Regression methods applied to the N-Body stability problem

Author
Alexandros Papamatthaiou
aip@princeton.edu

May 2019

Abstract

Computation and machine learning has become a useful tool for scientists in understanding the planetary system variations that are possible, by predicting the stability of N-Body systems through numerical integration. In this assignment, we will be analyzing the stability of planetary systems through machine learning methods. We will determine the most significant orbital and physical properties in the stability of planetary systems using Pearson and Spearman correlation, and predict planetary stability using regression methods. We will also be looking at a more advanced method, XGBoost, which has shown promising results in planetary system stability research and other disciplines. Our data set of about 25000 planetary systems with 120 variables will be randomly split into training and testing data.

1 Introduction

Before exoplanetary research first became possible with the discovery of 51 Pegasi b, our knowledge of planetary systems, formation and dynamics was limited by our own Solar System. Exoplanetary research has unveiled massive gaseous planets orbiting at the distance that Mercury is orbiting the Sun. "Hot-Jupiter", as well as other planetary marvels challenge our understanding of how and where planets may form and necessitate the creation of new ideas such as planetary migration and orbital capture by resonance.

Computation has offered a unique tool in understanding what type of planetary systems are stable and therefore what planetary formation theories and adjustments are possible. Additionally, computational processes offer a supplementary tool in corroborating observations. The emergence of computation in planetary science provides an important task for data science. What determines the stability of planetary systems? Machine Learning provides a useful

tool for predicting the most significant features in planetary orbits. We see some example features on Figure 1.

2 Related Work

The literature on machine learning methods applied to planetary stability problems is not very extensive. Dr. Tamayo and a research team in the University of Toronto, produced randomized three planet systems and integrated them over a ling period of time to test integrating and machine learning methods [1]. We used their data set for the purposes of our own paper, and used the XGBoost method which they found to be successful. We also looked at a similar paper by Lam and Kipping of Columbia University, who used Deep Neural Networks to model the stability of planetary systems [2]. We considered the features they used (mass, eccentricity and semi-major axis) and tested their approach, but we decided to find a more comprehensive approach to feature selection, using correlation algorithms. Additionally, we read Holman and Wiegert of York University who did some early work on the stability of planetary systems[3]. They placed a strong focus on Hill radius separation, which we are investigating among other features. Another result of their research was the significance of resonance orbits in the long term stability of system, a fact which we have incorporated into our explanation of features.

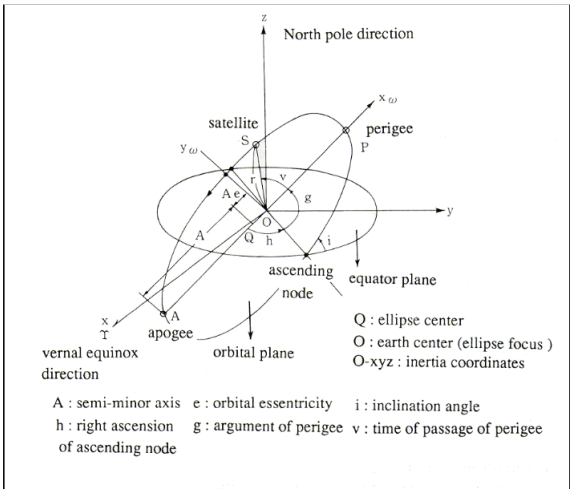

Figure 1: Some orbital parameters. From Japan Association of Remote Sensing website

3 Methods

3.1 Data set

Our data set is provided by Dr. Daniel Tamayo of Princeton University Department of Astrophysical Sciences and it follows a similar survey by a research team in University of Toronto.[2] The data set has 25000 sample planetary systems with three planets and a parent star. These systems have been integrated for a number of orbits before they became unstable. The variables that determine

the time of instability are orbital properties of the individual planets, (inclination, eccentricity, semi-major axis) as well as their relative properties (relative masses, Hill radius separation, relative orbital velocities). We are modelling 120 variables that contribute to the stability of planetary systems. The variables that we use to approximate the stability of systems are the inclination, eccentricity, beta angle, alpha angle, and other parameters of orbital geometry. The outcome that we are fitting to these features is the time (in years) it took for the system to become unstable, which occurs when a planet in the system is either slingshot out of the system, or falls into the star.

3.2 Modelling Methods

Feature Selection

1. Pearson Correlation: We use this technique for feature selection
2. Spearman Correlation: We use this technique to corroborate results from Pearson correlation.

Regression Methods

1. Ridge Regression: We use this method as a benchmark to compare XGBoost
2. XGBoost: This technique has been used widely in bibliography

We will be looking at the most significant features using two types of correlations and then applying this feature selection to our Ridge Regression algorithm. These methods will serve as benchmarks through which we will be testing XGBoost.

4 XGBoost: A Deeper Look

The XGBoost algorithm is one of the most popular machine learning algorithms of the last five years. Developed by Tianqi Chen and Carlos Guestrin of the University of Washington [4] it is known for its speed and accuracy, and its widely used in many disciplines, from sentiment analysis to high energy physics. XGBoost is an ensemble machine learning methods. It combines multiple models to provide accurate predictions according to the nature of the training data. XGBoost is based on a decision tree

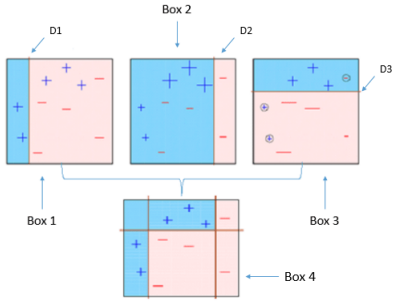

Figure 2: Representation of Bagging. From DataCamp website

structure. To decrease variability of results it employs bagging, a method that involves taking multiple decision trees and averaging their results. The algorithm also involves boosting, which means that the evaluation metrics that the algorithm uses, evolve during its fitting. A gradient descent algorithm is used to minimize errors.

4.1 The XGBoost Objective Function

The objective function that is optimized in the XGBoost algorithm consists of a training loss function and a regularization term which contributes to the complexity of the trees:

$$L = \sum_{\iota=1}^{n} l(y_\iota, u_\iota^{t-1} + f_t(x_\iota) + \sum_{t=1}^{K} \omega(f_t))$$

where y_i is the real value from the data set and u_i is defined as:

$u_i = \sum_{\kappa=1}^{n} f_k(x_\iota), f_k \in F$. We define ω_k as a regularization term, which can refer to the number of leaves on a node or an L2 term depending on the configuration of the algorithm.

4.2 Optimization

In the second step, a second-order Taylor series approximation is used:

$$L = \sum_{\iota=1}^{n} [l(y_\iota, u_\iota^{t-1} + g_\iota f_t(x_\iota) + 0.5 h_\iota f_t^2(x_\iota)] + \sum_{t=1}^{K} \omega(f_t))$$

where g_ι is the first and h_ι is the second partial derivative with respect to u_ι^{t-1} [5] The function is optimized a number of times depending on hyperparameters.

4.3 Hardware

Beyond its strong performance the XGBoost algorithm is known for its high speed. It employs block structure, using many cores within the CPU. The data is stored and reused, which saves time as they don't have to be regenerated at each iteration. It also uses usage beyond memory if data sets are too large.

5 Results

5.1 Pearson Correlation

Through Pearson correlation we found the linear relation between the variables and the time that each planetary system remained stable. We observe that there are some clusters of similar variables that are significant features. We find that

the most important features relate to the inclination of the second planet in the system. The normalized maximum and standard deviation of the inclination of both the second and the third planet in the system. This pattern is also true for the eccentricity. The normalized maximum and standard deviation of the eccentricity are important features. We also find that the Hill sphere separation of the planets is important for the stability of planetary systems.

These results are readily explainable. A high inclination and eccentricity for any planet means that the gravitational perturbation between the planets do not have stable patterns. Planetary systems, in which all planetary orbit are at roughly the same plane are believed to often form resonance relations. Resonance relations such as, for example, the 2:3 resonance of Pluto and Neptune's orbits, are believed to improve long term stability in the system. In systems that planets orbit in different planes and with different eccentricities, resonance relations are impossible to form and frequent gravitational perturbations often cause the smaller body to be slingshot-ed out of the system. Similarly the separation of Hill radius is important. The Hill sphere defines the sphere in which a planet has a greater gravitational influence than the parent star. The Hill sphere increases as a planet is more massive and orbits further away from the parent star. When a planet enter or approaches the Hill sphere of another, one perturbs the orbit of the other.

5.2 Spearman Correlation

We also used Spearman correlation to test the strength and sign of the monotonic relation between the variables and the stability of the planetary systems.

The results where similar to Pearson correlation and are visualized in Figure 2. To begin with, the line across the heat map corresponds to the correlation of each variable with itself, which is 1 and thus is green. We observe that there are three distinct squares in the heat map. They correspond to the features of the three planets in the system. We also observe smaller squares within each square. This means that similar variables (e.g. the standard deviation and the mean of a variable) are correlated with each other. The correlations of each variable with the stability time of each planetary system is observed in the edge of the box. We observe that additionally to inclination and eccentricity the semi-major axis fea-

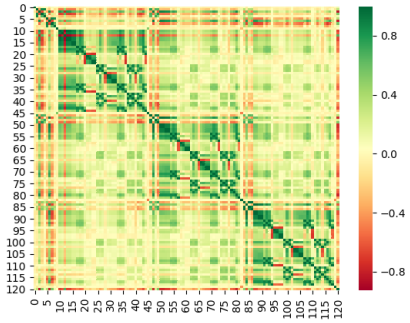

Figure 3: Heat map of Spearman correlation between all variables and the outcome (feature 121). We used the seaborn package.

ture is important (features 10-15). This makes sense, as the further away planets orbit from the parent star, the more significant and frequent gravitational perturbations become.

5.3 Ridge Regression

We began by training a Ridge Regression model on the data using the Pearson correlation data that we found to be the most meaningful. We used feature selection in order to prevent overfitting. We used a Ridge Cross validator estimator to arrive at a value of $\alpha = 1$. Our results are promising but there are some issues. To begin with, some of our predicted values are negative. This fact, of course, is concerning since all values refer to time and therefore they must be positive. We replaced all negative values with zero. Additionally, since the integration time for the training sample stops at the upper limit of 10^9 years, and many results have exactly that value, and no value above, we had to decrease the boundary of classification of stability from 10^8 to 5.1×10^8 years. This change was decided empirically, by finding the point at which false positive predictions where roughly equal to false negative predictions for the training sample.

5.4 XGBoost Regression

We trained the XGBoost Regression algorithm with all the features since it does not require preliminary feature selection. The algorithm significantly overperformed Ridge Regression, for both train and test cases, as observed in Figure 3.

We configured the XGBoost algorithm hyperparameters, first by choosing a learning rate of 0.1 and a number of estimators of 10, which according to literature would be enough for the dimensionality of our problem. For some hyperparameters, such as minimum child weight and gamma we used the default settings, while we tried multiple maximum depth and alpha values and arrived at 5 and 10 respectively through cross validation. We experienced roughly the same issues that we had with Ridge Regression. The predicted outcomes often had negative times and there were not many outcomes above the upper limit of 10^9 years so we had to lower the stability boundary of classification. Beyond high performance, XGBoost showed surprising results as its most significant features. Its results did not demonstrate the trends we observed with Pearson and Spearman Correlation. In Figure 4 we observe the importance of all 120 of the features used to train the algorithm.

	accuracy	precision	recall	f1	Stability Boundary
Ridge Regression					
train(Pearson 10)	0.78	0.73	0.72	0.75	5.10E+08
test(Pearson 10)	0.79	0.73	0.73	0.75	5.10E+08
train(Random 10)	0.44	0.47	0.47	0.47	4.10E+08
test(Random 10)	0.4	0.5	0.54	0.52	4.10E+08
XGBoost Regression					
train	0.93	0.91	0.92	0.91	3.80E+08
test	0.92	0.89	0.89	0.89	3.80E+08

Figure 4: Comparison of Ridge and XGBoost Regression

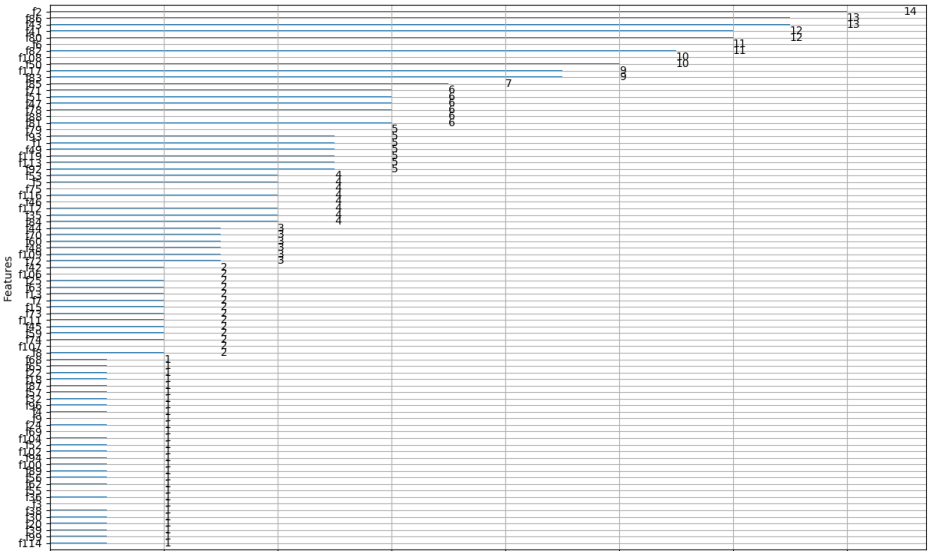

Figure 5: Feature Importances in XGBoost

While there is some structure in the features, as for example in the cluster of features from 40s and 80s, which correspond to the Hill separation of the first planet and the semi-major axis of the third planet, the importance of the features does not seem to be significantly clustered with specific features. This is not due to overfitting the sample, as we observe that the training and the testing show similarly high metrics.

6 Discussion and Conclusions

The results are promising. XGBoost overperforms Ridge Regression significantly, an event that confirms our expectations. The preliminary feature selection that we applied to Ridge Regression is based on purely monotonic feature importance, and is thus simplistic. The XGBoost algorithm executes feature selection on its own, reevaluating its fitting method at every iteration of the regression, and is thus more well equipped to study non-linear tasks such as that of planetary system stability. A surprising features is that in contrast with Pearson and especially Spearman, XGBoost does not demonstrate any structure in its choice of features. This would be readily explainable in the case there was overfitting, but since the algorithm is high-performing for both train and test sample, the explanation is more sophisticated and should be researched further. The accuracy we acquired (0.92 for the training sample is exceptional, given that in the bibliography we observed values around 0.85-0.90. [2] An explanation that covers both the exceptional performance and idiosyncrasy of feature importance is that since all samples were integrated by the same algorithm,

there could be a hidden bias that favors certain features over others. If there is a heuristic feature in the algorithm of integration, it has both astrophysical and computational importance.

An extension that could potentially be useful would be a wider comparison of simple regression methods such as Ridge, Lasso and Bayesian with more advanced ones such as XGBoost and H2O algorithms. Another potential extension would be more detailed fine-tuning of the hyperparameters to determine the best combination and its computational meaning. Sample wise, there should be more research done on different integration methods, and testing of the XGBoost algorithm on them. This expansion will allow researchers to confirm whether there is any significance to the disordered nature of feature importance in XGBoost.

Acknowledgments

A special thanks to Dr. Daniel Tamayo for introducing me to this very interesting problem and giving me access to the data set, that we used for this task.

1. Tamayo, Daniel Tamayo et al. A Machine Learns to Predict the Stability of Tightly Packed Planetary Systems. (2016).

2. Lam, Christopher Kipping, David. (2018). A machine learns to predict the stability of circumbinary planets.

3. Holman, M.J. Wiegert, Paul. (1996). Long-term Stability of Planets in Binary Systems. Astronomical Journal. 28. 1113.

4. Chen, T., Guestrin, C. 2016, arXiv:1603.02754

5. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

6. Japan Association of Remote Sensing

7. Manash, Pathak, Using XGBoost in Python, Data Camp