Regression Revisited:
Linearity 1-2: Supplemental

This note develops a few model-fitting scenarios where linear regression can be performed on simulated data to mimic real world research tasks. Kind of like the old Mathnet show on PBS "The data is made-up, but the problems are real." This pseudo-real situation gives us the ability to explore the problem at exactly the contact point between the mathematical procedure we are practicing and the application where it shows up. Furthermore since we are using simulated data-sets we have the 'correct answer' available so we can conduct precise analysis of any errors or procedural artifacts which arise from the mathematics. (This secondary analysis is generally not available to scientists and engineers without careful planning or further experimentation.)

Application: Stellar Parallax

The distance to nearby stars is estimated using a combination of elementary geometry, careful data collection and modern technology. In the 90's the Hipparcho's mission provided a catalogue of stellar positions repeatedly captured over a three year period. This data can be used to estimate the distances to nearby stars using the stellar parallax. The idea is a type of triangulation based on an angular version of similar triangles. One can explore the basic idea of parallax using thumb triangulation (See Figure 1). For stellar parallax the concept is the same, however the parallax is based upon the difference in position of the Earth in its rotation around the sun. (See Figure 2).

While the concept of stellar parallax was well understood by classical astronomers (think Copernicus, Kepler, Brahe on forward) the actual measurements are difficult to take accurately, so only about 100 or so stars had well known Parallax measurements until the early 90's. The Hipparchos mission from 89-93 carefully measured stellar positions reliably from orbit and helped create a much larger catalog of reliable parallax measurements, and thus a much more accurate and complete picture of our local stellar neighborhood.

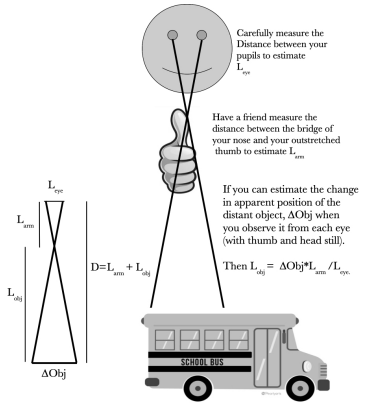

Figure 1: Thumb Parallax Fundamentals

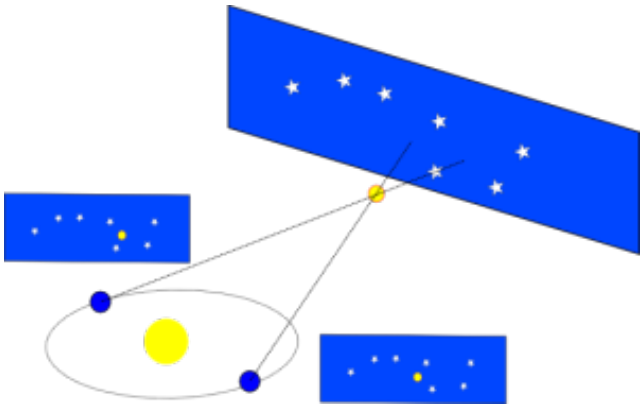

Figure 2: Stellar Parallax (scraped from U. Nebraska-Lincoln)

Nothing is Ever Easy:

Stellar Parallax measurements are angular measurements. While this might seem awkward or unwieldy to us earthbound pedestrians, the linear measurements on a starfield image are easily converted to angular measurements using careful calibration of the viewing window of the telescope or camera. (So we won't worry about that aspect at all) when one looks at the data for a single star viewed over the course of the Hipparchos mission the resulting measurements trace spiraling paths which are a result of the superposition of the relative proper motion of the star and our sun and the wiggle due to the parallax from our earth wobbling around the sun. (See Figure 3.)

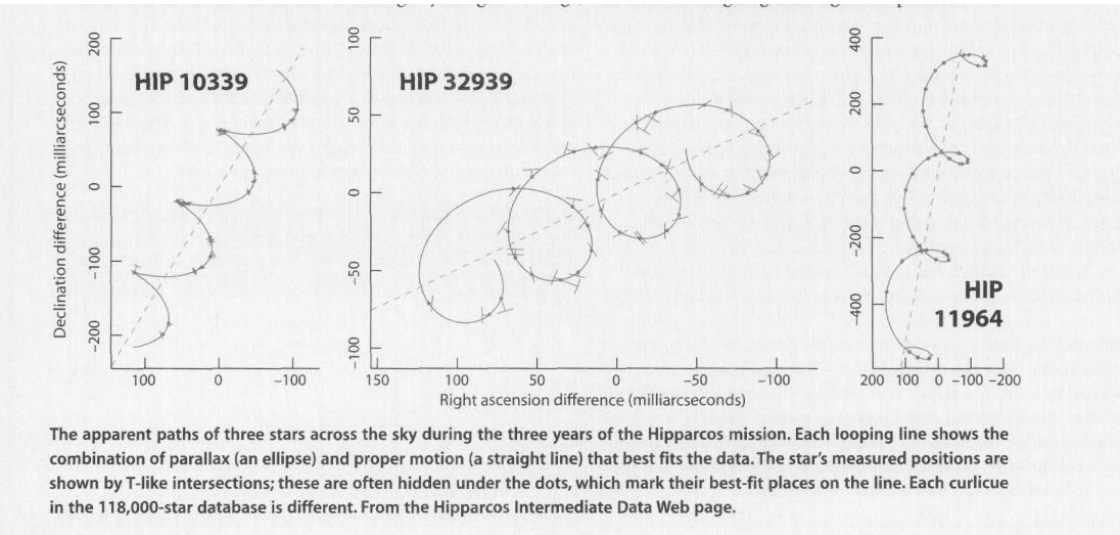

Figure 3: Hipparchos Intermediate Data (scraped from an old R.I.T. astronomy course)

Our goal with this project problem will be to extract an estimate for the parallax angle from simulated data sets which mimic this structure. Since the original data is came from a precisely controlled satellite, we will equivocate the parallax angle and the **major**-axis of the elliptical portion of the motion of the star in our data set.

Our data will be generated by a matlab script called

`parallaxfaker.m`

. If you look at the code in that script you will see exactly how the data is generated.

Our starting data sets will have the structure of time-series, and we will be assuming that the data is collected uniformly over the course of the experiment. To keep the data manageable we will assume that a star's position is measured exactly once a month, with the measurements uniformly spread across the year. Since Hipparchos was a three year mission, this means that a single data set will consist of 36 ordered pairs of x, y positions of a particular star.

We will begin with the assumption that the measurements are taken in a way which is coordinated with the observational position of each star's measurement. Thus we will start with the following discrete model for describing the evolving position of each simulated

star.

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} x_{i-1} \\ y_{i-1} \end{pmatrix} + \begin{pmatrix} v_x \\ v_y \end{pmatrix} + \begin{pmatrix} a\cos(\frac{\pi}{6}t_{i-1}) \\ b\sin(\frac{\pi}{6}t_{i-1}) \end{pmatrix}$$

We will use linear regression to find preliminary estimates for v_x, v_y, a, b.

1. Discuss and determine why $\frac{\pi}{6}$ was included as a frequency parameter in this positional model.

2. Manipulate the positional model equations (component-wise) to arrive at a linear equation (or equations) which is vulnerable to least squares regression.

3. Use Matlab to run parallax faker to generate a simulated data set (use $\sigma = 0$ to avoid any difficulties with phase at this point). Substitute the corresponding generated data into your model equations, run linear regression on your model (by explicitly finding the normal equations, or by relying on the \ operator) and verify that your model is producing reliable estimates when the data is perfect.

4. Now either edit the parallaxfaker.m script to add a small amount of random error to your simulated data (here we are just adding measurement error, but assuming the corresponding data is still collected at the correct times.) You can do this within the script itself, or after the fact by adding a *randn*() vector to the output. Once you include this error, the linear regression should not produce exact estimates for the unknown model parameters, so investigate how those errors behave when you vary the level of random noise/measurement error.

5. Finally consider the same problem but allow the phase to vary. (i.e. generate a data set where σ is not zero.) Discuss and determine a way to adjust the modeling process (or data) so that you can discover the phase. (This is **hard** since the phase does not naturally appear linearly, and linear regression cannot handle it as it appears. You might try playing with the sine and cosine functions, or you might try playing with the data itself to see if you can find a way to incorporate and discover a reliable sigma estimate.)

Tidal Prediction:

Overall tidal structure is a highly variable local phenomena which is fundamentally driven by a combination of the orbital interactions between Earth-Moon, and Earth-Sun orbits around the Rotating Earth, tempered and further complicated by the fluid-dynamics and local geography at each location on the coast. For us it provides a great window into a situation where the underlying idea (the level of 'high' and 'low' tides) is difficult to measure directly due to the constant interference of local variation (e.g. Waves.)

Good resources for the underlying tidal physics include: Physics of Continuous Matter, B Lautrup. CRC Press 2010; Solar System Dynamics, Murray and Dermott, Cambridge University Press. The original formulation is Laplace's theory of the tides.

The quasi-steady theory of tides (Basically the assumption that tides remain in equilibrium with the gravitational variations and pressure distributions as the Earth rotates around in space) predicts that the overall equilibrium tide structure is primarily composed of an average tidal height (which varies with position on the planet) influenced by both daily and semi-daily changes (which also vary with position). This simple theory is consistent with the following model for tidal prediction:

$$h(t) = \langle h_o \rangle + A_1 \cos(\Omega t + \phi) + A_2 \cos 2(\Omega t + \phi)$$

We can break up the waveforms to make all parameters appear linearly, however this permits the model an extra non-physical flexibility since the phases of the two wave forms are uncoupled in this formulation:

$$h(t) = \langle h_o \rangle + A_1 \cos(\Omega t) + B_1 \sin(\Omega t) + A_2 \cos 2(\Omega t) + B_2 \sin 2(\Omega t)$$

The value of Ω is chosen in concert with the choice of time unit in order to force the period of $\cos(\Omega t)$ to be equal to a day. (If the second model is intended to capture **only** the behavior of the first model, then an additional constraint needs to be added $2 \arctan\left(\frac{B_1}{A_1}\right) = \arctan\left(\frac{B_2}{A_2}\right)$)

If we permit our model the extra freedom, we can still use it as a reasonable model for short term tidal predictions.

(a) Find a time series of tidal measurements. Use the frequency of the tidal measurements to determine the corresponding Ω. Fit with linear regression.

(b) Find a time series of tidal measurements before and after a storm, evaluate the predictions under these circumstances.