (Fig. 8). We identify two subsets of the data: those with $|\epsilon|$ in the top 10% of all residuals and $\epsilon < 0$, and those with $|\epsilon|$ in the top 10% of all residuals and $\epsilon > 0$. A scatterplot of all data against two predictors is then overlaid with red circles around each of the extremely negative residuals, and blue circles around each of the extremely positive residuals. This method allows easy visualization of the density of extreme residuals in different planes of predictor space. The percentage of residuals circled can be easily raised or lowered from 10%. This technique performs poorly in particular dense regions of predictor space, which can be fixed by zooming in the plot on that particular region.

Our second method is a *heatmap of residuals* (Fig. 2). Here we choose two predictors and subdivide the region spanned by the two predictors into evenly spaced cells. The mean of the residuals in each cell is then evaluated. Each cell is colored depending on its mean of residuals, and a scatterplot is overlaid. To emphasize high magnitude residual points while avoiding overcrowding the plot, points in top 10% of residuals by absolute value are plotted larger than other points. This method works well for identifying regions in which the model systematically overestimates or underestimates the response.

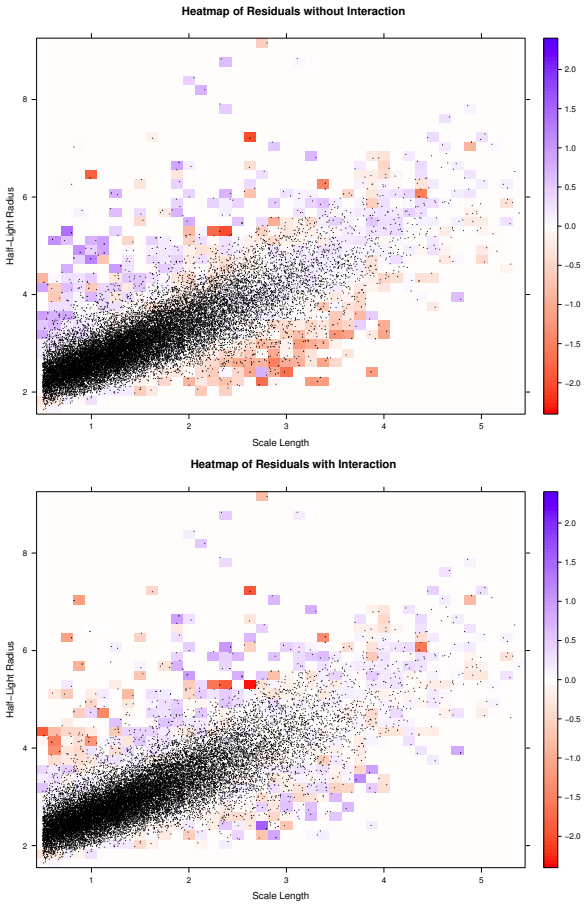

Figure 2. A heatmap of residuals plot of the half-light radius and scalelength in the GAM before including an interaction (top), and after including the interaction term (bottom).

III. Nonparametric Regression: Results

A. Binning by Redshift

Binning by redshift was attempted using the GAM, but found to have little effect on the predictive power of the model. Data with redshift less than 1.5 were partitioned into three subsets: those with redshift less than .5, those with redshift between .5 and 1, and those with redshift between 1 and 1.5. Those with redshift greater than 1.5 comprised only 7.7% of the total data and were not considered in this analysis.

A GAM was fit on each of the redshift bins, and the sum of the residuals was computed and compared to that of a model without redshift binning. The TSS (now excluding all points with redshift greater than 1.5 from the test sample) was found to decrease by .58% with redshift binning. This was a small improvement for the 3-fold increase in the complexity of the model, and binning by redshift was not used for the rest of the study.

B. Scale Length and Half-Light Radius Interaction

Using the graphical methods of Sec. II.F graphs of all combinations of predictors were examined. Only one combination of predictors showed a clear interaction: the scale length and the half-light radius. In both graphing methods two regions of systematic error in our model were apparent. The model was changed to include an interaction between the two predictors and heatmaps of residuals before and after including the interaction are shown in Fig. 2. It is clear visually that including the interaction improved the model in both regions of systematic error. A contour plot of the 2-variable nonparametric function of the half-light radius and scalelength confirms that the model detected this non-additive relationship between the variables (Fig. 3). The TSS on the test set was decreased from .293 to .278, and 5.12% decrease.

C. Comparison of Cross-Validation and AIC in PPR

1) Setup: We used each of the two methods of nonparametric regression implemented in R for PPR, super smoother and smoothing splines. Each of the methods contains a smoothing parameter, the span and the effective degrees of freedom respectively. These can be chosen for each ridge function through GCV or fixed for all ridge functions.

We tried five different PPR models. The first was using super smoother, with span chosen by GCV. We used K-fold CV to determine the number of ridge functions. Second we

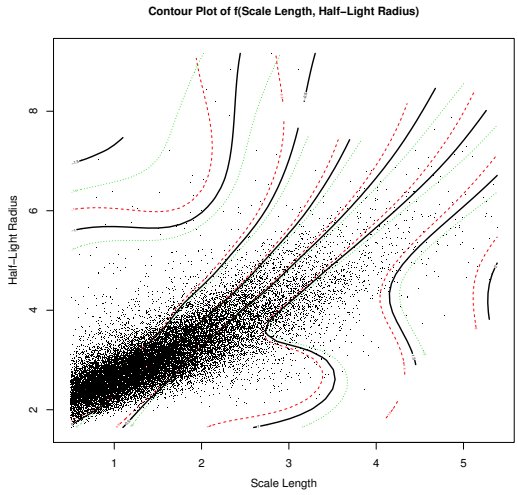

Figure 3. Contour plot of the nonparametric function in the GAM of the half-light radius and scale length overlaid on a scatterplot of the data.

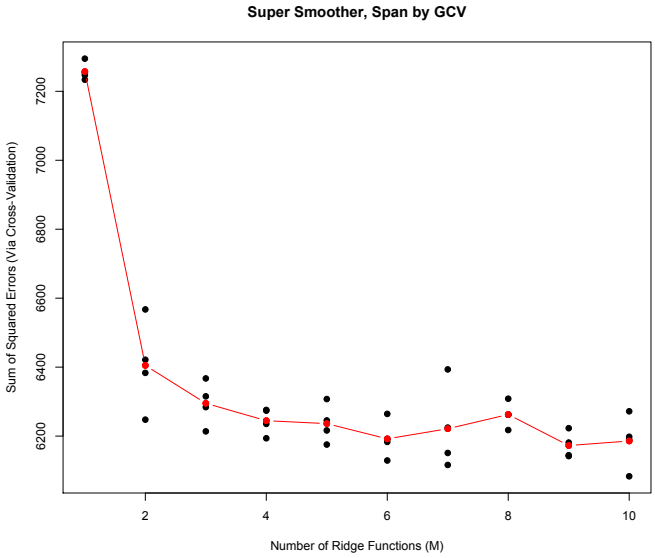

Figure 4. Results of cross-validation for PPR using super smoother with span chosen by GCV for each ridge function. Cross-validation was performed 4 times with different seeds in R's random number generator. Results of individual trials are in black, the mean of all trials is plotted in red. In this case $M = 6$ was found to be optimal.

considered super smoother with the span fixed. 20 different values of span were tested, from .05 to 1.0 in intervals of .05. K-fold CV is performed on each of them, and the combination of span and number of ridge functions which gave the lowest sum of squared errors in cross-validation was chosen for the model.

Next three different methods using smoothing splines were tried. The first was with the degrees of freedom chosen by GCV, and number of terms by K-fold CV (Fig. 4). We then tried varying the degrees of freedom from 1 to 20, and chose the combination of degrees of freedom and number of ridge functions using K-fold CV, as in super smoother. Our third method is choosing the smoothing parameters such that they minimize the AIC. This was done analogous to cross-validation with fixed degrees of freedom.

In all instances K-fold CV considered for PPR, K-fold CV was performed four times with the seed of R's random number generator reset to a different value each time. This was to reduce random variation in cross-validation results from the selection of folds. An average of the sum of squared errors for each K-fold CV was used in determining the optimal choice of smoothing parameters.

2) Results and Discussion: The chosen smoothing parameter values and number of ridge functions for each of the 5 methods were compiled into a table below. For both AIC and CV on smoothing splines two combinations of smoothing parameters were chosen: that which gave the minimum of each criterion, and that which gave the 2^{nd} minimum. In both cases the smoothing parameter values that gave the 2^{nd} minimum were lower (less complex) than those which gave the 1^{st} minimum, and thus were worth considering.

Method	CV or AIC	Specifications	TSS
supsmu	CV	span = .5, M = 7	.260
supsmu	CV	span by GCV, M = 6	.257
spline	CV	df by GCV, M = 2	.306
spline	CV	df = 5, M = 10	.280
spline	Both	df = 5, M = 6	.279
spline	AIC	df = 8, M = 8	.291

PPR with Super Smoother and span chosen by GCV (Fig. 4) was found to give the lowest TSS. Results of CV for fixed span using super smoother are shown in Fig. 5. It was found that the optimal number of terms in PPR increases with increasing span. This was expected; it is essentially a trade off between complexity in the number of ridge functions and the complexity of each individual ridge function. This was a common feature of varying both the nonparametric smoothing parameter and the number of ridge functions in PPR.

IV. Tree-Based Models: Methods

Tree-based models are another common method in prediction problems and machine learning. We used regression trees, implemented through `tree()` and `rpart()` in R, and an expansion of regression trees, random forests.

A. Regression Trees

In a regression tree, we recursively partition the data into small subsets, and in each subset model the response as the mean of the particular subset. This can be represented graphically as in Fig. 6. Starting at the top, at each node of the tree we follow the left branch if the condition is true, and the right branch if false. The prediction for each subset is

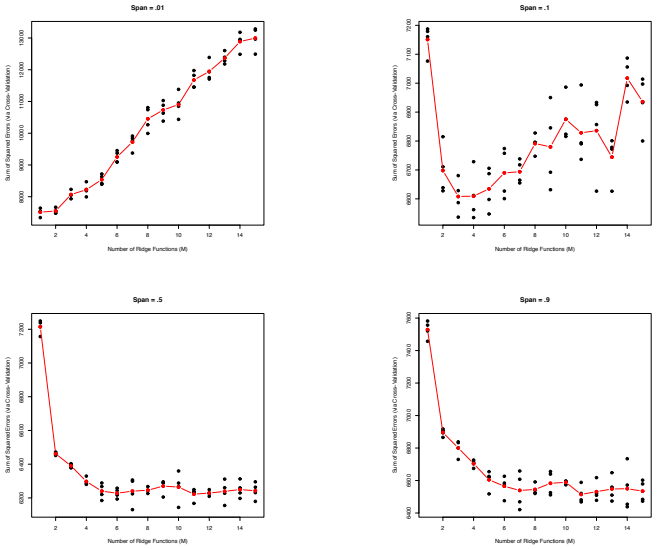

Figure 5. Results of cross-validation for PPR using super smoother with span fixed, for span = .01, .1, .5, and .9. Span = .5 with $M = 7$ was found to have the lowest sum of squared errors.

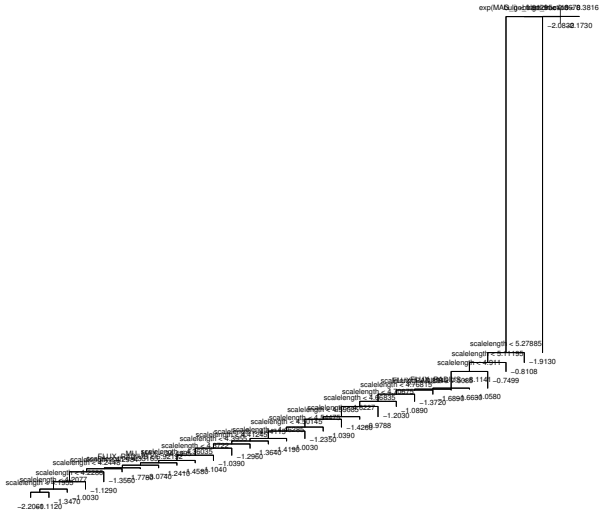

Figure 6. Graph of regression tree with parameters set just above their values at failure. The tree was generated using `rpart()`, with

displayed at the corresponding terminal node. An important limitation is that we only consider splits containing simple inequalities in one predictor.

There are two general steps to 'growing' a regression tree. A greedy algorithm is used to grow as complicated a tree as the limitations of R allow – ideally each data point would belong to it's own subset. Second, we prune the tree by selecting the 'optimal' subtree of the tree. In the `tree` function in R, subtrees are selected to minimize $\mathrm{RSS} + \alpha k$, where k is the number of nodes. α is a smoothing parameter and chosen through cross-validation. In the `rpart` function in R, a subtree is chosen by recursively 'snipping' off splits which do not reduce the deviance of the tree by `cp`. `cp` is also a smoothing parameter, and is chosen through cross-validation as implemented in `rpart()`.

B. Random Forests

Next we consider random forests, an extension of regression trees. To motivate this extension, we first discuss the concept of stability of a regression method. We define a regression method to be stable if a small change in the initial data produces a small change in the model's predictions. Regression trees as discussed in the previous section have been found to be unstable due to the greedy algorithm used to grow the full tree. Random forests attempt to reduce this instability by growing many different trees and averaging the predictions of each of them.

In the random forest method, we first draw B samples of size n with replacement from the initial data. Next, we fit a regression tree on each sample S_i using the same procedure as in the previous section, but with one alteration: when growing a tree, at each split we consider only splits from among m of the p predictors. This reduces correlation between the trees and increases the stability of the method. A prediction is found by averaging over the predictions of each of the B trees.

There are two tuning parameters to be varied in this model, B and m. We use a technique similar to cross-validation to choose the values of these parameters. Because the samples are chosen with replacement, for each sample there will be data points in the initial data which are left out of the sample. This set of left-out data is called the out-of-bag (OOB) sample. The out-of-bag error of i^{th} tree defined to be $\mathrm{OOBError}_i = \sum_{j \in S_i} w_j (y_j - \hat{y}_j)^2$. The total OOB error is the sum of the individual OOB errors. We choose B and m such that they will minimize the total OOB error.

V. Tree-Based Models: Results

A. Failure of Regression Trees

The `tree()` function in R failed because of strong linear relationships in our data. The function is limited to trees of depth 32 or less, and failed from this limitation for any reasonable choice of input parameters. A plot of the tree just before failing illustrates that nearly all splits were made with respect to the scale length (Fig. 6). In our earlier linear models, the scale length was found to be highly correlated with the ellipticity, more so than any other single predictor.

Due to the failure of `tree()`, a second implementation of regression trees in R, `rpart()`, was tried. `rpart()` uses a different algorithm for generating its full tree, and did not encounter any issues relating to the depth of the tree. A tree

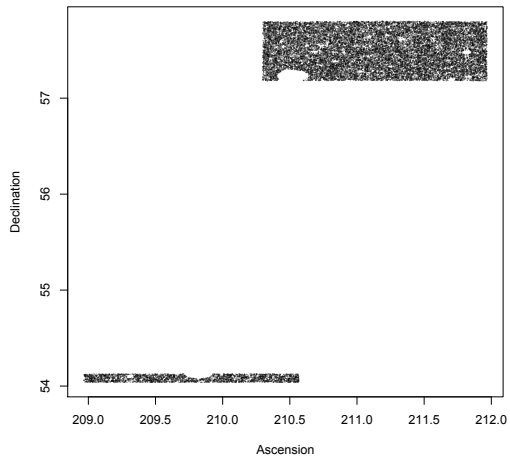

Figure 7. Scatterplot of ascension and declination. The upper region lies in the rectangle $57.18 < \delta < 57.80, 210.30 < \alpha < 211.97$. The lower region lies in the rectangle $54.04 < \delta < 54.13, 208.97 < \alpha < 210.57$. Both angles are measured in degrees.

was grown and pruned as described in the methods. The test sum of squares was computed to be $\mathrm{TSS} = .342$.

B. Ascension and Declination Tree

A scatterplot of the declination δ against the ascension α for our data is shown in Fig. 7. The galaxies observed lie in two rectangular regions.

The circled residuals graphing method (Sec. II.F.) was applied to a plot of the declination and ascension (Fig. 8). We see there are nearly entirely low (red) residuals in the lower region. Plots of the upper region showed a more balanced distribution of high and low residuals. Motivated by this observation, a tree was fit on the residuals of the GAM with the ascension and declination as predictors. The ascension-declination tree was used for all other nonparametric methods. This was found to reduce the TSS from .281 to .278 in the GAM, a 1.1% decrease.

C. Random Forests

We found $B = 500$ and $m = 7$ (out of 13 predictors) to give the lowest OOB error. Multiple trials with different seeds in R's random number generator were performed for our choice of m. Due to issues with computation time, our choice of m was based on results for the $\mathrm{OOBerror}$ on a sample of size 2,500, chosen at random from our training sample. Once m was chosen, our final random forest was fit using all data in the training sample. The resulting random forest gave $\mathrm{TSS} = .283$.

VI. Conclusion

A. Summary

The TSS for all methods discussed were computed and compiled in a table below. The (effective) degrees of freedom

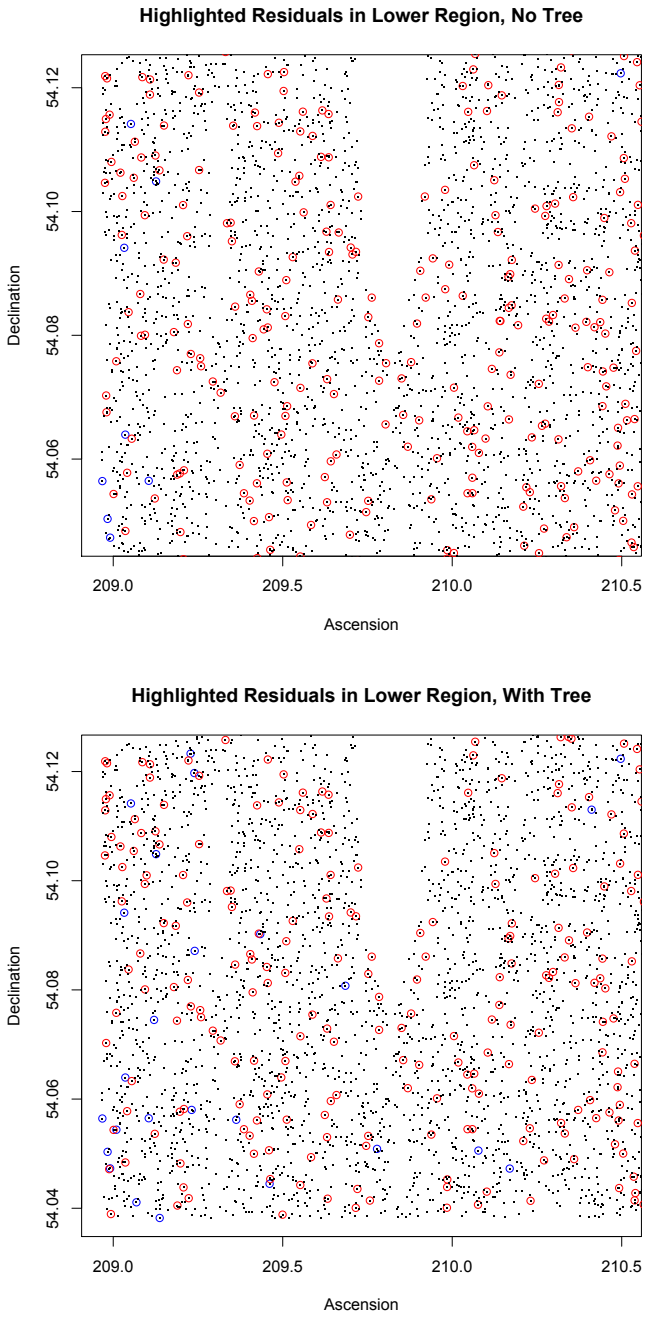

Figure 8. Highlighted residuals graphing method applied to the lower region of the ascension vs. declination plot in Fig. (NUMBER). The residuals were calculated from the GAM without the ascension-declination tree (top) and with the tree (bottom).

(EDF) for each model was also computed when possible to display the difference in complexity between models. Of all methods, PPR using super smoother, with span chosen through GCV and with 6 ridge functions, was found to have the lowest TSS. Other varieties of nonparametric regression also performed well.

Method	Specifications	TSS	EDF
Mean		.591	1
Linear		.332	12
GAM		.278	76
GAM	w/o interaction	.293	74
GAM	w/o tree	.281	62
PPR	supsmu, span = .5, M = 7	.260	
PPR	supsmu, span by GCV, M = 6	.257	
PPR	spline, df by GCV, M = 2	.306	56
PPR	spline, df = 5, M = 10	.280	180
PPR	spline, df = 5, M = 6	.279	108
PPR	spline, df = 8, M = 8	.291	168
Tree	prune w/ `cp` = .00056	.342	
Forest	$m = 7$	.282	

The effective degrees of freedom of PPR with smoothing splines was found to be much greater than that of the GAMs, despite having fewer ridge functions than GAM has predictors. This is due to the increased degrees of freedom that come with choosing the components of the projection direction vectors.

Tree-based models performed substantially worse than nonparametric models. Random forests were the only tree-based method which gave a TSS in the range of most nonparametric models. As discussed in Sec. V.A., this is likely due to strong continuous relations between certain predictors and the response in our data.

Plots of residuals against fitted values and actual against fitted values for the the best fitting PPR are shown in Fig. 9. As with all models, the distribution of residuals is skewed, with more extreme values less than 0. This might be a consequence of taking the logarithm of the ellipticity, with $0 \leq e < 1$, as our response.

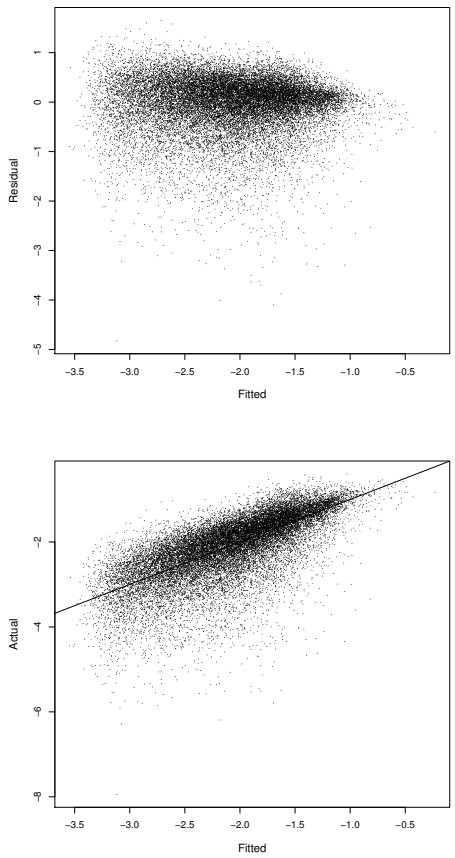

Figure 9. Plots of residuals against fitted values (top) and actual vs. predicted (bottom) for the best fitting model, PPR with super smoother and span chosen through GCV. The line $y = \hat{y}$ is displayed on the bottom plot.

B. Future Work

Increasing the number of data points used would increase the reliability of our model. This would also allow the data to encompass a wider range of ascensions and declinations, and our ascension-declination tree could be refit to incorporate the new regions.

Trees in combination with linear or nonparametric regression could be tried. This could fix the failure of `tree()` and produce a model with results comparable or better than nonparametric regression alone. One concern would be the computation time required for such a model.

The ellipticity can also be defined as $(e_1^2 + e_2^2)^{1/2}$, with e_1 and e_2 found as in Heymans et al. (2012), and provided in the CFHTLenS data. The two definitions are not identical. Our definition of ellipticity has the advantage that we know the error associated with each galaxies ellipticity, as the errors of A and B are provided in the data. Similar methods could be applied to this second definition of the ellipticity and results could be compared.

C. Acknowledgements

This work is based on observations obtained with MegaPrime/MegaCam, a joint project of CFHT and CEA/DAPNIA, at the Canada-France-Hawaii Telescope (CFHT) which is operated by the National Research Council (NRC) of Canada, the Institut National des Sciences de l'Univers of the Centre National de la Recherche Scientifique (CNRS) of France, and the University of Hawaii. This research used the facilities of the Canadian Astronomy Data Centre operated by the National Research Council of Canada with the support of the Canadian Space Agency. CFHTLenS data processing was made possible thanks to significant computing support from the NSERC Research Tools and Instruments grant program.

REFERENCES

[1] Erben T. et al., 2012, arXiv:1210.8156
[2] Friedman, J. H. (1984) A variable span scatterplot smoother. Laboratory for Computational Statistics, Stanford University Technical Report No. 5.

[3] Heymans C. et al., 2012, arXiv:1210.0032
[4] Hildebrandt H. et al., 2012, arXiv:1111.4434
[5] Miller L. et al., 2012, arXiv:0708.2340