Optimizing Parameter Searches using Gaussian Process Regression

David P. Fleming
Department of Astronomy
University of Washington, Seattle
Seattle, WA 98105
dflemin3@uw.edu

Matthew C. Wilde
Department of Astronomy
University of Washington, Seattle
Seattle, WA 98105
mwilde@uw.edu

Abstract

We present the results of an exploratory analysis which seeks to model the results of computationally expensive simulations as a function of simulation initial conditions. Using bootstrapping, we identify regions in parameter space where our models perform poorly and hence where additional simulations are required. We fit the results of 10,000 simulations using ordinary least squares and ridge regression and find comparable performance. We augment these fits with additional features that are solely functions of simulation initial conditions and find that in general, the fits improve but the bootstrapped uncertainty of the estimator also grows. Our next steps include implementing Gaussian process regression to fit the simulation results and model where the fit performs poorly according to the Gaussian process's covariance matrix.

1 Project Overview

Astronomers recently discovered a planet orbiting the closest star to the Sun, Proxima Centauri b [1]. The data hints at the possibility of another planet exterior to Proxima b's orbit, Proxima c. We simulated such a system using the code `VPLANET` [2] which models the tidal and orbital interactions of Proxima b, a theoretical Proxima c and the central star in order to constrain the orbit of Proxima c if it exists. To do so, we seek to maximize the time in which Proxima b lies within 3 standard deviations of its currently observed orbital parameters. Given the large parameter space and that the simulations are computationally expensive, it is infeasible to simulate a representative sample of all planetary configurations in order to place any statistical constraints on the existence of Proxima c and its potential orbit.

We seek to fit a model which predicts the time in which Proxima b lies within 3 standard deviations of its currently observed orbital parameters as a function of the initial conditions. Here, we fit the results of these simulations with simple linear models and use bootstrapping to identify regions in parameter space where the fit performs poorly, identifying points where new simulations are required. These simple methods will be useful benchmarks with which we can compare fits using Gaussian processes and their covariance matrices for prediction error estimates.

2 Feature Creation and Model Training

Since the goal of this project is to identify when a parameter search is complete and we can then use a fitted model instead of running additional simulations, our feature vector for a given sample must be a function of simulation initial conditions, such as planet orbital properties. Before a search is complete, we identify regions of parameter space where more simulations are required by noting where our fitted model is uncertain (see Section 3).

To maximize the accuracy of our model, we created additional features that are physically meaningful functions of initial conditions for a given simulation. For example, for each planet, we create total and normal angular momentum proxy features of the form $\sqrt{1-e^2}$ and $\sqrt{1-e^2}\cos i$ for orbital eccentricity e and inclination i. We created an additional 14 physically-motivated features and call this augmented feature set "Physical". We synthesized more features by transforming the Physical feature set to all monomials of degree 2 including all cross-terms. This transformation, still just a function of initial conditions, yielded a total of about 200 features per sample. We call this augmented feature set "Polynomial".

We fit our data using ordinary least squares (OLS) and ridge regression (RR) to gauge the importance of regularization. We trained each model on a training subset containing 8,000 simulations and tested on the remaining 2,000. For RR, we performed randomized 5-fold cross validation on the training set over 50 logarithmically spaced bins for $\alpha \in [10^{-10}, 10]$ to optimize the regularization parameter α. We found $\alpha \sim 10^{-7}$ yielded optimal performance. Once fit, we evaluated both models on the training and testing set and recorded the mean squared error (MSE) and the coefficient of determination, R^2 (see Tables 1 and 2). All fits and hyperparameter optimization made use of [3].

3 Bootstrapping and Model Comparison

In order to determine the sampling distribution we used the bootstrapping method [4] with 100 realizations of our data and computed the standard deviation. Areas in parameter space with a high bootstrapped standard deviation are regions where more simulations should be run to improve the fit of the given estimator. Specifically, we computed the standard deviation using the algorithm found by [5]. We explored how the number of bootstraps influenced our standard deviation calculation and found that using an order of magnitude more realizations did not significantly impact the fit.

We found that both linear models, Ordinary Least Squares and Ridge Regression, performed comparably as seen in Tables 1 and 2 implying, at least for these models, that regularization is not important. We do notice a somewhat significant improvement in the MSE and the R^2 values when using the Polynomial feature set compared to the Physical feature set. We find that the bootstrapped standard deviation is roughly a factor of two larger in the Polynomial set relative to the Physical set suggesting that bootstrapping error scales with feature number. These simple linear methods show promise in being able to predict the outcomes of our complex simulations. We are therefore optimistic that more advanced methods will succeed in replacing the need to run more simulations.

Table 1: Physical feature set fit results.

Model	Train MSE	Test MSE	Train R^2	Test R^2	Bootstrap Median Std
OLS	0.097308	0.099976	0.575670	0.561866	0.013340
RR	0.097311	0.099923	0.575658	0.562102	0.016322

Table 2: Polynomial feature set fit results.

Model	Train MSE	Test MSE	Train R^2	Test R^2	Bootstrap Median Std
OLS	0.083028	0.090569	0.637941	0.603094	0.035985
RR	0.083197	0.090388	0.637204	0.603885	0.033817

As an example, we visualized the inclinations of Proxima b and Proxima c shown in the figure Fig. 1 and colored each simulation by the bootstrapping-derived standard deviation of the fit, our proxy for how uncertain the fit is. The linear cut off is due to dynamical stability constraints and the numerical stability of our secular models, the maximum mutual inclination is 40 degrees. In general, systems with high mutual inclination are dynamically active and causing large orbital fluctuations. As expected, we observe a lower standard deviation where there is a higher density of simulations which happens at lower mutual inclination angles and a higher standard deviation where the simulations are sampled sparsely. Both inclinations are sampled from separate Gaussian priors.

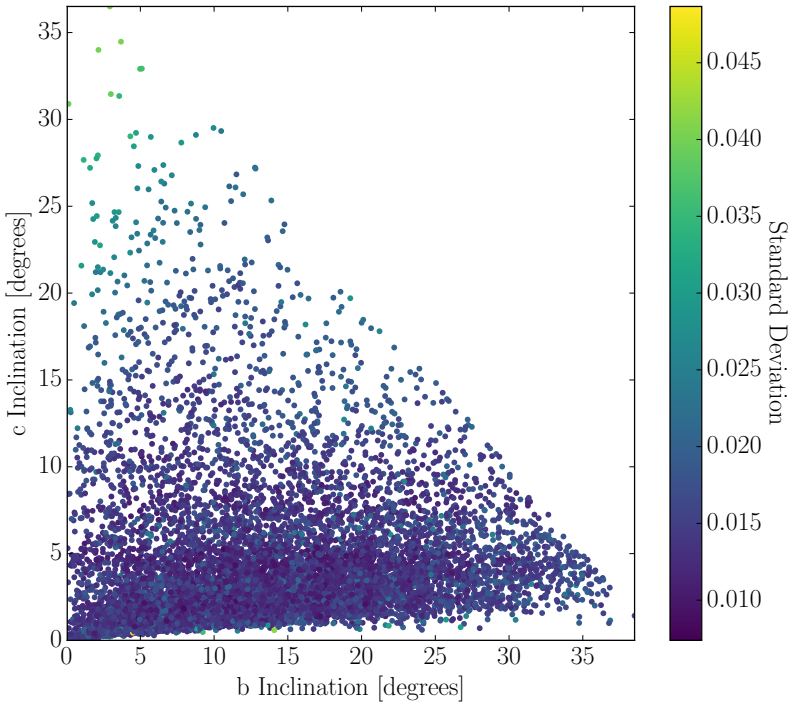

Figure 1: The inclination of Proxima b and c. Each point is colored by the standard deviation of a RR fit on the Physical feature set derived from our bootstrapping procedure. Lighter points reflect higher standard deviations.

4 Future Work

With our linear model and bootstrapping benchmarks in hand, we will now implement Gaussian process (GP) regression to model the results of our simulations (see similar work by [6], [7]). Given the computational expense of GP regression for high-dimensional data, we will partition simulation space using the `scipy` KD tree implementation and fit GPs on small batches of $N \sim 100$ data points using the GP regression functionality in [3]. As done with bootstrapping linear models, we will sample local points to identify where the fit performs poorly according to the local GP's covariance matrix and compare these estimates to our bootstrapping heuristic.

After identifying where additional simulations need to be ran, running those simulations and repeating, a natural question to ask is when is the fit good enough? To answer this question, we will augment the linear regression and GP fits with more robust ensemble methods such as a random forest regressor and the `xgboost` algorithm presented in [8] to see if they can yield an improved fit. Ideally, these properly trained ensemble methods will fit the data sufficiently well such that they will replace running computationally expensive simulations.

References

[1] G. Anglada-Escudé, P. J. Amado, J. Barnes, Z. M. Berdiñas, R. P. Butler, G. A. L. Coleman, I. de La Cueva, S. Dreizler, M. Endl, B. Giesers, S. V. Jeffers, J. S. Jenkins, H. R. A. Jones, M. Kiraga, M. Kürster, M. J. López-González, C. J. Marvin, N. Morales, J. Morin, R. P. Nelson, J. L. Ortiz, A. Ofir, S.-J. Paardekooper, A. Reiners, E. Rodríguez, C. Rodríguez-López, L. F. Sarmiento, J. P. Strachan, Y. Tsapras, M. Tuomi, and M. Zechmeister. A terrestrial planet candidate in a temperate orbit around Proxima Centauri. , 536:437–440, August 2016.

[2] R. Barnes, R. Deitrick, R. Luger, P. E. Driscoll, T. R. Quinn, D. P. Fleming, B. Guyer, D. V. McDonald, V. S. Meadows, G. Arney, D. Crisp, S. D. Domagal-Goldman, A. Lincowski, J. Lustig-

Yaeger, and E. Schwieterman. The Habitability of Proxima Centauri b I: Evolutionary Scenarios. *ArXiv e-prints*, August 2016.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] A. K. Jain, R. C. Dubes, and C. C. Chen. Bootstrap techniques for error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):628–633, Sept 1987.

[5] B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

[6] Richard R. Picard and R. Dennis Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, 1984.

[7] Pritam Ranjan, Ronald Haynes, and Richard Karsten. A computationally stable approach to gaussian process interpolation of deterministic computer simulation data. *Technometrics*, 53(4):366–378, 2011.

[8] T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. *ArXiv e-prints*, March 2016.