# Optimizing Parameter Searches using Gaussian Process Regression

David P. Fleming, Matthew C. Wilde,

Russell Deitrick, Rory Barnes, Jake VanderPlas

## Abstract

For simulations involving a large number of initial conditions, it is infeasible to sufficiently sample parameter space so methods other than grid searches or random sampling must be employed. We model the results of 10,000 simulations involving tidal and dynamical interactions between theoretical planets in the Proxima Centauri (Anglada-Escude et al. 2016) system as a function of initial conditions. Using fit uncertainties derived from bootstrapping and Gaussian process (GP) regression, we identify regions in parameter space where our models are uncertain and hence where additional simulations are required. We find our linear regression bootstrapping heuristic decently approximates the more robust GP uncertainties. We consider a parameter sweep complete if a model can accurately predict the results of unseen data. We test numerous models and find that ensemble methods perform well with XGBoost (Chen & Guestrin 2016) yielding the best performance.

## Dataset and Feature Generation

### Dataset

Our dataset consists of 10,000 simulations created using the code VPLANET (Barnes et al. 2016) which models the orbital and tidal interactions of potential realizations of the Proxima Centauri system. In these simulations, we try to infer the presence of a planet exterior to Proxima Centauri b based on their mutual interactions and hence track how each body's orbital parameters evolve. Our target variable is how long the orbital parameters of Proxima Centauri b lie within $3\sigma$ of its observed properties as maximizing this will help us place constraints on the orbital properties of a potential exterior planet.

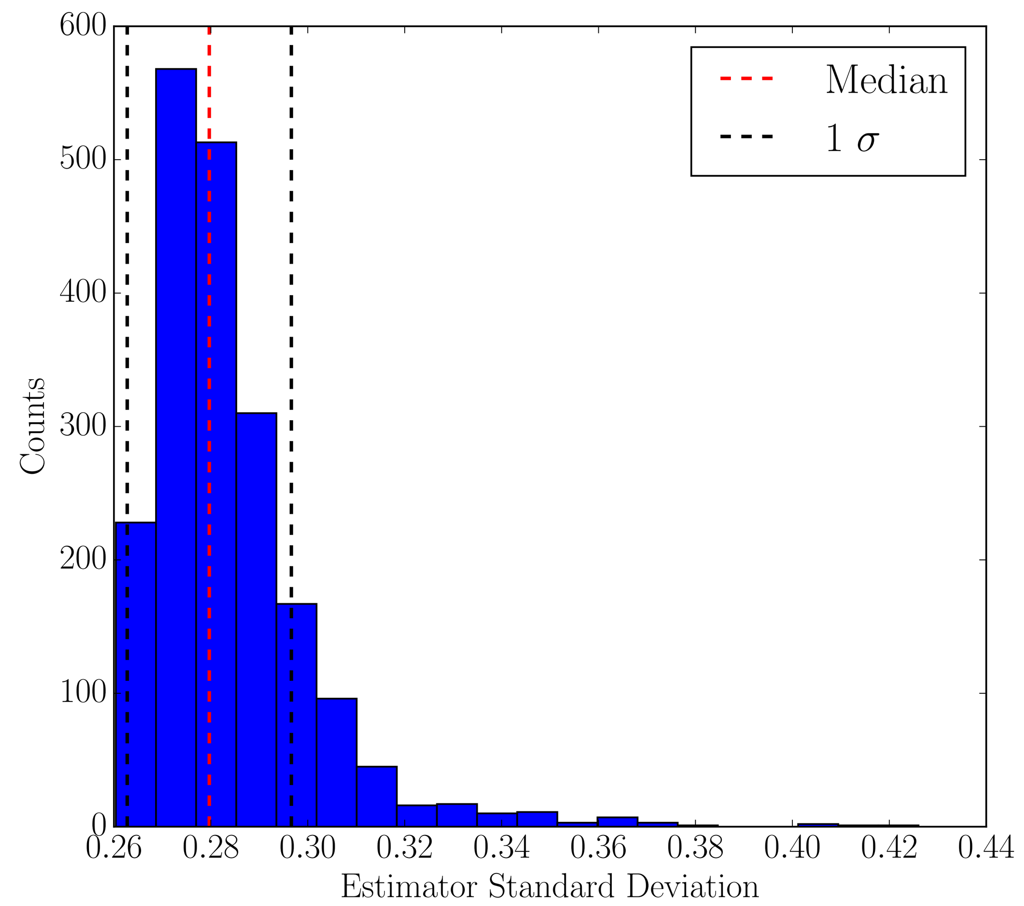
### Feature Generation

In order to maximize the the accuracy of our models, we created an additional 14 features that are physically meaningful functions of initial conditions to complement the original 9. We call this augmented feature set "Physical". We synthesized more features to probe the bias-variance tradeoff by transforming the Physical set to all monomials of degree 2 including cross-terms. This transformation, deemed "Polynomial", yielded about 200 total features.

## Where do we simulate next?

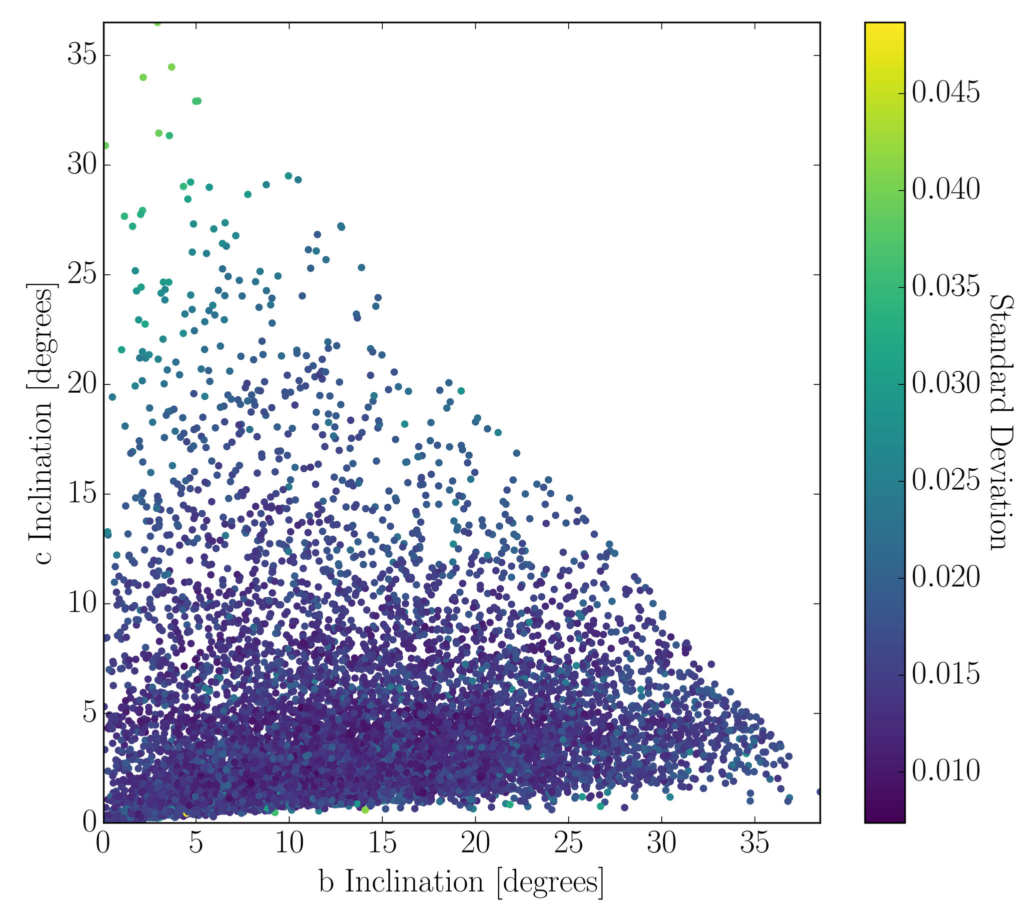
### Gaussian Process

We fit GPs with RBF, Matern and Rational Quadratic kernels and derive uncertainties on the testing set predictions via the standard deviation of the predictive distribution. If the uncertainty on a prediction is greater than $1\sigma$ above the median uncertainty, we say that that region of parameter space requires additional simulations. The top figure displays a histogram of the testing set uncertainties for a GP with a Rational Quadratic kernel and shows that our heuristic identifies the few points in parameter space where more simulations are required.




### Bootstrapping

Due to the computational cost of fitting GPs, $O(N^3)$, we used bootstrapping, fitting on various resampled realizations of the training set with replacement, and predicted on the testing set using each bootstrap. For each predicted point, we computed the standard deviation of all the bootstrapped model predictions as our uncertainty proxy. We performed bootstrapping with linear regression and ridge regression. In the bottom figure, we plot a 2D slice of our data colored by the uncertainty of the bootstrapped ridge regression estimator. As expected, the model is more uncertain in the sparser regions of this space.




## Gaussian process uncertainties versus bootstrapping

Table 3: Model Estimator Error Comparison.

| Truth | OLS | RR | GP Matern | GP RBF | GP RQ |
|---|---|---|---|---|---|
| GP Matern | 0.125 | 0.125 | 0 | 0.045 | 0.031 |
| GP RBF | 0.081 | 0.081 | 0.045 | 0 | 0.075 |
| GP RQ | 0.156 | 0.156 | 0.031 | 0.075 | 0 |

Fractional disagreement between a "truth" estimator and another evaluated on the testing set. Given the uncertainties inherent in GP regression, we consider those models "truth". We find that the GP fits tend to agree with each other while the linear models perform surprisingly well with less than 10% disagreement with the RBF kernel fit!

## When can we stop?

Ideally, an appropriately trained model will fit the data and could replace running computationally expensive simulations. We fit our dataset using numerous methods such as ordinary least squares (OLS), ridge regression (RR), ensemble methods like Random Forest (RF) and XGBoost, and GP regression. To quantify model performance we used scikit-learn's score ($R^2$) where score of 1 is the best possible and also mean squared error (MSE). We trained each model on 8,000 simulations and tested on the remaining 2,000. We tuned all hyperparameters using k=5 fold cross-validation.

Table 1: Physical feature set fit results.

| Model | Train MSE | Test MSE | Train $R^2$ | Test $R^2$ | Estimator Median Std |
|---|---|---|---|---|---|
| OLS | 0.097308 | 0.099976 | 0.575670 | 0.561866 | 0.013340 |
| RR | 0.097311 | 0.099923 | 0.575658 | 0.562102 | 0.016322 |
| RF | 0.014023 | 0.033080 | 0.941388 | 0.861286 | - |
| GP Matern | 0.050634 | 0.081951 | 0.788365 | 0.656356 | 0.279772 |
| GP RBF | 0.065208 | 0.083292 | 0.727450 | 0.650734 | 0.281678 |
| GP RQ | 0.046516 | 0.081516 | 0.805579 | 0.658183 | 0.279678 |
| XGBoost | 0.005621 | 0.028202 | 0.976505 | 0.88174 | - |

Table 2: Polynomial feature set fit results.

| Model | Train MSE | Test MSE | Train $R^2$ | Test $R^2$ | Estimator Median Std |
|---|---|---|---|---|---|
| OLS | 0.083643 | 0.093350 | 0.650397 | 0.608556 | 0.049163 |
| RR | 0.083747 | 0.092761 | 0.649963 | 0.611030 | 0.046600 |
| RF | 0.013843 | 0.033862 | 0.942140 | 0.858009 | - |
| XGBoost | 0.00871 | 0.028451 | 0.963597 | 0.880698 | - |

As expected, GPs perform better than linear models while we find that ensemble methods perform the best. We find that XGBoost consistently yields the best performance. Even with cross-validation tuned regularization parameters, more complex models seem to overfit. With more data to train the XGBoost method, we could perhaps get to an improved performance around a score of 0.9 - 0.95 where overfitting is less of an issue and we would no longer have to run simulations.

## References and Acknowledgements

Anglada-Escudé, G., Amado, P. J., Barnes, J., et al. (2016), A terrestrial planet candidate in a temperate orbit around Proxima Centauri, Nature, vol. 536 (7617) , pp. 437-440.

Barnes, R., Deitrick, R., Luger, R., Driscoll, P., Quinn, T., Fleming, D., Guyer, B., McDonald, D. V., Meadows, V. S., Arney, G., Crisp, D., Domagal-Goldman, S. D., Lincowski, A., LustigYaeger, J., Schwieterman, E. (2016). The Habitability of Proxima Centauri b 1: Evolutionary Scenarios. Astrobiology, submitted.

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

This work was supported by the NASA Astrobiology Institute's Virtual Planetary Laboratory under Cooperative Agreement number NNA13AA93A. DPF is supported by an NSF IGERT DGE-1258485 fellowship. We thank G. Anglada-Escudé for leading the Pale Red Dot team and for sharing the results. This work was facilitated through the use of advanced computational, storage, and networking infrastructure provided by the Hyak supercomputing system at the University of Washington.