Regression methods applied to the N-Body stability problem

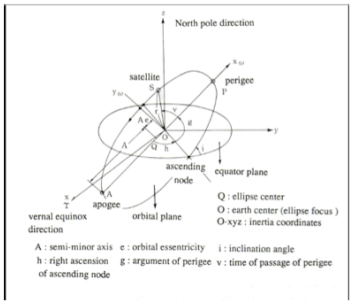
Alexandros Papamatthaiou '21
aip@princeton.edu

Abstract

Computation and machine learning has become a useful tool for scientists in un- derstanding 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 look- ing 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.

Background and Approach

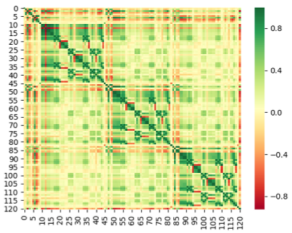
N-Body problem: What factors affect the stability of a planetary system?



Some example orbital parameters

Spearman correlation of all variables

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 dis- tinct 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.

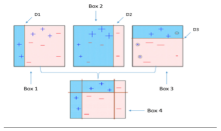


Heatmap of spearman correlation

XGBoost Algorithm

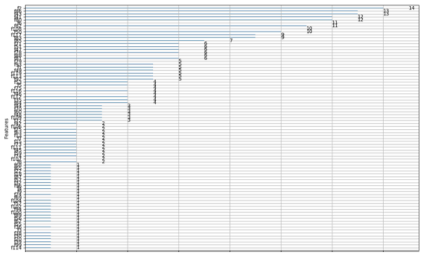
What is it?

The XGBoost algorithm is one of the most popular machine learning algo- rithms of the last five years. De- veloped by Tianqi Chen and Carlos Guestrin of the University of Wash- ington [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 meth- ods. It combines multiple models to provide accurate predictions accord- ing to the nature of the training data. XGBoost is based on a decision tree 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.



Representation of Bagging

Results



Feature Importances

How XGBoost Compares with Ridge Regression

	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

Discussion

The results are promising. XGBoost overperforms Ridge Regression signifi- cantly, an event that confirms our expectations. The preliminary feature selec- tion 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 explana- tion 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 compari- son of simple regression methods such as Ridge, Lasso and Bayesian with more advanced ones such as XGBoost and H2O algorithms. Another potential exten- sion 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 XG- Boost 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 in- teresting problem and giving me access to the data set, that we used for this task.

References

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

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

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

Manash, Pathak, Using XGBoost in Python, Data Camp