CMSC 25025 / STAT 37601

Machine Learning and Large Scale Data Analysis

LSD PROJECT 4

Due: Tuesday, June 4th, 2013

This is the fourth and final large scale data project. In this project, you will work with the Kepler light curve dataset. This dataset consists of light curves of 166,904 stars, and is divided into 338 subsets: `0.part.zip` to `337.part.zip`.

This project contains four parts. In the first part, you will implement a method for kernel regression and apply it to some selected light curves. In the second part, you will extend this code to detect planets and eclipsing binary stars using a thresholding technique. In the third part, you will build on this approach to differentiate between exoplanet transits[1] and eclipsing/occluding of binary stars. In the fourth part, you will get bonus points by developing your own novel approach to exoplanet detection. To get scalable implementation, we suggest you read the *programming hints* before beginning to write your code. This section also provides the code to load the light curves.

An overview of the Kepler data processing pipeline is given in this article: `http://iopscience.iop.org/2041-8205/713/2/L87/pdf/2041-8205_713_2_L87.pdf`

PART I. KERNEL REGRESSION (20 points)

Implement the *Nadaraya-Watson kernel estimator* and fit three light curves, one from each of the `ep`, `fp`, and `conf` examples below. Experiment with different kernels that are commonly used:

- *Boxcar* $K(x) = \frac{1}{2} I(x)$
- *Gaussian* $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$
- *Epanechnikov* $K(x) = \frac{3}{4}(1 - x^2) I(x)$
- *Tricube* $K(x) = \frac{70}{81}(1 - |x|^3) I(x)$

where

$$I(x) = \begin{cases} 1 & \text{if } |x| \le 1 , \\ 0 & \text{otherwise.} \end{cases}$$

Try different values of bandwidth h, for example $\{0.1, 0.2, \ldots, 1\}$. You may wish to try other values of h as well. Plot the results and comment on the influence of different kernels and bandwidths.

Pass in your code as `lsd_project4_prob1.ipynb` in your `submissions/lsdproj4` directory.

[1] `http://en.wikipedia.org/wiki/Transit_(astronomy)`

PART II. DETECTING TRANSITS AND ECLIPSING BINARY STARS (50 points)

Your objective in this part is to find exoplanet transits, and eclipses and occultations[2] of binary stars. In this dataset, a few of the stars are categorized:

- `conf`: the star is confirmed to have planets
- `fp`: confirmed to have no planet
- `eb`: confirmed to be an *eclipsing binary star system*[3] which has no planet

Note that some light curves, such as for Kepler 16, may have exoplanet transits, binary star eclipses and occultations. Such a star would be categorized as `conf`. For each type, we have provided light curves of some example stars; see the plots below.

For `conf` stars, transits coincide with periodic drops in the light flux. Such spikes are not observed in light curves of `fp` stars. The `eb` stars are more interesting. Their light curves will have two periodic flux drops, corresponding to the eclipses and occultations.

In this part your goal is to identify stars of type `conf` and `eb`. Note that approximately 0.005% of the light curves are `conf`, and about 1% are `eb`. You are asked to implement the algorithm we used in some preliminary research with these data. But as you will see, there is significant room for improvement.

Process each of the light curves as follows.

1. Fit the light curve using the regression function you implemented in Part I. Use the kernel you select. Choose the bandwidth by cross validation.
2. Compute the residual r as $r_i = y_i - \widehat{y}_i$, where $y = (y_1, \ldots, y_n)^\top$ is the response (light curve value) and $\widehat{y}$ is the fitted value.
3. Standardize the residual so that it has zero mean and variance one:

$$r_i \longleftarrow \frac{r_i - \sum_i^n r_i/n}{\sigma},$$

where σ is estimated using the *median absolute deviation* (MAD):

$$\widehat{\sigma} = 1.4826\,\mathrm{MAD}(\mathbf{X}) = \mathrm{MAD}(\mathbf{X})/\Phi^{-1}(3/4)$$
$$\mathrm{MAD}(\mathbf{X}) = \mathrm{median}(|\mathbf{X} - \mathrm{median}(\mathbf{X})|).$$

[2] `http://en.wikipedia.org/wiki/Occultation`
[3] `http://en.wikipedia.org/wiki/Binary_star`

4. Compute the *universal threshold* $\beta = \sqrt{2 \log n}$. Threshold the residual, where we set r_i to zero if it is greater than or equal to $-\beta$:

$$\widetilde{r}_i = \begin{cases} r_i & \text{if } r_i < -\beta \\ 0 & \text{otherwise.} \end{cases}$$

5. Compute the ℓ_1 norm of thresholded residual: $\|\widetilde{r}\|_1 = \sum_{i=1}^{n} |\widetilde{r}_i|$.

Now, rank the stars in order of decreasing norm $\|\widetilde{r}\|_1$. Top ranked stars are considered likely to be `conf` or `eb`.

To understand this algorithm, we suggest you inspect the residual of the example stars. Feel free to use any other threshold or norm that you think is appropriate.

Submit a file named `lsd_project4_rank_a.txt`, which contains the ids (one id per line) of each star in rank order. Try to process as many stars as you can. If you don't manage to compute over the whole dataset, turn in the ranking based on light curves you processed. You could also complete the ranking with a random ranking for the unprocessed stars. If you are unable to process all of the stars, tell us the number that you are able to process.

To help you check your results, we will provide you with a utility function that computes the area under the precision-recall curve for a given ranking. You will have 10 tokens to call this function.

PART III. DETECTING PLANETS (30 points)

Your algorithm above is capable of separating `conf` and `eb` from the others, but it is not designed to separate `conf` and `eb`.

In this part you are asked to develop your own ideas and implement an algorithm for distinguishing `conf` and `eb`—in other words, you will attempt to actually find planets.

As discussed in class, one idea for doing this is to look for two types of peaks, corresponding to eclipses and occultations, which indicate a binary star system. You may wish to use your top-ranked stars from above to visualize curves for likely `eb` stars.

Prepare a second ranking, called `lsd_project4_rank_b.txt`, with the goal of placing light curves with planet transits near the top, followed by light curves for eclipsing binaries. Note that there are fewer than 100 `conf` stars and fewer than 2,000 `eb` stars in this dataset.

We will again provide you with a utility function that computes the area under the precision-recall curve for a given ranking, restricted to `eb` and `conf` stars.

Pass in your code for both parts above as `lsd_project4_prob2.ipynb`. Briefly describe your algorithm.

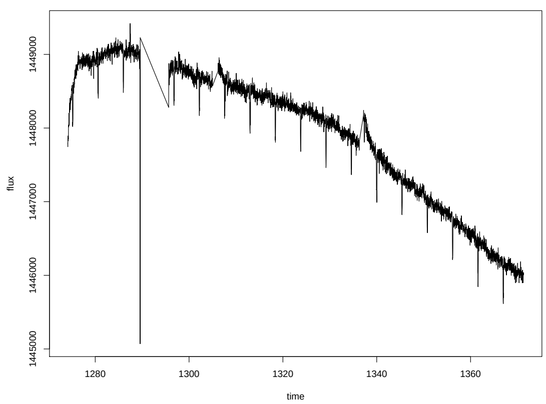
PART IV. MY LITTLE PLANET HUNTER

As extra credit, you can help us improve Part II, by developing a new approach to transit detection.

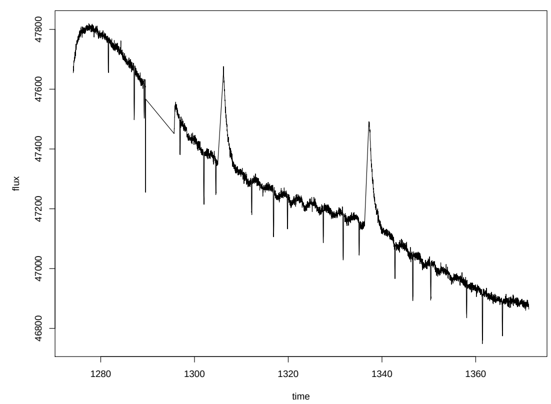
See `http://en.wikipedia.org/wiki/Methods_of_detecting_extrasolar_planets` for some example methods. One possible direction worth exploring is to exploit the periodicity of transits.

Pass in your code as `lsd_project4_prob4.ipynb` together with a `lsd_project4_rank_c.txt` file. Briefly describe your method in the iPython notebook.

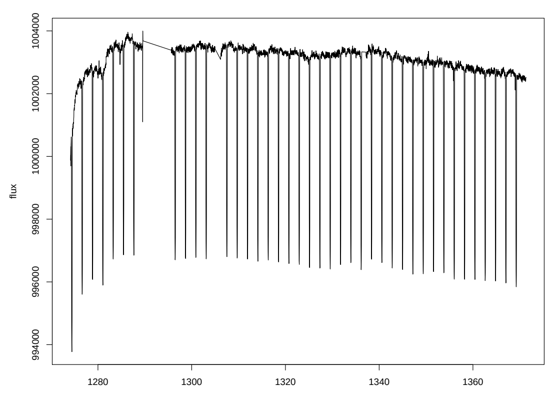
LIGHT CURVES OF EXAMPLE STARS



(a) subset 189, id 914193

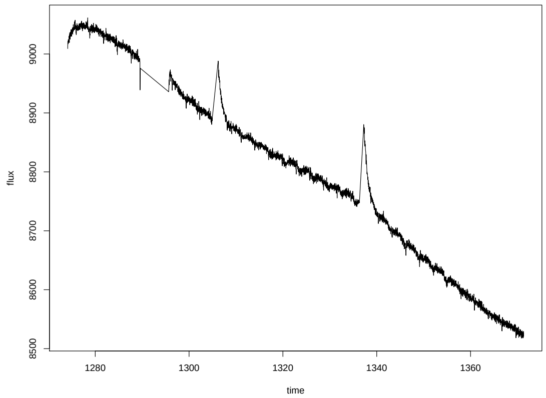


(b) subset 126, id 2042713

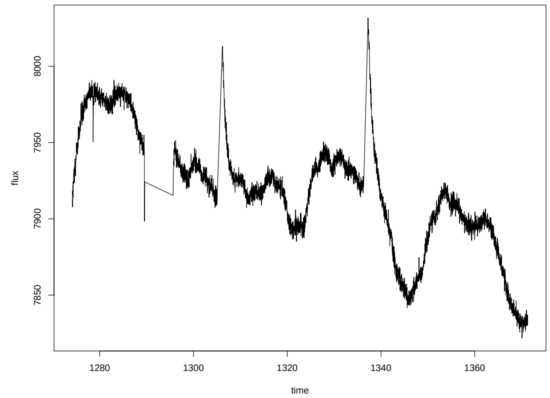


(c) subset 276, id 6852096

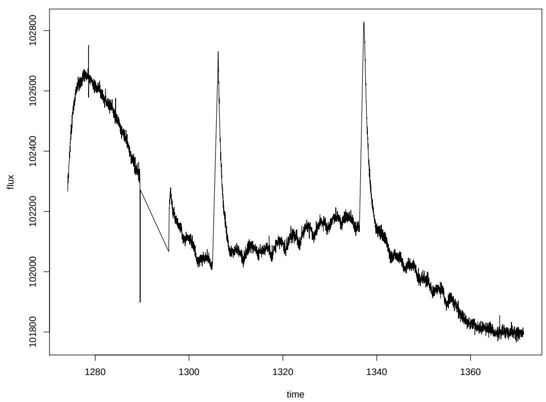
Figure 1: `conf`–light curves for stars with confirmed planet transits



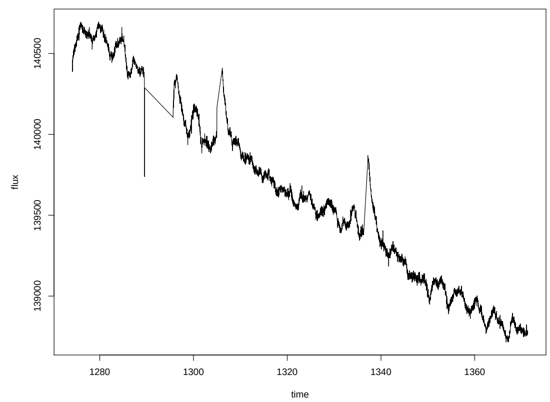
(a) subset 37, id 3972533



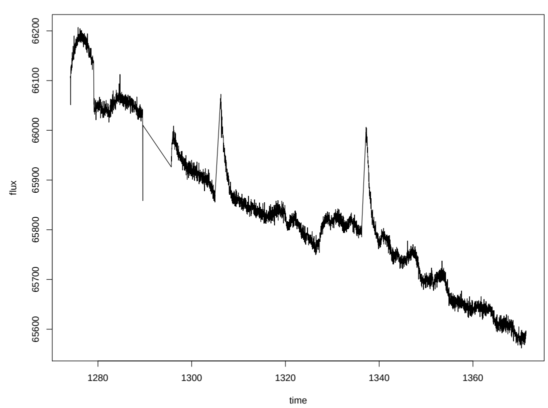
(b) subset 245, id 1255171



(c) subset 211, id 5004731

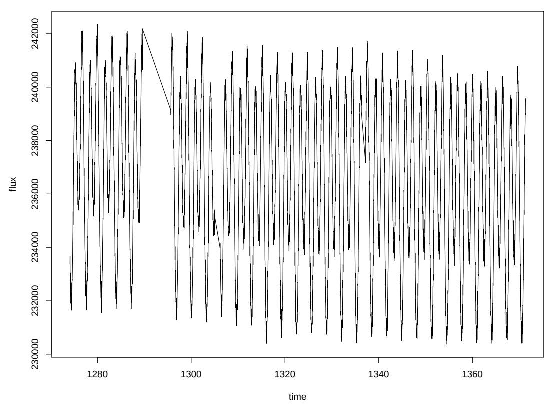


(d) subset 199, id 4848370

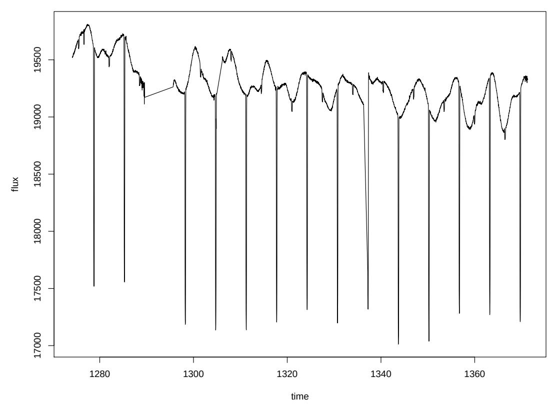


(e) subset 64, id 6258272

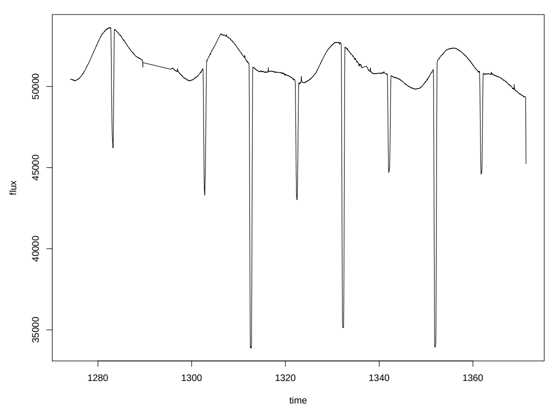
Figure 2: `fp`–light curves for stars that were false positives; no planet transits



(a) subset 102, id 3096237



(b) subset 313, id 7412246



(c) subset 231, id 485585

Figure 3: `eb`–light curves for eclipsing binary stars