A Weighted Least Squares Method for Time Delay Estimation in Gravitationally Lensed Photon Streams

Michał Staniaszek

Supervisor: Peter Tiňo

March 8, 2013

Abstract

In this report, we present a system for the estimation of the time delay Δt between multiple realisations of a Poisson process with the underlying function $\lambda(t)$, with particular application to gravitationally lensed photon streams. We build on the weighted least squares approach to develop a linear estimator which we use to estimate $\lambda(t)$. We then introduce a numerical and probabilistic method for estimating Δt using the function estimates. Finally, we compare the performance of our linear estimator to kernel regression on simulated data and data from real quasars.

Keywords: Poisson process, gravitational lensing, machine learning, linear estimation

Contents

1   Introduction

With continued advances in computing and sensing technologies, the amount of data that can be gathered from both everyday objects and scientific experiments has increased rapidly. However, more data is not always a blessing—it must be stored and analysed for it to have any use, and this is not an easy task when one has terabytes of data to deal with. The Large Hadron Collider at CERN is one perhaps extreme example, producing on the order of five terabytes of data each second. Storing this amount of data, let along analysing it is impossible, and so multiple stages of intelligent filtering are applied, reducing the throughput to 100 gigabytes per second, and then further to around 200 megabytes per second, where it is finally stored, producing almost two CDs each second [11]. This project focuses on creating the foundations for a system to do such intelligent filtering, but in the context of astronomical data. The volume of data produced by modern telescopes, while not on the same scale as the LHC, is nonetheless overwhelming. Image sizes of one to two gigabytes are not uncommon, and deciding what data is actually relevant is not a trivial task [15]. Using intelligent filtering algorithms, it should be possible to flag up interesting-looking data for further study. While there are many areas in which such capabilities would be useful, we are particularly interested in finding candidates for images of gravitationally lensed objects. In order to do this, it is necessary to find pairs of observations of photon flux which appear to have the same underlying function. More precisely, given a set of data containing the time of arrival of photons from a particular source, henceforth called a *stream*, we wish to find another stream which, when shifted in time by some value Δ, has similar numbers of photons arriving in a given interval as the first stream. We call Δ the *delay* between the two streams. In this project, we develop a system which can generate simulated photon streams using Poisson processes, use linear regression to estimate the underlying function of a given stream, and, given the function estimates of two streams, estimate the time delay between them. Knowing the value of the time delay has many applications in astrophysics, and with more precise estimates, more accurate calculations can be made to increase our understanding of the universe we live in.

In section 2 we discuss the concepts underpinning the project in more detail, with a more in-depth explanation of the issues surrounding the calculation of the time delay and its uses. In section 3 we introduce our method of generating photon streams from Poisson processes. Section 2.3 shows our approach to estimating the underlying function of a given stream of photons. Our methods of calculating the time delays between multiple photon streams are explained in section 5. Section 6 gives detailed information on the design and development of the system, including the software and project management aspects. Finally, in section 7 we present experimental data from both simulated and real data and discuss the relative effectiveness of our methods.

2 Background

2.1 Gravitational Lensing

In an eight-year period starting in 1907 and ending in 1915 with the publication of a paper on field equations of gravitation [6], Albert Einstein wrote many papers developing a new theory of gravitation, his general theory of relativity. This generalisation of special relativity and Newton's law of universal gravitation led to a revolution in the field of physics, and remains one of the most important scientific discoveries to date. The theory describes how spacetime is affected by the presence of matter and radiation, and this idea has many important consequences, but one of the effects in particular is important in the context of this report.

According to the theory, objects with mass, or massive objects, cause spacetime to curve around them. A simple way to visualise this effect is to imagine dropping a ball onto a sheet of cloth which has been pulled taut. The ball will eventually come to a stop in the centre of the cloth, and cause it to sag. Here, the sheet represents spacetime, and the ball represents anything from planets, to stars, or even entire galaxies. Depending on the weight of the ball, the shape of the cloth will be affected to different degrees—a ping pong ball will have hardly any effect at all, but if we drop a bowling ball onto the sheet, the effect will be significant. In a similar way, the amount that spacetime curves around a massive object depends on its mass. An object with high mass will cause a large amount of curvature, whereas a lower mass object will cause less. If a second ball, lighter than the first, is introduced to the system, what happens? With no initial velocity, it will roll in a straight line towards the first ball sitting at the centre of the sheet. This is one way of thinking about gravity and its relationship with spacetime—an object's gravitational attraction is a result of its mass curving spacetime, and the strength of the attraction is proportional to the mass. While objects with no mass, such as photons, cannot be affected by gravity directly, they *are* affected by the curvature of spacetime. This bending of light rays is known as *gravitational lensing*.

The first person to study the effects of gravitational lensing was Orest Chvolson, publishing a short note to *Astronomische Nachrichten* in 1924 [4]. However, the concept was largely unknown until a short calculation by Einstein was published in *Science* in 1936 [7]. Interestingly, Chvolson's note appears directly above a note from Einstein, but there appears to be no evidence that Einstein had ever seen it [1, 12]. The first gravitationally lensed object to be identified was the twin quasar SBS 0957+561, in 1979, and since then, over a hundred such objects have been discovered [16, 8]. The effect of gravitational lensing is, as the name suggests, similar to that of a lens, such as that of a camera. Unlike a camera lens, however, gravitational lenses do not have a focal point, but instead a focal line, resulting in images such as that shown in Figure 1a if the source (the object being lensed), the lensing object (the massive object around which the light is being bent) and the observer lie on a straight line. This effect is relatively rare, however, and in gen-
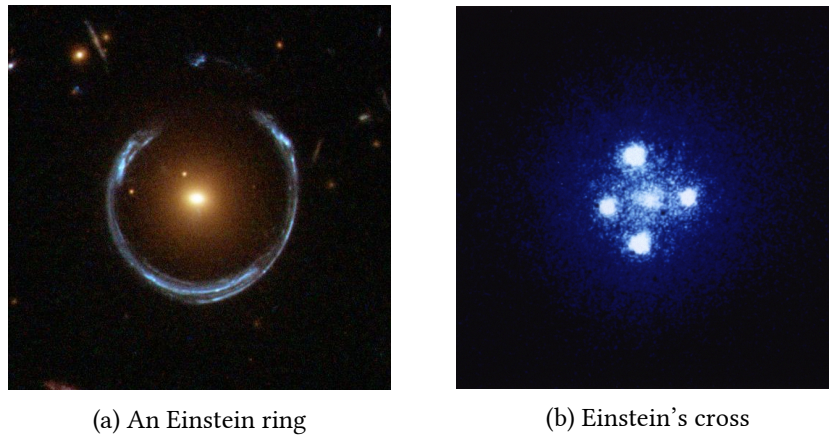
(a) An Einstein ring

(b) Einstein's cross

Figure 1: Two examples of strong lensing effects. a) shows light from a distant blue galaxy being distorted by the central galaxy LRG 3-757 [5]. b) shows four images of a distant quasar being lensed by a foreground galaxy [10].

eral rather than a ring, multiple images of the source can be observed. In these so called *strong* lensing effects, the distortion is very clearly visible. However, two other classes of lensing exist—*weak lensing* and *microlensing*. The effects of weak lensing cannot easily be observed visually, but statistical techniques can show the distortion produced. Microlensing works on even smaller scales than the other two classes, and can be used to detect planets and stars. It has also been proposed as a method to find objects such as black holes and brown dwarfs, which are otherwise difficult to detect [14].

2.1.1 Importance of the Time Delay

In gravitationally lensed systems, there is a delay between photon streams coming from images of the source due to the bending of light. Light from one source may have had to travel a slightly longer distance than that from the other, and while photons travel extremely fast, over astronomical distances the delay can become quite large.

- strong lensing p86

- Talk generally about the problem of time delay estimation

- refer to physics papers attempting to make estimates of the delay

- talk about time delay estimation in particular, refer to kundic et al, many others

- talk about how better estimates benefit the scientific community

- refer to peter's paper about the efficacy of kernel regression

3

- better estimators are necessary to increase the accuracy of estimates

- this is an experiment to see whether this method has any use

- build on technique introduced in massey et al

2.2 Poisson Processes

In certain situations, there are many benefits of having good models of the numbers of events that occur in a given period. For example, being able to estimate the number of incoming requests to a server, the number of calls made to emergency services, and the rate of radioactive decay at any given time are all useful in different applications. Poisson processes are *stochastic processes* that can be used to do just that. A stochastic process is a way of representing the evolution of a random value or system over time by using collections of random variables. Most such processes do not evolve in a *deterministic* way. That is, the way they change as time passes is not predictable.

A Poisson process is one such process which counts the number of events and the time at which they occur in a given time interval, and have been used to model all of the above examples [9, 3, 2]. In their basic form, Poisson processes have the following important properties [13]:

1. $N(0) = 0$.

   - $N(t)$ represents the total number of events that occurred up until time t. Thus, if $N(0) = 0$, it follows that the process begins at $t = 0$.

2. The numbers of events occurring in disjoint time intervals are independent.

   - The *independent increment* assumption. This states that $N(t)$, the number of events that occur up to time t is *independent* of the number $N(t + s) - N(t)$, i.e. the number of events in the time interval between t and s. In other words, the number of events that occur in one interval does not have an effect on the number of events in any other time interval.

3. The probability distribution of the number of events that occur in a given interval is dependent only on the length of the interval.

   - The *stationary increment* assumption. The implication of this is that the probability distribution of $N(t + s) - N(t)$ is the same for all values of t. That is, the likelihood of a number of events n occurring in the above time interval does not change, regardless of the value of t.

4. No counted occurrences are simultaneous.

- For all events that occur in the duration of the process, no two events will occur at the same time.

The most important thing about Poisson processes is the *rate parameter*, λ. This value represents the number of events that occur in each time interval. As we are counting events, it is clear that the rate parameter can never go below zero—there cannot be a negative number of occurrences in a given time interval. There are two types of Poisson processes, *homogeneous* and *non-homogeneous*. In a homogeneous Poisson process (HPP), the rate parameter is constant throughout the running of the process. This means that in every interval, the same number of events are likely to occur. In contrast, a non-homogeneous Poisson process (NHPP) has a rate parameter which varies. This means that the rate at which events occur varies during the running time of the process.

2.3 Function Estimation

2.3.1 Linear Regression

Linear regression is a statistical technique used to fit lines or curves to data points in order to find some sort of relationship between them. The number of variables in the data is important. One of the variables is called a *dependent* variable. We want to find the relationship between this variable and the other variables, called *independent* variables. What makes one variable dependent and another independent? Consider the expression $y = f(x)$. If $f(x)$ is some function of the variable x, then we know that the value of y depends on the value of x. This is where the names come from. In this simple example, x is the independent variable, and y is the dependent variable. There can be multiple independent variables.

Linear regression is used in many different fields to find the trend between variables. It is heavily used in economics to make predictions about what happens in many economical situations. Finding trends in data is useful to many people in different ways.

2.3.2 Kernel Density Estimation

This is another method which can be used to estimate functions, but which applies specifically to the probability density function of random variables. This technique uses *kernels* to estimate the function densities.

3 Simulation of Photon Streams

The first step in building the system was the development of a photon stream simulator. The ability to simulate photon streams means that the system can be tested on many different stream types, so that we are able to determine where its strengths and weaknesses lie. While many simulation tools are very complex, our

system does not require complex simulation of quasars or the movement of photons, as we are only interested in their arrival time. A quasar can be represented by some random variable X, which changes over time. This means that a NHPP can be used to create similar effects. In order to simulate arrival times, a function that describes how the rate parameter λ changes over time is needed.

3.1 Function Generation

3.2 Generating Streams from Functions

4 Function Estimation

4.1 Baseline Estimation

4.1.1 Optimum Least Squares

4.1.2 Iterative Weighted Least Squares

4.1.3 Piecewise Iterative Weighted Least Squares

Initially, we thought that it may be possible to decide whether to extend the line or not based on the difference in slope between the estimate from the previous time interval and the estimate of the next. If the previous estimate was positive, and the next negative, and vice versa, clearly the line should not be continued. The intercept parameter was considered to be much less important. However, this assumption was highly flawed. Due to the nature of poisson processes, it is perfectly possible that although the function has changed significantly after the end of the previous interval, the estimate for the interval that we are trying to extend the line into could return very similar values to that of the previous interval. Because of this, we extend the line when we should not be doing so. There are several potential solutions to this problem. First, rather than forming a new estimate, we extend the line and then check how much the error has increased. If it goes over a certain threshold, then we reject the extension attempt and try again, this time with a shorter extension. Another potential way of improving the piecewise estimation in general would be to require the estimate for the next time period to start from the end point of the last time period. This would prevent the intercept parameter from changing, and would force the estimator to find the best estimate given a specific starting point, rather than giving it free reign to find the estimate which actually minimises the error.

- coding issues
  what to do with the issue of minimum length of intervals? Sometimes not extending the original gives a better estimate of the line than re-estimating the interval extended, or adding the short interval onto the end of the previous one and using its estimate. See data in the $\min_{\text{intervallength}}$ folder in data. The better fitting line is the baseline estimate of that with no minimum, and the other set is the estimate with minimum interval length applied

5   Time Delay Calculation

5.1   Area Method

5.2   Probability Mass Function Method

6   System

6.1   System Structure

6.1.1   Overall Structure

6.1.2   Estimators

6.1.3   Generators

6.1.4   Experimenter

6.2   Development

6.2.1   Development Process

- first draft up code in notebook to get down the concept

- write a basic code skeleton and add tests to make sure that it works as intended - particularly for mathematics and the like

- flesh out the code and integrate it with the system

- make code as modular as possible to make it easy to add stuff in later

6.2.2   Version Control

- branching strategy

- commit frequency

- using issues on github

6.2.3   Project Management

- keep changelog

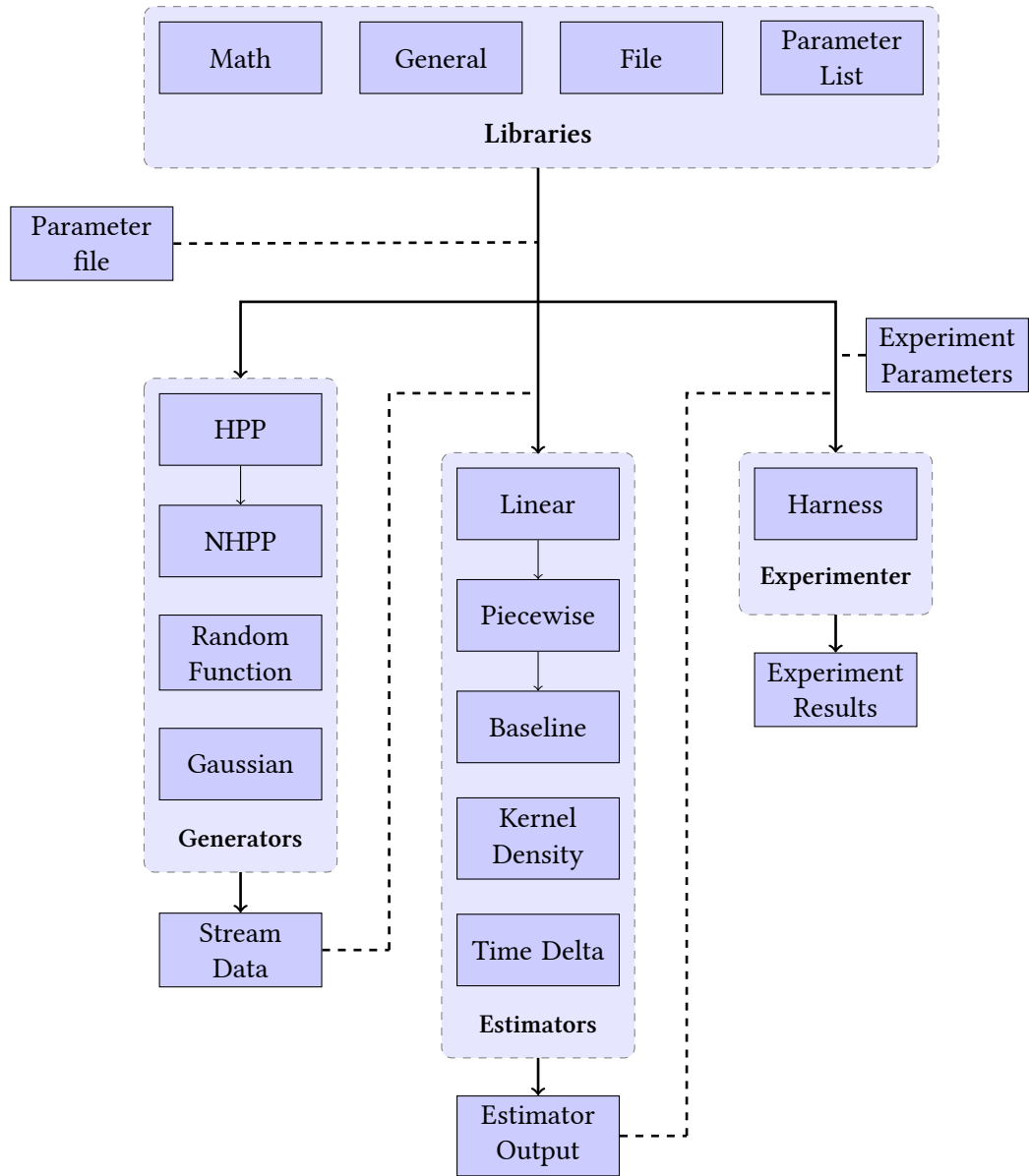- writing up and planning layout in notebook

Figure 2: System structure

7 Evaluation

7.1 experimentation on simulated data

7.2 experimentation on real-world data

8 Conclusion

8.1 Future Work and Improvements

References

[1] SAO/NASA Astrophysics Data System (ADS). *Über eine mögliche Form fiktiver Doppelsterne*. URL: http://articles.adsabs.harvard.edu/full/1924AN....221..329C.

[2] Martin F Arlitt and Carey L Williamson. "Internet web servers: Workload characterization and performance implications". In: *IEEE/ACM Transactions on Networking (ToN)* 5.5 (1997), pp. 631–645.

[3] F Cannizzaro et al. "Results of the measurements carried out in order to verify the validity of the poisson-exponential distribution in radioactive decay events". In: *The International Journal of Applied Radiation and Isotopes* 29.11 (1978), 649–IN1.

[4] Orest Chwolson. "Über eine mögliche Form fiktiver Doppelsterne". In: *Astronomische Nachrichten* 221 (June 1924), p. 329.

[5] NASA Astronomy Picture of the Day. *A Horseshoe Einstein Ring from Hubble*. URL: http://apod.nasa.gov/apod/ap111221.html.

[6] Albert Einstein. "Die Feldgleichungen der Gravitation". In: *Preußische Akademie der Wissenschaften, Sitzungsberichte*. 1915.

[7] Albert Einstein. "Lens-Like Action of a Star by the Deviation of Light in the Gravitational Field". In: *Science* 84 (Dec. 1936), pp. 506–507.

[8] CFA-Arizona Space Telescope Lens Survey of Gravitational Lenses. *CASTLeS Survey*. URL: http://www.cfa.harvard.edu/castles/.

[9] Amir Hajjam, J-C Creput, and Abder Koukam. "Approach for optimized and dynamic medical emergency management". In: *Service Systems and Service Management, 2008 International Conference on*. IEEE. 2008, pp. 1–6.

[10] HubbleSite. *First ESA Faint Object Camera Science Images the Gravitational Lens G2237 + 0305*. URL: http://hubblesite.org/newscenter/archive/releases/1990/20/image/a/.

[11] Worldwide LHC Computing Grid Project. *What is WLCG: Data Processing*. 2011. URL: http://lcg-archive.web.cern.ch/lcg-archive/public/data-processing.htm.

[12] Jürgen Renn and Tilman A Sauer. *Eclipses of the stars*. Max-Planck-Inst. für Wissenschaftsgeschichte, 2000.

[13] Sheldon M Ross. *Simulation*. Academic Press, 1997.

[14] Peter Schneider, Christopher Kochanek, and Joachim Wambsganss. *Gravitational Lensing: Strong, Weak and Micro: Saas-Fee Advanced Course 33*. Vol. 33. Springer, 2006.

[15] Jean-Luc Starck and Fionn Murtagh. *Handbook of Astronomical Data Analysis*. 2002.

[16]   D Walsh, RF Carswell, and RJ Weymann. "0957+ 561 A, B- Twin quasistellar objects or gravitational lens?" In: *Nature* 279.5712 (1979), pp. 381–384.

Appendices

Appendix A Installation

A.1 MuParser

download package
run `./configure --prefix=/usr`, followed by `make && make install` (may require sudo) this installs muparser so that headers can be found in `/usr/include`

```
 sudo apt-get install libgsl0-dev check
```

Appendix B Usage

B.1 Creating functions for experimentation

Generate 10 random functions using gaussians, and output the transforms as well so they can be plotted

```
    ./launcher -g ../data/params.txt -r -t 2 -c 10
```

Generate two streams from each of these generated functions using the gaussian generator

```
    ./launcher -g ../data/params.txt -f rand -n 2 -c 10
```

Generate stuttered streams from files in this directory, so that you can perform model fitting to find the best parameters to use on the generated set.

```
  ./launcher -x ../data/exp_params.txt -p ../data/params.txt -c 10 -n 2 -s -i .
```