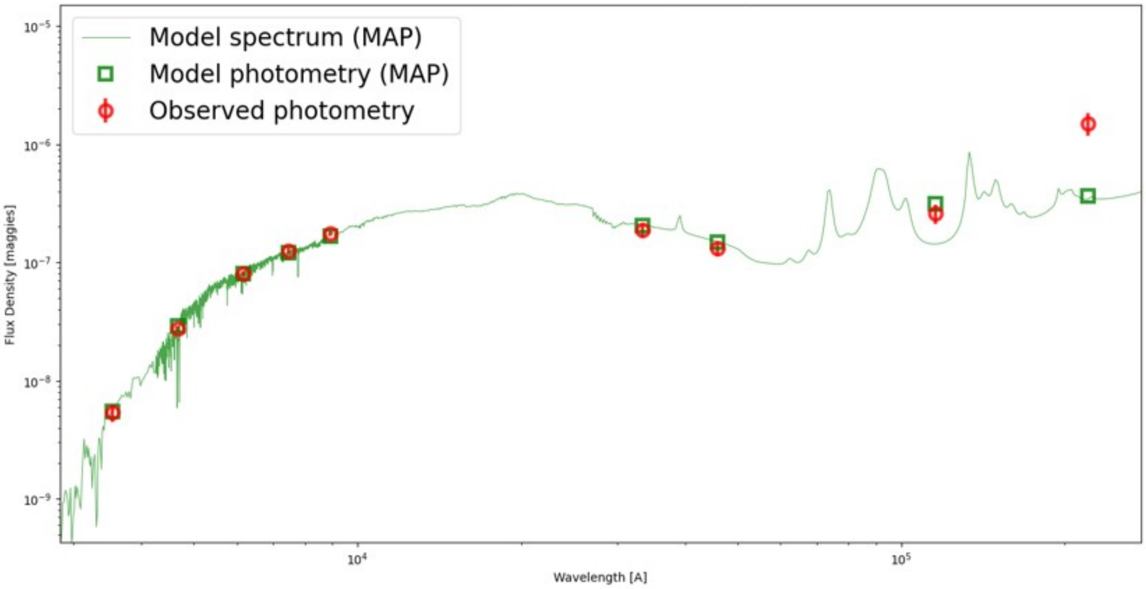
PHYS 788 Report

Estimating Galaxy Properties using SED Fitting, Machine Learning, and MCMC



Written By: Justin Marchioni

April 22nd, 2024

1 Assignments

Following report details a summary of the main tasks completed in the assignments during the course. A subset of the primary results from the assignments will be of focus.

A1 Prospector SED Fitting

The first assignment involved analyzing constraints on the model parameters estimated by Prospector. To do this, Prospector was run on a single galaxy. This galaxy was randomly selected from the entire sample. The model fit by Prospector consisted of 5 free parameters related to the galaxy's properties: (1) the mass M, (2) the metallicity Z, (3) the dust optical depth κ, (4) the SFR delay timescale τ, and (5) the age t_{age}. The redshift z of the galaxy could also be set as a free parameter, or it could be fixed to the spectroscopically measured value from SDSS. After running the model, the results were compared to that from a paper by [1], where the authors used an SED fitting code MAGPHYS along with known redshift to estimate galaxy properties.

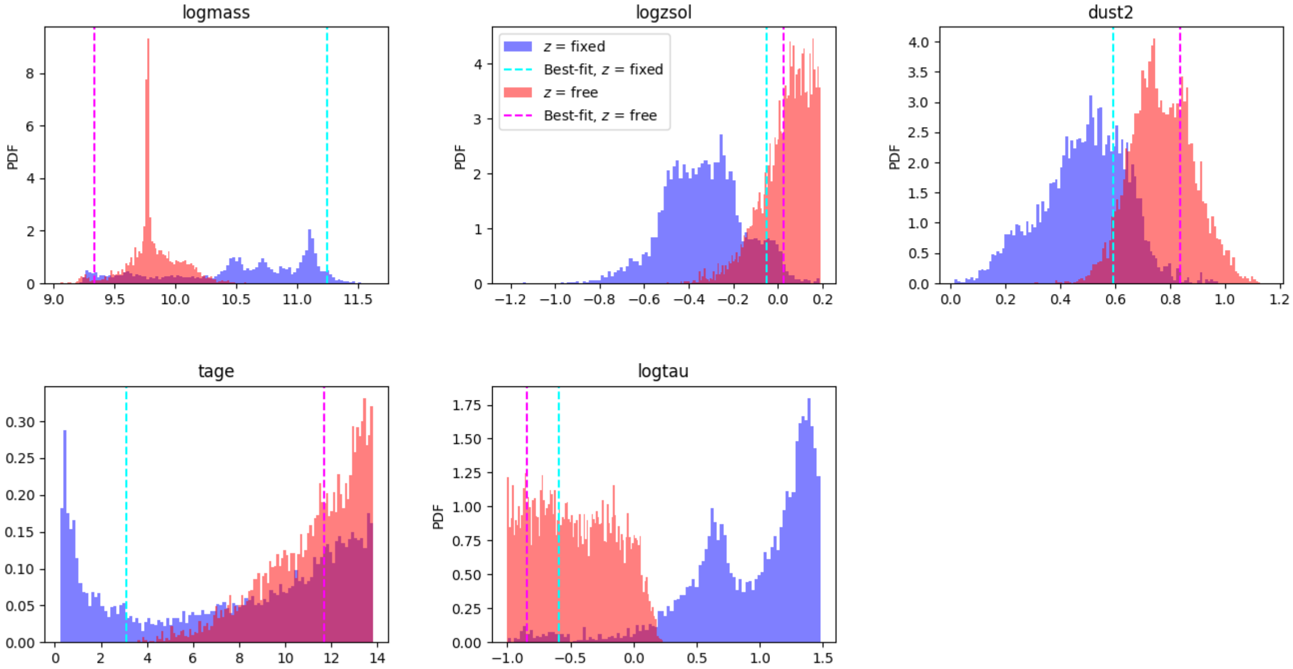


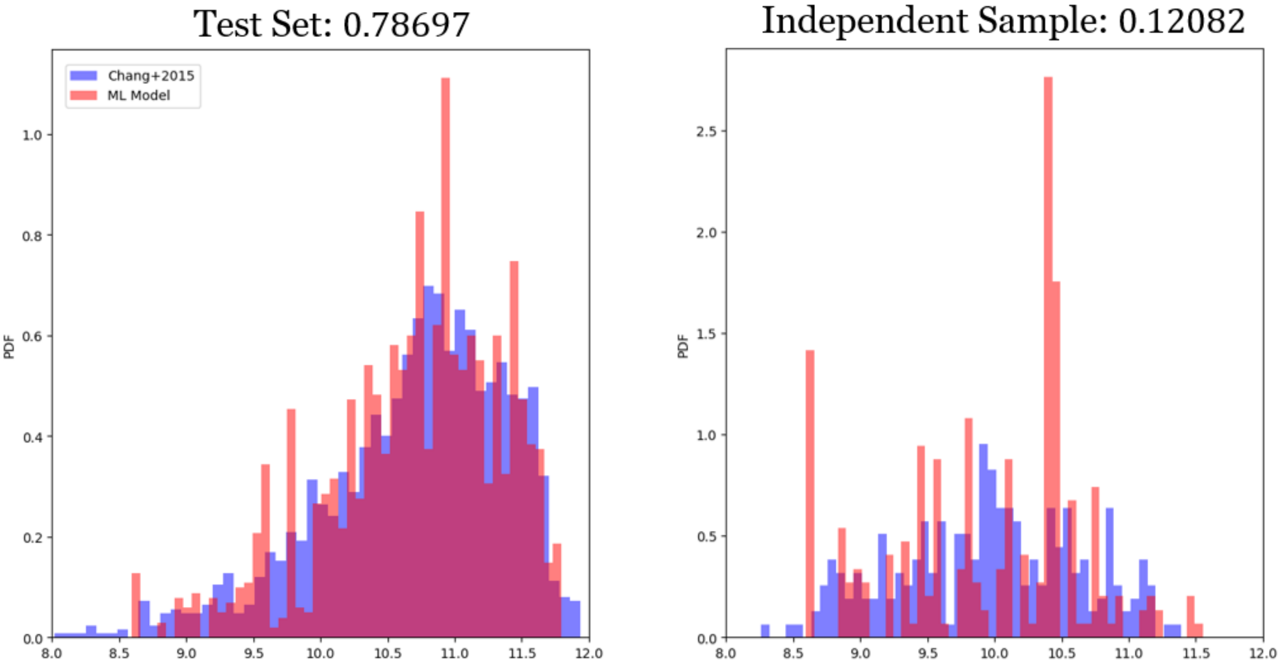
Figure 1: Sampled distributions on the model parameters for an example galaxy obtained from Prospector. Two models are considered: (1) fixed z in blue and (2) free z in red. The vertical dashed lines show the best-fit values, where these are the maximum of the likelihood of the samples.

Figure 1 shows estimated posterior distributions on all 5 model parameters considering fixed and free redshift models. Prospector uses MCMC sampling to generate these distributions. For this particular galaxy, Z and κ have similar best-fit values in both cases, while t_{age} and τ, have different best-fit values. In all four cases, the distribution for the parameter in the fixed redshift case is broader than when the redshift is a free parameter. The largest difference in the two cases is the estimated mass of the galaxy. For fixed redshift, the best-fit mass is $\sim 10^{11.25} M_\odot$ compared to $\sim 10^{9.34} M_\odot$ when redshift is a free parameter. This is likely caused by the best-fit redshift in the free case being far from the true redshift. The best-fit redshift was found to be 0.012 compared to the true redshift of 0.184. This caused the best-fit stellar mass to be much lower than expected.

The other primary tasks for this assignment involved calculating the covariance matrix for the model parameters, and estimating uncertainties on mass estimates in the output catalogue. From the covariance matrix, $t_{\rm age}$ was found to have the largest variance. Mass uncertainties were approximated using Jackknife. The optimal number of subsets to split the dataset into that resulted in the most accurate errors was $N = 200 - 1000$. When the number of subsets was either very small or very large, the errors were consistently underestimated by a significant fraction in several bins.

A2 Machine Learning (ML) Techniques

The second assignment focused on using machine learning to relate galaxy properties to their measured photometry without using a physical model to link the two. The first step required choosing a subset of galaxies and normalizing the input data. The input data consisted of 9 flux bands and their errors. Each band was normalized by subtracting off the mean and dividing by the standard deviation. Then, an ML model was trained in order to predict the stellar mass and redshift of galaxies.


Test Set: 0.78697

Independent Sample: 0.12082

Figure 2: Results of the decision tree ML model used to predict the stellar masses of galaxies from their measured fluxes and errors. The target masses from the output catalogue in [1] are shown in blue, while the ML predictions are shown in red. Left: Predictions for the test set split from the training set. Right: Predictions for an independent sample not considered in the training/test split.

A decision tree was chosen for the model. I attempted to train a linear model through both Lasso and Ridge regression, but the predictions for the training set were poor. In hindsight, this likely would have worked better had the input values been log(flux) to transform the values onto the same linear scale. The decision tree model had two hyperparameters: (1) "min_samples_leaf" which controls the minimum number of samples at each leaf node and (2) "max_depth" which sets the maximum number of levels to the tree. Roughly 1000 combinations of the hyperparameters were tested and the best choice was identified using the built-in method score. This method calculates the coefficient of determination for the prediction. The best model had the highest score, and this was found to be min_samples_leaf = 15 and max_depth = 17.

Figure 2 depicts results of the decision tree model used to predict the stellar masses of galaxies. This model was trained using 80% of galaxies in our initial sample. The test set consisted of the remaining 20% of galaxies. From the plot, we find the test set has pretty good agreement with the output catalogue. However, the model has much worse predictions for the stellar masses of galaxies in the independent sample. This is reflected by the much lower accuracy score on aggregate. It's likely that the decision tree was overfit to the training data. The test set still had good agreement with the output catalogue since the flux distribution of galaxies in the test set coincided with the training set. On the other hand, the input data for the independent sample did not look like the training set. Decision trees are poor at extrapolation resulting in these inaccurate predictions. The decision tree also had predictions for redshift. These results were similar but slightly worse than that for stellar mass shown above.

In addition, this assignment involved compressing the input data into fewer dimensions, identifying clusters in this lower-dimensional space, and seeing how galaxy properties varied between clusters. Data compression was done through UMAP and clusters were identified through Spectral Clustering. UMAP was used since the data seemed to have non-linear structure and PCA would be less effective in this case. Spectral Clustering was applied since the clusters after UMAP reduction had complex shapes. Other algorithms like K-means prefer to create circular clusters which would be undesired in this case. The number of clusters the data should be split into was evaluated using silhouette scores. These scores quantify how well a data point fits into its assigned cluster and how distinct it is from other clusters. My results indicated that the data should be separated into 5 clusters, one more than what was visually determined after performing UMAP. Futhermore, I found that redshift had the most statistically significant differences between the clusters.

A3 Markov Chain Monte Carlo (MCMC) Parameter Estimation

The third and final assignment looked at estimating posterior distributions on galaxy properties using a selected pair of MCMC sampling techniques rather than using the built-in MCMC methods in Prospector. The two methods chosen were EMCEE (ensemble sampler) and Dynesty (nested sampler), although I will focus on the nested sampling results. Nested sampling has a few distinct advantages. First, there is no burn-in phase. Each sample that makes up the posterior distribution is weighted by an "importance weight". Unimportant points that would have a burn-in like affect are minimally weighted when creating the posterior. In addition, the Bayesian evidence is computed along with the parameter constraints. When setting up the sampler, uniform priors on the model parameters were used. The prior ranges were set to the default values in the "Prospector Example" notebook.

Figure 3 illustrates constraints on stellar mass and redshift for a galaxy computed using MCMC sampling. For stellar mass, the MCMC best-fit estimate and the ML estimate are similar distances from the result in [1] with the ML result being slightly more accurate. The opposite is true for redshift, although the three values are in much better agreement.

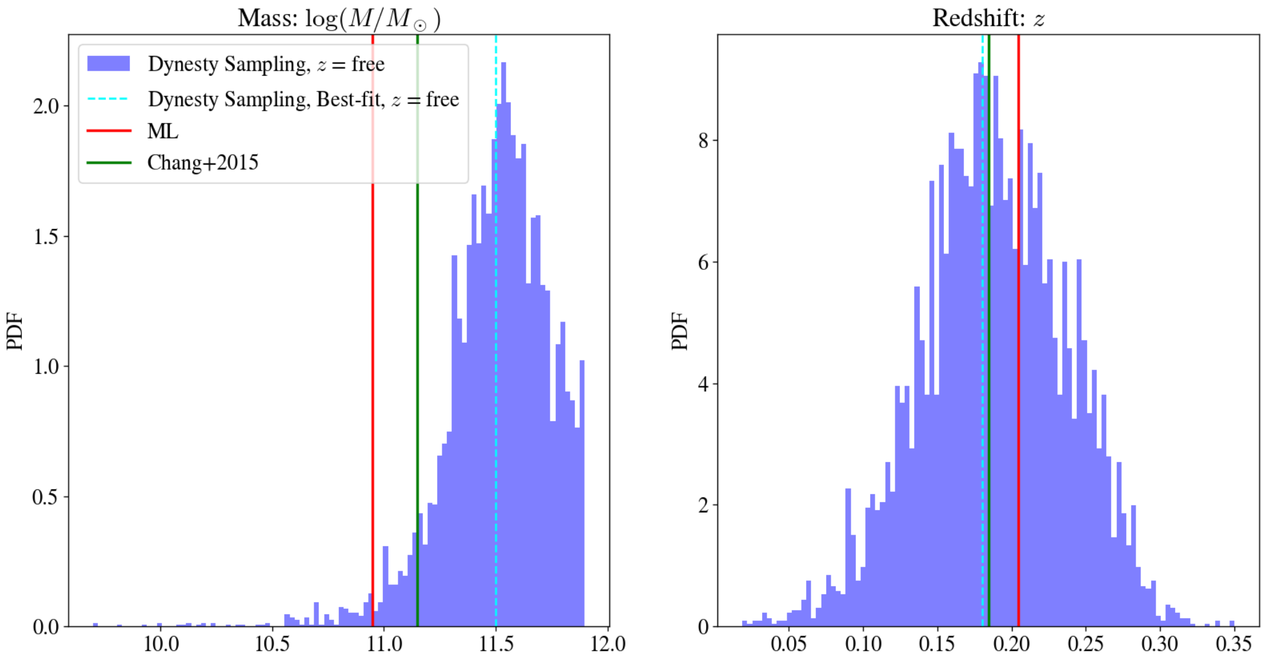


Figure 3: Stellar mass (left) and redshift (right) posterior distributions for a galaxy computed using the Dynesty MCMC sampler shown in blue. The ML prediction in red and the "true" value from [1] in green are overplotted for comparison.

There are several advantages to using MCMC over machine learning. First, the ML results are much more limited than the galaxy properties inferred from MCMC. The ML results for mass and redshift are single values compared to MCMC, where we obtain estimates for the full posterior distribution of each parameter. Since we used a decision tree, the ML model will function poorly for galaxies with properties that fall outside of the training set. We do not suffer from this restriction with MCMC. One advantage the ML model has compared to MCMC is the time required for generating galaxy properties. The ML model operates much faster than MCMC sampling, and the same ML model can be used to estimate properties for every galaxy in a full sample. Generating full posteriors for each galaxy using MCMC is much more computationally expensive compared to running the ML model.

In this assignment, we were also tasked with evaluating which of two models the data preferred. The two models considered were ones with fixed and free redshift. The Bayes factor was used to compare models, which encodes the plausibility of two different models, $M_1 = z_{\text{fixed}}$ and $M_2 = z_{\text{free}}$, given the observed data. It has the form $K = Z_1/Z_2$, where Z_1 and Z_2 are the Bayesian evidence (obtained from Dynesty) for models M_1 and M_2, respectively. For the data, I computed a value of $K = 6.419$, indicating a preference for the fixed redshift model over the free redshift model.

2 Application to Research

My research primarily focuses on describing how the structural parameters of dark matter haloes vary as they undergo binary major mergers. These haloes can be described by many structural properties including concentration, shape, and spin. Here I will discuss the scale radius which sets the radius where the slope of the halo density profile changes.

In my research, I attempted to implement a machine learning model to predict how the scale radius of the remnant halo changes during the merger. I chose to use XGBoost as the underlying model. It operates similarly to a random forest where many decision trees are generated and the model prediction is weighted by the outputs of all trees. The input data for the model is the orbital properties (mass ratio, orbital energy and angular momentum). The output is the remnant response with time.

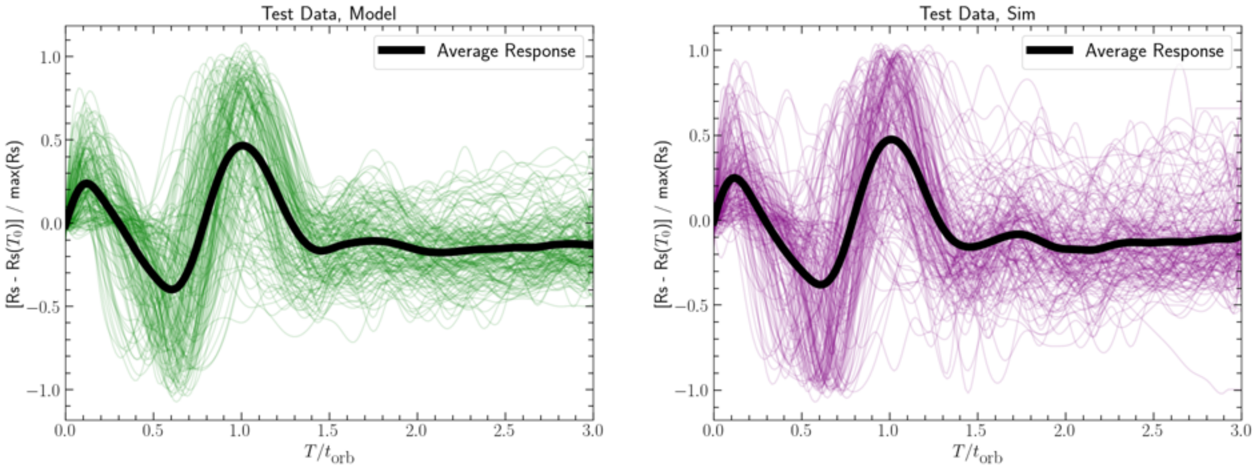


Figure 4: Testing a machine learning model relating the initial orbital properties of the merger to the remnant halo scale radius trained using XGBoost. For the test set, green curves illustrate model predictions for the scale radius, purple the response directly from the simulations, and black the average response.

Figure 4 compares model predictions for the scale radius to the output we are trying to match from the simulations. Both of these results are for a test set not used in training the ML model. The plot demonstrates that the model does not predict the individual responses of each merger well, but it can isolate when the oscillations occur. The model can also replicate the average response of all mergers accurately. This shows that it is possible to relate the initial orbital properties of a merger to how the scale radius of the remnant changes on average. In theory, one could also train models to predict other structural parameters.

Additionally, my research involves fitting models for the orbital evolution of a satellite halo about the remnant halo during the merger, as well as the density profiles of the two haloes. This is usually done through χ^2 minimization, but other techniques learned in class could be applied as well. These techniques include maximum likelihood estimation (MLE) and MCMC sampling. In my case, MLE is likely more applicable than MCMC since I do not require full posteriors for each model parameter at each timestep of the orbit. This is likely too costly to compute and my main focus is on how the best-fit values change with time.

References

[1] Y. Chang, A. van der Wel, E. da Cunha, and H. Rix, "Stellar Masses and Star Formation Rates for 1M Galaxies from SDSS+WISE." *The Astrophysical Journal Supplement Series*, vol. 219, no. 1, A8, 2015.