Optimizing Parameter Searches Using Gaussian Process Regression

David P. Fleming & Matthew Wilde

October 24, 2016

Project Description

Astronomers recently discovered a planet orbiting the closest star to the Sun, Proxima Centauri b (Anglada-Escude 2016). The data hints at the possibility of another planet exterior to Proxima b's orbit, Proxima c. Simulations ran using the code VPLANET (Barnes et al. 2016) model the tidal and orbital interactions of Proxima b and a theoretical Proxima c in order to constrain the orbit of Proxima c if it exists. Simulating such a system requires specifying at least 10 parameters such as stellar mass and orbital properties for each planet. 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 use Gaussian processes (GPs) to model the results of the simulations as a function of the initial conditions. We will fit the results of these simulations with GPs and identify regions in parameter space where the fit performs poorly according to the GP's covariance matrix and hence identify points in parameter space where new simulations need to be ran. We will test this method against a naive regularized linear regressor where the error on the fit is estimated via bootstrapping.

Dataset

We have 10,000 simulations of the tidal and dynamical orbital interactions of planets in a theoretical Proxima Centauri planetary system. These computationally expensive simulations produce times series data of how each planet's parameters evolve. The dataset can parsed to yield the target variable, what fraction of time Proxima b lies in current orbit, as a function of tens of initial simulation conditions that are randomly sampled from physically-informed priors on the dynamics of the posited planetary system. The model parameters are likely to be correlated given the physics involved and hence a GP is a natural heuristic for modeling such a dataset.

Software

For this analysis, we will need to write code which allows us to perform batch GP since performing a regression over the entire data set is infeasible given the GP regression complexity of $O(N^3)$ for N simulations and that these fits must be ran many times to optimize the GP hyperparameters. We will write code that partitions 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 `scikit-learn`. For each fit, we will monitor the covariance length scale to see how it varies across the parameter space and adapt accordingly. Also for each fit, we will sample local points to identify where the fit performs poorly according to the local GP's covariance matrix and hence where we must perform additional simulations to improve the next GP fit.

Milestone

At the milestone, we will have completed the regularized linear regression bootstrapping error estimate portion. With this method complete, we will use it to estimate high error regions in parameter space and produce visualizations for interpretation. For example, we will be able to plot the linear regression estimate error as a function of Proxima c's orbital properties. This will be a useful base case against which we may test our more robust GP regression estimate.

Papers to Read

There papers we will read explore potential uses for Gaussian processes in interpolation for simulated data ([1], [2]) and the bootstrapping method for error estimation ([3]).

References

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

[2] 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.

[3] 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.