MNRAS **000**, 1–7 (2016) Preprint 14 December 2016 Compiled using MNRAS LATEX style file v3.0

Identification of Kuiper Belt Populations Using Machine Learning

Rachel A. Smullen,[1]★
[1] *Steward Observatory, University of Arizona, Tucson, AZ 85721, USA*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT
The classification of Kuiper belt objects into the varied populations is currently an arduous process that requires human intervention to confirm the fit. I apply machine learning algorithms, specifically the Gradient Tree Boosting Classifier, to place the 1459 Kuiper Belt objects into their 11 constituent classes. Using refinement techniques, the algorithm achieves a 94% accuracy in reproducing the original classes. Those that are incorrectly classified tend to have insecure orbit classifications, suggesting that the objects should be looked at again. I also investigate the effect of including objects that do not belong to the labeled classes and investigate the inclusion of only four major Kuiper belt populations. As current Kuiper belt surveys finish and the next generation surveys come on line, it will be critical to have an autonomous methodology for classifying objects based on orbital characteristics; machine learning is the ideal tool for this job.

1 INTRODUCTION

The Kuiper Belt is the disk of debris past Neptune; it consists of millions of icy bodies ranging from a few meters in size to thousands of kilometers. The Kuiper belt has four main populations: the cold classical belt, the hot resonant populations, objects recently scattered off of Neptune, and a detached population. The cold classical belt is fairly primordial; most of the objects lie between the 3:2 and 2:1 resonances with Neptune (~40 and 48 AU, respectively) and have low orbital eccentricity. The scattered population are Kuiper belt objects (KBOs) that have recently interacted with Neptune in such a way that their orbits have been greatly modified. The detached population all have large semi-major axes and moderate orbital eccentricities; their pericenters (point of closest approach to the Sun) are outside the 2:1 resonance.

The resonant populations were likely formed early in the Solar System's history. A plethora of works beginning from Malhotra (1993, 1995) have explored the early history of the Solar System and the sculpting of the Kuiper belt via giant planet migration. In these scenarios, Neptune and Pluto/other KBOs begin closer to the Sun than they are today. Neptune then migrates outward to its current orbit and KBOs are swept into resonances. During this process, the orbits of the resonant populations gain both eccentricity and inclination.

Orbital resonance is defined as two bodies being having an integer period ratio. In the Kuiper belt, there are many orbital resonances identified with Neptune. The most famous example, of course, is Pluto, which orbits twice for every two orbits of Neptune. However, being in resonance is more than just a semi-major axis. In the Kuiper belt, for instance, resonant and non-resonant KBOs are found at the same location. Objects in resonance have a resonant angle that is a combination of integer multiples of the three orbital angles (argument of pericenter, line of nodes, and mean anomaly) of both bodies. The average resonance angle is fixed, but the instantaneous value of the resonant angle can librate (oscillate in time) around this mean value. It is this resonant center and libration that truly defines membership in a resonance.

In this paper, I aim to classify Kuiper belt objects using information from short numerical integrations input as features into machine learning (ML) algorithms. In Section 2.1, I discuss the current classification techniques for KBOs and the catalog of objects I will be using for this work. Sections 2.2 and 2.3 describe the ML algorithms naively applied to the catalog and the refinement of the Gradient Tree Boosting Classifier chosen for the final analysis.. Section 3 shows the results of the ML classification and contains different scenarios for applying the classification. Finally, Section 4 presents conclusions and final thoughts.

2 METHODS

2.1 The Data

The methodology used to classify the orbits of observed KBOs is as follows. First, the object is observed to move over the course of a span of time. Sometimes, the arc of motion can be extended by searching through previous data. The arc is then run through an orbit fitting software that returns the orbital elements semi-major axis a, eccentricity e, inclination i, argument of pericenter ω, line of nodes Ω,

and time of pericenter passage T. There is an associated covariance matrix and error space (likely underestimated) that is returned from the fit. The orbit fit is then tested by drawing three orbits (two extreme and one middling) from the error space and integrating them forward in time for 10Myr. Finally, a person confirms by eye that the fit is reasonable. The multitude of steps, number of known objects, and the need for human intervention makes this problem the ideal application of machine learning.

Data used in this project consist of 1593 Kuiper belt objects whose solutions and classes have been determined with 10Myr numerical integrations and rigorous tests for orbital resonance. The list of classes and the number of bodies in each class are listed in Table 1. Orbits are deemed "secure" (S) or "insecure" (I) based on the agreement or lack thereof of three test orbits drawn from the error space of the original arc fit. There are 65 individual classes output for the 1593 objects, but the majority have one or two objects in the class and are not suitable for inclusion in the full data. Only classes containing more than 15 objects have been considered for the analysis. 1459 objects remain for the final analysis.

For remaining objects, the following data were output: initial arc fit a, e, i, Ω, ω, and T, final orbit fit from numerical integration a, e, and i, and resonant angle ϕ and resonant angle amplitude $\Delta\phi_{\rm res}$. The resonant angle is a combination of integer multiples of the object's orbital angles and Neptune's orbital angles. For a given resonance, the resonant angle and amplitude should be similar for all member objects. However, a resonance must be assumed to calculate the angles. Using the numerical integrations for each object, I used basic statistics at relevant time intervals to provide extra information to my classifier. This was inspired by Tamayo et al. (2016), who found that information from short integrations increased the accuracy of the classifier significantly. I output statistics at 5kyr (20 orbital periods for a Plutino), 50kyr, 500kyr, 1Myr, and 10Myr (the length of time normally utilized for classification). At each time interval, I took the mean, maximum, minimum, standard deviation, and maximum difference from time zero to the interval for a, e, i, Ω and ω.

2.2 Choosing the Classifier

The first challenge in applying machine learning to Kuiper belt classification was choosing a classifier. All classifiers used are part of the `scikit-learn` package from Pedregosa et al. (2011), which is an open source ML package for Python. I explored all classifiers that could be used with no additional parameters. The twelve used are listed in Table 2. They briefly fall into five categories: tree methods (GBC, RFC, ABC, and ETC), support vector machine methods (SVC and LVC), linear methods (LR, PAC, RC, and SGDC), clustering methods (KNC), and perceptron methods (MLPC).

At each of the five time intervals described in Section 2.1, I applied the classifier as shown in Figure 1, which presents the accuracy at each time step for each classifier. Both hold-out and k-fold cross validation were applied to prevent over-fitting. Typically, the tree methods obtained better accuracy, while the linear methods were among the worst. Figure 2 presents a magnified view of the top 20% accuracy; the Gradient Tree Boosting Algorithm (GBC) was the top performer in both cross validation schemes. Surprisingly, including more data from the numerical integrations did not always correspond to an increase in accuracy. Table 2 lists the time interval corresponding to peak accuracy and the maximum accuracy for both validation methods. The GBC algorithm was most accurate with a time interval of 50kyr, or about 200 orbits for a Plutino; at this interval, GBC returned an accuracy of 93.7% with the 5-fold cross validation. It is possible that other algorithms, with the proper tuning, could outperform the GBC algorithm, but the computational expense of that search precluded that investigation.

Table 1. Classifications of KBOs considered in this work. The first column gives the class, the second gives the total number of objects in the category, the third give the number of those that have secure classifications, while the fourth column shows the number of objects whose classifications are not as secure. The last column gives other names and information for the different populations.

Class	$N_{\rm Tot}$	$N_{\rm S}$	$N_{\rm I}$	Notes
classical m 0	703	584	119	Classical between 3:2 and 2:1
Nresonant 3 2	241	228	13	"Plutinos" in 3:2
detached 0 0	137	64	73	aka scattered disk objects
scattering 0 0	133	72	61	Still active
Nresonant 2 1	47	41	6	"Twotinos" in 2:1
Nresonant 7 4	46	25	21	
Nresonant 5 3	44	34	10	
classical i 0	40	33	7	Classical interior to 3:2
Nresonant 5 2	38	32	6	
Nresonant 4 3	20	17	3	
classical o 0	16	11	5	Classical exterior to 2:1

2.3 Refining the Classifier

The next step required was to refine the GBC algorithm at the interval 5×10^4yr (which gives 15 features used to classify and object) to obtain the best fit. I split the data into a 70% training data set and a 30% training dataset. Using smaller or larger splits results in a worse accuracy because for a smaller testing set, there aren't enough of each class to easily classify, while for a larger testing set, the training set loses sensitivity to the less-populated classes. the I used a grid search with cross validation to find the highest accuracy solution; the parameters I iterated over are listed in Table 3. Further refinement was attempted but did not increase accuracy. The final accuracy obtained by the refined algorithm was 94.3%, an increase of 0.6% over the default configuration.

The importance of the different features is shown in Figure 3. The most important features pertain to the semi-major axis of the body. The zeroth order classification of a KBO is based on its mean semi-major axis, so the mean semi-major axis is the most important feature. To identify membership in a resonance, we ideally want to identify the resonant angle and amplitude of the change in the resonant angle. However, in order to calculate a resonant angle, we must first assume a resonance. I did not want to bias the classification by assuming a resonance, so the next best identifier is the variation in semi-major axis that occurs because of res-

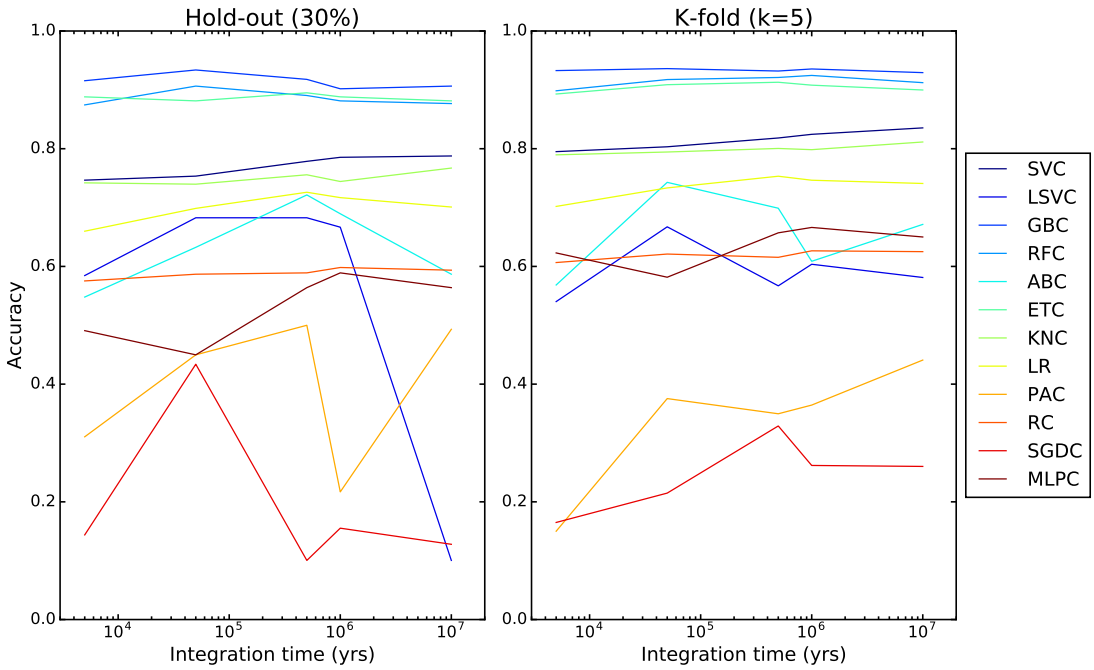

Figure 1. Classifier accuracy vs. numerical integration time for the twelve `scikit-learn` classifiers presented herein. The left panel shows the results for a 30% hold-out cross validation scheme, while the right panel shows the results for a 5-fold cross validation scheme. Overall, the tree methods, including the best performing Gradient Tree Boosting Classifier, are among the top performers.

Table 2. All classifiers tested, sorted by the maximum accuracy attained by the classifier. The names are those given by `scikit-learn`. All have been run with default parameters. The classifiers were tested on data at 5kyr, 50kyr, 500kyr, 1Myr, and 10Myr; the time with the best accuracy is listed. The first two numerical columns show the results for a hold-out cross validation scheme in which 30% of the data was used for testing purposes. The second two columns show results for a k-fold cross validation scheme with five folds. The Gradient Tree Boosting Classifier, a tree method, performed the best in both cases and was chosen for further refinement in Section 2.3.

Name	Acronym	30% Hold-out CV Time (yr)	30% Hold-out CV Max Accuracy	5-fold CV Time (yr)	5-fold CV Max Accuracy
Gradient Boosting Classifier	GBC	5×10^4	0.94	5×10^4	0.94
Random Forest Classifier	RFC	5×10^4	0.90	5×10^5	0.93
Extra Trees Classifier	ETC	5×10^4	0.89	5×10^4	0.91
Support Vector Classifier	SVC	1×10^7	0.79	1×10^7	0.84
K-Neighbors Classifier	KNC	1×10^7	0.77	1×10^7	0.81
Logistic Regression	LR	5×10^5	0.73	5×10^5	0.75
AdaBoost Classifier	ABC	5×10^5	0.72	5×10^4	0.74
Multi-layer Perceptron Classifier	MLPC	1×10^7	0.59	5×10^4	0.68
Linear Support Vector Classifier	LSVC	5×10^5	0.69	1×10^6	0.64
Ridge Classifier	RC	1×10^6	0.60	1×10^6	0.63
Stochastic Gradient Descent Classifier	SGDC	1×10^6	0.49	5×10^4	0.43
Passive Aggressive Classifier	PAC	5×10^5	0.46	1×10^6	0.42

onant interactions. A resonant KBO will have more variation in semi-major axis than a KBO in the classical belt. Thus, the minimum and maximum semi-major axis are the next most important features in the classifier. The populations of the Kuiper belt also have correlations with inclination and semi-major axis, as is shown by their moderate importance in the features. Resonant populations tend to be more eccentric than their classical components, while the trend in inclination is more subtle.

Finally, the receiver operating characteristic (ROC) plot is shown in Figure 4, which shows the true positive rate (correct classifications out of all members of the class) against the false positive rate (things falsely thought to be in the class over all object not in the class) for different threshold values in the classifier. The color denotes each of the 11 KBO classes. A perfect classifier will look like a step function and have an area under the curve (AUC) of 1. The legend of Figure 4 lists the class, the AUC, and the true number of members in the class. All classes have greater than 97% AUC values, meaning that the classifier is robust against changes in the threshold value.

In addition to the twelve features previously mentioned,

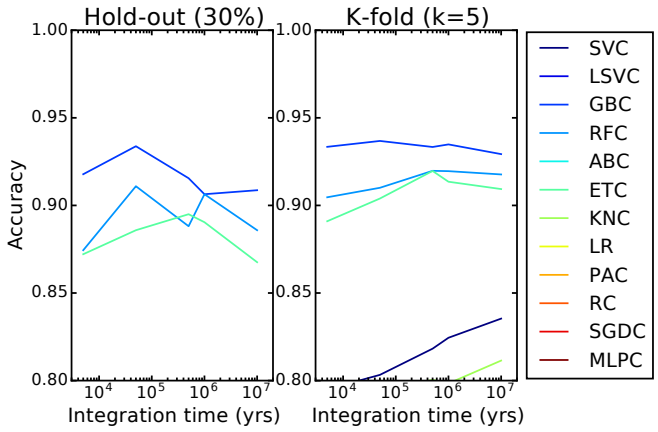

Figure 2. Same as Figure 1, but zoomed in to the top 20%. In both cases, the Gradient Tree Boosting Classifier works the best, and the best accuracy with this method is achieved using the data from a 5×10^4 year integration.

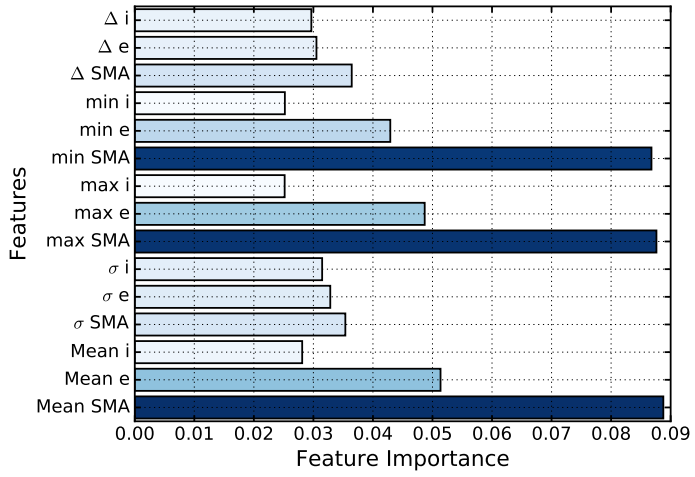

Figure 3. Relative importance of orbit features in classification. Each feature is labeled along the vertical axis. Color also corresponds to feature importance (larger numbers and darker colors are more important). The changes in semi-major axis are the most important features in this classifier, while inclination is the least important.

I also tried the classification with the inclusion of reported resonant angles and resonant angle amplitudes for each object. I do not include them as part of the main results because calculation of the resonant angle assumes a classification already. The accuracy did not change much when these values were included; it increase by 0.4%. Because most of the information given by the resonant angle is already captured in the semi-major axis variations, I was not expecting great gains in accuracy.

3 RESULTS

As stated above, the accuracy of classification of the GBC algorithm was 94.3%. Out of the 438 objects in my testing

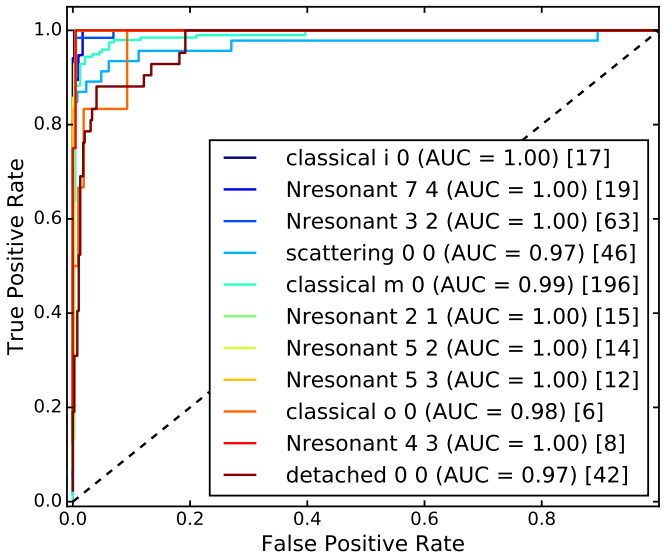

Figure 4. Receiver operating characteristic plot for each of the 11 classes identified. The color corresponds to the class. All of the colored curves are well above the black dashed one-to-one line, so the classifiers were successful. The legend also lists the area under the curve in parentheses (AUC; a value of 1 is perfect) and the number of objects in the testing set in brackets. Because most of the AUC values are close to 1, the classifier is fairly robust against changes in the threshold value.

Table 3. Grid search parameters to refine the GBC. Other parameters were left at their default value.

Parameter	Tested Range	Best fit
loss	(deviance)	deviance
learning rate	$0.1 \leq x \leq 0.9$ by 0.1	0.2
$n_{estimators}$	$50 \leq x \leq 200$ by 50	100
max depth	$1 \leq x \leq 5$ by 1	3
max features	(None, auto, sqrt, log2)	sqrt

data set, 25 were misclassified. These are shown in Figures 5 and 6, which plot eccentricity against semi-major axis. The color of the points correspond to the identified class. Circles were correctly identified, while squares indicate an incorrect classification. The color of the square denotes the GBC's chosen class and the color of the x indicates the true class. In almost all cases the the semi-major axis–eccentricity space of the KBO is consistent with both the ML and true classification.

Figure 7 shows the fraction of misclassifications as a function of class. The true number of objects in each class is shown at the top of the plot, and the colors are the same scheme as used in Figures 4, 5, and 6. Over half of the classes have errors less than 10%, while 80% of classes have error less than 20%. The misclassified objects in the 5:3 resonance tended to be placed into the classical main belt class; all have semi-major axes in the outer limits of the resonance. The other class that had many errors was the outer classical belt. These objects tended to be misclassified as detached objects; due to the sparseness of that area of space, the classes are difficult to differentiate.

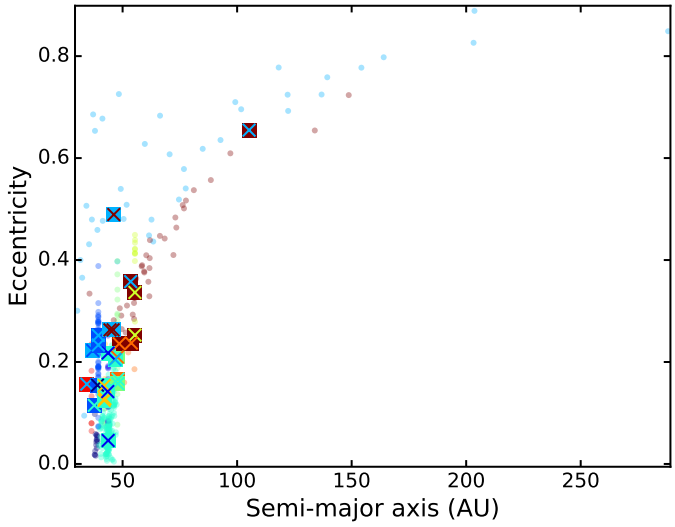

Figure 5. Eccentricity vs. semi-major axis for the 438 KBOs in the testing dataset. Colors correspond to the class and is the same coloring as Figures 4 and 7. Small circles are points that were identified correctly, and the squares with exes are things classified incorrectly. The color of the square denotes the predicted class, while the color of the x indicates the true color. Overall, the method had a 94% accuracy.

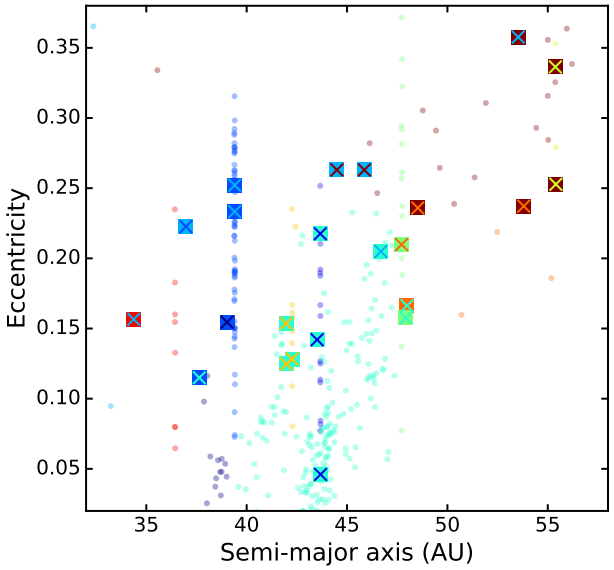

Figure 6. Same as Figure 5 but magnified to show detail of the main belt and dominant resonant populations. Many of the wrongly-classified objects are consistent with being in the GBC-identified class; as 68% of the misclassified bodies have insecure classes from numerical integrations, the machine learning algorithm may call into question the correct classification. Alternatively, if a later interval of the integration is used, the objects might be reclassified.

About 75% of KBOs in the full sample have a secure class, meaning that three orbits drawn from the error space agree. However, 70% of misclassified objects have an insecure class, indicating that at least one of the orbits preferred a different class for the object. Thus, it may be worthwhile to take a closer look at the misclassified objects. The disagreement in the two classes may indicate that the original classification is indeed incorrect (although, I wouldn't place money on the reclassification).

3.1 Folding in Removed Data

To see how the algorithm would classify any data, I added in information from the data removed in Section 2.1. The results are shown in Figure 8, which plot the new data in black triangles. Most are placed in the classical belt although they are likely in higher-order resonances whose semi-major axes lie in between the 3:2 and 2:1. Resonances are most easily identified by eye by looking for vertical lines; although small number statistics make this difficult, some of the newly-included data look like they may populate unclassified resonances. Many of the outer objects appear to be part of the detached population.

I plot the probability of the identified class membership for the core data from Section 2.1 (circles) and the data removed from the full catalog (triangles) in Figure 9 to see how certain membership is estimated to be. The removed data tend to have lower probabilities of membership but are still consistent with their identified classes at more than 95% confidence. Surprisingly, few objects have class membership probabilities less than 90%. Rigorous cuts in probability will be needed if this classifier is used for classification of newly-found KBOs.

The largest concern about folding in this extra data is that there are obvious and important resonances in the Kuiper belt that don't have many members. Thus, because these classes aren't labeled, they are falsely classified as something completely different.

3.2 Four Class Classification

As mentioned in the previous section, the concern of unlabeled resonances causes a problem when applying this classifier to new KBOs. To address this problem, I re-ran my analysis with only four classes: resonant, scattering, detached, and classical. Without any further fine-tuning of parameters, the GBC algorithm achieved 90.4% accuracy. I did add in five new features that may help inform membership in a resonance: $\dot{a}$, $\dot{e}$, $\dot{i}$, $\dot{\Omega}$, and $\dot{\omega}$. These represent time derivatives of previously defined quantities and should have non-zero values for resonant objects because resonance causes precession in orbital angles. Additionally, I normalized the pre-existing features by dividing by the mean a, e, and i so that all objects can be compared independent of true location in the Kuiper belt.

Figure 10 shows the feature importance for the twenty features in this fit. While features relating to the semi-major axis are still important, precession of the orbital angles inform most about class membership.

The ROC plot presented in Figure 11 is not quite as pristine as the previous example; AUC values range from 94–98%. However, all four classes perform similarly, so the classifier is robust against changes in threshold. This plot suggests that refining the classifier may provide strong gains in accuracy; indeed, it may even be useful to revisit the choice of relevant time interval.

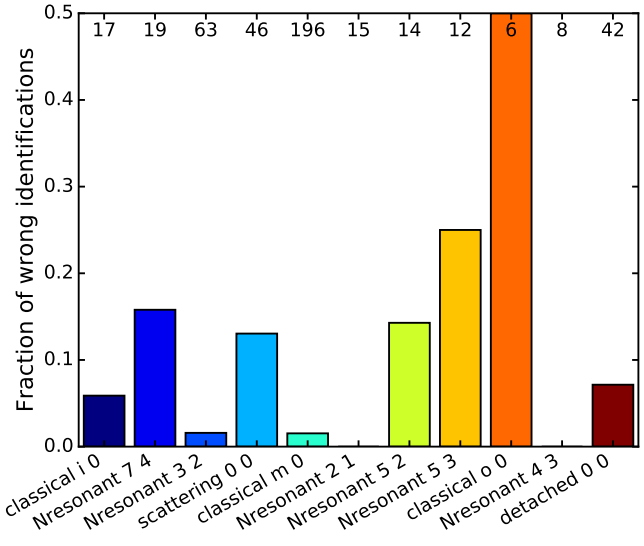

Figure 7. The fraction of wrong identifications for each class. The colors are the same as in the previous three figures. The true number of bodies in each class is listed at the top of the plot. Most of the classes had a less than 15% misclassification. The outer classical belt had a 50% misclassification, but that is likely due to the presence of only six bodies in the sample.

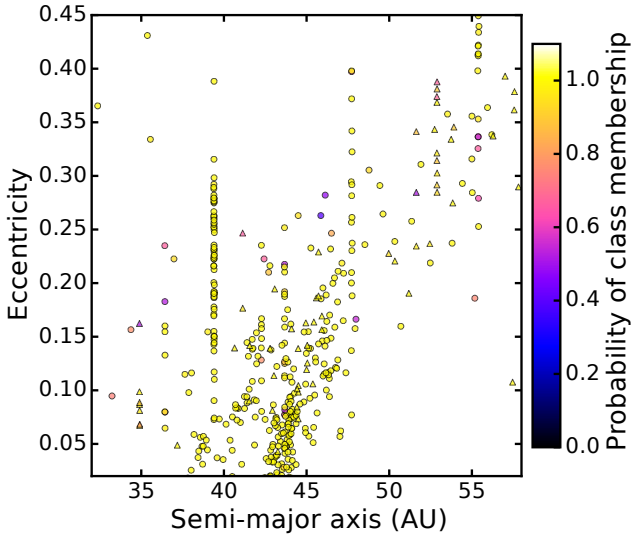

Figure 9. Eccentricity vs. semi-major axis colored by probability of class membership for both the main data (circles) and the re-included data (triangles). Very few objects have probabilities of class membership less than 90%; only a handful of the full sample have probabilities less than 50%.

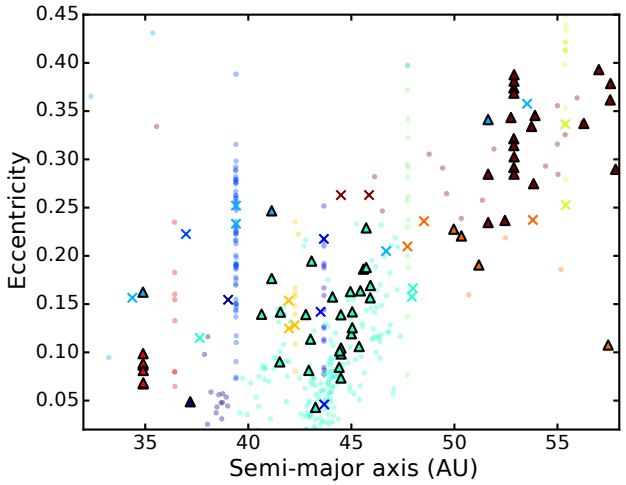

Figure 8. Same as Figure 6 but with the data removed in Section 2.1 reinserted in the colored triangles. The triangles are colored according to the predicted class membership. The misclassifications from Figure 6 are shown with exes.

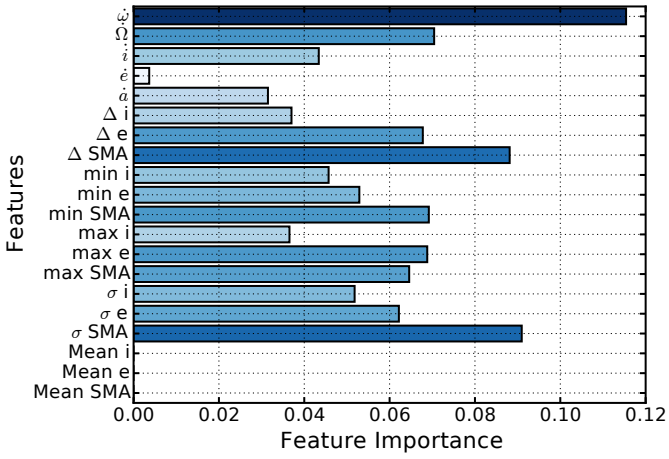

Figure 10. Relative importance of orbit features in classification for four classes only. This plot is the same as Figure 3 with the inclusion of five new features: $\dot{a}$, $\dot{e}$, $\dot{i}$, $\dot{\Omega}$, and $\dot{\omega}$.

The four classes and their identified members are shown in Figure 12. There is a lot of degeneracy between members of the outer classical belt and the detached population. Additionally, many of the resonant populations embedded in the main classical belt are difficult to distinguish. Again, exploring different time intervals may assist in creating better classifications because the resonant precession will be more apparent.

3.3 Other work

I also used unsupervised methods to see if additional correlations in the data not described by the identified classes could be found. I used a k-means clustering algorithm with the number of clusters ranging from 5–20. There was little difference in the output of the algorithm with different numbers of clusters, and many times the clusters were completely nonsensical in a–e space. I also attempted a t-SNE reduction in both 2-d and 3-d; while there was some structure in the output, I was not able to easily identify clusters and structures in a meaningful way.

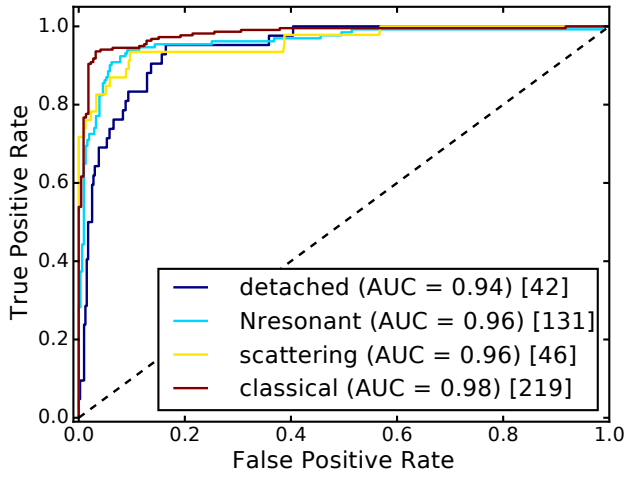

Figure 11. ROC plot for four classes only. This plot is the same as Figure 4.

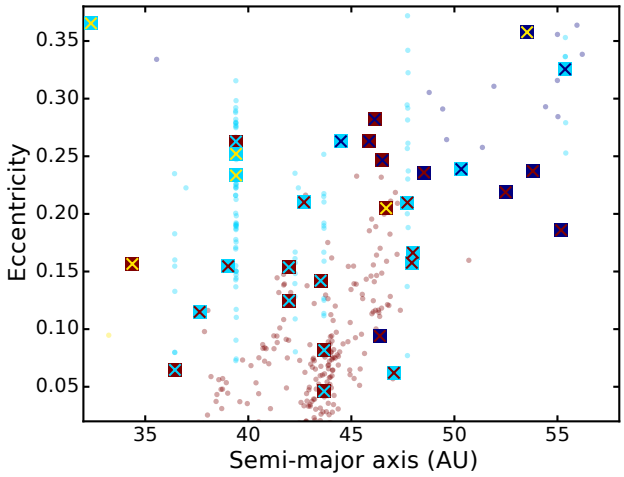

Figure 12. Same as Figure 6 but for four classes only. The classes are colored in the same scheme as Figure 11.

4 CONCLUSIONS AND DISCUSSION

This work has used ML techniques, mainly the Gradient Tree Boosting Classifier from `scikit-learn`, to identify populations of the Kuiper belt with 94% accuracy using short numerical integrations. I have investigated the properties of misclassifications, the impact of including previously discarded data, and the results obtained when only four classes are considered.

Some of the misclassifications may be due to the different methods of finding a class: the original class is found at the end of a 10Myr integration, while the ML-identified classes incorporate data from only 50kyr. It may be possible to get better agreement by using a "best of three" mindset. If we run the classifier many times on several different 50kyr chunks throughout the data, we should be able to capture changes in behavior such as falling out of resonance or scattering off of Neptune. Then, the most commonly identified class should be the correct one.

It may be most valuable for real applications to data to change the threshold value to reduce the number of falsely classified resonant KBOs. Most of the identified KBOs are in the main classical belt, where there also happen to be many resonances. One of the major bottlenecks in current KBO classification is identifying membership in a specific resonance. If a KBO can be identified with high probability as being a main classical belt KBO, then resonances do not need to be searched for, thereby freeing up computational power and manpower.

A significant benefit of using ML to classify KBOs is that the orbit fit, while not always correct, will return the same data with the same biases for all objects identified. Thus, we can agglomerate KBOs discovered with any survey and the ML algorithm will classify them with the same accuracy.

With the looming onset of the Large Synoptic Survey Telescope (LSST), we simply cannot manage the volume of KBOs about to be identified with current techniques. Thus, the application of machine learning to Kuiper belt object identification is a crucial step to ensure that our science can keep up with our data.

ACKNOWLEDGMENTS

My sincerest thanks to Kathryn Volk for giving me the data and answering all my questions about Kuiper belt dynamics. I also want to thank Leon Palafox for first teaching me everything I know about machine learning and then being a sounding board over the course of this project.

REFERENCES

Malhotra R., 1993, Nature, 365, 819
Malhotra R., 1995, AJ, 110, 420
Pedregosa F., et al., 2011, Journal of Machine Learning Research, 12, 2825
Tamayo D., et al., 2016, ApJ, 832, L22

This paper has been typeset from a TEX/LATEX file prepared by the author.