A FIRST LOOK AT CREATING MOCK CATALOGS WITH MACHINE LEARNING TECHNIQUES

XIAOYING XU$^{1}$, SHIRLEY HO$^{1}$, HY TRAC$^{1}$, JEFF SCHNEIDER$^{1,2}$, BARNABA POCZOS$^{2}$, AND MICHELLE NTAMPAKA$^{1}$
$^{1}$ McWilliams Center for Cosmology, Department of Physics, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
$^{2}$ School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Received 2013 February 25; accepted 2013 June 4; published 2013 July 17

ABSTRACT

We investigate machine learning (ML) techniques for predicting the number of galaxies ($N_{\text{gal}}$) that occupy a halo, given the halo’s properties. These types of mappings are crucial for constructing the mock galaxy catalogs necessary for analyses of large-scale structure. The ML techniques proposed here distinguish themselves from traditional halo occupation distribution (HOD) modeling as they do not assume a prescribed relationship between halo properties and $N_{\text{gal}}$. In addition, our ML approaches are only dependent on parent halo properties (like HOD methods), which are advantageous over subhalo-based approaches as identifying subhalos correctly is difficult. We test two algorithms: support vector machines (SVM) and $k$-nearest-neighbor (kNN) regression. We take galaxies and halos from the Millennium simulation and predict $N_{\text{gal}}$ by training our algorithms on the following six halo properties: number of particles, $M_{200}$, $\sigma_r$, $v_{\text{max}}$, half-mass radius, and spin. For Millennium, our predicted $N_{\text{gal}}$ values have a mean-squared error (MSE) of $\sim 0.16$ for both SVM and kNN. Our predictions match the overall distribution of halos reasonably well and the galaxy correlation function at large scales to $\sim 5\%$–$10\%$. In addition, we demonstrate a feature selection algorithm to isolate the halo parameters that are most predictive, a useful technique for understanding the mapping between halo properties and $N_{\text{gal}}$. Lastly, we investigate these ML-based approaches in making mock catalogs for different galaxy subpopulations (e.g., blue, red, high $M_{\text{star}}$, low $M_{\text{star}}$). Given its non-parametric nature as well as its powerful predictive and feature selection capabilities, ML offers an interesting alternative for creating mock catalogs.

Key words: galaxies: halos – large-scale structure of universe – methods: numerical

Online-only material: color figures

1. INTRODUCTION

As we enter the era of large-scale structure experiments such as LSST, WFIRST, and Euclid, the creation of reliable mock galaxy catalogs will become increasingly more important. Such catalogs are essential for characterizing the expected errors in the analyses of these data sets, calibrating analysis pipelines, and ultimately measuring cosmological parameters (such as the dark energy equation of state) from galaxy clustering (e.g., Anderson et al. 2012). Making mock catalogs for different subpopulations of galaxies (e.g., blue versus red, high $M_{\text{star}}$ versus low $M_{\text{star}}$, etc.) to study their clustering properties is also of utmost importance for understanding galaxy formation and evolution (e.g., Coil et al. 2008; Guo et al. 2012). Although these mock catalogs can be generated relatively quickly using perturbation-theory-based approaches such as the dark energy equation of state from galaxy clustering, there are still many challenges. For instance, the subhalo identification problem is crucial for HOD-based methods, and it is a very difficult problem in its own right due to resolution limitations (see, e.g., Behroozi et al. 2013). It then assumes that each subhalo contains a single galaxy with a stellar mass or luminosity that is monotonically related to a subhalo property.

While both methods have been shown to produce mock galaxy catalogs that match observations reasonably well, it would be unreasonable to assume that these methods are perfect. The sophisticated non-parametric regression algorithms that form a subcategory of “machine learning” (ML) are ideal for this purpose. To obtain the mapping from halos to $N_{\text{gal}}$, the only assumptions that these model-independent algorithms require are that such a relationship exists and that it is a continuous function of the halo parameters. They then proceed to construct a model from the data itself and hence do not impose any pre-supposed relationships onto the data. We note that although ML algorithms are non-parametric in the sense that we do not need to assume a known relationship between data parameters, they do often require us to assume some operational parameters such as how severely poor predictions are penalized. These, however, can be optimized as described in Section 3.1.

In addition, the ML-based approaches we propose here, like HOD-based methods, rely only on the properties of the parent dark matter halo. Hence, we can circumvent the difficulties in subhalo identification. Another point worth highlighting is that, in principle, these techniques can be trivially extended to understand how halo properties map onto different subpopulations of galaxies. This provides a method for making mock catalogs for these different subpopulations as well.

ML algorithms do, however, need to be trained on large, accurate data sets in order for them to learn robust mappings between halo properties and galaxy properties such as $N_{\text{gal}}$. More recently, subhalo abundance matching (SHAM) has become very prevalent (e.g., Conroy et al. 2007; Vimal et al. 2012). This method relies on being able to correctly identify the subhalos within a halo, a very difficult problem in its own right due to resolution limitations (see, e.g., Behroozi et al. 2013). It then assumes that each subhalo contains a single galaxy with a stellar mass or luminosity that is monotonically related to a subhalo property.
Training on observations of galaxy clusters is therefore ideal, however, at present a sufficiently large and accurate data set does not exist, even if we combine observations from multiple telescopes. The key advantage of HOD and SHAM methods over ML methods is that they can be tuned to match observations which reflect the actual universe. The ML methods we propose here, however, can only build mock catalogs based on the output from cosmological simulations of the universe with both N-body and hydrodynamics. In fact, such simulations are rare to come by due to their computational complexity, but current advances in simulation techniques are promising. For example, a similar simulation with a slightly larger box size than the medium-resolution run in Dolag & Sunyaev 2013 (a 300 Mpc h⁻¹ box which resolves at least 10 galaxies in a 10¹⁴ Mₜₜ h⁻¹ halo) should serve as a reasonable starting point for creating mocks for current on-going surveys. Due to its public accessibility, for this study we use the Millennium simulation with semi-analytic galaxy formation as a proof-of-concept.

A final point of interest rests in the observation that most HOD methods typically use the halo mass as the only parameter in their halo-to-galaxy mappings. An important topic to study is the sensitivity of N_gal to other halo parameters. For example, there have been investigations hinting that the environment of the halo is also an important factor in determining how many galaxies will form (e.g., Gao et al. 2005; Croft et al. 2012). This information can be gleaned using ML techniques as well, through performing a “feature selection” which picks out the halo properties that best predict N_gal.

In Section 2, we describe the data set we use derived from the Millennium simulations. In Section 3, we describe the two ML techniques we employ to learn the mapping between halo properties and N_gal. In Section 4, we describe our results, i.e., how well our predictions match the actual values from Millennium. This section also includes a discussion on using ML to make mocks for different subpopulations of galaxies. We conclude in Section 5.

2. DATA SET

We construct our data set from halo and galaxy catalogs at z = 0 derived from the Millennium simulation (Springel 2005; Springel et al. 2005). These catalogs are obtained via querying the Millennium online database (Lemson & the Virgo Consortium 2006). The halo catalogs were generated using a friends-of-friends (FoF) algorithm with linking length of 0.2 and the semi-analytic galaxy prescription used to populate these halos is described in Croton et al. (2006), De Lucia et al. (2006), and De Lucia & Blaizot (2007). The Millennium simulation is run with 2160³ particles in a 500 h⁻¹ Mpc box. The cosmology employed has Ω_m = 0.25, Ω_b = 0.045, Ω_Λ = 0.75, h = 0.73, n_s = 1, and σ₈ = 0.9.

To obtain our data set, we search through the Millennium halo catalog and extract all primary halos (FoF groups) with mass greater than 10¹² h⁻¹ Mₜₜ (at present, we are unlikely to observe anything less massive except in the local universe). We then match galaxies from the semi-analytic catalog to these halos, keeping only the primary galaxies of a halo or subhalo (i.e., those flagged 0 or 1 in the Millennium database). Hence, we emerge with a halo catalog listing the following seven parameters: number of particles in the halo N_p, M₂₀₀, velocity dispersion σ_v, maximum circular velocity v_max, half-mass radius R₁/₂, spin, and number of galaxies in the halo N_gal. Our goal is to train an ML algorithm to predict N_gal using the other six halo parameters.

The semi-analytic model used to populate the Millennium halos with galaxies is dependent on various thresholds (such as for gas accretion and star formation) that are also evolved through time. This quality makes Millennium an adequate testing ground for ML applications because the mapping from halo parameters to N_gal is much more complicated than the straightforward functions normally employed in methods such as HOD and SHAM. The complexity of these mappings should be closer to the level we expect from actual N-body simulations with hydrodynamics.

There are 395,832 halos (with 445,983 total galaxies) in our sample which we use for our basic tests of the ML algorithms. However, since the Millennium semi-analytic model provides b, v, r, i, z magnitudes and stellar masses for the galaxies, we can also use these same halos to learn the mapping between halo properties and N_gal for different subpopulations of galaxies. We perform tests of the ML algorithms after splitting the halo sample on color and stellar mass (a proxy for luminosity) in Sections 4.1 and 4.2, respectively.

3. MACHINE LEARNING ALGORITHMS

We test two different ML algorithms for predicting N_gal from the halo parameters N_p, M₂₀₀, σ_v, v_max, R₁/₂, and spin. The first is a support vector machine (SVM) and the second is a k-nearest-neighbor (kNN) routine, both described below. They work by “learning” a relationship between a set of input features X and the value we are interested in predicting Y. SVM and kNN are both non-parametric in the sense that we do not need to assume a model as traditional methods for populating halos with galaxies do. The mapping is constructed using information in the data itself; the data pick the model best suited to it. In addition, we avoid the messy problem of subhalo finding which is a required step in any SHAM-based approach; like HOD-based models, our proposed ML techniques operate on the parent halo itself.

The learning process is accomplished by “training” the algorithm on a set of training data where Y is known. The learned mapping can then be applied to a test set of X values with known Y to verify its accuracy. If the learned relationship appears robust, it can be applied to a set of X with unknown Y to make predictions.

In testing the accuracy of the predicted values, we draw on the mean-squared error (MSE) defined as

\[
\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (Y_{i,\text{test, true}} - Y_{i,\text{test, predicted}})^2,
\]

where N is the number of test data points. This effectively measures a combination of variance (the scatter in the predicted values) and bias (how different from truth the predicted values are).

3.1. Support Vector Machines

SVMs work by mapping the features X to a higher dimensional space and attempting to separate them into regions that map onto specific Y values using a set of hyperplanes (Cortes & Vapnik 1995). In its original form, it is a classification scheme but can be generalized to a regression algorithm (Drucker et al. 1997).

The general idea behind SVM is best illustrated through the case of a binary classifier. The binary SVM is trained on a set of input features X = \{X₁, X₂, X₃, ..., X_d\}, where d is the total number of features, to classify data into one of two classes
Consider a set of \( N \) training data, each with a corresponding column vector of features \( X_i \) and a corresponding class \( Y_i \). An SVM attempts to separate the training data into their appropriate classes using two parallel hyperplanes in a high-dimensional space. These planes can be written as

\[
W \cdot X - b = \pm 1,
\]

where \( W \) is the normal vector to the hyperplane and \( b \) is some constant scalar analogous to a \( y \)-intercept in two dimensions (2D). The two planes must satisfy the condition that no points fall in between them, i.e.,

\[
W \cdot X_i - b \geq 1 \quad \text{for } X_i \text{ of the first class (i.e., } Y_i = 1) \quad \text{or}
\]

\[
W \cdot X_i - b \leq -1 \quad \text{for } X_i \text{ of the second class.}
\]

The region bounded by these two planes is known as the margin, which has width

\[
w = \frac{2}{||W||}.
\]

The best classifier is obtained through maximizing this distance or minimizing \( ||W|| \) subject to the constraint in Equation (5) with respect to the Lagrangian

\[
L(\alpha_i) = \frac{1}{2} ||W||^2 - \sum_{i=1}^{N} \alpha_i [y_i(W \cdot X_i - b) - 1] \quad \text{subject to the constraints } \alpha_i \geq 0.
\]

Taking the derivative of this Lagrangian with respect to \( ||W|| \) yields the solution

\[
||W|| = \sum_{i=1}^{N} \alpha_i Y_i X_i.
\]

Values for the \( \alpha_i \) can be obtained by substituting this solution back into Equation (7), which yields

\[
L(\alpha_i) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j Y_i Y_j X_i \cdot X_j
\]

and maximizing \( L(\alpha_i) \) with respect to the \( \alpha_i \). It turns out that only a few of the \( \alpha_i \) are non-zero. These correspond to the training points that satisfy the equality condition in Equation (5). Such points are called the support vectors and set the value of \( b \), i.e., \( b = W \cdot X_i - Y_i \). In practice, \( b \) is taken to be an average over the support vectors, that is

\[
b = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} (W \cdot X_i - Y_i),
\]

where \( N_{SV} \) is the number of support vectors.

It is often useful to introduce some slack (quantified by \( \xi_i \) below) into the SVM classifier. This amounts to replacing the constraint in Equation (5) with the new constraint

\[
Y_i(W \cdot X_i - b) \geq 1 - \xi_i.
\]

The effective reduction of the margin width in the above equation allows for some misclassification of the data. We then minimize

\[
\frac{1}{2} ||W||^2 + C \sum_{i=1}^{N} \xi_i,
\]

where \( C \) is a parameter that determines the penalty for any misclassification. If we solve the Lagrangian for this case, we will find that \( \xi_i = \alpha_i/C \) is required.

In addition, one can see that Equation (9) contains the term \( W \cdot X_i \), which is just the dot product between two vectors in feature space. We can then imagine generalizing this dot product to a space spanned by a nonlinear mapping \( \Phi \). We can then imagine generalizing this dot product to the space. We can then imagine generalizing this dot product to the space spanned by a nonlinear mapping \( \Phi \) of the features (Aizerman et al. 1964; Boser et al. 1992). This mapping can be used to take the features into a higher dimensional space, making them more easily separable. The dot product \( \Phi(X_i) \cdot \Phi(X_j) \) can be thought of as a kernel function \( k(X_i, X_j) \).

Commonly used kernels like the polynomial, Gaussian (or radial basis function, rbf), and sigmoid functions are popular due to their simplicity. These have the forms

\[
k_{\text{poly}}(X_i, X_j) = (X_i X_j)^m,
\]

where \( m \) is the degree of the polynomial,

\[
k_{\text{rbf}}(X_i, X_j) = \exp(-\gamma ||X_i - X_j||^2),
\]

and

\[
k_{\text{sigmoid}}(X_i, X_j) = \tanh(X_i X_j + r),
\]

where \( \gamma \) and \( r \) are parameters of the kernel.

The simple binary SVM described above can be generalized to support vector regression (SVR; Drucker et al. 1997) which is the algorithm we employ in this study. For the regression problem, we seek hyperplanes satisfying the equations

\[
Y_i - (W \cdot X_i - b) \leq \epsilon + \xi_i
\]

\[
(W \cdot X_i - b) - Y_i \leq \epsilon + \xi_i^*.
\]

Here, \( \epsilon \) is a tolerance parameter, i.e., there is no penalty assigned to predictions that fall within \( \epsilon \) of the true value. \( \xi_i > 0 \) and \( \xi_i^* > 0 \) are slack variables corresponding to upper and lower constraints on the system output.

The quantity the SVR must minimize is

\[
\frac{1}{2} ||W||^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*)
\]

which closely resembles Equation (12). The Lagrangian takes the form

\[
L(\alpha_i, \alpha_i^*) = \frac{1}{2} ||W||^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*)
\]

\[
- \sum_{i=1}^{N} \alpha_i [\epsilon + \xi_i - y_i + (W \cdot X_i + b)]
\]

\[
- \sum_{i=1}^{N} \alpha_i^* [\epsilon + \xi_i^* + y_i - (W \cdot X_i + b)],
\]
where $a_i$ and $a_i^*$ are Lagrange multipliers subject to the constraints $0 \leq a_i \leq C$, $0 \leq a_i^* \leq C$, and $\sum_{i=1}^{N}(a_i - a_i^*) = 0$.

For our analysis, we use the SVR algorithm implemented in the scikit-learn Python library (Pedregosa et al. 2011). We use the default $\epsilon = 0.1$ and find that changing this value does not make a noticeable difference in our predictions or the resulting MSE. The algorithm takes in a value for the penalty parameter $C$ from Equation (18) and the kernel. $m$, $\gamma$, and $r$ need also be specified depending on what kernel is being used. Optimal parameter values can be determined by splitting our sample of halos into three equal parts: a training set, a validation set, and a test set. We can then train the SVM using some pre-determined grid values: $C = \{100, \ldots, 10^3\}$ and $\gamma$ or $r = \{0.1, \ldots, 10^{-3}\}$ in powers of 10, and calculate the MSE of the validation set. The parameter values that give the minimum MSE are chosen for our analyses, where we evaluate how well the ML predictions match truth using the test set.

### 3.2. k Nearest Neighbors

The kNN algorithm is much simpler than the SVM to understand and can also be used for both classification and regression. The kNN routine calculates "distances" to the points that we are interested in predicting $Y_i$. This distance is often just a simple Euclidean distance between the features $X$, however, one can imagine using other definitions as well. The predicted $Y_i$ is then just the average of the $k$ nearest training set $Y$ values. This average can be weighted according to distance such that points further away have less impact on the predicted value. Mathematically, one can represent this as

$$Y_i = \frac{\sum_k w(d(X_i, X_k))Y_k}{k}, \quad (20)$$

where $d(X_i, X_k)$ is the distance between $X_i$ and one of its kNNs $X_k$, and $w(d)$ is the weight corresponding to that distance.

We again use the scikit-learn implementation of kNN (Pedregosa et al. 2011). The algorithm takes in a value for $k$. Again, the optimal value can be found by stepping through a pre-determined set $k = \{3, 6, \ldots, 21, 24\}$ and picking the value with the lowest MSE when the algorithm is applied to the validation set.

### 3.3. Feature Selection

Feature selection is the process by which we select which features are the most relevant for predicting $Y$ from $X$. A simple approach to this is forward feature selection which relies on an initial comparison to the base MSE, or the MSE calculated by taking all $Y_{i, \text{test, predicted}} = \langle Y_{\text{train}} \rangle$ in Equation (1), i.e.,

$$\text{Base MSE} = \frac{\sum_{i=1}^{N}(Y_{i, \text{test, true}} - \langle Y_{\text{train}} \rangle)^2}{N}. \quad (21)$$

This is just the MSE one would obtain by doing the most naive thing: predicting all $Y$ values to be the mean $Y$ of the training set (denoted as $\langle Y_{\text{train}} \rangle$ above).

Forward feature selection starts by training an ML algorithm to predict $Y$ using only a single feature. We repeat this for each individual feature and calculate its MSE value from a test set. If the minimum MSE is less than the base MSE, then we are doing better than the naive prediction, indicating that the features do contain information that is correlated with and can help us predict $Y$. We then "select" the feature that produced the minimum MSE and individually add each other feature to it and repeat the train/test procedure to calculate MSE values. At the end of this round, if the minimum MSE is smaller than the minimum MSE in the previous step, we again select the feature that produced this minimum MSE and repeat the previous procedure. At each step, if adding in an additional feature decreases the minimum MSE further, we continue. Otherwise, we stop and deem the remaining features as not having much predictive power beyond the “selected” features.

Such feature selection schemes are useful for identifying the halo parameters that are most relevant to inferring $N_{\text{gal}}$.

An interesting question arises regarding the effectiveness of ML feature selection versus more commonly used statistical decompositions such as principal component analysis (PCA) which can also be used to find the most correlated quantities (or linear combinations of properties) with $N_{\text{gal}}$. However, it is much more illuminating to think of these two methods as being mutually supportive rather than in competition. PCA can be used to first decompose a feature space into its most significant modes. As these modes are orthogonal to each other, using these as input features to an ML algorithm often allows for a cleaner classification or regression.

### 4. RESULTS

As described in Section 2, our tests use a sample of 395,832 halos from the Millennium simulation to assess the ML algorithms detailed in the preceding section. We randomly split this halo sample into three equal parts and use the first part for training, the second part for validation, and the third part for testing. The base MSE of the test set is $\sim 0.505$. We first look at the results obtained through training the ML algorithms on all six halo features.

As mentioned in Section 3.1, we do a grid-search to find the SVM training parameters ($C$, $\gamma$, and kernel) that return the minimum MSE on the validation set. This procedure selected the rbf kernel with $C = 1000$ and $\gamma = 0.0001$. We then use the SVM trained using these values to make predictions from the test set. For kNN, we use a similar search technique (described in Section 3.2) and find that using $k = 12$ gives the minimum MSE on the validation set. The kNN test set results below are derived using this value.

A set of 2D histograms in $N_{\text{gal, true}}$ versus $N_{\text{gal, predicted}}$ from our test set is shown in Figure 1. The top panel shows the SVM result and the bottom panel shows the kNN result. One can see that in both cases, the MSE is dramatically improved over the base MSE which indicates that the ML algorithms are learning some information about $N_{\text{gal}}$ from the input features as expected. The MSE values from the two different methods are very similar and hence SVM and kNN appear to be equally good for inferring $N_{\text{gal}}$ from halo features. However, we note that upon careful inspection of the 2D histograms, one sees that there is a slight bias toward underpredicting $N_{\text{gal}}$ which is discussed more below.

ML algorithms appear to match the expected distribution of $N_{\text{halo}}$ as a function of $N_{\text{gal}}$ quite well as shown in Figure 2. The panels show histograms where the y-axes correspond to the fraction of halos with $N_{\text{gal}}$ and the x-axes correspond to $N_{\text{gal}}$. The top panel was obtained using the SVM method and the bottom panel shows the analogue for kNN. We slightly overpredict the number of halos with one galaxy, and underpredict elsewhere, especially when using the SVM. The true number of galaxies in the test set is 149,064; for SVM we predict 140,519 and for kNN we predict 144,346. This phenomenon was pointed out previously and will be elaborated on below.
The galaxy correlation function $\xi(r)$ can also be used to test the robustness of the ML results. This test is especially interesting as $\xi(r)$ is the principal observable for large-scale structure analyses, the key motivation behind constructing mock galaxy catalogs. We create a mock galaxy catalog using the $N_{\text{gal}}$ values predicted by our ML algorithms. We place these galaxies randomly within their host halos according to a Navarro–Frenk–White profile in the radial direction and random point generation on a sphere in the angular directions. We calculate the correlation function for the training, true test set, and predicted test set galaxies in $5 \ h^{-1}$ Mpc bins in the range $5$–$60 \ h^{-1}$ Mpc (suitable for redshift–space distortion analyses from large-scale structure). Figure 3 shows the resulting $\xi(r)$ using SVM (top) and kNN (bottom). Each plot has two panels, the upper panel shows the actual correlation functions and the bottom panel shows the percent difference (i.e., $100 \times [\xi_{\text{predicted}}(r) - \xi_{\text{true}}(r)]/\xi_{\text{true}}(r)$) between the correlation function calculated from the predicted galaxies in the test set versus that calculated from the true galaxies in the test set. Note that differences between the $\xi(r)$ of the training set and the true test set are due to sample variance. Since the $N_{\text{gal}}$ values used to produce the predicted test set galaxies are based on the training sample, we should not expect the predicted and true test set $\xi(r)$ to agree better than this sample variance limit.

In the SVM case we see that at small scales ($\sim 5 \ h^{-1}$ Mpc), the predicted correlation function has a lower amplitude than the true correlation function by a significant amount ($\sim 40\%$). For kNN, the difference is fairly benign ($\sim 10\%$). At these small scales, the one-halo term is still important and hence the fact that we underpredict the number of halos with $N_{\text{gal}} > 1$ as shown in Figure 2 will result in lower clustering amplitudes than expected. However, for analyses of large-scale structure, we are more interested in $\xi(r)$ at $r > 30 \ h^{-1}$ Mpc where our predicted $\xi(r)$ is only $\sim 5\%$–$10\%$ lower than the true correlation function for the most part. As the amount of underprediction is fairly constant, one can even imagine implementing a simple scaling correction for crude applications. Hence, the potential for making large-scale structure mocks using SVM or kNN remains worthy of further investigation.

As mentioned above, there is a slight bias toward underpredicting $N_{\text{gal}}$. This could be due to the fact that the number of halos with $N_{\text{gal}} = 1$ dominates the overall distribution while halos with higher $N_{\text{gal}}$ are much rarer (see Figure 2). The range of values spanned by $N_{\text{gal}} = 1$ halos shows significant overlap with higher $N_{\text{gal}}$ halos for all six of the halo properties we consider (as shown in Figure 4). Hence, taking kNN as an example, even if a halo should have a larger number of galaxies, its nearest neighbors may still be dominated by $N_{\text{gal}} = 1$ halos. This will lead the algorithm to underpredict as it is essentially taking...
weighting to the bias, i.e., penalizing the MSE more if the bias is high. This can potentially reduce the amount of underprediction we see, however, we will pay the price of having a larger scatter in our predictions.

For SVM we underpredict the total number of galaxies by \(\sim 6\%\) and for kNN we underpredict by \(\sim 3\%\). These are both small and do not appear to significantly alter the correlation function shown in Figure 3 at scales relevant to large-scale structure analyses. At these scales, the correlation function is dominated by the two-halo term which comes mostly from halos containing a single galaxy. As discussed above, such halos are the most abundant by far. In addition, Figure 2 indicates that the number of halos we predict with \(N_{\text{gal}} = 1\) matches the true distribution reasonably well. Hence, it is not surprising that our predicted \(\xi(r)\) is in fair agreement with the true \(\xi(r)\) at large \(r\).

We can also look at the distribution of \(N_{\text{gal}}\) as a function of the various features. Figure 4 shows this in scatter plot form for the kNN test. The analogous plots for SVM are largely the same. Here we have randomly subsampled the number of points plotted to 3000. The black points show true \(N_{\text{gal}}\) versus features while the red points show predicted \(N_{\text{gal}}\). One can see that overall the span of the points overlaps quite well between the true and predicted sets, again indicating the general statistical agreement between the two. However, the black points do show slightly more spread overall, especially at low \(N_{\text{gal}}\).

For \(M_{200}\), this difference can be larger than a factor of two. For example, without subsampling, the domain of values in \(M_{200}\) for \(N_{\text{gal}} = 1\) spans \(\sim 2.1 M_\odot h^{-1}\) in the training data, but only \(\sim 0.9 M_\odot h^{-1}\) in the predicted test data. For \(N_{\text{gal}} = 3\), the training set spans \(\sim 5.2 M_\odot h^{-1}\) but the predicted test set only spans \(\sim 2.2 M_\odot h^{-1}\); and for \(N_{\text{gal}} = 5\), the spreads are \(\sim 6.2 M_\odot h^{-1}\) and \(\sim 2.8 M_\odot h^{-1}\), respectively. This indicates that the halo properties we have chosen to use here may not fully encapsulate the mapping to \(N_{\text{gal}}\), i.e., we are still missing some information that is relevant to the halo–galaxy relationship. This should not be surprising as most of our parameters (\(N_p, M_{200}, \sigma_v, v_{\text{max}}\)) are effectively mass tracers. The ratio of \(R_{1/2}\) and \(R_{200}\) (calculated easily from \(M_{200}\)) can be thought of as a measure of concentration, effectively the ratio of two radii that are defined in the same way for all halos. Concentration is known to trace mass (Navarro et al. 1996), but has also been found to correlate with halo environment (Wechsler et al. 2006; Gao & White 2007; Maccio et al. 2007). One can imagine that in addition to mass, spin, and environment, there are additional factors (branching from the full merger history of the halo) that might affect \(N_{\text{gal}}\). Fortunately, it appears that any other factors are not completely orthogonal to the halo properties used in this study. As shown above, our parameters seem to capture most of the halo–galaxy mapping, at least in the context of the Millennium simulations.

To identify the most predictive halo property for \(N_{\text{gal}}\) within the framework of Millennium, we perform a feature selection. We employ a forward feature selection algorithm using SVM to do this. To ensure the stability of our results, we re-perform the feature selection 10 times with different random draws of the training, validation, and test sets. Key numbers are summarized in Table 1. The MSE values quoted for each feature under the column heading “first round” correspond to the median MSE values of the 10 trials in the first round of the forward feature selection. One can see that \(R_{1/2}\) has the smallest MSE in the first round of selection which indicates that it is the best predictor for \(N_{\text{gal}}\). Using \(R_{1/2}\) as the only feature for training gives a median MSE of 0.163 which is very close to the MSE obtained using all

![Figure 3. Correlation functions \(\xi(r)\) of the training, true test set, and predicted test set galaxies. We create a mock galaxy catalog from the ML-predicted galaxies in order to calculate their \(\xi(r)\). This is an excellent test of the ML predictions as \(\xi(r)\) is a key observable used for large-scale structure analysis, a major motivator for generating mock galaxy catalogs in the first place. The bottom panels of each plot show the percentage difference between the \(\xi(r)\) calculated from the actual test set galaxies and the ML predictions. The gray dashed line marks 0% to help guide the eye. The correlation functions of the ML-predicted galaxies match the true correlation function well overall. However, in the SVM case, due to the underprediction of halos with \(N_{\text{gal}} > 1\) at small scales (around \(\xi(r) h^{-1}\) Mpc) where the one-halo term is still important, the clustering amplitude appears to be 40% smaller than expected. This is not a major deterrent from using ML for making large-scale structure mocks since for these analyses we are mostly interested in scales greater than \(\sim 30 h^{-1}\) Mpc where the difference between true and predicted \(\xi(r)\) is mostly \(<10\%\). In addition, the kNN case demonstrates better agreement between predicted and true \(\xi(r)\) at small scales. Hence, ML provides a potentially interesting alternative for creating large-scale structure mocks.](image-url)
as discussed above, can be related to halo environment through
is interesting; it contains information about the halo mass and,
values ranging from 0.189 (second round) which lists the median MSE values obtained using
parameter are shown in Column 1 and the median MSE values obtained by using only the listed
parameter are shown in Column 2 (first round). One can see that
R_{1/2} has the smallest median MSE and hence it should be the best predictor of N_{gal}
in the context of the Millennium simulations. Adding in additional parameters
does not significantly change the median MSE as indicated in the third column
(second round) which lists the median MSE values obtained using R_{1/2} plus the
parameter listed in Column 1.

features as shown in Figure 1. The other parameters yield MSE values ranging from 0.189 (M_{200}) to 0.275 (spin).

The values quoted under “second round” correspond to the median MSE obtained by adding in another feature on top of R_{1/2}. One can see that this does not significantly change/improve on the minimum MSE from the first round. Hence, it appears that most of our constraint on N_{gal} is coming from R_{1/2} in the Millennium simulations. The selection of R_{1/2} is interesting; it contains information about the halo mass and, as discussed above, can be related to halo environment through M_{200}. The fact that introducing other parameters in addition to R_{1/2} does not improve the MSE suggests that our five halo properties show a fair amount of degeneracy in terms of their feature–N_{gal} relationship.

Finally, we compare our ML results with the results from an HOD-like prescription obtained in the following way. We bin
the halos in the training set according to halo mass. We use log bins with d \log(M) = 0.2 (to ensure that each bin has at least one halo) and calculate the mean N_{gal} in each bin. We then perform a similar binning on the halos in the test set and for each halo, we predict its N_{gal} to be the corresponding average N_{gal} from the training set for the same bin. This is similar in spirit to HOD fitting, but gives the model even more flexibility. HOD fitting is restricted to power-law-like models in M and do not allow for additional curvature. After obtaining a predicted N_{gal} for the test set in this way, we calculate the MSE for this HOD-like approach. The MSE we obtain is \sim 0.199 which is indeed higher than the MSE values obtained using the ML approaches (\sim 0.16) as shown in Figure 1. This suggests that the other features we have considered contain information about N_{gal} beyond that contained in halo mass.

However, it is possible that our HOD model is merely not using the best halo property to find the mapping to N_{gal}. As our feature selection suggests, just using R_{1/2} correlates with N_{gal} very well without the aid of other halo parameters. As is such, we also use R_{1/2} (in linear bins of dR_{1/2} = 0.1) to construct a similar HOD-like model as described above using the same 10 randomizations of the training and test data as in our feature selection. The median MSE obtained this way is \sim 0.177 which is larger than the MSE obtained using all features as well as the median MSE for R_{1/2} from the feature selection as shown in Table 1. Hence, non-parametric ML techniques appear to glean additional information about the mapping to N_{gal} than traditional parametric approaches. Given the large observational data sets expected in the future, ML techniques should provide an interesting alternative for understanding the link between halos and galaxies.

4.1. Color-dependent Mocks

The Millennium semi-analytic galaxies come with b, v, r, i, z absolute magnitudes which we can use to define colors and
split galaxies into blue and red subpopulations. This allows us to study whether or not ML-based methods for learning the halo–galaxy mapping and, most importantly, making mock catalogs can be directly extended to subpopulations of galaxies that have different colors. We define blue galaxies to have \((v - r) < 0.7\) and red galaxies to have \((v - r) > 0.7\). This gives 175,177 blue galaxies and 270,792 red galaxies.

We again split the 395,832 halos equally into training, validation, and test sets for each case (blue or red) and repeat the previous tests using kNN. Note that many of our halos now have 0 red or blue galaxies reducing the size of our effective training sample. In the case of the blue galaxies, the MSE we obtain after applying kNN is 0.186 as compared to a base MSE of 0.334. For the red galaxies, we obtain an MSE of 0.293 as compared to a base MSE of 0.738. In both cases, there is a significant reduction in MSE compared to the base MSE which indicates that the algorithm is learning information about \(N_{\text{gal}}\) from our input features.

Correlation functions derived from our predictions are shown in Figure 5. One can see that the predicted \(\xi(r)\) agrees fairly well with the results. Again at small scales ~5 \(h^{-1}\) Mpc, we see that the predicted \(\xi(r)\) has a lower clustering amplitude, especially for the red galaxies (~20%). If we look at the distribution of halos with \(N_{\text{gal}}\), we again see that there is a small underprediction in the number of halos with \(N_{\text{gal}} > 1\) that can cause this effect. This is slightly more problematic here as the main goal of studying the dependence of clustering on color is to understand galaxy formation and evolution mechanisms which rely on information at small scales. However, such problems can be mitigated if we had a larger set of training data. We are now also approaching the point where the number of galaxies observed is large enough for us to begin understanding the differential clustering of blue and red galaxies at large scales. The agreement between the correlation function derived from our ML-predicted galaxies and the true correlation function at these scales is better (~10%) and hence ML-based mock catalogs can be useful for these purposes.

### 4.2. Stellar Mass-dependent Mocks

Since the Millennium semi-analytic galaxy models also supply us with a stellar mass, we can split our galaxies into high stellar mass \(M_{\text{star}}\) and low stellar mass samples. This can help us...
understand how effectively we can extend ML-based approaches to understanding the halo–galaxy mapping as a function of stellar mass, including their potential for constructing mock catalogs for these distinct subpopulations of galaxies. We make the cut at $10^{11} \, M_\odot \, h^{-1}$ which gives 71,573 high-$M_{\text{star}}$ galaxies and 374,410 low-$M_{\text{star}}$ galaxies. Stellar mass and luminosity are correlated with each other and hence by performing this split we are effectively separating our galaxies into low- and high-luminosity samples.

Once again, we split the halos randomly and equally into training, validation, and test sets. We then run kNN to predict $N_{\text{gal}}$ for each of the high and low stellar mass cases. Note that in the high-$M_{\text{star}}$ case, most of our halos now contain 0 galaxies so we have reduced the effective size of the training set by a large amount here. For the high-mass case, we obtain an MSE of 0.119 after applying kNN as compared to a base MSE of 0.250. For the low-mass case, our MSE is 0.214 as compared to a base MSE of 0.283. Again, the MSE after putting the data through a kNN algorithm is smaller than the base MSE, indicating that the algorithm is learning about $N_{\text{gal}}$ from the input features.

A plot of the correlation functions derived from the ML-predicted galaxies is shown in Figure 6. The low $M_{\text{star}}$ $\xi(r)$ agrees well with truth at large scales ($\lesssim 5\%$ deviation), however it is again low near $5\, h^{-1} \, \text{Mpc}$. Like in the above case where we split by color, this is slightly troublesome since studying the luminosity dependence of clustering is also mostly focused on understanding galaxy evolution through the clustering at small scales. Nonetheless, mocks produced using our predicted $N_{\text{gal}}$ can still benefit studies of luminosity-dependent clustering at large scales. The agreement in the high-$M_{\text{star}}$ case is poor with deviations $\gtrsim 20\%$. However, this is because there are very few high stellar mass galaxies. With only a small number of halos with non-zero $N_{\text{gal}}$, it is not surprising that the algorithm has some difficulty with the regression. Looking at the overall distribution of halos with $N_{\text{gal}}$ in this case reveals an underprediction of halos with $N_{\text{gal}} > 0$ (including $N_{\text{gal}} = 1$). Having fewer galaxies that break the $N_{\text{gal}} = 1$ threshold effectively increases the galaxy bias and hence the clustering amplitude due to the two-halo term which begins to become important near $\sim 5\, h^{-1} \, \text{Mpc}$ and is dominant at larger scales.

5. CONCLUSIONS

We have made some preliminary investigations into using ML techniques to populate dark matter halos from $N$-body simulations with galaxies. Since it is very computationally expensive to run cosmological $N$-body simulations with hydrodynamics, and perturbation theory approaches tend to have problems on small scales (such as treating redshift–space distortions correctly), ML serves as a powerful alternative for creating large numbers of mock galaxy catalogs. These are a key ingredient in large-scale structure analyses, a quickly emerging area with the advent of large galaxy surveys such as LSST, WFIRST, and Euclid which will have effective survey volumes of $\sim 10$–$100 \, h^{-3} \, \text{Gpc}^3$.

Most importantly, ML brackets a subclass of non-parametric algorithms which provide a unique way to construct the mapping from halo properties to galaxies. Unlike other techniques such as HOD, we do not need to pre-suppose a known model for the halo–galaxy relationship. The only assumption that must be made is that a function taking halo properties to number of galaxies per halo does exist and that it is smooth. It also allows us to circumvent the problems in subhalo identification which affect SHAM-based approaches.

We test two ML algorithms, SVMs and kNN, on the halos and semi-analytic galaxies in the Millennium simulation. We use six halo properties: number of particles $N_p$, $M_{\text{500}}$, $\sigma_5$, maximum circular velocity $v_{\text{max}}$, half-mass radius $R_{1/2}$, and halo spin, to characterize the mapping between halo properties and $N_{\text{gal}}$ (the number of galaxies that reside in the halo). We find that both ML algorithms give MSEs of $\sim 0.16$ for the predicted $N_{\text{gal}}$, which is much smaller than the base MSE 0.505. We also compare our ML results with those obtained using an HOD-like prescription and find that the MSE from the ML algorithms is lower in all cases. This suggests that ML is able to glean more information about the halo–galaxy mapping from the halo properties, possibly due to its non-parametric nature.

The overall distribution of number of halos as a function of $N_{\text{gal}}$ is also matched well by the ML predictions. We use our ML-predicted galaxies to create a mock galaxy catalog and calculate the correlation function from it. While in the SVM case we see a deficiency in clustering amplitude at small scales ($\sim 5\, h^{-1} \, \text{Mpc}$) by $\sim 40\%$, the predicted correlation function tracks that calculated from the true Millennium galaxies to $\lesssim 10\%$ at the scales most relevant to large-scale structure analyses. This is very important as large-scale structure science is the key motivator behind the construction of mock galaxy catalogs.

Due to the rarity of halos with high $N_{\text{gal}}$, we do find that there is a slight bias toward underpredicting $N_{\text{gal}}$. This can lead to the minor deficit in power at small scales in $\xi(r)$ as mentioned above, however, this does not appear to significantly deter our ability to make mock catalogs. As previously stated, we obtain a reasonably good matching between the predicted and test correlation functions at scales relevant for large-scale structure studies.

We see that the predicted and true $N_{\text{gal}}$ values as a function of the features are similar in spread. However, for a given $N_{\text{gal}}$, the spread in the features is slightly larger for the true values. This suggests that the features we have used here do not fully capture the mapping between halo properties and $N_{\text{gal}}$, although they do come very close. One should not be surprised by this since our features mostly trace halo mass and environment. While key, the full merger history of the halos is likely to impact properties of the halo beyond just mass and environment.

We also demonstrate a simple feature selection procedure on our halo properties. Feature selection is merely the process by which we use ML to identify the halo property most predictive in the mapping to $N_{\text{gal}}$. In the context of the Millennium simulations, our feature selection algorithm identifies $R_{1/2}$ as the most relevant parameter. This is interesting as $R_{1/2}$ is germane to both halo mass and environment.

Finally, we investigate direct extensions of our ML algorithms to understanding the halo–galaxy mapping and making mock catalogs for various subpopulations of galaxies (i.e., blue, red, low $M_{\text{star}}$, and high $M_{\text{star}}$). We find that in general the agreement between the predicted and true correlation functions is fair. However, once again we observe an underprediction of power at small scales in most cases. As studies of differential clustering in various subpopulations are aimed at understanding galaxy formation and evolution which draw on information contained in the small scales of the correlation function, this is slightly more problematic here. However, we are entering an era where the number of observed galaxies is large enough to engage in studies of differential subpopulation clustering at large scales. Our ML-predicted $\xi(r)$ matches truth to $\sim 5\%$–$10\%$ at these scales.
scales and hence provides an interesting alternative for creating mocks for such studies. The key advantage of ML is that it offers a method of inferring the halo–galaxy mapping in a model-independent manner. In addition, it is computationally inexpensive: for example, to train an SVM on \( \sim 170,000 \) points as done in this study takes \( \lesssim 1 \) hr on a single core. If we can run a large cosmological simulation (\( N \)-body plus hydrodynamics) with galaxy formation calibrated against available observations, we should be able to use the ML algorithms tested here to learn the mapping from halo properties to \( N_{\text{gal}} \) very quickly. We can then create large sets of mock galaxy catalogs from pure \( N \)-body simulations which are much less computationally expensive than running a large number of these fully armed cosmological simulations.

There also exist a number of avenues for future investigation including the use of ML-based approaches to predict not only \( N_{\text{gal}} \) but also the positions and velocities of galaxies within the parent halo. This is much more complex as it requires predicting a distribution (i.e., multiple galaxy positions \((x, y, z)\) and velocities \((v_x, v_y, v_z)\)) for each halo. One can also imagine devising methods to mitigate the bias toward underprediction (i.e., penalizing the MSE more for biased predictions or giving more weight to high-\( N_{\text{gal}} \) neighbors in a kNN implementation). These will all aid in our quest to generate mock catalogs reliably and efficiently.

We thank Daniel Eisenstein and Martin White for helpful discussions. X.X. is supported by a McWilliams Center for Cosmology Postdoctoral Fellowship made possible by the Bruce and Astrid McWilliams Center for Cosmology. H.T. is supported in part by NSF grant AST-1109730. M.N. is supported in part by the M. Hildred Blewett Fellowship of the American Physical Society, http://www.aps.org. The Millennium simulation databases used in this paper and the web application providing online access to them were constructed as part of the activities of the German Astrophysical Virtual Observatory.

REFERENCES

Aizerman, M. A., Braverman, E. M., & Rozonoer, L. I. 1964, Autom. Remote Control, 25, 821
Anderson, L., Aubourg, E., Bailey, S., et al. 2012, MNRAS, 427, 3435
Behroozi, P. S., Wechsler, R. H., & Wu, H.-Y. 2013, ApJ, 762, 109
Boser, B. E., Guyon, I. M., & Vapnik, V. N. 1992, in 5th Annual ACM Workshop on COLT, A Training Algorithm for Optimal Margin Classifiers, ed. D. Haussler (Pittsburgh, PA: ACM Press), 144
Carlson, J., White, M., & Padmanabhan, N. 2009, PhRvD, 80, 043531
Coil, A., Newman, J. A., Croton, D., et al. 2008, ApJ, 672, 153
Conroy, C., Wechsler, R. H., & Kravtsov, A. V. 2007, ApJ, 668, 826
Cortes, C., & Vapnik, V. 1995, Mach. Learn., 20, 273
Croft, R. A. C., Di Matteo, T., Khandai, N., et al. 2012, MNRAS, 425, 2766
Croton, D. J., Springel, V., White, S. D. M., et al. 2006, MNRAS, 367, 864
De Lucia, G., & Blaizot, J. 2007, MNRAS, 375, 2
De Lucia, G., Springel, V., White, S. D. M., Croton, D., & Kauffmann, G. 2006, MNRAS, 366, 499
Dolag, K., & Sunyaev, R. 2013, MNRAS, 432, 1600
Drucker, H., Burges, C. J., Kaufman, L., Smola, A., & Vapnik, V. 1997, in Advances in Neural Information Processing Systems, ed. M. C. Mozer, M. I. Jordan, & T. Petsche (Cambridge, MA: MIT Press), 155
Gao, L., Springel, V., & White, S. D. M. 2005, MNRAS, 363, L66
Gao, L., & White, S. D. M. 2007, MNRAS, 377, L5
Guo, H., Zehavi, I., Zheng, Z., et al. 2013, ApJ, 767, 122
Lemson, G., & the Virgo Consortium 2006, arXiv:astro-ph/0608019
Macciò, A. V., Dutton, A. A., van den Bosch, F. C., et al. 2007, MNRAS, 378, 55
Manera, M., Scoccimarro, R., Percival, W. J., et al. 2013, MNRAS, 428, 1036
Navarro, J. F., Frenk, C. S., & White, S. D. M. 1996, ApJ, 462, 563
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, JMLR, 12, 2825
Springel, V. 2005, MNRAS, 364, 1105
Springel, V., White, S. D. M., Jenkins, A., et al. 2005, Natur, 435, 629
Vinum, S., Weinberg, D. H., Davé, R., et al. 2012, MNRAS, 423, 3458
Wechsler, R. H., Zentner, A. R., Bullock, J. S., Kravtsov, A. V., & Allgood, B. 2006, ApJ, 652, 71
Zheng, Z., Zehavi, I., Eisenstein, D. J., Weinberg, D. H., & Jing, Y. P. 2009, ApJ, 707, 554