Contents

INQUIRY PROPOSAL FORM

Sameer Singh

December 15, 2020

1 Research Question and Goals

Ultimately, the objective of my work is to develop an algorithm that is able to discern exoplanets in photometric data. Below is a breakdown of this lofty goal:

1. Ensure that the algorithm is able to work on photometric data obtained from all major sources, to wit: *Kepler*, *K2*, and *TESS*.
2. Ensure that the algorithm is robust enough to work even in cases where the S/N is unusually low.
3. Ensure that the algorithm is able not only to detect exoplanets, but to categorize false positives as well as non-transiting phenomena.
4. Ensure that the code is readable enough that it is possible for citizen scientists who have at least some experience with the Python programming language to modify it for their own use (MIT Licensing).
5. Ensure that the time complexity of the algorithm does not exceed $\mathcal{O}(n \log n)$ and that the storage and RAM requirements of the algorithm are minimal.
6. It is widly unlikely to happen, but I would like to run the code on large batches of light curves in the hopes of detecting an exoplanet missed in previous surveys.

2 Motivation and Rationale

The prospect of finding extraterrestrial habits conducive to life has tantalized scientists and laymen alike for decades. The prospect first became reality in 1992, when astronomers discovered the first exoplanet (Wolszczan Frail, 1992); since 1992, nearly 4200 exoplanets have been discovered (National Aeronautics and Space Administration [NASA], 2020). Prior to 2006, Doppler spectroscopy—that is, the detection of periodic changes in radial velocity shifts in spellar spectra—was the dominant method for the discovery of exoplanets; now, however, "transit photometry" has come to dominate the field (Fischer et al. 2015). Transit photometry, simply put, involves the detection of a periodic "dip" in the light curve of a star, caused by the "transit" of an exoplanet in front of said star. With the launch of the Kepler Space Telescope in 2009—which eliminated problems such as turbulence mixing, traditionally associated with ground-based observations—the effectiveness of transit photometry was only reaffirmed (Koch et al., 2010). Coincidentally with the rise of transit photometry, astrophysicists have found various ways to incorporate machine learning into their work, using it to recognize patterns that have allowed for the identification of quasars and variable stars in photometric data (Carrasco et al., 2015; Naul et al., 2017). Although effective algorithms do exist today, complexity, high storage requirements, and lack of replicability have plagued them. Although their developers have made good-faith attempts to make these algorithms accessible to the public, they have often fallen short, despite the fact that citizen scientists have directly contributed to the discovery of exoplanets in numerous cases. The results of my model will be replicable and its source code relatively accessible.

3 Sources of Data/Information

All data collected by the *Kepler* mission is made available to the public, whose tax dollars funded the expedition. The data is available at the Barbara Ann Mikulski Archive for Space Telescopes and at Harvard University's Center for Astrophysics, in the form of FITS (Flexible Image Transport System) files at the former, and in the form of txt (Text) files at the latter. *Wget* scripts can be run in the shell to execute a mass download of data onto an external solid state drive in both cases. Information about confirmed planets and their properties, as well as candidates and their potential properties, will come from the NASA Exoplanet Archive, which consolidates all confirmed exoplanets and their parameters into one database.

4 Cursory Review of Literature

1. Shallue, C. J., Vanderburg, A. (2018). Identifying exoplanets with deep learning: A five-planet resonant chain around Kepler-80 and an eighth planet around Kepler-90. The Astronomical Journal, 155(2), 94.

    (a) Vanderburg & Shallue automated the process of exoplanet identification using convolutional neural networks. Convolutional neural networks – which are especially well-suited to image recognition – read in images as inputs, assign relative importance to each part of the image, and differentiate these parts of the images from one another. Their model was able to classify exoplanets with a recall of 0.95 at a precision of 0.90, meaning that it successfully classified 95 percent of real planets as planets, and that 90 percent of its classifications were real planets.

2. Vanderburg, A., Johnson, J. A. (2014). A technique for extracting highly

precise photometry for the two-wheeled Kepler mission. Publications of the Astronomical Society of the Pacific, 126(944), 948.

   (a) While Kepler was able to achieve photometric accuracies as high as 10 parts per million per 6 hours for 10th magnitude stars, few other sources have this capability. Some models estimate that K2 – because of the inherent jitteriness that it possessed as the result of its missing two stabilizing components – produced data three to four times less precise than that produced by the initial Kepler mission. By accounting for the non-uniform pixel response function of the Kepler detectors by correlating flux measurements with the spacecraft's pointing and removing the dependence, Vanderburg & Johnson were able to extract corrected light curves from raw *K2* photometry.

3. Sarkar, S., Argyriou, I., Vandenbussche, B., Papageorgiou, A., Pascale, E. (2018). Stellar pulsation and granulation as noise sources in exoplanet transit spectroscopy in the ARIEL space mission. Monthly Notices of the Royal Astronomical Society, 481(3), 2871-2877.

   (a) Stellar pulsation and granulation can introduce significant amounts of statistical noise into the light curves of Sun-like stars being orbited by exoplanets with long orbital periods and small atmospheric scale heights

4. McCauliff, S. D., Jenkins, J. M., Catanzarite, J., Burke, C. J., Coughlin, J. L., Twicken, J. D., ... & Cote, M. (2015). Automatic classification of Kepler planetary transit candidates. The Astrophysical Journal, 806(1), 6.

   (a) McCauliff et al. created the "Autovetter Project" based on random forest classifiers to automatically classify the shape of light curves as either belonging to exoplanets or false positives. Running their algorithm on Kepler Threshold-crossing events, they were able to distinguish between systematic noise, eclipsing binaries, and exoplanet candidate signatures with an error rate of just 5.85 percent, reduced further to 2.81 percent when distinctions between astrophysical false positives and systematic noise were reduced.

5. Aigrain, S., Favata, F., Gilmore, G. (2004). Characterising stellar microvariability for planetary transit searches. Astronomy & Astrophysics, 414(3), 1139-1152.

   (a) Frequently, such astrophysical phenemona as high mass ratio eclipsing binaries or hierarchical triple-star systems can generate light curves that very closely resemble the u-shaped dips in stellar light curves generated by transiting exoplanets. They key distinction is that these tend to produce dips more closely resembling ”v”s.

5 Proposed Methodology

A very brief synposis of how I plan to build my model and analyze my data follows:

1. Run the *wget* scripts necessary to mass download light curves off the Harvard-CfA website onto an external SSD. Because this data is already SFF-corrected as described in 2014 by Vanderburg & Johnson, one extra step is averted in the pre-processing stage.

2. Normalize all light curves such that the median brightness value is approximately equal to one. This will allow for uniform comparison across light curves, whose fluxes (given in units of $\frac{electrons}{second}$ might otherwise vary enough to introduce a confounding effect into the analysis.

3. Fit a spline to each normalized light curve and then divide each light curve by its spline. Splines are piecewise-defined polynomials that rely on least squares methods and derivative-based penalty parameters to model noisy data. This will allow for the detrending and flattening of data, which will ultimately ensure that transits are the only prominent signals in a given light curve.

4. Remove from the normalized and detrended light curve any high outliers, as determined by the "median + 1.5 * IQR" criteria. Although outlier removal traditionally entails the removal of both high and low outliers, it is illogical to do so here, as low outliers, if they exist, likely represent transits.

5. Construct for each point remaining in the light curve a 40-point neighborhood. Then, compute the mean value and standard deviation of the flux values in that neighborhood. If in that neighborhood there are any points whose flux is less than $\overline{x} - 2\sigma$, they will be labelled as potential transits.

6. Using the points identified as potential transits, construct a list of successive differences between points. These will serve as approximations for the orbital period of the potential exoplanet. Remove from the list any values whose modulo with respect to another value is approximately equal to the value of any other difference to remove duplicates.

7. If the successive differences list contains values not identified as duplicates, then a variant of the aforementioned modulo process must be again repeated to get an approximation for how many exoplanets are contained within a given stellar system.

8. If there are n multiple planets identified in a single stellar system, then n light curves will need to be created from the parent light curve. The traditional means by which to do this is to do as follows:

    (a) Calculate the time from the nearest transit of the planet you want to get rid of.

  (b) Find all of the points more than about half a transit duration from the time of the nearest transit.

  (c) Remove all points near the transit of the other planet

9. For each light curve, whether it be a child light curve or a parent light curve for a single-planet system, the period is approximated as the median of the successive differences between relative minima. An array of potential period is then created, starting at ten percent below the approximation, and ending at ten percent above the approximation (in other words, $P_{orb} - 0.1P_{orb}$ to $P_{orb} + 0.1P_{orb}$).

10. The array of potential periods is used as an input to the Box Least Squares Algorithm, which models transits as boxcars. It is a statistical means for guessing what the most likely period associated with a given light curve is, and the period that has the highest power is chosen as the true period of the phenomenon. Box Least Squares is superior to other similar methods such as the Lomb-Scargle Periodogram Algorithm because it captures accurately the non-sinusoidal nature of exoplanet transits.

11. Using the true period as calculated by Box Least Squares, the light curves are phase folded. That is to say, transits are stacked on top of one another with respect to the period, such that the magnitude of each transit is amplified and the S/N is increased.

12. At this point, all processing of the light curves is complete. "Garbage" signals are most likely rooted out. We now turn our attention to distinguishing between false positives and true positives. Ultimately, exoplanet signals are generally u-shaped, and signals associated with similarly periodic phenomenon are v-shaped. Mathematically, we can model the dip associated with an exoplanet as $f(t)$ and the dip associated with an eclipsing binary as $g(t)$ as follows. The second derivatives of the functions are listed as well.

$$f(t) = \begin{cases} 1 & \text{if } -1 < t < -0.5 \\ x^2 + 0.75 & \text{if } -0.5 < t < 0.5 \\ 1 & \text{if } 0.5 < t < 1.0 \end{cases}$$

$$g(t) = \begin{cases} 1 & \text{if } -1 < t < -0.5 \\ -t + 0.5 & \text{if } -0.5 < t < 0 \\ t + 0.5 & \text{if } 0 < t < 0.5 \\ 1 & \text{if } 0.5 < t < 1.0 \end{cases}$$

$$f''(t) = \begin{cases} 0 & \text{if } -1 < t < -0.5 \\ 2 & \text{if } -0.5 < t < 0.5 \\ 0 & \text{if } 0.5 < t < 1.0 \end{cases}$$

$$g''(t) = \begin{cases} 0 & \text{if } -1 < t < -0.5 \\ 0 & \text{if } -0.5 < t < 0 \\ 0 & \text{if } 0 < t < 0.5 \\ 0 & \text{if } 0.5 < t < 1.0 \end{cases}$$

13. By computing the rate of change of the rate of change between points for every $(t_1, f_1), (t_2, f_2)$ in the light curve, and storing all of these rates of change in a list, we generate a list of values with which to compare the second derivatives. Because the distances between any given t_1 and t_2 are extremely small, the approximation is generally valid. If the general pattern observed is more similar to that associated with $f''(t)$ than it is to that associated with $g''(t)$, we can say we identified an exoplanet; if the opposite is true, then we can say we identified an eclipsing binary.

14. The algorithm as described above will return the following outputs:

    (a) The EPIC ID – or unique astronomical identifier – of each star system.

    (b) Whether or not the star system contains any exoplanets (Yes - No).

    (c) If so, how many exoplanets it contains.

    (d) If not, whether or not the star system was identified as a false positive (e.g. eclipsing binary), or a totally insignificant system.

6 Equipment and Resources

I expect to be making use of Python 3.6. An object-oriented programming language with extensive support for numerical and scientific computing is necessary, and although Java and C++ might meet these criterion, they are both less readable and have less community support for astrophysics than Python. C and FORTRAN could be used to optimize certain calculations, and R is generally considered the gold standard for statistics in the world of programming, but Python reconstructs all the functionality of all these languages using packages such as pandas, numpy, and scipy. Moreover, it has built in support for certain astronomical functions with open-source libraries such as astropy and lightkurve, the scale of which is not nearly matched in any other programming language. I will be writing my code in Jupyter Notebooks, which are considered the gold standard in data science for their cloud-based accessibility. My laptop is a 2019 MacBook Air and I am supplementing it with a 500 GB SSD in case storage space is needed for extraneous data. The reason for using MacOsX rather than Windows is that the Mac provides quick access to the command line for running scripts in the shell via its Terminal application. The Windows Powershell can replicate the same functionality but is slightly more difficult to use. All code will be stored in a private repository on Github – to be made public when ready – and all version control will be done through Git. SQL might be useful for large-scale queries of the NASA Exoplanet Archive – which is, after all, a relational database – but web crawlers can be built in Python to replicate the functionality of SQL.

7 Potential Challenges

I suspect that the most significant challenge is ultimately going to be the nature of *K2* data and transit photometric data in general. As stated earlier, some models estimate that *K2* – because of the inherent jitteriness that it possessed as the result of its missing two stabilizing components – produced data three to four times less precise than that produced by the initial Kepler mission (Vanderburg Johnson, 2014). That is to say, *K2* light curves actually contain more noise than they do signal. Moreover, the depth-of-transit, given as $(R_p/R_s)^2$, where R_p is the planetary radius, and R_s is the stellar radius, typically has a value of less than 0.01, or 1 percent – the 1 percent threshold exists for planets on the order of Jupiter or Saturn (Carter & Winn 2010), which themselves comprise only about 1 percent of all exoplanets. The model will need to be extremely sensitive and robust in the face of large quantities of highly noisy data, and will need to pick up dips that are, at best, only one percent lower than the median value for the flux associated with their star.

8 Expected Approval

I have reached out to Dr. Andrew Vanderburg – the librarian and creator of the Harvard-CfA database – to ensure that I can use his SFF-corrected light curves without infringing on his intellectual property rights. I have ensured that all Python packages and frameworks I will be using are open-source and usable for academic purposes. There is no human or animal experimentation involved in this project, nor is there any contact with volatile compounds.