Estimating Time Delays between Irregularly Sampled Time Series

by

Juan Carlos Cuevas Tello

A thesis submitted to

The University of Birmingham
for the degree of Doctor of Philosophy

School of Computer Science
The University of Birmingham
Birmingham B15 2TT
United Kingdom

April 2007

Abstract

The *time delay* estimation between time series is a real-world problem in gravitational lensing, an area of astrophysics. Lensing is the most direct method of measuring the distribution of matter, which is often dark, and the accurate measurement of time delays set the scale to measure distances over cosmological scales. For our purposes, this means that we have to estimate a time delay between two or more noisy and irregularly sampled time series. Estimations have been made using statistical methods in the astrophysics literature, such as interpolation, dispersion analysis, discrete correlation function, Gaussian processes and Bayesian method, among others. Instead, this thesis proposes a kernel-based approach to estimating the time delay, which is inspired by kernel methods in the context of statistical and machine learning. Moreover, our methodology is evolved to perform model selection, regularisation and time delay estimation globally and simultaneously. Experimental results show that this approach is one of the most accurate methods for gaps (missing data) and distinct noise levels. Results on artificial and real data are shown.

In the memory of my parents
Catalina Tello Briones and
Juan Cuevas Leija, who
started the PhD with me,
but unfortunately they
could not arrive to this end
– *requiescant in pace.*

Acknowledgements

First I would like to thank my supervisor Peter Tiňo for introducing me to this challenging area of machine learning and astrophysics, and for his valuable advice and support during my studies. I also want to thank Somak Raychaudhury for sharing his knowledge in astrophysics. It has been a pleasure to work with such a team. I thank Markus Harva for providing his data and all the results of his Bayesian method. The LaTeX style was provided by my schoolmate Gregorio, whom I also thank. My gratitude goes also to my thesis group, schoolmates and friends in Birmingham because their discussions have all contributed to this thesis in some way or another; I do not list them because they are too many. I thank the staff of the English for International Students Unit (EISU) for their one-to-one consultations. This work was possible thanks to my sponsors the PROMEP (PROgrama de MEjoramiento del Profesorado de Educación Superior) and the UASLP (Universidad Autónoma de San Luis Potosí). Finally and most important, I am deeply grateful to my family – Araceli, Karla and all my relatives – for sharing, motivating and inspiring me in good and bad moments.

Contents

1 Introduction **1**

1.1 Background 1

1.2 Motivation 3

1.3 Contribution 4

1.4 Thesis Organisation 4

1.5 Publications from this Thesis 4

2 Problem Statement and Astronomical Data **6**

2.1 Gravitational Lensing 6

2.2 Cosmological Significance of Time Delays 8

2.3 Gravitational Lens: Q0957+561 10

2.3.1 Radio Data 12

2.3.2 Optical Data 12

2.4 Artificial Data 14

2.4.1 DS-500 14

2.4.2 DS-5 15

2.5 Chapter Summary 18

3 Literature Review and Other Methodologies **19**

3.1 Time Delay Estimates: Q0957+561 19

3.2 Methods 21

3.2.1 Interpolation 21

3.2.2 Cross Correlation 22

3.2.3 PRH Method 23

3.2.4 Dispersion Spectra 25
3.2.5 A Bayesian Estimation Method 27
3.3 Chapter Summary 28

4 Machine Learning and Kernel-based Method 30
4.1 Machine Learning 30
4.1.1 Supervised Learning 31
4.1.2 Kernels, Kernel Methods and Kernel Machines 32
4.1.3 Other Disciplines Related to Kernels 34
4.1.4 Regression and Motivation 35
4.2 Kernel-based Method for Time Delay Estimation 36
4.2.1 Weights $\{\alpha_j\}$ 38
4.2.2 Kernel Parameters 39
4.3 Time Complexity 42
4.4 Chapter Summary 43

5 Evolved Kernel-Based Method 44
5.1 Introduction 44
5.2 Evolution Strategies 45
5.3 Evolutionary Algorithm 45
5.3.1 Regularisation 45
5.3.2 Representation 47
5.3.3 Fitness Function 48
5.3.4 Fitness Landscape 49
5.3.5 Evolution Operators 50
5.4 Literature Review 51
5.5 Time Complexity 52
5.6 Chapter Summary 52

6 Experimental Results 54
6.1 Artificial Data 54
6.1.1 DS-500 54

6.1.2 DS-5 . . . 63
6.1.3 PRH Data . . . 68
6.1.4 Harva Data . . . 78
6.2 Real Data: Q0957+561 . . . 79
6.2.1 Radio Data . . . 79
6.2.2 Optical Data . . . 85
6.2.3 Q0957+561 Summary . . . 92
6.3 Chapter Summary . . . 94

7 Conclusions and Future Work **95**
7.1 Conclusions . . . 95
7.2 Future Work . . . 101
7.2.1 Speedup . . . 101
7.2.2 Theoretical Analysis . . . 103
7.2.3 Superimposed Light Curves . . . 103
7.2.4 Multiple Time Delays . . . 105
7.2.5 Other Applications . . . 106
7.3 Chapter Summary . . . 106

A Statistical Analysis **109**

B Notation **112**

Bibliography **115**

Index **125**

List of Algorithms

4.1 Cross-Validation 41
4.2 Time Delay Estimation 43
5.1 Fitness Function: CV 48
5.2 Evolutionary Algorithm 53

List of Figures

2.1 Gravitational Lensing 7
2.2 Gravitational Lenses 9
2.3 Real Data: Q0957+561 11
2.4 Artificial Data: DS-500 16
2.5 Artificial Data: DS-5 17

3.1 PRH Data 26
3.2 Harva Data 28

4.1 Set of Gaussian Kernels 33

5.1 Patterns on DS-5 and DS1 46
5.2 Fitness Landscape 49

6.1 Results on DS-500. Part I 57
6.2 Results on DS-500. Part II 58
6.3 95% CI on DS-500-1-N-0 60
6.4 t-test on DS-500 62
6.5 MSE on DS-500 63
6.6 AE on DS-500 64
6.7 $\hat{\mu}$ on DS-500 65
6.8 $\hat{\sigma}$ on DS-500 66
6.9 Results on DS-5 68
6.10 t-test on DS-5 71
6.11 MSE on DS-5 72

6.12 MSE on DS-5 73
6.13 $\hat{\mu}$ on DS-5 74
6.14 $\hat{\sigma}$ on DS-5 75
6.15 Reconstructions on Radio Data 80
6.16 Results on 4 cm from EA-CV (100 Generations) 85
6.17 Results from Dispersion Spectra $D^2_{4,2}$ on Optical Data 86
6.18 Reconstructions on Optical Data 88
6.19 Patterns on Optical Data 89

7.1 Q Curve on Artificial Data Set DS-500-5-G-0-N-0 106
7.2 Reconstructions of Superimposed Light Curves and Their Components for DS-500-5-G-0-N-0. 108

List of Tables

2.1 Radio Data: 6 cm 13
2.2 Optical Data: g-band 14
2.3 Artificial Data: DS-500 15

3.1 Time Delay Estimates: Q0957+561 20

6.1 DS-500: Statistical Analysis 59
6.2 95% CI on DS-500 61
6.3 t-test on DS-500 61
6.4 DS-5: Statistical Analysis 67
6.5 95% CI on DS-5 69
6.6 t-test on DS-5 69
6.7 DS-5: Results by Noise Level 70
6.8 Results on PRH Data from PRH SF* 76
6.9 Results on PRH Data from PRH SF+ 76
6.10 Results on PRH Data from EA-M-CV 77
6.11 PRH Data Results: Bias and Variance 77
6.12 Harva Data Results 78
6.13 Results on Radio Data: K-F and K-V 81
6.14 Results from EA-M-CV on 4 cm 82
6.15 Results from EA-M-CV on 6 cm 82
6.16 Results from EA-M-CV on 6*cm 83
6.17 Results from EA-R-CV on 4 cm 83
6.18 Results from EA-R-CV on 6 cm 84

6.19 Results from EA-R-CV on 6*cm 84
6.20 Results on Observed Radio Data 85
6.21 Results on Observed Optical Data 87
6.22 Results from MC Simulations on Optical Data 87
6.23 Results from EA-R-CV on DS1 90
6.24 Results from EA-M-CV on DS1 91
6.25 Results from EA-R-CV on DS2 91
6.26 Results from EA-M-CV on DS2 92
6.27 Results from EA-R-CV on DS3 93
6.28 Results from EA-M-CV on DS3 93
6.29 Results from EAs on Observed Data 93
6.30 Q0957+561 Summary of Results 94

Chapter 1

Introduction

1.1 Background

THE Einstein's General Theory of Relativity showed that the presence of matter locally distorts the fabric of space-time. Since light travels along geodesics, rays of light (radio waves or X-rays) are bent as they pass in the vicinity of massive objects in space. Geodesics are the shortest path between two points, i.e., the generalisation of straight lines in curved space. Depending on the angle of deflection, images of distant objects can be dramatically distorted, and even broken up into multiple images, due to massive objects like galaxies or galaxy clusters along the line of sight (see §2). This is known as *gravitational lensing* [79, 56].

Gravitational lensing can produce various spectacular effects in astronomical observations. On the one hand, microlensing is caused by star-sized massive compact halo objects (MACHOs) between a source and the observer, which increase the magnification of the observed images [13, 60]. On the other hand, galaxy clusters, which have masses of 10^{15} times the mass of stars, can cause distant galaxy images to be distorted in arcs and arclets.

A variation of gravitational lensing is known as strong lensing, where an astronomer observes two or more images of the same distant celestial object (quasar), when light travelling from it passes close to the centre of a massive galaxy on the way. Quasars are highly variable sources, and the variation can be represented as a random

process. The level of variation depends on the frequency of observation, so that the variability in optical images, for instance, is different from that in X-ray images. The light paths from the common source to the multiple images are in general different for each image, so a variation in the source will be observed at different times in the different images. An accurate measurement of the time delay is one of very few direct ways of measuring distances in the universe, which is essential in measuring such parameters of the universe as its expansion rate, mass density and the Hubble constant [74, 75]. With these parameters, it is possible to discover the age and future of the universe [79, 56]. Therefore, the time delay is the most direct method of measuring matter in the universe [74, 75, 43].

In practice, a gravitationally lensed quasar is monitored such that its brightness at some wavelength (optical, radio or X-ray) is measured as a function of time. These light curves are represented as noisy time series. The problem here is to find the *time delay* between any given pair of these time series. Depending on the type of telescope, however, the recorded observations have different levels of errors due to the observational process. Related to the process of observation, the time series are also irregularly sampled and can have large gaps.

The earliest discovered gravitational lens, Q0957+561, discovered in 1979 [92], is also the most studied so far. This system has two images of the same quasar, referred to as images A and B (see Fig. 2.3). Since the discovery, many attempts to estimate the time delay between the light curves obtained from images A and B, in both radio and optical images, have been made, e.g., see [71, 64, 65, 58, 69, 83, 49, 32, 59]. Controversy raged from the first report in 1981 until 1997, when a time delay of 417 ± 3 days was published [49], which apparently stopped the controversy. The history of this controversy is well reviewed in Haarsma *et al.* [31], which tabulates different time delays (in the range of 300 to 1000 days) claimed over different data sets using distinct methods. One of many recent publications is Kochanek *et al.* [43], which judges [49] to be correct. However Ovaldsen et. al. [59], with new and more accurate photometry, report a time delay of 424.9 ± 1.2 days, which is one of the latest and best-measured optical data set for the source 0957+561. Previous and recent publications such as Oscoz *et al.* [57], Burud *et al.* [9], Colley *et al.* [13],

among others, report values of the time delay as 422.6 ± 0.6, 423 ± 9 and 417 ± 0.07 days respectively. Therefore, the time delay for Q0957+561 is still not agreed upon by the various research groups studying it, even if a lot of them use the same set of observations. This is largely due to the limitations of the different methods used by these groups.

Next we address the question: Why is the measurement of time delay complicated? Basically there are two reasons: (a) the data are irregularly sampled with big gaps (or missing data), and (b) the data have appreciable levels of noise. If we have to do better than existing methods, we have to directly address these two aspects of the data.

1.2 Motivation

Having introduced briefly the time delay problem, we cite Ovaldsen *et al.* [59] (page 904), who state:

> "A longer observational base line and maybe more statistical techniques could shed new light on [the] time delay issue."

Haarsma *et al.* [32] (page 69) who made the last analysis using radio data also state:

> "The 0957 radio light curves will continue to be a useful data set for studying systematic effects and time-delay analysis techniques."

Talking about the future of time delay measurements, Kochanek *et al.* [44] bring attention to the problem of estimating the Hubble constant and the structure of the mass distribution of lens galaxies. Then, they state

> " The simplest way to clarify this problem is to measure accurate time delays for many more systems" (quasars).

Moreover, Pindor [70], studying time delays as a gravitational lens searching technique, states

> "... the best possible method for identifying time delays [remains] a matter open to investigation."

The above citations, and the failure to reach agreement on the value of the time delay between the two images of Q0957+561, have been the main motivation of this research.

1.3 Contribution

Here we propose a kernel-based approach to estimating time delays in the context of kernel methods [86] and statistical and machine learning from the standpoint of computer science [54, 35]. This approach can be defined as a data-driven approach. Through artificial data sets, we answer such questions as: What is the effect of noise in the time delay estimation? What is the influence of gaps? What is the effect of features? The answers to these questions are useful before launching an observational campaign to study a specific gravitational lens in order to estimate a time delay accurately. Moreover, this thesis compares the proposed method with other methods such as linear interpolation, correlation-based methods, Dispersion spectra, PRH method (Gaussian processes) and Bayesian method.

1.4 Thesis Organisation

The remainder of this thesis is organised as follows: Chapter 2 gives a broad description of the time delay problem and our real and artificial astronomical data. Chapter 3 contains the literature review and other methods, including the synthetic data associated with them. Chapter 4 gives a brief introduction to kernels and introduces the kernel-based approach. Chapter 5 presents the evolved version of our approach. Chapter 6 has the experimental results on artificial and real data. Some conclusions are drawn in Chapter 7.

1.5 Publications from this Thesis

Some results of this thesis have been published. The list in chronological order is below:

- Cuevas-Tello, J.C., Tino, P. and Raychaudhury, S. (2005) *Determining time delays in gravitational lensing: How significant are the results.* RAS National Astronomy Meeting 2005 (poster), Birmingham, UK.

- Cuevas-Tello, J.C., Tino, P. and Raychaudhury, S. (2005) *Time delay estimation in gravitational lensing: a new approach.* RAS National Astronomy Meeting 2005 (talk), Birmingham, UK.

- Cuevas-Tello, J.C., Tino, P. and Raychaudhury, S., (2005) *Kernel-based method applied on irregularly sampled time series*, The Analysis of Patterns Workshop (poster), Erice, Italy (Ettore Majorana Centre for Scientific Culture). Directors: N. Cristianini, R. Cerulli and J. Shawe-Taylor.

- Cuevas-Tello, J.C., Tino, P. and Raychaudhury, S. (2006) *How accurate are the time delay estimates in gravitational lensing?.* Astronomy & Astrophysics, 454:695-706.

- Cuevas-Tello, J.C., Tino, P. and Raychaudhury, S. (2006) *A kernel-based approach to estimating phase shifts between irregularly sampled time series: an application to gravitational lenses.* Machine Learning: ECML 2006, Lecture Notes in Artificial Intelligence (LNAI 4212), Springer-Verlag, pp. 614-621.

- Cuevas-Tello, J.C., Tino, P., Raychaudhury, S., Yao, X. and Harva, M. (2007) *Evolved kernel approach for noisy and irregularly sampled time series: an application to time-delays in astronomy* . IEEE Transactions on Evolutionary Computation, submitted.

Chapter 2

Problem Statement and Astronomical Data

THIS chapter, besides introducing gravitational lensing and the time delay problem, also defines some concepts in astrophysics. We introduce gravitational lensing in order to discover where the data come from, especially the delay Δ, because in practice one works directly with the time series and previous knowledge of gravitational lensing is not needed; hence, §2.1 may be skipped at discretion. We also describe specific aspects of observational radio and optical data in §2.3.

2.1 Gravitational Lensing

Figure 2.1 illustrates gravitational lensing in detail. On the left-hand side is the source plane where the quasar (or bright source) is located (shaded circle); the mass deflecting the light from the source is in the middle, lens plane; the observer plane, on the right, is where the telescope is located. Here one assumes that the deflection of light happens in one plane, which is the lens plane. The astronomer observes two images from the source, denoted by empty circles in the source plane, giving rise to an interesting cosmic illusion. This is due to the fact that photons of light (or any other form of electromagnetic radiation, e.g., radio or x-rays), are affected by a

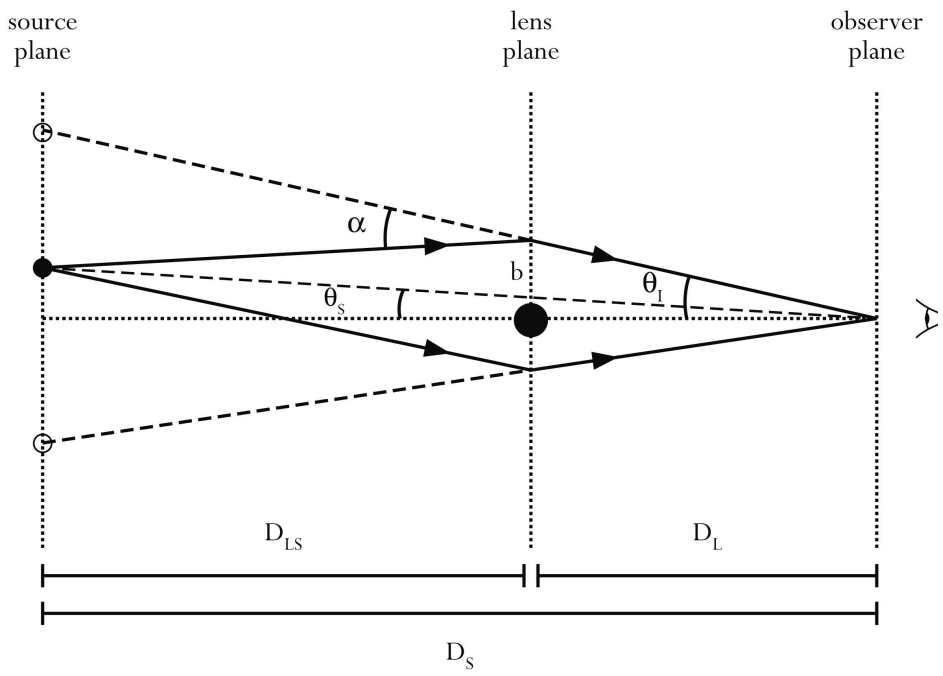


Figure 2.1: Gravitational Lensing. See §2.1 for details. This figure was obtained from http://en.wikipedia.org/

gravitational field, described by the general relativity [79, 56].

The effect of gravity on photons can be described by

$$\alpha = \frac{4G\mathcal{M}}{c^2 b} \tag{2.1}$$

whereas α is the angular amount of deflection (see Fig. 2.1), G is the gravitational constant [56], $\mathcal{M}$ is the mass in the lens plane, c is the speed of light, and b is the closest distance from the source to the mass $\mathcal{M}$. This is why gravitational lensing is the most direct method of measuring matter, which is often dark. The mass $\mathcal{M}$ may be clumps of matter such as stars or galaxies.

Also from general relativity and Fig. 2.1, a photon coming from the source plane to the observer plane passing by a point mass $\mathcal{M}$ from a direction θ_I relative to the lens, will be *delayed* by

$$-\frac{4G\mathcal{M}}{c^3} \ln \theta_I \tag{2.2}$$

due to the gravitational influence of the lens. Therefore, the photons making up the two distorted images obtained at the observer plane arrive at distinct times, differing by Δ, because of the geometry of the lens and the distribution of gravitational potential within the lens [79].

The source plane in Fig. 2.1 shows two apparent images from a single source, but this is only to illustrate this phenomenon. One can in fact observe more than two images at the observer plane. For instance, when a point mass is interposed exactly along the line of sight to the source, one can observe a spectacular ring image, known as the Einstein ring [79, 56, 20]. In practice, such rings are rarely observed. In Fig. 2.2 are some gravitational lenses with either two or four images[1] within the Einstein ring.

2.2 Cosmological Significance of Time Delays

Over a hundred systems of lensed quasars are currently known[2], and about 10 of these have been monitored for long periods. In some of these cases, the measurement of a time delay Δ has been claimed.

To measure a time delay Δ, a monitoring campaign must produce light curves from a individual lensed quasar [43]. These light curves are well sampled compared to the time delays. Then, the source must have measurable features (brightness fluctuations) on time scales shorter than the monitoring period. Finally, a time delay estimation method is used to measure the delay.

Since the time delay between light curves depends on the mass of the lens, it is the most direct method to measure the distribution of matter in the Universe [75, 43]. Therefore, through time delays, observations of gravitational lenses yield estimates of the masses of galaxies and of clusters of galaxies [29]. However, the main motivation to study time delays is the estimation of the cosmological constant, the Hubble constant, from gravitational lens [79, 29, 43]. The Hubble constant is useful

[1]For B1938+666 and B1608+656, the apparent sources are clearly visible in [42] after removing the lens galaxies $\mathcal{M}$.

[2]A growing list of multiply-imaged gravitationally lensed quasars can be found at http://www.cfa.harvard.edu/castles

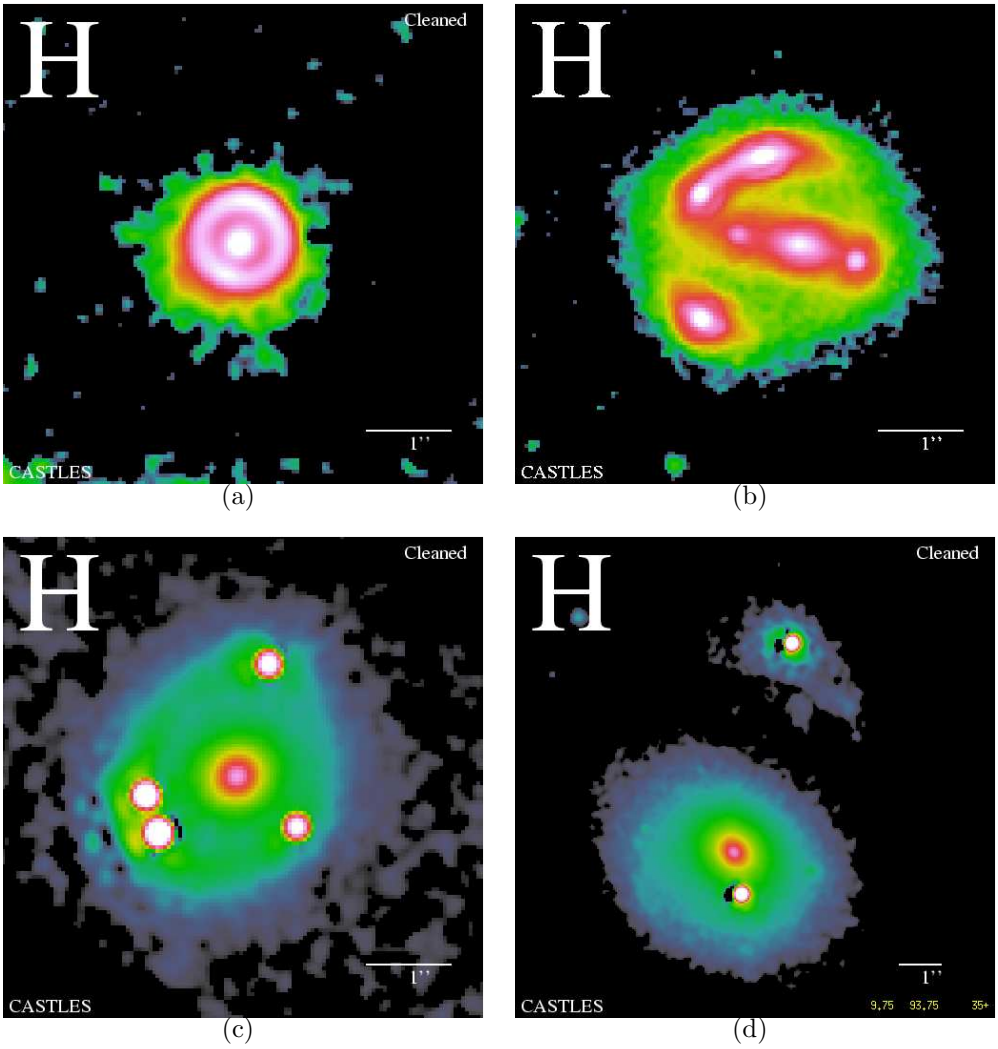


Figure 2.2: Gravitational Lenses. Images observed at H-band by the Hubble Space telescope (http://www.cfa.harvard.edu/castles). At each false[a] colour image **(a)-(d)**, a point image (white) corresponds to the observed image from the source, and the extended objects are $\mathcal{M}$; i.e., colours apart from black and white. **(a)** Two-image B1938+666 lens. **(b)** Four-image B1608+656 lens. **(c)** Four-image PG1115+080 lens. **(d)** Two-image Q0957+561 lens.

[a] i.e., colours are modified to obtain an enhanced image.

in measuring the parameters of the universe such as the expansion rate, mass density, age and future.

Moreover, recent investigations concentrate on discovering gravitational lenses

through measurements of their time delays [70]. This is also one of the objectives of the ongoing Sloan Digital Sky Survey[3] (SDSS) and the Large Synoptic Survey Telescope[4] (LSST).

2.3 Gravitational Lens: Q0957+561

Quasar Q0957+561, an ultra-bright galaxy with a super-massive central black hole, in Fig. 2.2d, was the first lensed source to be discovered and has so far been the most studied. Therefore, we concentrate our study only on this quasar. The source is 3.3×10^{10} light-years away from us, being lensed by a galaxy (visible in Fig. 2.2d), along the line of sight, only 0.62×10^{10} light-years away. The brightness of quasars varies on the time-scale of days. The lens is a galaxy with a mass $\mathcal{M} \sim 10^{42}$ kg, which is about 10^{12} times the mass of the sun.

The observations have been made by both radio and optical astronomers, since theory predicts that they should measure the same time delay between the light curves obtained from the two images. This is because gravitational lensing is a purely gravitational effect, which cannot affect the frequency of an electromagnetic wave, and thus the time delay is independent of the frequency of observation.

For our purposes, the data are available as two unevenly sampled time series of fluxes (or logarithm thereof) of the two images; see Fig. 2.3 where the three plots, data sets, on the left are radio data and those on the right are optical data sets; for more details on these data see §2.3.1 and §2.3.2 respectively. The data sets depicted in Fig. 2.3 are our *real data*. The observations are made at irregular intervals due to weather conditions, equipment availability, object visibility, among other practical considerations [21].

Some time delay estimates for this quasar are given in the next chapter, where the time delay Δ is around 400 days.

[3] http://www.sdss.org/

[4] http://www.lsst.org/

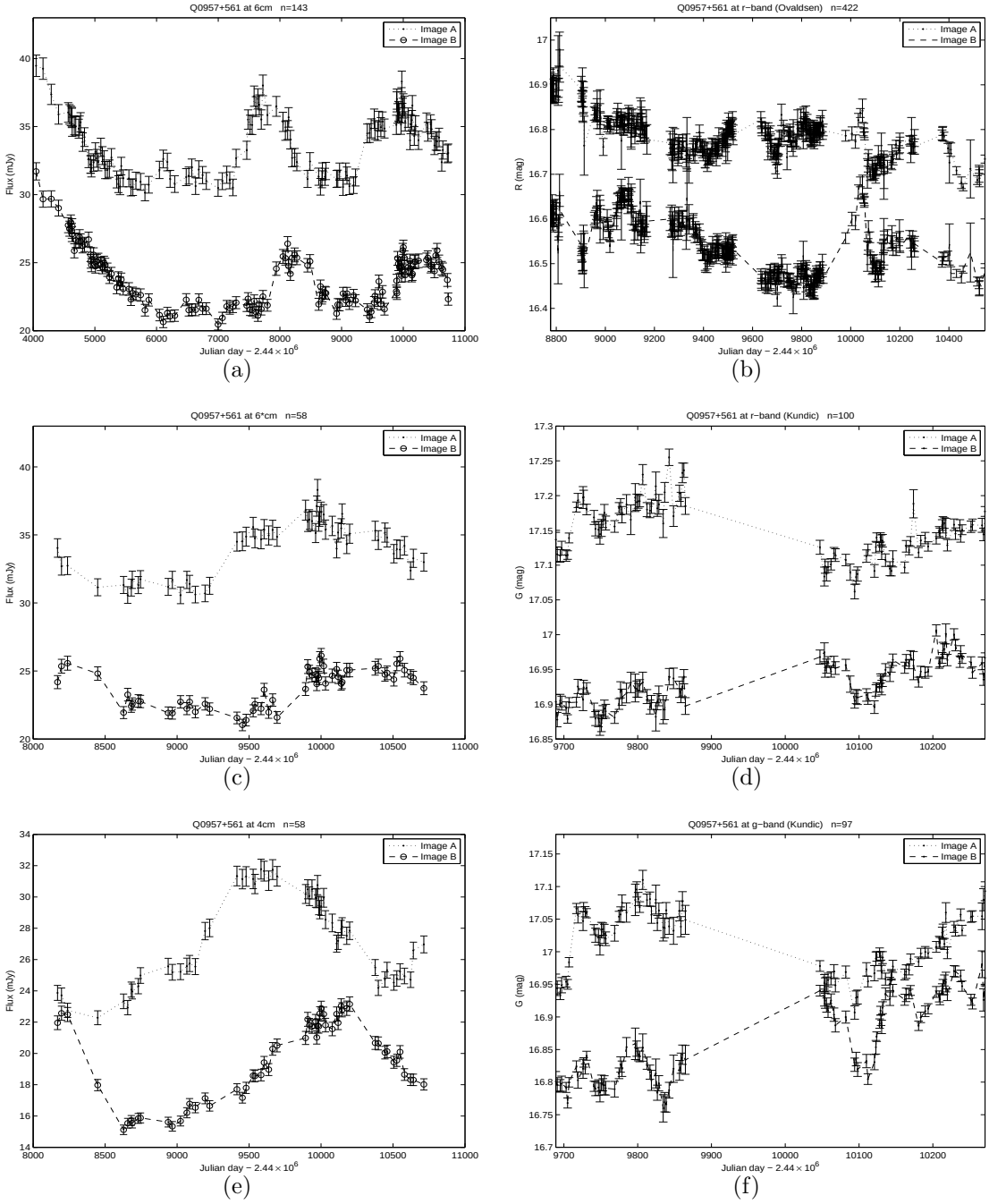


Figure 2.3: Real Data: Q0957+561. Error bars represent the observational error. **(a)** 6 cm. **(b)** DS3; image A has been shifted upwards by 0.2 mag for visualisation. **(c)** 6*cm. **(d)** DS2. **(e)** 4 cm. **(f)** DS1. For more details see §2.3.

2.3.1 Radio Data

The radio data to be used in this thesis are reported as time series, and the error involved is assumed to be 2% of the density flux [32]. A data set at 6 cm wavelength is shown in Table 2.1, which contains in its first column the observation number, in the second column the calendar date and in the third the date represented in Julian days. The Julian day (JD) is the decimal number of days that have elapsed since Noon, Greenwich Mean Time (GMT), of Monday, 1st of January 4713 BC. This is used as a way of representing the date as a continuous real variable. The last two columns have the density flux of images A and B. In radio data the flux is reported in linear scale, milliJanskys (mJy), where 1 mJy=10^{-29} W m $^{-2}$ Hz^{-1}. In practice, we need only the last three columns. We use mainly two data sets at two different wavelengths: 4 cm and 6 cm [32]. For the 6 cm data set[5], we use the light curve with four points from Spring 1990 removed, as in [32]. The 4 cm data set [32] has $n = 58$ observations from 4 October 1990 to 22 September 1997. These data sets were gathered from the National Radio Astronomy Observatory[6], the Very Large Array radio telescope (VLA).

Since we have generated a new data set, 6*cm contains 6 cm observations only at the observation times of the 4 cm data set. In other words, we keep a 6 cm observation when there is a 4 cm observation at the same time t . Therefore, both 4 cm and 6*cm data sets contain 58 observations and gaps of the same size.

The three data sets are depicted on the left-hand side of Fig. 2.3.

2.3.2 Optical Data

The optical data are also reported as time series, but in a different way. The Table 2.2 gives an example. This table contains the standard deviation of measurement errors at each observation for each density flux (A and B). This characteristic makes these

[5]Data are available at http://space.mit.edu/RADIO/papers.html. Note that a record of the 6 cm data set is not included in the published papers and the observation on 11th April 1994 is recorded a day earlier in previous studies.

[6]Its total cost was US$78,578,000 (in 1972), roughly US$1 per taxpayer at the time – http://www.vla.nrao.edu/.

Table 2.1: Radio Data: Q0957+561 at 6cm

#	Calendar date	Time[a] t	Image A	Image B
1	23 Jun 1979	4,047.50	39.26	31.71
2	13 Oct 1979	4,160.16	39.26	29.67
3	23 Feb 1980	4,292.79	37.37	29.69
...	...	...	...	...
$n = 143$	6 Oct 1997	10,728.18	33.06	22.32

[a] Julian date - 2.44×10^6

data more attractive than radio data, because they are more precise; about 0.006% – 0.474% of its flux, i.e 0.001–0.08 mag. This is an advantage over radio data (2%). However, the disadvantage is that optical data contain more oscillations than radio data, and they are sensitive to other sources of noise such as microlensing [13, 60]. In Fig. 2.3, on the right-hand side, are depicted three data sets (hereafter referred to as DS1, DS2 and DS3, from bottom upwards). DS1 is an optical data set at g-band [49] with $n = 97$ observations from 2 December 1994 to 8 April 1996. DS2 is also an optical data set at r-band [49] with $n = 100$ observations during same period of time as in DS1. And DS3 refers to optical data at r-band [59] with $n = 422$ observations from 2 June 1992 to 8 April 1997. In both DS1 and DS2, there is a gap of 182 days because a time delay of about 420 days was known a priori [49].

Contrary to radio data, optical astronomers measure the brightness of a source (flux) using imaging devices (e.g., Charge-Coupled Device – CCD), with filters to restrict the range of wavelength/frequency of light observed. The flux f of light from a source is expressed in logarithmic units known as magnitudes (mag), defined as $\text{mag} = -2.5 \log_{10} f + \text{constant}$, where f can be represented in mJy (see radio flux units above). The errors on mag are mainly measurement errors, assumed to be zero-mean Gaussian. The green (g-) and red (r-) bands represent measurements obtained with filters in the wavelength range 400–550 nm and 550–700 nm, respectively.

Table 2.2: Optical Data: Q0957+561 at g-band

Time[a] t	Image A	Error A	Image B	Error B
9,689.009	16.9505	0.0152	16.8010	0.0152
9,691.007	16.9439	0.0111	16.7957	0.0111
9,695.001	16.9356	0.0090	16.7949	0.0090
...	...	...	...	...
10,253.672	17.0544	0.0084	16.9206	0.0084
10,266.665	17.0544	0.0205	16.9808	0.0205
10,268.642	17.0798	0.0170	16.9261	0.0170
10,270.652	17.0928	0.0145	16.9597	0.0119

[a] Julian date - 2.44×10^6

2.4 Artificial Data

Since the exact time delay of Q0957+561 is unknown (see §1 and §3), we use artificial data sets to perform a set of controlled large-scale experiments in order to measure the accuracy of time delay estimation techniques on gravitational lens systems. We generate simulated data sets with different levels of noise and varying sizes and locations of observational gaps.

2.4.1 DS-500

For this data, DS-500, the basic signal was constructed by superimposing twenty Gaussian functions with centres and widths generated randomly. The width was allowed to vary from zero up to a quarter of the duration of the entire monitoring campaign. Next, two artificial fluxes were created by scaling and shifting the basic signal in the flux density and time domains, respectively. The amplitude and flux densities are similar to radio data, 4 cm [32]. The flux ratio was set to $M = 1/1.44$ and the temporal shift was equal to $\Delta = 500$ days. The time goes from 0 to $T_S \cdot \Delta$ days with s_1 samples per Δ days ($T_S = 10$ and $s_1 = 5$), i.e., if the samples had been regularly sampled, we would have had a separation of $z = \Delta/s_1$ days between samples. To irregularly sample, we disturbed the regular observation times with a

Table 2.3: Artificial Data: DS-500

	Gap size					
Noise	**0**	**1**	**2**	**3**	**4**	**5**
0%	1	10	10	10	10	10
1%	50	500	500	500	500	500
2%	50	500	500	500	500	500
3%	50	500	500	500	500	500
Sub-Total	151	1510	1510	1510	1510	1510

Total = 7,701 data sets per underlying function.

5 underlying functions yield 38,505 data sets.

random variable uniformly distributed in $[-P \cdot z, +P \cdot z]$, $P = 0.49$. Moreover, we simulated continuous gaps in observations by imposing five blocks of missing data. The blocks were located randomly with at least one sample between them. We used block lengths, *gap size*, of $1, 2, ..., 5$ (see Table 2.3).

Three levels of noise were used to contaminate the flux signal: 1%, 2% and 3% of the flux; these represent our measurement errors $\sigma_A(t_i)$ and $\sigma_B(t_i)$, which are standard deviations of the flux distribution at each observation time. Fig. 2.4 depicts an example of a couple of scaled and shifted artificial fluxes[7].

We use 5 different underlying functions (basic signals). For each underlying function, there are 50 realisations for each noise level by adding a Gaussian noise to the underlying functions. For each such a data set, there are 10 realisations of missing observational blocks. Overall, DS-500 contains 38,505 distinct data sets; 7,701 data sets per underlying function (see Table 2.3).

2.4.2 DS-5

These artificial data, DS-5, are generated as above, but with an observational season of 1.3 years, 50 irregular samples, a true time delay of 5 days, an offset $M = 0.1$. Three levels of noise are involved 0.03%, 0.106% and 0.466% of mag; they come from Image

[7]More plots can be found at http://www.cs.bham.ac.uk/∼jcc/artificial/

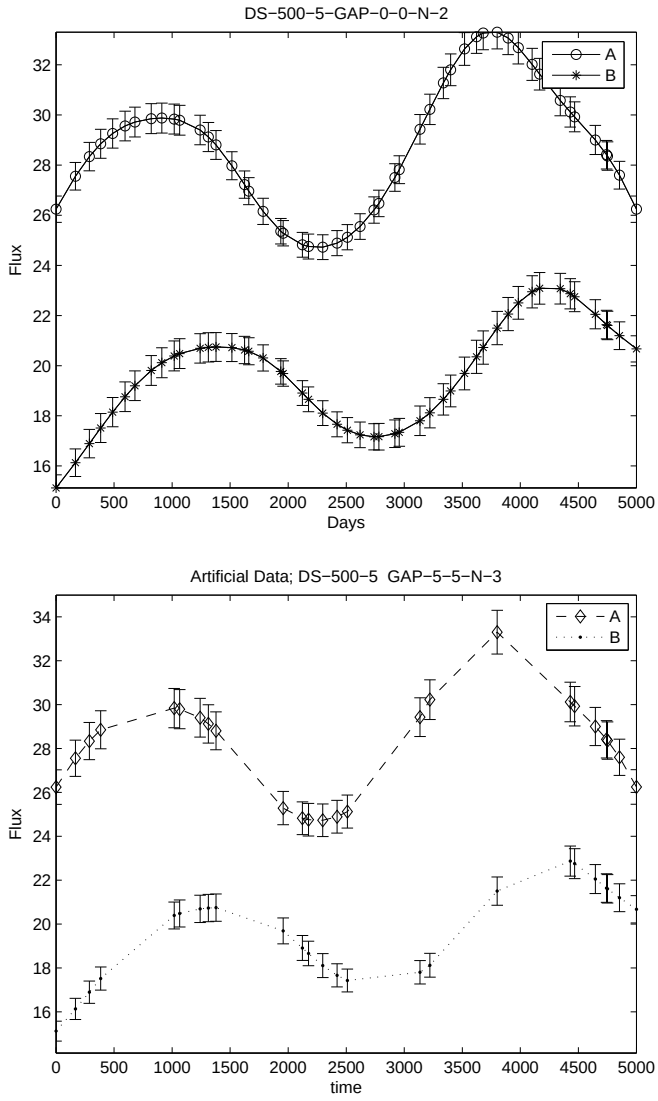


Figure 2.4: DS-500 Artificial Data. At the top, this plot shows the underlying function DS-500-5 without observational gaps but irregularly sampled, and the error bars of 2% of the flux are shown. At the bottom are the same noise-free fluxes with imposed observational gaps of length 5; here error bars are 3% of flux.

A of DS3: minimum (0.005 mag), average (0.0177 mag) and maximum (0.078 mag), respectively. Gaps are simulated as above. DS-5 employs five different underlying functions[8], 50 realisations per level of noise and ten realisations per gap size. This yields 38,505 data sets under analysis. Thus, these data sets simulate optical data

[8]Plots are available at http://www.cs.bham.ac.uk/∼jcc/artificial-optical/

with low time delay and low offset with high precision [59]. Figure 2.5 shows two plots with distinct error bars; with and without gaps.

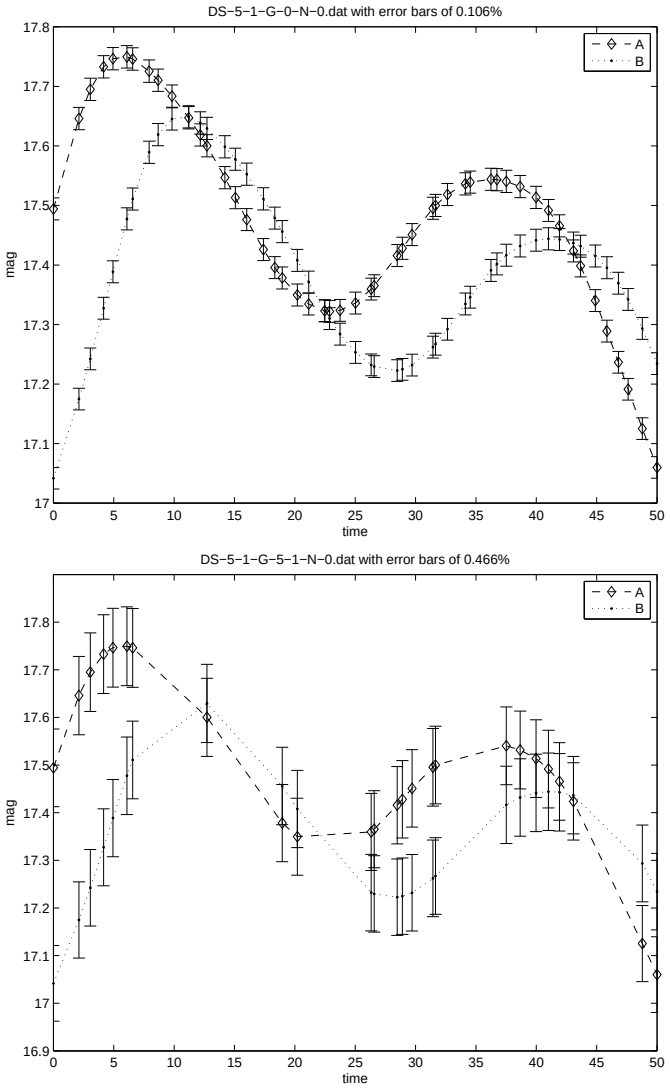


Figure 2.5: Artificial Data: DS-5. At the top is depicted the first underlying function (DS-5-1) without noise and no gaps. Error bars represent 0.106% of mag. At the bottom, this data set corresponds to the same underlying function without noise and gap size equals to five (first realisation). Error bars represent 0.466% of flux.

2.5 Chapter Summary

We have introduced the concepts of gravitational lensing and gravitational lens, which are the source of our data. The time delay problem has been given, i.e., the time delay Δ between pairs of light curves – in Q0957+561, the image B is delayed with respect to the image A (see Fig. 2.3a). The time delay is estimated directly from the time series, for optical or radio observations, and the data sets are shown in Figure 2.3, which correspond to the quasar under study in this thesis. A snapshot of this quasar is in Fig. 2.2d. The importance of studying the time delay is also presented in §2.2. Finally, we described the artificial data, DS-500 and DS-5. These data will be used to compare the performance of all the methods given in the next two chapters. Furthermore, with these data, it will be possible to measure bias and variance, among other statistical estimators (see Appendix A).

Chapter 3

Literature Review and Other Methodologies

HERE we present the literature survey based on Q0957+561; some time delay estimates for this quasar are given from 1997 to 2005 along with their methods; estimates before 1997 can be found in Haarsma *et al.* [31]. Further, a review of the most important methods is introduced, finally, more artificial data are described: PRH data and Harva data. Strictly speaking, we start the literature review in §1 and it continues in the following chapters.

3.1 Time Delay Estimates: Q0957+561

Table 3.1 contains a review in chronological order of the more recent time delay estimates of the quasar Q0957+561 and the methods employed. The first column indicates the type of data set used, either optical or radio. The second one displays the year when the time delay estimate was published. In third column are the method(s) used for time delay estimation. The last column has the estimates (in days) with their confidence intervals (CI). These CI are estimated in distinct ways; for instance, it can be 95% or 68% (1σ) from Monte Carlo simulations [72](§15.6) or bootstrap methods [35]. Note that these estimates have been adopted by their authors, and

Table 3.1: Review of Time Delay Estimates of Q0957+561 from 1997.

Data	**Year**	**Method(s)**	**Time delay**
Optical[g,r]	1997	Linear, Cross correlation, PRH method and Dispersion spectra	417±3 [49]
Optical[g]	1997	Cross correlation and Dispersion spectra	427±3 [58]
Optical[r]	1997	SOLA	425±17 [69]
Optical[g,r]	1998	Dispersion spectra	416.3±1.7 [66]
Radio[4,6]	1999	PRH method and Dispersion spectra	409±30 [32]
Optical[g,r]	2001	Linear, Cross correlation and Dispersion spectra	422.6±0.6 [57]
Optical[r]	2001	χ^2 algorithm	423±9 [9]
Optical[r]	2003	PRH method	417.09±0.07 [13]
Optical[r]	2003	Dispersion spectra and χ^2 algorithm	424.9±1.2 [59]
Radio[4,6]	2005	Bayesian method	394 ± 8 [33]
Optical[r]	2005	Bayesian method	423.5 ± 0.5 [33]

[g] g-band; [r] r-band; [4] 4 cm; [6] 6 cm

therefore rarely converge all methods on all data sets to the reported time delay.

This gravitational lens Q0957+561 is the most extensively monitored one so far, being the first to be discovered (see §1.1). As is evident in Table 3.1, a whole range of estimates (with varying uncertainty bounds) for the gravitational lens is available; the problem is that we do not know the actual time delay. Therefore, one of the aims of this thesis is to study the reliability of several time delay estimation methods in a large set of controlled experiments on artificially generated data with realistically modelled observational noise and mechanisms for missing measurements (see §2.4). Only after learning lessons from such a study does it make sense to offer yet another batch of time delay estimation claims.

Table 3.1 has the main time delay estimation methods that have been used on gravitational lens data. The Cross correlation method [49, 58], PRH method [71], Dispersion spectra method [65] and Bayesian method [33, 34] are described in §3.2; these first three methods have been widely used in the literature. Therefore, we

employ them as base-line models when reporting the performance of our methods.

Of the methods mentioned in Table 3.1, the Linear method uses chi-squared (χ^2) fitting [72](§15.1). Since the data are irregularly sampled, linear interpolation in the observational gaps is performed [49].

The method of Subtractive Optimally Localised Averages (SOLA) has been proposed as a method for solving inverse problems. The method was adopted by Pijpers [69] who formulated time delay estimation as an inverse problem. It is worth noting that SOLA employs kernels, called averaging kernels. However, SOLA differs from our approach in several respects: *i*) SOLA makes a symmetrical treatment of the two estimated signals (image A is fixed and image B is varied to match A and vice versa); *ii*) the reported time delay is the mean of the estimated time delays in the two symmetric cases; and *iii*) a free parameter is used to adjust the relative weighting of the errors in the variance-covariance matrix. It is argued that parameter estimation in SOLA is problematic [51, 73], therefore, this method has been not often used.

The χ^2 algorithm [9, 59] is a χ^2-based method similar in principle to our model, in that it also uses the notion of an underlying model curve when fitting the two observed images. However, the underlying model is assumed to be regularly sampled. It is regularised using a smoothing term [9](Eq. 3). Confidence intervals on the delay are estimated by performing Monte Carlo simulations [9].

3.2 Methods

Let us denote the observed signals from two lensed images A and B of the same distant source, as two time series $x_A(t_i)$ and $x_B(t_i)$, where $t_i, i = 1, 2, ..., n$ are discrete observational times. Observational errors are modelled as zero-mean normal distributions $N(0, \sigma_A(t_i))$ and $N(0, \sigma_B(t_i))$.

3.2.1 Interpolation

Perhaps, this is the more straightforward method to think about. If one sees Fig. 2.3a, where is clearer that a time delay exists, it is possible to shift one of the time series in two directions, time and flux, keeping the other one fixed. Then, the fitting

can be measured through a loss function; e.g., mean squared error. Therefore, the less the error, the better the time delay and the flux shift. But, because irregular sampling occurs, it is not possible to measure the error at certain times t_i given a time shift. Therefore, interpolation is needed. The problem of this formulation is that, depending on the desired time resolution, one can come up with too many samples due to interpolation. For instance, the 6 cm data set has 143 observations, so with a time resolution of 0.01 one obtains 668,069 samples. The amount of samples grows exponentially, and this makes intractable this method; in particular when huge amounts of data sets are involved. Besides, it is necessary to define a range of trial time shifts where the error will be measured.

However,the interpolation method needs to be defined before use. We have found that linear interpolation gives reasonably good results compared with other methods, such as splines and nearest neighbours.

This methodology is rarely used but is still valid [46, 34]. In theory, it is better to avoid it, since interpolation adds more uncertainty to unseen data, where observational errors are not easy to discover.

3.2.2 Cross Correlation

Basically, there are two versions of the methods based on cross correlation: the Discrete Correlation Function (DCF) [19] and its variant, the Locally Normalised Discrete Correlation Function (LNDCF) [52]. Both calculate correlations directly on discrete pairs of light curves. These methods avoid interpolation in the observational gaps. They are also the simplest and quickest time delay estimation methods.

First, time differences (lags), $\Delta t_{ij} = t_j - t_i$, between all pairs of observations are binned into discrete bins. Given a bin size $\Delta\tau$, the bin centred at lag τ is the time interval $I_\tau = [\tau - \Delta\tau/2, \tau + \Delta\tau/2]$, where $P(\tau)$ is the number of observational pairs in the bin centred at τ. The DCF at lag τ is given by

$$DCF(\tau) = \frac{1}{P(\tau)} \sum_{i,j}^{t_i, t_j \in I_\tau} \frac{(x_A(t_i) - \bar{a})(x_B(t_j) - \bar{b})}{\sqrt{(\sigma_a^2 - \sigma_A^2(t_i))(\sigma_b^2 - \sigma_B^2(t_j))}}, \tag{3.1}$$

where $\bar{a}$ and $\bar{b}$ are the means of the observed data, $x_A(t_i)$ and $x_B(t_j)$, and the variances are σ_a^2 and σ_b^2, respectively.

Likewise,

$$LNDCF(\tau) = \frac{1}{P(\tau)} \sum_{i,j}^{t_i,t_j \in I_\tau} \frac{(x_A(t_i) - \bar{a}(\tau))(x_B(t_j) - \bar{b}(\tau))}{\sqrt{(\sigma_a^2(\tau) - \sigma_A^2(t_i))(\sigma_b^2(\tau) - \sigma_B^2(t_j))}}, \tag{3.2}$$

where $\bar{a}(\tau)$, $\bar{b}(\tau)$, $\sigma_a^2(\tau)$ and $\sigma_b^2(\tau)$ are the lag means and variances in the bin centred at τ.

The time delay Δ is found when $DCF(\tau)$ and $LNDCF(\tau)$ (3.1)-(3.2) are greatest; i.e., at the best correlation [19, 52].

3.2.3 PRH Method

This method is widely used for time delay estimation. Its fundamentals are based on the theory of stochastic processes and Wiener filtering [71]. Given two light curves $\vec{x_A}$ and $\vec{x_B}$, the PRH method combines them into a single time series $\vec{y}$ by assuming a trial time delay $\Delta_t = [\Delta_{min}, \Delta_{max}]$ and a constant offset/ratio M between $\vec{x_A}$ and $\vec{x_B}$. Thus, for each of the two images, we end up with a new data set of $2n$ observations; half is interpolated using the other image. The parameter M can be estimated as a difference between the weighted means of the observed images, $\vec{x_A}$ and $\vec{x_B}$; the weights are derived from the quoted observational errors, $\vec{\sigma_A}$ and $\vec{\sigma_B}$.

The optimal time delay Δ is estimated by minimising

$$\chi^2(\Delta_t) = \vec{y}^{\mathrm{T}} \left(\vec{A} - \frac{\vec{A}\vec{E}\vec{E}^{\mathrm{T}}\vec{A}}{\vec{E}^{\mathrm{T}}\vec{A}\vec{E}} \right) \vec{y}, \tag{3.3}$$

which is a measure of the goodness of fit of measurements from a Gaussian process [71]. Here, $\vec{y}$ is the combined signal[1], $\vec{E}$ is a column vector of ones, and

$$\vec{A} = \left\{ C_{ab} + \langle \sigma_a^2 \rangle \delta_{ab} \right\}^{-1} \tag{3.4}$$

where

$$C_{ab} = \langle y(t_a) y(t_b) \rangle \equiv C(t_a - t_b) \equiv C(\tau_{ab}) \tag{3.5}$$

[1]Note that Press *et al.* [71] refer to $\vec{y}$ as a component rather than a combination of the components, image A and image B. The same occurs with the matrices $\vec{A}$ and $\vec{C}$ in Eq. (3.4) and (3.5).

is a covariance model estimated from the data (angle brackets denote the expectation operator); t_a, t_b, $a, b = 1, ..., 2n$, are sample times of the combined light curve; δ_{ab} denotes the Kronecker delta [86](pp.52). Press *et al.* [71] suggest finding $C(\tau_{ab})$ through a first-order structure function $V(\tau_{ab}) = \langle s^2 \rangle - C(\tau_{ab})$ with the assumption that s is stationary, where s is the source (clean data) of $\vec{y}$. Then, the structure function $V(\tau_{ab})$ is computed from the data, using a single image, by determining lags[2]

$$\tau_{ij} \equiv |t_i - t_j| \tag{3.6}$$

and values

$$v_{ij} \equiv (x_S(t_i) - x_S(t_j))^2 - \sigma_S^2(t_i) - \sigma_S^2(t_j). \tag{3.7}$$

where $S = \{A, B\}$ denotes that $\vec{x}$ comes from either image A or image B.

All pairs (τ_{ij}, v_{ij}) are sorted with respect to τ_{ij} and binned into 100 bins [71]. The values of τ_{ij} and v_{ij} in each bin are averaged and finally a power-law model is built to fit the binned list,

$$V(\tau_{ij}) = \mathcal{B}\tau_{ij}^{\mathcal{A}}. \tag{3.8}$$

Note that this model is linear in logarithmic scale,

$$\ln(V(\tau_{ij})) = \ln(\mathcal{B}) + \mathcal{A}\ln(\tau_{ij}). \tag{3.9}$$

Therefore parameters $\mathcal{A}$ and $\mathcal{B}$ of the structure function can be determined using a simple line-fitting algorithm[3]. Remember that $\vec{V}$ is estimated on a single signal, and one would naturally expect that estimates on image A would be similar to those on image B. However, this is often not the case. Press *et al.* [71] claim that it does not matter which image is chosen. Our experience suggests that this may be an overoptimistic expectation. Moreover, the matrix $\vec{A}$ (3.4) is often ill-conditioned and we regularise the inversion operation through SVD.

[2]Note that τ_{ij} denotes lags Δt_{ij}; i.e., to be consistent with authors.

[3]We have noticed that in some cases a negative slope $\mathcal{A}$ is found. Also we note that a negative $\mathcal{B}$, y-intercept, in Eq. (3.9), and $\tau_{ij} = 0$ leads to numerical overflow. In such cases we apply a shift upwards in Eq. (3.7), and we set τ to a small positive number; e.g., $\min \tau_{ij}$ in (3.6).

Artificial Data: PRH data

A possible criticism of our testing framework in §2.4 may be that we construct artificial underlying functions as linear superpositions of Gaussian functions, while our proposed model is a linear superposition of Gaussian kernels – discussed in the next chapter. However, widths of the Gaussian functions used to construct the underlying functions are much greater than the widths of Gaussian kernels in our model formulation, and thus this criticism is less important. Still, in order to properly address this issue, we let the PRH method "play at its own game" by constructing a set of underlying functions using the PRH method[4] with a specified structure function (SF). We refer to such data sets as PRH data.

These are Monte Carlo time series, generated exactly as described in [71](§5.2), with a fixed SF given by $\mathcal{B} = 1/5.36\times10^5$ and $\mathcal{A} = 0.246$ [90, 70]. We use a monitoring campaign length of 8 months with an irregular sampling rate, every two days with periodic gaps of fifteen days [21] yielding $n = 61$ samples. The ratio between light curves is $M = 1$.

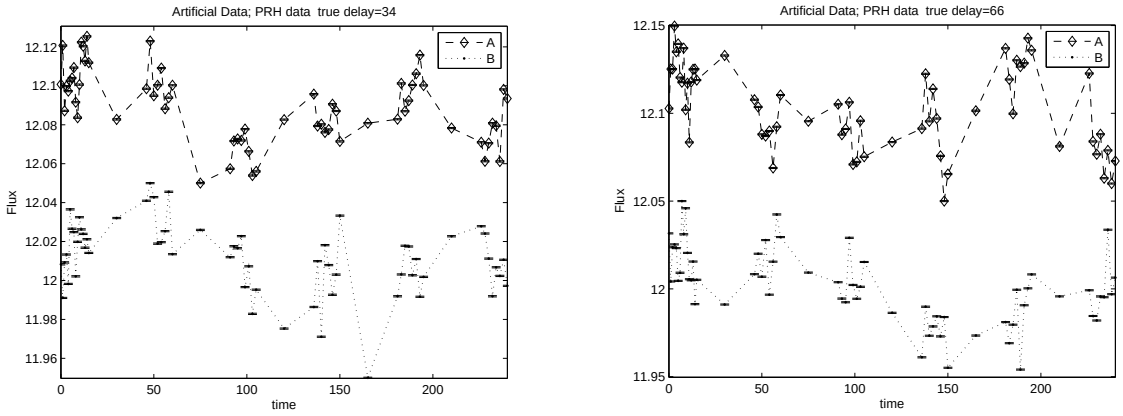
We randomly choose seven time delays in the range of 30–100 days and then we generate 100 Monte Carlo data sets. To simulate observational errors, we use a fixed variance of 1×10^{-7} in order to obtain distinguishable shapes by eye, i.e., low noise. Two data sets are shown in Fig. 3.1.

3.2.4 Dispersion Spectra

Dispersion spectra is a weighted sum of squared differences between $x_A(t_i)$ and $x_B(t_i)$ [65, 66]. The method is similar to those based on DCF (see §3.2.2). However, it measures the dispersion of the time series of two light curves in a different way by combining them (given a trial time delay Δ_t and offset/ratio M) into a single signal, $\vec{y}$, as in the PRH method (§3.2.3). We employ two versions of this method [66]:

$$D_1^2(\Delta_t) = \min_M \frac{\sum_{a=1}^{2n-1} w_a \left(y(t_{a+1}) - y(t_a)\right)^2}{2\sum_{a=1}^{2n-1} w_a} \tag{3.10}$$

[4]We are grateful to the A&A anonymous reviewer for this suggestion.



(a) (b)

Figure 3.1: PRH Data. The error bars represent a variance of 1×10^{-7}. **(a)** This is a realisation for a true delay of 34; image A has been shifted upwards by 0.08 for visualisation. **(b)** In this realisation, the true delay is 66. Image A has been shifted upwards by 0.1 for visualisation.

and

$$D^2_{4,2}(\Delta_t) = \min_M \frac{\sum_{a=1}^{2n-1}\sum_{c=a+1}^{2n} S^{(2)}_{a,c} W_{a,c} G_{a,c} \left(y(t_a) - y(t_c)\right)^2}{2\sum_{a=1}^{2n-1}\sum_{c=a+1}^{2n} S^{(2)}_{a,c} W_{a,c} G_{a,c}}, \tag{3.11}$$

where

$$w_a = \frac{1}{\sigma^2(t_{a+1}) + \sigma^2(t_a)}, W_{a,c} = \frac{1}{\sigma^2(t_a) + \sigma^2(t_c)} \tag{3.12}$$

are the statistical weights taking in account the measurement errors, where $G_{a,c} = 1$ only when $y(t_a)$ and $y(t_c)$ are from different images, and $G_{a,c} = 0$ otherwise.

$$S^{(2)}_{a,c} = \begin{cases} 1 - \frac{|t_a - t_c|}{\delta}, & \text{if } |t_a - t_c| \leq \delta \\ 0, & \text{otherwise.} \end{cases} \tag{3.13}$$

The estimated time delay Δ is found by minimising D^2 over a range of time delay trials, as above.

Compared with D^2_1, the $D^2_{4,2}$ method has an additional parameter, the *decorrelation length* δ, which signifies the maximum distance between observations that we are willing to consider when calculating the correlations [65].

So far, $D^2_{4,2}$ has been widely used; e.g., see [21, 70]. Furthermore, it is simple and fast.

3.2.5 A Bayesian Estimation Method

Given the two observed time series $\vec{x_A}$ and $\vec{x_B}$, again, these are combined into a single time series $x(t_a), a = 1, ..., 2n$; the correspondence is denoted by $k(a) \in \{1, 2\}$ [33, 34]. This combination procedure is similar to the Dispersion spectra and PRH methods. But this method models $\vec{x_A}$ and $\vec{x_B}$ from the source $\mathbf{S} = \{s(t_a)|a\}$ with scale and shift factors, $a_{k(a)}$ and $b_{k(a)}$ respectively. For $\vec{x_B}$, the shift is bidirectional, in time Δ and in flux $b_{k(a)}$.

The modelled observations, as noisy versions of the source, are given by

$$x(t_a) \sim N(a_{k(a)}s(t_a) + b_{k(a)}, e^{v_{k(a)}}) \tag{3.14}$$

where $N(\mu, \sigma^2)$ denotes the Gaussian distribution with mean μ and variance σ^2, and $v_{k(a)}$ denotes the log variances of the noise. Additionally, the source $\vec{s}$ is assumed to be temporally correlated and modelled as a Markov process

$$\begin{aligned} s(t_1) &\sim N(0, 10^2) \\ s(t_a) &\sim N(s(t_{a-1}), (t_a - t_{a-1})^{\gamma_M} e^{\omega_M}) \quad a > 1 \end{aligned} \tag{3.15}$$

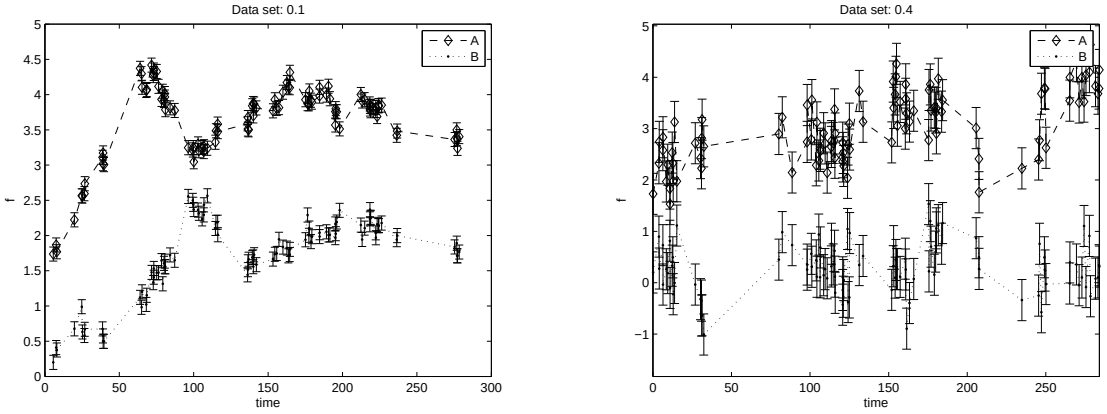
where the set of parameters is $\mathbf{\Theta} = \{a_k, b_k, v_k, \gamma_M, \omega_M\}$, except Δ.

When the uncertainties of observations $\sigma_y(t_a)$ are known, the data are referred to as $\mathbf{Y} = \{y(t_a)|a\}$.

The aim of this methodology is to find the posterior distribution of the delay $p(\Delta|\mathbf{Y})$, i.e., the distribution of Δ given the data $\mathbf{Y}$. The posterior distribution is estimated by sampling $p(\Delta, \mathbf{\Theta}|\mathbf{Y})$ through the MCMC method and Metropolis algorithm. The marginalisation is made to obtain $p(\mathbf{Y}|\Delta, \mathbf{\Theta})$ from $p(\mathbf{Y}, \mathbf{S}|\Delta, \mathbf{\Theta})$. Thus, $p(\mathbf{Y}|\Delta, \mathbf{\Theta})$ is used (with the prior) to sample from $p(\Delta, \mathbf{\Theta}|\mathbf{Y})$. The marginalisation over the source $\mathbf{S}$ is a crucial part in the method, since otherwise sampling would be almost impossible. The derivation of likelihood and priors are in [34]. A compact presentation of this approach is also in [33].

Artificial Data: Harva data

The data were generated by the above model (3.14)-(3.15) with $\gamma_M = 2$, $\omega_M = 2 \ln 0.05$, $a_1 = 1$, $b_1 = 0$, $a_2 = 0.8$ and $b_2 = 0.2$ [34]. These data simulate three levels of noise with 225 data sets per level of noise, where each level of noise represents the variance; 0.1^2, 0.2^2 and 0.4^2. Each data set has $n = 100$ samples. They are irregularly sampled and the true time delay in all the cases is 35 units. Some examples are shown in Fig. 3.2.



(a) (b)

Figure 3.2: Harva Data. **(a)** First realisation of the data set with variance 0.1^2, image A has been shifted upwards by 1.5 for visualisation. Error bars represent a standard deviation of 0.1. **(b)** First realisation of the data set with variance 0.4^2, image A has been shifted upwards by 2.9 for visualisation. Error bars represent a standard deviation of 0.4.

3.3 Chapter Summary

We started with a review of time delay estimates of Q0957+561. At the same time, we listed the principal methods used for time delay estimation, where the DCF, LNDCF, PRH and Dispersion spectra methods are the most popular, and therefore the methods to beat; this is the reason why we presented them in detail. Two kinds of artificial data were also described: PRH data and Harva data. In the next chapter, we introduce our kernel-based method. In §6, we will present the results of the above

methods on real and artificial data. In §7, we discuss the results in detail by pointing out the advantages and disadvantages of each method, and some conclusions are also given.

Chapter 4

Machine Learning and Kernel-based Method

FIRST we contextualise our kernel-based method in §4.1, and start by introducing some relevant concepts in statistical and machine learning. Then, we introduce our kernel methodology to deal with irregularly sampled time series in gravitational lensing in §4.2.

4.1 Machine Learning

Many authors refer to Machine Leaning (ML) as a branch of Artificial Intelligence (AI); for instance Tom Mitchell's book [54], which is a good book to introduce ML topics, is well known in AI. It is not intended here to discuss whether ML either derives from AI or from statistical learning theory [35], or from other sources. The fact is that ML techniques are those that implement learning from data in an automatic way (algorithm), and include neural networks, evolutionary computation, reinforcement learning, Bayesian networks, and more recently support vector machines and kernel methods just to mention the most popular approaches. Nowadays, there are several conferences and journals that deal with ML and all types of learning.

Our interest is to develop an automatic method, an algorithm, to estimate the

time delay given a set of observations, so ML is a promising approach to explore because its main scope is noisy data.

Across ML literature, one finds three types of learning: supervised learning, unsupervised learning and reinforcement learning [54, 35, 86]. What kind of learning to use will depend on the problem to solve. In our case, our problem deals with supervised learning, which is described in the next subsection.

4.1.1 Supervised Learning

Supervised learning occurs when we have labelled data for a given phenomenon. It is called "supervised" because of the presence of the outcome variable to guide the learning process [35]. In the unsupervised learning problem, there are not measurements of the outcome (i.e., no labels). Depending on the types of outputs, supervised learning can perform either classification or regression. Classification if the outputs are a finite number of categories, and regression if the outputs are real numbers; here we deal with regression. Supervised learning can also produce preferences, which is known as preference learning.

In theory, supervised learning may give us models for an observed phenomenon, i.e., the problem to solve. The observations that describe such phenomenon are divided basically into two sets: training set and test/validation set. The training set is used to model the system and the test/validation set, as its name suggests, is used to test/validate the model in order to obtain generalisation, so one can deal with the problem of overfitting. The test set and validation set may be the same data set. Some authors refer the test set as validation set; e.g., see [54]. Others define clearly three data sets: training, validation and test [35], where validation is used to estimate prediction error for model selection, and, the test set is used for assessment of the generalisation error of the final model. With the use of test/validation set, we have a robust model that describes the phenomenon. The division of data sets occurs when there are a lot of observations, and when we remove some of them, this does not affect the model.

In our case, we have few observations and we cannot remove any of them because we are adding more gaps in the time series, so the training set and the test set are

the same. To obtain generalisation, other approaches are explored in the learning process.

4.1.2 Kernels, Kernel Methods and Kernel Machines

In general, a kernel[1] is a two variable function $K(t', t)$, and the mathematical theory of kernels is relatively old (about 1909); e.g., see review in [86](§3.6). In kernel methods, the basic idea is the transformation to the feature space ϕ, so

$$K(t', t) = \langle \phi(t'), \phi(t) \rangle \tag{4.1}$$

where $K : \mathcal{L} \times \mathcal{L} \mapsto \Re$ and $t', t \in \mathcal{L}$; thus $\langle \cdot, \cdot \rangle$ denotes the dot product. Then the map $\phi : \mathcal{L} \mapsto \mathcal{H}$ is the so-called reproducing kernel Hilbert space (RKHS) and Eq. (4.1) known as *kernel function* [86, 84]. This transformation allows us to deal with nonlinearity through the linear space $\mathcal{H}$ (feature space), and it is also known as the kernel trick [84].

The above is a general formulation of kernels. In practice, there are several types of kernels including polynomial, Gaussian and sigmoid kernels [86, 84]. This thesis concentrates only on Gaussian kernels $K(t', t) = \exp(-|t, t'|^2/\omega^2)$, because few parameters are involved (centres t' and width ω) and they are widely used in both theory and practice [22, 38, 2, 6, 50, 91, 41].

Given a training set $T = \{t_1, ..., t_n\}$ and a kernel function $K(\cdot, \cdot)$, there is a Gram matrix

$$K_{ij} = K(t_j, t_i), \text{ for } i, j = 1, ..., n. \tag{4.2}$$

which is the core ingredient in the theory of kernel methods, and it is the main data structure in their implementation. The larger the training set T, the larger the Gram matrix K_{ij}. Therefore, kernels are also considered to be memory-based methods [35].

With kernels, one is able to create complicated kernels from simple building blocks (4.2); see [86](§3.4). Moreover, there are other kernel constructions such as graph

[1] In operating computer systems, the concept of kernel is distinct since it means the core of the system.

kernels, string kernels, P-kernels among many others [86].

Since this thesis deals with regression, the kernels are represented as (known as representer theorem [84])

$$f(t) = \sum_{j=1}^{n} \alpha_j K(t_j, t) \tag{4.3}$$

which is a linear combination of kernels (basis functions) and $\alpha_j \in \Re$. For example, let $T = \{10, 15, 25, 32, 38, 43\}$ and $K(t_j, t) = \exp(-|t - t_j|^2/\omega^2)$ be a training set of size six and a Gaussian kernel with $\omega = 3$, respectively. The time t goes from 0 to 50 with a resolution of 0.1, and $\vec{\alpha} = [0.5, 1, 0.5, 1, 1.5, 1]$. This set of Gaussian kernels $K(t_j, \cdot)$ scaled by α_j is shown in Fig. 4.1. Then, T has the centres of Gaussian functions and $\vec{\alpha}$ contains the heights. Now it is clear that if one wants to fit an arbitrary curve $g(t)$, we may do it through $f(t)$ in (4.3). In the above example, we fixed centres and weights $\vec{\alpha}$. However, the most common is to locate centres at observations t_i, but kernels are quite flexible and they can be located anywhere. The weights $\vec{\alpha}$, or heights, play an important role, and here is where the concept of *learning* is involved.

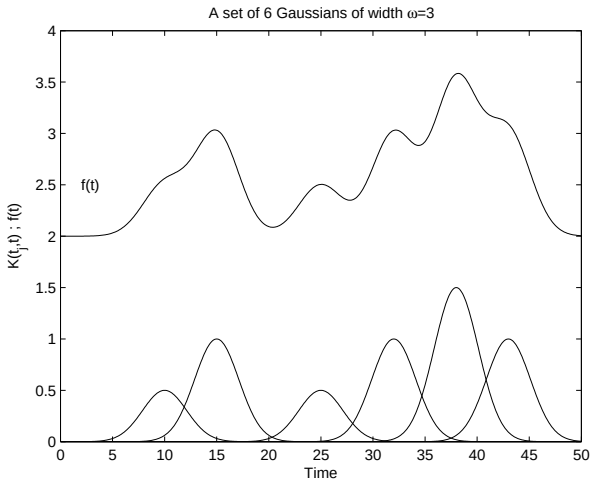


Figure 4.1: A Set of Six Gaussian Kernels. The centres are at $\{10, 15, 25, 32, 38, 43\}$, $\omega = 3$ and $\vec{\alpha} = [0.5, 1, 0.5, 1, 1.5, 1]$. At the top is $f(x)$, which is a linear combination of Gaussian kernels (4.3); it has been shifted upwards by two units for visualisation. At the bottom are the set of Gaussian kernels (basis functions).

On-line learning is processing the training data one at a time as it is received

[22], and in real-time applications is a very important issue. In the above kernel formulation, we process all training data at once, which is batch learning.

Again, given a training set $T = \{t_1, ..., t_n\}$ and a kernel function $K(\cdot, \cdot)$, the Gram matrix is straightforward, but $\vec{\alpha}$ is not, and it needs to be learned from the data. There are a number of different directions, but these can be grouped in two: eigen-decompositions and convex optimisations [86]; the latter leads to Support Vector Machines (SVM) [35, 86, 84]. In this thesis, we explore only eigen-decompositions; see §4.2.1.

Finally, the concept of kernel machines comes from learning machines [8] and SVM [36, 2].

4.1.3 Other Disciplines Related to Kernels

This brief review aims to cover those disciplines that are not referenced in the ML literature, specifically in the kernels literature [86, 84].

Radial Basis Function (RBF) networks are similar to kernels, but developed in a different field; i.e., artificial neural networks [36, 55]. In fact, a Gaussian kernel is sometimes referred to as RBF Kernel; e.g., see [68, 50]. Recalling Eq. (4.3), it can be seen as a RBF network where $\vec{\alpha}$ are the weights and $K(\cdot, \cdot)$ the activation functions. In neural networks, the activation functions are usually chosen to be sigmoid. However, if there is a single hidden layer and Gaussian functions as activation functions, this leads to RBF networks. Consequently, one can see Eq. (4.3) plus a regularisation term as a RBF network [54]. Moreover, the normalised RBF network corresponds to the Nadaraya-Watson regression estimator; for a study presenting the relations among kernels, RBF networks and Nadaraya-Watson regression estimators, see [36](§5.12).

The example illustrated in Fig 4.1 is also related to inverse problems and ill-posed problems, where inverse theory has been mainly developed in geophysics [61, 82]. This theory also involves linear combinations of kernels, defined as $G(x, y)$, and it is also known as the Backus-Gilbert method [61, 72].

A problem is considered ill-posed when one or more conditions for well-posed problems are not satisfied. The conditions are *existence*, *uniqueness* and *continuity*; see [36](§5.4). In practice, the existence condition may be violated when an output

is not available for a given input (e.g., gaps). The uniqueness condition is likely to be violated if there may not be as much information in the training sample to reconstruct the input-output mapping uniquely. Finally, the unavoidable presence of noise adds uncertainty, and therefore there is likelihood for the continuity condition to be violated. The regularisation theory of Tikhonov is widely used in RBF networks for solving ill-posed problems.

Other related topics are general linear least squares [72](§15), where the concept of design matrix is similar to the Gram matrix (4.2), and wavelet analysis, which also combines a set of basis functions (scaled and shifted versions of a mother wavelet) to fit a curve; i.e., $g(t)$ [30, 72]. Still more generally, since we are contextualising, other disciplines come in; for instance if $g(t)$ is treated as a signal, one deals with signal processing. At the same time, signal processing overlaps with image processing when an image is studied as a signal.

4.1.4 Regression and Motivation

Several approaches appear for regression problems. The literature is vast on regression methods including linear, non-linear, parametric and non-parametric regression [36, 35, 72], which we will not attempt to summarise here. Some of them on the notion of interpolation (e.g., RBF networks) and others on the notion of density estimation (e.g., kernel regression).

This thesis concentrates on kernel regression; see next section. Besides the motivation given in §1.2, we propose our kernel formulation because

1. linearity in parameters enables us to use tools of linear algebra in parameter fitting and regularisation,

2. Gaussian kernel formulation using variable kernel widths is natural in cases of irregularly sampled data,

3. parameter sharing in (4.6) and (4.7) provides a transparent tool for coupling the two observed images.

As we have shown in the previous section, the RBF networks, Nadaraya-Watson

regression estimator and our kernel formulation belong to the same class of kernel regression techniques.

4.2 Kernel-based Method for Time Delay Estimation

We model the observed data from two lensed images A and B of the same distant source (see §2), as two time series

$$\begin{aligned} x_A(t_i) &= h_A(t_i) + \varepsilon_A(t_i) \\ x_B(t_i) &= h_B(t_i) \ominus M + \varepsilon_B(t_i), \end{aligned} \tag{4.4}$$

where $\ominus = \{\times, -\}$ denotes either multiplication or subtraction. Hence, M is either a ratio (for radio data) or an offset (for optical data) between the two images, and $t_i, i = 1, 2, ..., n$ are discrete observation times. The observation errors $\varepsilon_A(t_i)$ and $\varepsilon_B(t_i)$ are modelled as zero-mean Normal distributions

$$N(0, \sigma_A(t_i)) \quad \text{and} \quad N(0, \sigma_B(t_i)), \tag{4.5}$$

respectively. Now,

$$h_A(t_i) = \sum_{j=1}^{N} \alpha_j K(c_j, t_i) \tag{4.6}$$

is the "underlying" light curve that underpins image A, whereas

$$h_B(t_i) = \sum_{j=1}^{N} \alpha_j K(c_j + \Delta, t_i) \tag{4.7}$$

is a time-delayed (by Δ) version of $h_A(t_i)$ underpinning image B.

The functions h_A and h_B are formulated as in §4.1.2. Each function is a linear superposition of N kernels $K(\cdot, \cdot)$ centred at either c_j, $j = 1, 2, ..., N$ (function f_A), or $c_j + \Delta$, $j = 1, 2, ..., N$ (function f_B). The model (4.4)-(4.7) has N free parameters α_j, $j = 1, 2, ..., N$, that need to be determined by (learned from) the data. We use Gaussian kernels of width ω^2: for $c, t \in \Re$,

$$K(c,t) = \exp \frac{-|t-c|^2}{\omega_c^2}. \tag{4.8}$$

The kernel width $\omega_c > 0$ determines the 'degree of smoothness' of the underlying curves h_A and h_B. We describe setting of $\omega_j = \omega_{c_j}$ and regression weights α_j in the next subsections. In this study, we position kernels on all observations, i.e., $N = n$.

Finally, our aim is to estimate the time delay Δ between the temporal light curves corresponding to images A and B. Given the observed data, the likelihood of our model reads

$$P(\text{Data} \mid \text{Model}) = \prod_{i=1}^{n} p(x_A(t_i), x_B(t_i) \mid \Delta, \{\alpha_j\}), \tag{4.9}$$

where

$$\begin{aligned} p(x_A(t_i), x_B(t_i) \mid \Delta, \{\alpha_j\}) \;=\; & \frac{1}{2\pi\sigma_A^2(t_i)\sigma_B^2(t_i)} \\ & \exp\left\{\frac{-(x_A(t_i) - h_A(t_i))^2}{2\sigma_A^2(t_i)}\right\} \\ & \exp\left\{\frac{-(x_B(t_i) - M \ominus h_B(t_i))^2}{2\sigma_B^2(t_i)}\right\}. \end{aligned} \tag{4.10}$$

The negative log-likelihood (without constant terms) simplifies to

$$Q = \sum_{i=1}^{n} \left(\frac{(x_A(t_i) - h_A(t_i))^2}{\sigma_A^2(t_i)} + \frac{(x_B(t_i) - M \ominus h_B(t_i))^2}{\sigma_B^2(t_i)} \right). \tag{4.11}$$

To avoid extrapolation when we apply a time delay to our underlying curve, we do not evaluate the goodness of fit over all observations (avoiding border effects [25]):

$$Q = \sum_{u=1}^{n-b_1} \frac{(x_A(t_u) - h_A(t_u))^2}{\sigma_A^2(t_u)} + \sum_{v=b_2}^{n} \frac{(x_B(t_v) - M \ominus h_B(t_v))^2}{\sigma_B^2(t_v)}, \tag{4.12}$$

where b_1 is the greatest index satisfying $t_{n-b_1} \leq t_n - \Delta_{max}$, and b_2 is the smallest index satisfying $t_{b_2} \geq t_1 + \Delta_{max}$. Here, Δ_{max} is the maximum possible time delay we are willing to consider (fixed in advance).

We determine the model parameters and evaluate Eq. (4.12) for a series of trial values Δ_t. The time delay is then estimated as the value of Δ with minimal cost

(4.12). Note that if the errors cannot be modelled as Gaussian, Eq. (4.12) would need to be rewritten using an appropriate noise term.

4.2.1 Weights $\{\alpha_j\}$

We rewrite Eq. (4.11) as

$$Q = \sum_{i=1}^{n} \left(\left[\frac{x_A(t_i)}{\sigma_A(t_i)} - \frac{h_A(t_i)}{\sigma_A(t_i)} \right]^2 + \left[\frac{x_B(t_i)}{\sigma_B(t_i)} - \frac{M \ominus h_B(t_i)}{\sigma_B(t_i)} \right]^2 \right). \tag{4.13}$$

We replace Eqs. (4.6) and (4.7) into (4.13), and we obtain

$$\vec{K}\vec{\alpha} = \vec{x}, \tag{4.14}$$

where

$\vec{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N)^T$,

$$\vec{K} = \begin{bmatrix} K_A(c_1,t_1) & \cdots & K_A(c_N,t_1) \\ \vdots & \ddots & \vdots \\ K_A(c_1,t_n) & \cdots & K_A(c_N,t_n) \\ \hline K_B(c_1,t_1) & \cdots & K_B(c_N,t_1) \\ \vdots & \ddots & \vdots \\ K_B(c_1,t_n) & \cdots & K_B(c_N,t_n) \end{bmatrix}, \qquad \vec{x} = \begin{bmatrix} \frac{x_A(t_1)}{\sigma_A(t_1)} \\ \vdots \\ \frac{x_A(t_n)}{\sigma_A(t_n)} \\ \hline \frac{x_B(t_1)}{\sigma_B(t_1)} \\ \vdots \\ \frac{x_B(t_n)}{\sigma_B(t_n)} \end{bmatrix}, \tag{4.15}$$

and the kernels $K_A(\cdot,\cdot)$, $K_B(\cdot,\cdot)$ have the form:

$$K_A(c,t) = \frac{K(c,t)}{\sigma_A(t)}, \qquad K_B(c,t) = \frac{M \ominus K(c+\Delta,t)}{\sigma_B(t)}. \tag{4.16}$$

Hence,

$$\vec{\alpha} = \vec{K}^{+}\vec{x}. \tag{4.17}$$

We regularise the inversion in (4.17) through singular value decomposition (SVD), $\vec{K} = \vec{U} \cdot \vec{W} \cdot \vec{V}^T$, and $\vec{K}^{+} = \vec{V} \cdot [\text{diag}(1/w_i)] \cdot \vec{U}^T$ is the pseudo inverse (or Moore-Penrose inverse) [28, 72, 67]. SVD has some interesting properties such as $\vec{W}$ is a diagonal matrix with positive or zero elements (known as singular values), where

$w_i \in \vec{W}$, $\vec{U}$ and $\vec{V}$ are orthogonal so $\vec{U}^T \cdot \vec{U} = \vec{V}^T \cdot \vec{V} = 1$, and $\vec{V}$ is also square and row-orthonormal; i.e., $\vec{V} \cdot \vec{V} = 1$.

Since Eq. 4.14 is a noisy overdeterminated system [72](§2.6), ill-posed[2] (see §4.1.3), a regularisation procedure is needed. Therefore, the best way to compute $\vec{K}^+$ is SVD so singular values w_i less than a tolerance λ are set to zero [72]. This allows us to deal with ill-conditioning (or singularity); i.e., the condition number, which is the ratio between the largest and the smallest singular value, can be decreased.

4.2.2 Kernel Parameters

In general, in order to use Gaussian kernels (4.8) in generalised linear regression (4.4)-(4.7), the kernel positions c_j, as well as kernel widths ω_j, need to be determined [86]. Several approaches have been taken in the literature. For instance, those who use radial basis function (RBF) networks employ e.g., k-means clustering, or EM algorithm and Gaussian mixture modelling [36, 35][3]. We have explored two approaches to kernel positioning:

1. the centres c_j uniformly distributed across the input range and
2. the centres c_j positioned at input samples t_j, $j = 1, 2, ..., n$.

The latter approach leads to superior performance and the results reported in this thesis were obtained using kernels centred at observation times t_j.

As for the kernel widths, we propose two approaches:

1. fixed width ω, and
2. variable widths ω_j, where $j = 1, 2, ..., n$.

Both are described in the following subsections.

[2]note that ill-conditioning can be also due to roundoff error [72], not only to noise and missing data.

[3]Some approaches attempt to simultaneously optimise the number of kernels.

Fixed Kernels Width

The width of the kernels determines the degree of smoothing for the underlying flux curves (4.6) and (4.7). Finding 'appropriate' values of smoothing parameters is one of the challenges in non- and semi-parametric regression. We use cross-validation [35](§7.10) to find the optimal kernel width ω. In particular, we invoke a variant of five-fold-cross-validation. We start by dividing the data set uniformly into five blocks. In the first step, we construct a validation set V as a collection of the first elements of each block, where V has five elements. The training set $T = A - V$ is formed by the remaining observations; i.e., the observations not included in the validation set, where A is the set of all observations with cardinality $|A| = n$; each observation can be represented by a triple $(t_i, x_A(t_i), x_B(t_i))$. We fit our models on the training set T and determine the mean square error (MSE) over a range of delay values Δ_t on the validation set V. In the next step, we construct a new validation set as a collection of the second elements of each block. The new training set is again formed by the remaining observations. As before we fit our models on the training set and determine MSE on the validation set. We repeat this procedure r times, where r is the number of observations in each block. Finally, the mean of all such mean square errors (there are r of them), MSE_{CV}, is calculated. The kernel width ω selected using the cross-validation is the kernel width yielding the smallest MSE_{CV}. This scheme is summarised in Algorithm 4.1.

Variable Kernels Width

Rather than considering a fixed kernel width ω, here we allow variable width Gaussian kernels of the form

$$K(c_j, t_i) = \exp \frac{-|t_i - c_j|^2}{\omega_j^2} \; ; \; K(c_j + \Delta, t_i) = \exp \frac{-|t_i - (c_j + \Delta)|^2}{\omega_j^2}.$$

We determine each ω_j through a smoothing parameter $k \in \{1, 2, ..., k_{max}\}$. Parameter k is the number of neighbouring observations t_i on both sides of c_j (boundary conditions need to be taken into account). In particular, since we centre a kernel on each observation time, i.e., $c_j = t_j$, we have the cumulative kernel width

Algorithm 4.1: Cross-Validation

```
/* Bounds for ω and M must be found separately; T = A − V */
Fix M, LowerBound and UpperBound ;                                  // O(1)
Fix Blocks ← 5 ;                                                    // O(1)
Fix PointsPerBlock ← min({b_1, n − b_2})/Blocks ;                   // O(1)
for λ ∈ {10^{−1}, 10^{−2}, ..., 10^{−6}} do
    for ω ← LowerBound to UpperBound do
        for l ← 1 to PointsPerBlock do
            Remove the l^{th} observation of each block and include it in the
            validation set V ;                                      // O(n)
            for Δ_t ← Δ_min to Δ_max do
                Obtain weights α⃗ on T, Eq. (4.17) ;                 // O((n − 5)^3)
                Compute h_A(t_u) and h_B(t_v) on T ;                // O((n − 5)^2)
                Obtain MSE_CV on the validation set V;              // O(5)
                S(Δ_t) ← MSE_CV ;                                   // O(1)
            R(l) ← mean(S)
        Best(ω, λ) ← mean(R)
ω ← argmin_{ω,λ}(Best)
```

$$\omega_j = \sum_{d=1}^{k}(t_j - t_{j-d}) + (t_{j+d} - t_j) = \sum_{d=1}^{k}(t_{j+d} - t_{j-d}). \tag{4.18}$$

The optimal value of k can be estimated using a cross-validation procedure analogous to Algorithm 4.1.

4.3 Time Complexity

This analysis is based on asymptotic notation [15], specifically on the O-notation which is an upper bound. In other words, we are interested on the order of growth of the running time of an algorithm. Because we are looking at the input size (of training data) to find the upper bound of the running time (time complexity), we are studying the algorithm efficiency [15].

From our formulation in §4.2, hereafter, we will refer to two methods: K-F and K-V. That is, K-F corresponds to Gaussian kernels centred at observations with fixed width, and K-V has variable width. Both methods use Algorithm 4.1 to estimate their parameters, ω and k, respectively.

Returning to our model formulation (4.4)–(4.12), we are interested in the time delay Δ between a pair of time series. Algorithm 4.2 illustrates both methods K-F and K-V as pseudocode with the running times as upper bounds $O(\cdot)$. Now, let us analyse the running times in line 1, which come from Algorithm 4.1. We assume that $\omega \mapsto L \subset \Re$ (line 5) and $\Delta_t \mapsto D \subset \Re$ (line 8), with cardinality $n_\omega = |L| \sim n$ and $n_{\Delta_t} = |D| \sim n$, respectively; line 4 is $O(6)$ and line 6 is $O(n/5)$. Therefore, the complexity of K-F is

$$\begin{aligned} O(n^6) &= 6n_\omega \tfrac{n}{5} n_{\Delta_t}[(n-5)^3 + (n-5)^2 + 5 + 1] \\ O(n^6) &= n_\omega \tfrac{n}{5} n_{\Delta_t}(n^3 + n^2), \end{aligned} \tag{4.19}$$

and for K-V is

$$\begin{aligned} O(n^5) &= 6 \times 15 \tfrac{n}{5} n_{\Delta_t}[(n-5)^3 + (n-5)^2 + 5 + 1] \\ O(n^5) &= \tfrac{n}{5} n_{\Delta_t}(n^3 + n^2), \end{aligned} \tag{4.20}$$

Because $\omega \equiv k$, one evaluates $k = 1, 2, ..., 15$ only; i.e., $O(15)$. In that sense, the parameter estimation of K-V is cheaper than K-F.

Returning to Algorithm 4.2 with the above assumption for Δ_t, the upper bound of running time of K-F is $O(n^6) = n^6 + n_{\Delta_t}(n^3 + n^2 + n)$, and for K-V is $O(n^5) = n^5 + n_{\Delta_t}(n^3 + n^2 + n)$. Note that n_ω and n_{Δ_t} may be greater than n, so the above time complexities may be underestimated. But, if n_ω and n_{Δ_t} are constants and much lower than n, then the complexities reduce to $O(n^4)$ for both K-F and K-V, where $\Omega(n^4)$ is the lower bound of running time. Therefore, the critical parts are at line 9 in Algorithm 4.1 and at line 3 in Algorithm 4.2; i.e., the SVD inversion which is $O(n^3)$ [28, 72, 67].

Algorithm 4.2: Time Delay Estimation: K-F or K-V

Input: $\{t_i|i\}$, $\vec{x_A}$, $\vec{x_B}$; i.e., A

Output: $\hat{\Delta}$

1 Fix parameters either ω or k by Algorithm 4.1 ; // $O(n^6)$ or $O(n^5)$

2 **for** $\Delta_t \leftarrow \Delta_{min}$ **to** Δ_{max} **do**

3 Obtain weights $\vec{\alpha}$ on all observations A, Eq. (4.17) ; // $O(n^3)$

4 Compute $h_A(t_u)$ and $h_B(t_v)$; // $O(n^2)$

5 Obtain $Q(\Delta_t)$ via (4.12) ; // $O(n)$

6 $\hat{\Delta} \leftarrow \text{argmin}_{\Delta_t}(Q)$

4.4 Chapter Summary

We have introduced some concepts of machine and statistical learning such as supervised learning and regression; see §4.1, §4.1.1 and §4.1.2. We aimed to provide the context of our kernel-based method. Therefore, in §4.1.2 the concept of kernel and its background are described, which are the preamble to our method. Our methodology for time delay estimation was presented in §4.2, including regularisation and Gaussian width estimation as our automatic model selection technique. Finally, §4.3 shows the analysis of our methodology in terms of running time or time complexity.

Chapter 5

Evolved Kernel-Based Method

WE start introducing the evolutionary computation terminology, then we present an evolutionary algorithm for time delay estimation. This approach is based on kernel methods, presented in the previous chapter. We use two types of representation: reals and mixed types (reals and integers). In this chapter, we also come up with a variation of the regularisation approach in §4.2.1.

5.1 Introduction

The algorithm to be presented here is an evolutionary algorithm (EA) [87, 80], which performs artificial evolution. This kind of algorithms belong to a new growing area, natural computation [95]. Several evolutionary computational models have been proposed: *i*) genetic algorithms [37, 27], *ii*) evolutionary programming [24, 23], *iii*) evolution strategies [85] and *iv*) genetic programming [47] mainly. Other recent approaches have been introduced, but they are variants or improvements of the above evolutionary algorithms; e.g., see [39, 48].

These models share the common ingredient in evolution, that is the evolutionary operators such as selection, recombination and mutation. Some use either recombination or mutation, others put more emphasis on specific operators, but all of them are inspired by natural evolution.

The EA proposed here comes from genetic algorithms (GAs) with real and integer

representation; the typical representation in GAs is binary. Moreover, we compare it with an evolutionary strategy, (1+1)ES [77]; see the next section.

5.2 Evolution Strategies

Typically an evolutionary strategy (ES) uses a single parent and an offspring; this is denoted as (1+1)ES. Therefore, the population size is one, and the selection is between the parent and the offspring. The latter is generated via mutation. General versions[1] of ES are (γ, ρ)ES and ($\gamma + \rho$)ES, where γ denotes the number of parents and ρ the number of children with $\rho > \gamma$. The first method replaces parents with the best γ children. The second method allows the best parents and children to survive by keeping the parent population of size γ. Thus, ($\gamma + \rho$)ES is elitist, but (γ, ρ)ES is not.

To compare with our EA, we use a continuous optimisation approach, (1+1)ES [77], which is based on the Gray-code neighbourhood distribution and uses real representation. Rowe and Hidovic [77] have shown superior performance of their (1+1)ES over Improved Fast Evolutionary Programming (IFEP) on some benchmark problems and on a real-world problem (medical tissue optics). IFEP is also a continuous optimisation approach [94].

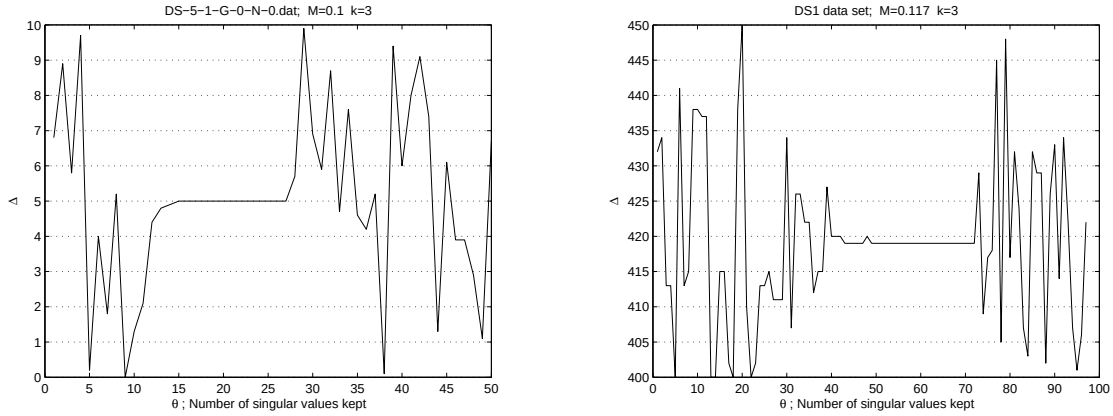
5.3 Evolutionary Algorithm

We evolve the parameters of our kernel-based formulation in §4.1.2, and a different regularisation approach is used.

5.3.1 Regularisation

In §4.2.1 we introduced a regularisation procedure to invert $\vec{K}$ in (4.17); i.e., the threshold λ tells us how many singular values to set to zero. Thus, for a given Δ the amount of singular values to keep may vary. We illustrate this through Fig. 5.1. We can see a well defined pattern in the range θ= [49, 72] ($\Delta = 419$), where θ is the

[1] These are referred as (μ, λ)ES and ($\mu + \lambda$)ES, but this conflicts with our notation.



(a) (b)

Figure 5.1: Patterns on DS-5 and DS1. In each relation (Δ, θ) the best time delay has been plotted. The best time delay is found through K-V method, Eq. 4.12. **(a)** DS-5-1-G-0-N-0. This data set has no noise and no gaps (see §2.4.2); $\Delta = [0, 10]$ with increments of 0.1; $M = 0.1$ and $k = 3$. The pattern is at $\theta = [5, 27]$, where $\Delta = 5$ (true value). **(b)** DS1: 0957+561 optical data at g-band (see §2.3.2); $\Delta = [400, 450]$ with unitary increments; $M = 0.117$ and $k = 3$. The pattern is at $\theta = [49, 72]$, where $\Delta = 419$.

number of singular values to set to zero. Thus, if one can find a proper λ that falls in this range, then one can claim that the estimation of Δ is "robust". But the range of this pattern may change for other M and k parameters. Then there is no warranty that the estimated λ falls in this range. Moreover, whatever method for goodness of fit is used, if we test Δ in a specific range with a fixed λ, we may come up with different θ – some inside the pattern, some outside, none inside, etc.

Rather, we use θ as a regularisation parameter. In fact, EA aims to perform as an automatic algorithm with global search, through all parameters, so it finds the proper θ that falls in the pattern.

A review of other general regularisation techniques for inverse problems can be found in Cowan [17] and Haykin [36].

5.3.2 Representation

Following our kernel-based approach in §4.1.2, we have three parameters: i) the time delay Δ, ii) the variable width k and iii) the amount of singular values to keep θ. Besides, we have the fitting measurement; e.g., log-likelihood or any loss function. This give us a third-dimensional search space Ψ. We follow an EA to avoid local minima [5, 27, 87] because one does not know anything about Ψ; i.e., its shape.

We define as our population

$$\vec{P_1} = \left[\begin{array}{ccc|c} \Delta_1 & \theta_1 & k_1 & f_1 \\ \Delta_2 & \theta_2 & k_2 & f_2 \\ ... & ... & ... & ... \\ \Delta_x & \theta_x & k_x & f_x \\ ... & ... & ... & ... \\ \Delta_{n_p} & \theta_{n_p} & k_{n_p} & f_{n_p} \end{array}\right] \tag{5.1}$$

where each row in $\vec{P_1}$ is a hypothesis commonly referred as individual or chromosome, which is a set of parameters $\{\Delta_x, \theta_x, k_x\}$ initialised randomly. Then we have n_p hypotheses [54]. Each hypothesis x is evaluated by f_x that is a measure of fitness pointing the best hypothesis out. Then, we apply artificial genetic operators such as selection, crossover, mutation and reinsertion (elitist strategy) to generate $\vec{P_2}, ..., \vec{P_{n_g}}$ populations. At the n_g generation, we choose from $\vec{P_{n_g}}$ the best set of parameters (or individual) according to its fitness; i.e., with minimum f_x. This process leads to artificial evolution, which is a stochastic global search and optimisation method based on the principles of biological evolution [27, 87].

For mixed types, we represent every population $\vec{P_1}$ to $\vec{P_{n_g}}$ as two linked populations of the same size n_p, $\vec{P_1} = [\vec{P_1^1}\ \vec{P_1^2}]$. Hence, $\vec{P_1^1}$ uses reals to represent Δ_x, and $\vec{P_1^2}$ employs integers to represent θ_x and k_x.

Hereafter, we define two EAs: *i*) EA-M, that uses mixed types and *ii*) EA-R, that uses real representation only. Therefore, we perform two kinds of flooring for integers: in population and in fitness function (default).

We employ a population size of $n_p = 300$ individuals and $n_g = 50$ generations

unless other values are given. We use the Genetic Algorithm Toolbox[2] for MATLAB [12, 11].

5.3.3 Fitness Function

We use as a measure of fitness (or objective function): *i*) negative log-likelihood (LL) and *ii*) cross-validation (CV). The first one is given by Eq. 4.12. For the latter, the mean squared error (MSE) is given by CV as described in Algorithm 5.1, where $T = A - V$ is the training set; A is the set of all observations, and V is the validation set.

These objective functions are the same regardless of the representation used. To denote representation and fitness function, we add the suffix CV or LL to either EA-M or EA-R (e.g., EA-M-CV and EA-R-LL); EA refers simply to any of these combinations. When only reals are used, a step is added to floor reals to integers before obtaining the fitness for each individual. In the following chapter, we will show the performance of both fitness functions on artificial data.

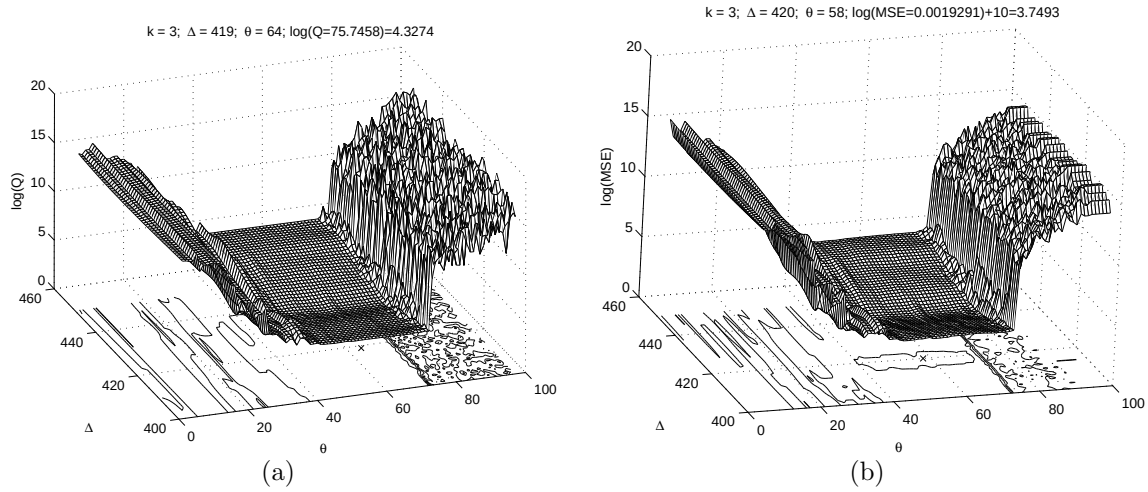
Algorithm 5.1: Fitness Function (A, Δ_x, k_x, θ_x)

```
/* A is the set of all observations; its cardinality is n      */
Fix Blocks ← 5 ;                                                // O(1)
Fix PointsPerBlock ← n/Blocks ;                                 // O(1)
for l ← 1 to PointsPerBlock do
    Remove the l^th observation of each block and include it in the validation
    set V ;                                                     // O(5)
    Compute h_A and h_B for the training set T = A − V ;        // O((n − 5)^3)
    Obtain MSE_CV on the validation set V ;                     // O(5)
    R(l) ← MSE_CV ;                                             // O(1)
f_x ← mean(R) ;                                                 // O(n/5)
return f_x
```

[2]which is available online with a good documentation

5.3.4 Fitness Landscape

In Figure 5.2 are shown the fitness landscapes (search space) of the above fitness functions. To reduce the dimensions (for visualisation), we fixed k to 3 in both cases. We use $\Delta_{min} = 400$ and $\Delta_{max} = 450$ with unitary increments, and $\theta = 1, 2, ..., n$, where $n = 97$. Because DS1 is the cleanest data set, we use it to show the landscape; see §2.3.2. Therefore, one expects that if the noise increases the error surfaces become worse; e.g., more local minima. We can see that from $\theta = 80$ to n the error surface is quite complicated for simple search algorithms; e.g., gradient descent or hill climbing search regardless of the fitness function. There are also more local minima when $\theta < 45$. In the θ-Δ plane is a mark (x) showing the best parameter combination; i.e., minimum Q or MSE_{CV}. To smooth the surface, we use a logarithmic scale for both fitness functions.



(a) (b)

Figure 5.2: Fitness Landscape. **(a)** LL fitness function landscape. **(b)** CV fitness function landscape; the surface is shifted upwards by 10 units for visualisation. See §5.3.4 for details.

5.3.5 Evolution Operators

Selection

We use the basic roulette wheel selection method for both $\vec{P_1^1}$ and $\vec{P_1^2}$, which is stochastic sampling with replacement. Consequently, this is a mechanism to probabilistically select individuals from a population based on their fitness function. The higher the fitness value, the larger the interval in the wheel [12]. From the initial population, we select half of the population so we work with this population during recombination, mutation and evaluation (see Algorithm 5.2). Finally, we reinsert the best individuals to the initial population to obtain a population of size n_p for the next generation. In other words, we perform reinsertion of offsprings [12].

Other selection methods were tested such as tournament selection and stochastic universal sampling, but roulette wheel selection gave us the better results on artificial and real data.

Recombination

Four methods for recombination (or crossover) have been tested: discrete, intermediate, linear and double-point recombination. The last one is only for integer representation. They all lead to similar results on DS1 and on some artificial data sets so we adopt linear recombination for reals and double-point for integers.

Linear recombination can generate an offspring on a slightly longer line than that defined by its parents. Whereas $o = p_1 + \alpha \times (p_2 - p_1)$, such as $\alpha = U[-0.25, 1.25]$ is uniformly distributed, and o is the offspring with parents p_1 and p_2 [12].

Double-point recombination involves selecting uniformly at random two integer positions to exchange the variables in those positions. Typically this method is used for binary representation, but integers also can be used [12].

Mutation

For reals, we tested two methods for mutation: Gaussian mutation and *mutbga* (as in Breeder Genetic Algorithm [11, 12]). Both lead to similar performance. Therefore, we adopt *mutbga* as our mutation operator; whereas a mutated variable can be obtained

by $m_M = v + s_1 \times r \times s_2 \times \delta^M$, where m_M is the mutated variable, v is the variable to mutate, $s_1 = \pm 1$ with a probability given by a mutation rate in the range [0,1], $r = 0.5 \times d$ (d is the domain of variable) and $\delta^M = \sum_{i=0}^{m-1} \alpha_i^M 2^{-i}$; $\alpha_i^M = 1$ with probability $1/m$, else $0 - m = 20$. For integers, we use 0.5 as mutation rate.

5.4 Literature Review

This section reviews some recent papers related to Kernel methods and EA across the evolutionary computation community. They evolve either only the weights or all the parameters, and they are mainly applications for classification problems.

GK SVM uses Genetic Programming (GP) to evolve a kernel for a Support Vector Machine (SVM) classifier [38]. This approach chooses a Polynomial, Gaussian or Sigmoid kernel in order to deal with the problem of kernel selection, that is to find the proper kernel for a given data set. Howley and Madden [38] present a good review of methods which evolve the complexity parameter and the width of Gaussian kernels in SVM via either GP or Evolutionary Strategies (ES). However, this approach has been tested only on classification problems, and GP explores a huge search space making it impractical for use with many real problems, in particular for large data sets.

Another implementation of SVM claims that a single Gaussian kernel $K(\cdot, \cdot)$ is replaced by a linear combination of Gaussian kernels. Then ES evolve the new weights and widths per Gaussian in such a combination [68]. However, the SVM parameters are not evolved and the optimisation procedure to obtain the support vectors is not specified.

A GA/SVM approach to the selection and classification of high dimensional DNA Micro-array data uses a Genetic Algorithm (GA) to select genes [6]. The fitness function is the classification rate given by a SVM, where the SVM parameters are estimated experimentally.

The above references differ from ours in different ways: i) most of them deal with SVM, i.e., convex optimisation rather than eigen-decomposition [86]; ii) They deal neither with regression nor with time series.

5.5 Time Complexity

First, the running time of our main fitness function CV (Algorithm 5.1) is

$$\begin{aligned} O(n^4) &= 1+1+\tfrac{n}{5}(5+(n-5)^3+5+1)+\tfrac{n}{5} \\ O(n^4) &= \tfrac{n}{5}(n^3); \end{aligned} \tag{5.2}$$

the critical part is at line 5 of Algorithm 5.1, where SVD is involved. The time complexity of LL fitness function is $O(n^3)$, where the weights $\vec{\alpha}$ are obtained by pseudo inverse; i.e., SVD.

Second, let us find the overall running time of our EAs. If we assume that $n_p \sim n$ and $n_g \sim n$, then the time complexity for EA-CV[3] is (see Algorithm 5.2)

$$\begin{aligned} O(n^6) &= n_p + n_p n^4 + n_g(n_p + 0.5n_p + 0.5n_p + 0.5n_p n^4 + (n_p + 0.5n_p)) \\ O(n^6) &= n^5 + n(n^5), \end{aligned} \tag{5.3}$$

and for EA-LL is

$$\begin{aligned} O(n^5) &= n_p + n_p n^3 + n_g(n_p + 0.5n_p + 0.5n_p + 0.5n_p n^3 + (n_p + 0.5n_p)) \\ O(n^5) &= n^4 + n(n^4). \end{aligned} \tag{5.4}$$

With the above assumptions, EA-CV is as costly as K-F, and EA-LL as K-V. However, if one considers to n_g as constant since $n_g = 50$, the time complexity is reduced to $O(n^5)$ and $O(n^4)$ for EA-CV and EA-LL, respectively, where EA-CV is cheaper than K-F and similar to K-V. But, EA-LL is cheaper than both K-F and K-V.

Moreover, if we treat $n_p = 300$ as constant rather than $n_p \sim n$, then the lower bound of EA-CV is $\Omega(n^4)$ and for EA-LL is $\Omega(n^3)$; this is, because in practice n_p is fixed. The lower bound of EA-CV is similar to K-F and K-V methods, but EA-LL bound is lower.

5.6 Chapter Summary

Through this chapter has been introduced the evolutionary computation terminology. It has also been presented an evolutionary strategy, (1+1)ES. Here, our evolved

[3]Regardless the representation, EA-M or EA-R.

Algorithm 5.2: Evolutionary Algorithm

```
/* for mixed types or real representation                    */
Initialise population ;                                      // O(n_p)
Evaluate population ;                                        // O(n_p n^3) or O(n_p n^4)
for generation ← 1 to n_g do
    Select ;                                                 // O(n_p)
    Recombine ;                                              // O(0.5 n_p)
    Mutate ;                                                 // O(0.5 n_p)
    Evaluate ;                                               // O(0.5 n_p n^3) or O(0.5 n_p n^4)
    Reinsert ;                                               // O(n_p + 0.5 n_p)
```

kernel-based approach is described. The parameters to evolve, representation, fitness functions and evolutionary operators have also been described across this chapter. The results of this evolutionary algorithm and our kernel methods are in the next chapter. Finally, we made a time complexity analysis of this evolutionary algorithm for two fitness functions, and we compared it with the time complexity of methods in the previous chapter.

Chapter 6

Experimental Results

THROUGH this chapter, we will show all main results from real and artificial data described in §2 and §3. We also present results from several methods introduced in §3.2, as well as results from our methodology introduced in §4 and §5. We stress that this is not an exhaustive analysis of all methods on all data sets, but we do compare our methods against all other methods, and we test them on all data sets; that is, real and artificial data. Note that the discussion on results is presented in the following chapter.

6.1 Artificial Data

Remember that the idea of generating synthetic data is that one knows the true time delay, therefore one can measure distinct statistics over the estimates and the true values, such as bias, variance, absolute error, mean squared error, t-test, etc. These statistics are presented in the Appendix A.

6.1.1 DS-500

These data are described in §2.4.1, the true time delay is 500 days for all the involved data sets. The ratio between image A and image B is fixed to its true value $M = 1/1.44$. Here, results from linear interpolation, DCF, LNDCF, Dispersion spectra,

PRH, K-F, K-V, EA-M-CV, EA-R-CV and EA-R-LL methods are presented; see §3.2, §4.2 and §5.3.

Experimental Set Up

To interpolate the light curves, we use linear interpolation with a time resolution of 0.01 days. Therefore, for any given t_i we can find the delayed version $t_i + \Delta$ of image A in image B.

For DCF and LNDCF, a bin size of 100 days is used, which is the average lag in these data. The best time delay is found by searching the maximum correlation between 400 and 600 days, where if there is not a bin containing a delay of 400 days then we search at the previous one.

A decorrelation length of $\delta = 100$ is used for Dispersion spectra method, $D^2_{4,2}$. This value give us accurate results on selected data sets (free noise cases). Time delay trials between $\Delta_{min} = 400$ and $\Delta_{max} = 600$ are generated with unitary increments. The ratio M is set to its true value 1/1.44.

When estimating the structure function of PRH method for each data set, we use bins in the range 100–700 days [32]. Linear regression is used to estimate the structure function given by $\mathcal{A}$ and $\mathcal{B}$ in (3.8). The image A (data) is only used to estimate the structure function so results by using both image A and image B are in our journal paper; see §1.5. To obtain $\vec{A}$ in (3.4), we use SVD as in §4.2.1 with $\lambda = 0.001$, because zero noise and duplicate times may occur leading to singularity. Consequently, fast methods to obtain $\vec{A}$ are not used [78]. When combining both images into $\vec{y}$ (see §3.2.3), we use time delay trials as above with the ratio $M = 1/1.44$.

The parameter setting of the Bayesian method is as in Harva and Raychaudhury [33, 34].

For K-F, we set the bounds of ω_c to $LowerBound = 900$ and $UpperBound = 1,200$ with increments of 10; see Algorithm 4.1. For K-V, we set $LowerBound = 1$ and $UpperBound = 15$ with increments of 1. For K-F and K-V, we use also time delay trials as above, and M is set to 1/1.44.

The Δ bounds for EA-M and EA-R are also as above; $k = [1, 15]$ and $\theta = [1, n]$.

Results

The results are shown in Figs. 6.1 and 6.2. Each plot is divided in two parts: Δ_μ (top) and Δ_σ (bottom), whereas Δ_μ is the mean of all estimates grouped by noise level and gap size, and Δ_σ is the standard deviation of those estimates; the standard deviation of all estimates of each underlying function is computed, then the mean over the five underlying functions. To be fair in the comparison, in each plot, the scale of y-axis is the same, apart from DCF and LNDF in Figs. 6.1b–c where their bounds are larger than others; otherwise, at the top of Fig. 6.1a and 6.2 we will see a straight line. We also point out that the time delay estimates were obtained exactly on the same collection of data.

Since the true delay is 500 days, the best solution is a straight line with zero slope at $\Delta_\mu = 500$. Therefore, Δ_μ is the bias of all the estimates. The uncertainty is measured by Δ_σ; i.e., the variance. In Figs. 6.1 and 6.2, there is a clear tendency to increase Δ_σ as the gap size increases. The same occurs when the noise level is increased. As expected, the lower the noise and gap size, the more accurate the results. The aim of this comparison is to find the method with low bias and low variance; i.e., high accuracy.

The results are interesting because linear interpolation has an outstanding performance compared with the most used methods: PRH method, D_1^2, $D_{4,2}^2$, DCF and LNDCF. Nevertheless, the best results are for our kernel-based methods: K-F, K-V, EA-M-CV and EA-R-CV. In Fig. 6.2, one can compare the performance of the distinct versions of our methodology. Comparing K-F and K-V, both lead to similar performance but K-V is less costly; see §4.3. Therefore, we adopt K-V as our best method of this class. Now, comparing our evolutionary algorithms: EA-R-LL, EA-M-CV and EA-R-CV. Those based on CV are more accurate than EA-R-LL. However, we adopt EA-M-CV as our best method because it uses a proper representation. Moreover, the difference in the performance for this class of algorithms is not significant regardless the representation.

In Fig. 6.2, the comparison is not easily seen for some methods graphically. Thus, we carried out a statistical analysis, see Appendix A, over all estimates in order to perform a quantitative analysis of results from the best methods in Figs. 6.1 and 6.2.

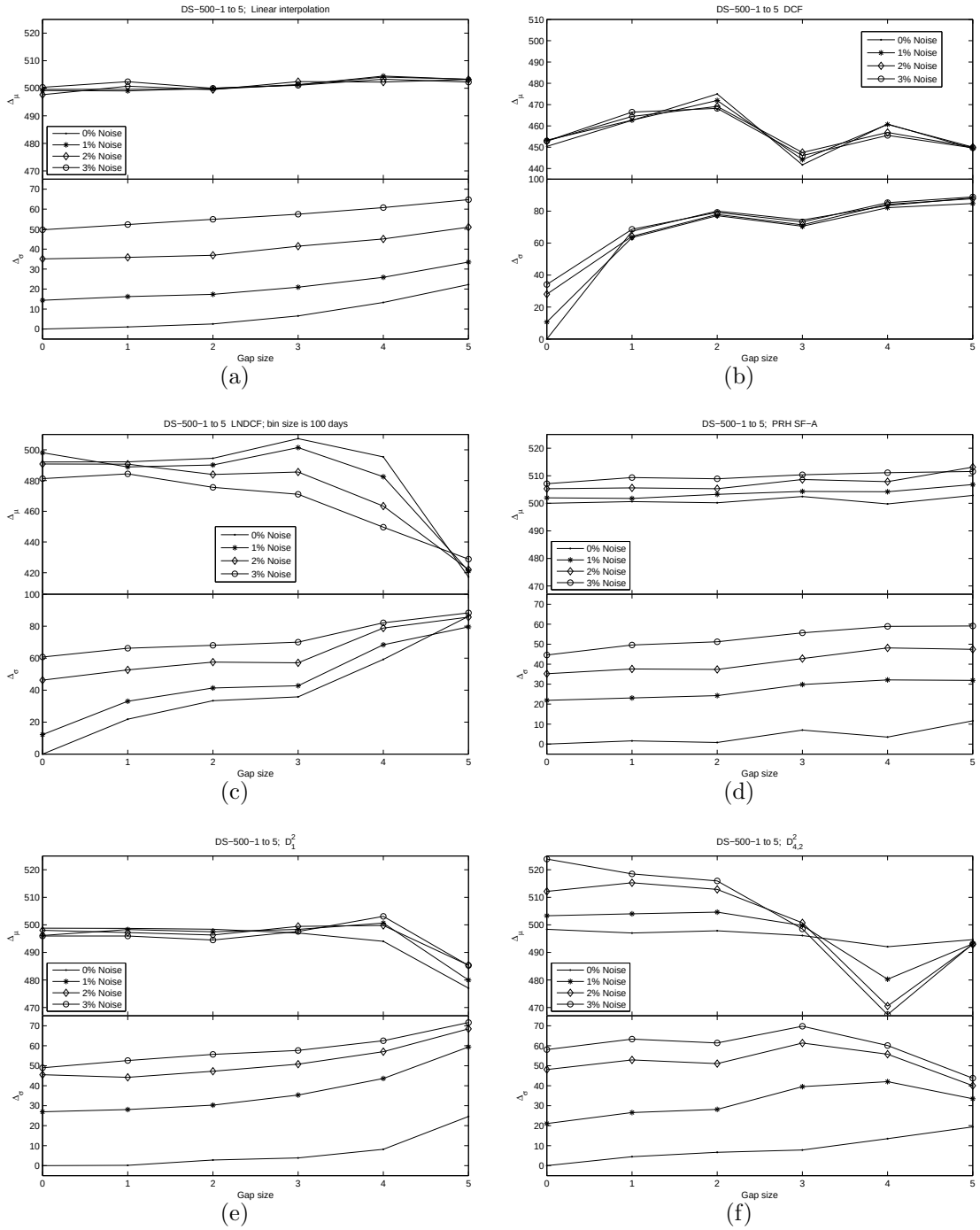


Figure 6.1: Results on DS-500 from: **(a)** Linear interpolation, **(b)** DCF, **(c)** LNDCF, **(d)** PRH method, **(e)** D_1^2 and **(f)** $D_{4,2}^2$; see §6.1.1 for details. Note that only **(b)**-**(c)** have different y-axis scale.

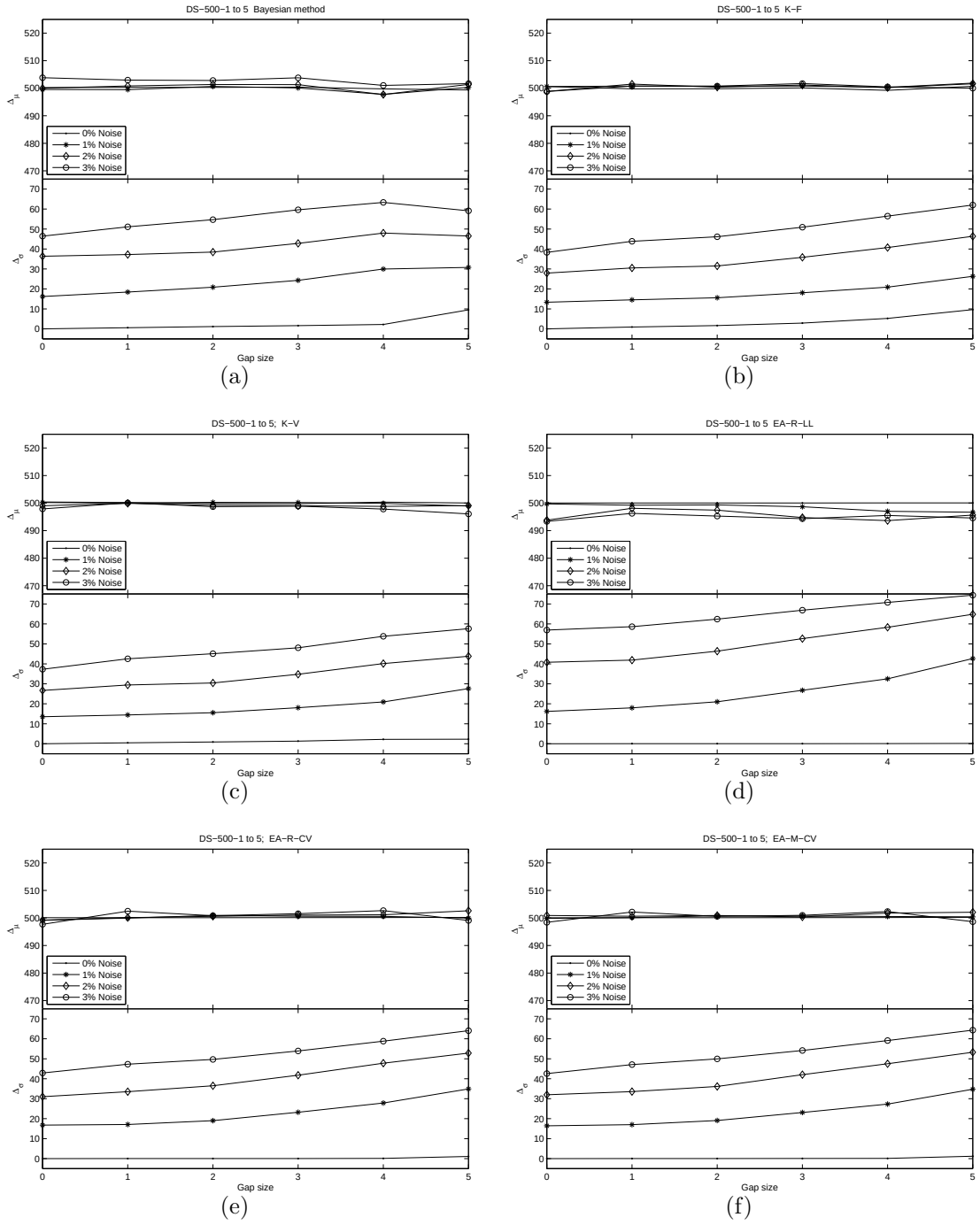


Figure 6.2: Results on DS-500 from: **(a)** Bayesian method, **(b)** K-F, **(c)** K-V, **(d)** EA-R-LL, **(e)** EA-R-CV and **(f)** EA-M-CV; see §6.1.1 for details.

Table 6.1: DS-500: Statistical Analysis of all the time delay estimates ($\eta = 38,505$).

Statistic	Interp[a]	D_1^2	$D_{4,2}^2$	PRH	K-V	EA-M-CV	Bayes[b]
95% CI	[501.1, 502.0]	[494.7, 495.8]	[497.5, 498.6]	[506.9, 507.8]	[498.7, 499.5]	**[500.4, 501.2]**	[500.4, 501.3]
CI range	0.90	1.05	1.07	0.88	**0.75**	0.79	0.89
MSE	1986.4	2803.6	2916.2	2019.3	**1417.4**	1905.1	2017.6
AE	33.8	43.9	43.6	34.0	**27.9**	33.11	33.19
$\hat{\mu}$	501.62	495.30	498.12	507.38	499.13	**500.83**	500.87
$\hat{\sigma}$	44.5	52.7	53.9	44.3	**37.6**	43.6	44.9

[a] linear interpolation; [b] Bayesian method.

We grouped estimates regardless of level of noise, gap size and underlying function, therefore, the number of degrees of freedom is 38,504 ($\eta = 38,505$). The results are in Table 6.1 from selected methods. The aim of this analysis is to present various ways to measure the performance of methods. The 95% confidence interval (CI) is shown in the first three rows. MSE is the mean squared error (A.3), and AE is the mean absolute error (A.5). The estimators $\hat{\mu}$ and $\hat{\sigma}$ are the mean and standard deviation (A.1)-(A.2), respectively. In bold fonts are highlighted the best results. Again, the details of these statistical estimators are in Appendix A.

Figure 6.3 only shows the 95% CI for the first underlying function (DS-500-1) with 0% of noise rather than grouping all time delay estimates as in Table 6.1. Then, we count the number of cases where the true delay ($\Delta = 500$) falls within the interval. Consequently, we obtain five cases for Linear interpolation method (shaded points) and one case for Bayesian estimation method (circles). In Table 6.2 are summarised the quantity of cases by following the above procedure for all the underlying functions, all levels of noise and five selected methods. The best results are in bold fonts.

We also performed the t-test on time delay estimates from five selected methods where the hypothesis to test is H_0: $\mu_0 = 500$ (the true delay); see the Appendix A. The results are shown in Fig. 6.4, where the estimates are grouped by underlying function, level of noise and gap size. Since $\mathcal{T}$ (A.6) follows a Student's t-distribution, which is centred at zero, those values close to zero are statistically significant [1, 18, 3]. The

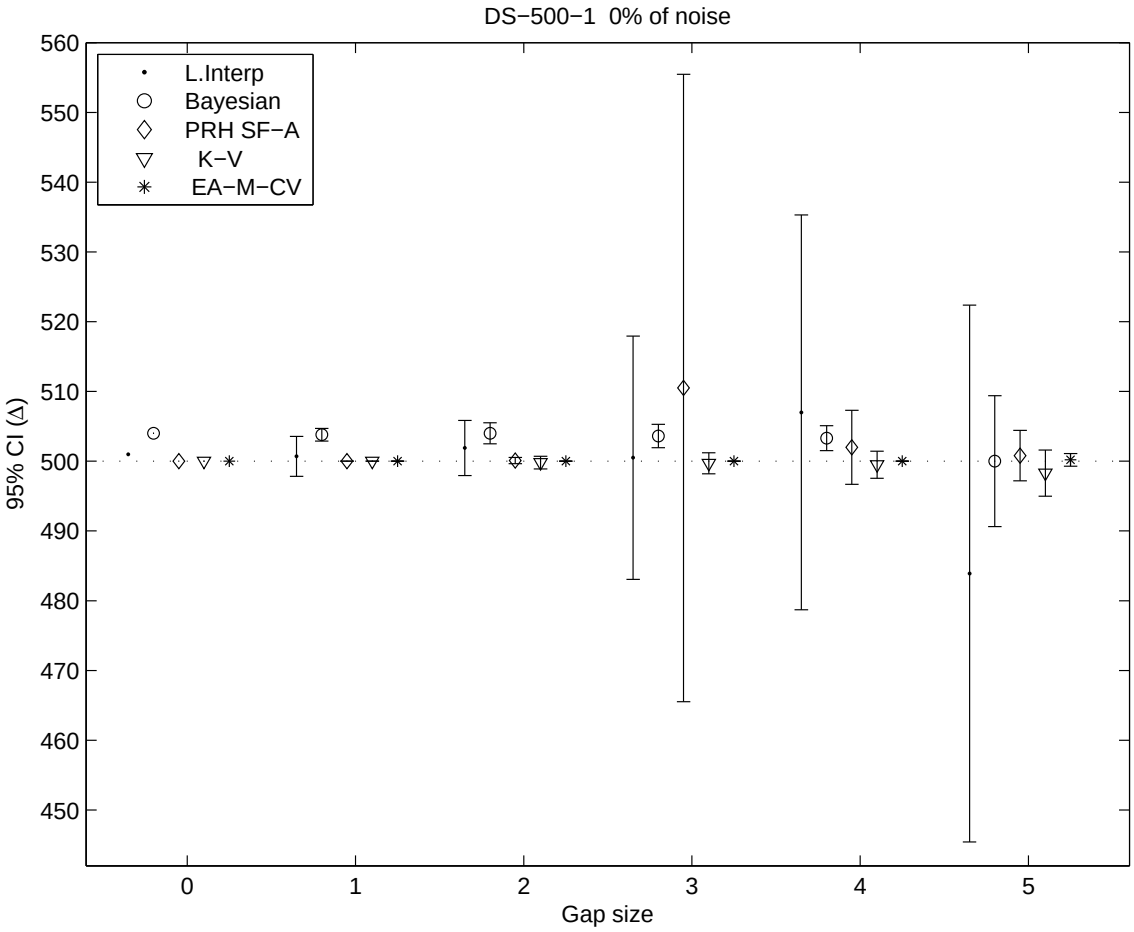


Figure 6.3: 95% CI on DS-500-1-N-0. The intervals correspond to five methods and grouped at each gap size. Only time delay estimates on the first underlying function (DS-500-1) and 0% of noise are shown. See §6.1.1 for details.

horizontal dotted line shows the threshold for a significance level of 95%, $\alpha = 0.05$; i.e., when $\mathcal{P} < \alpha$, where $\mathcal{P}$ (A.7) denotes the cumulative probability from a Student's t-distribution. Thus, the threshold values for $|\mathcal{T}|$ in Fig. 6.4 are 2.2, 2 and 1.9 for $\nu = \{9, 49, 499\}$, degrees of freedom, respectively; see Table 2.3.

In Table 6.3 are shown the quantity of cases that satisfy the above threshold values. The results are grouped by noise level, and the best ones are highlighted with bold fonts.

In Fig. 6.5 are shown the results of MSE, where the estimates are grouped as above. The AE statistic gives similar results; see Fig. 6.6. Figures 6.7 and 6.8 show

Table 6.2: 95% CI on DS-500. Quantity of cases that are within the 95% Confidence Intervals per method and level of noise.

	Noise Level			
Method	0%	1%	2%	3%
Linear interpolation	21	13	20	22
Bayesian estimation	17	17	17	17
PRH	**28**	15	11	11
K-V	25	**24**	**28**	24
EA-M-CV	**28**	**24**	27	**26**

See §6.1.1 for more details.

Table 6.3: t-test on DS-500. The quantities mean the number of cases that are significant (95%).

	Noise Level			
Method	0%	1%	2%	3%
Linear interpolation	19	13	20	22
Bayesian estimation	16	16	17	16
PRH	17	15	10	11
K-V	**21**	**24**	**28**	24
EA-M-CV	5	23	27	**25**

See §6.1.1 for more details.

the results from $\hat{\mu}$ and $\hat{\sigma}$ statistics respectively.

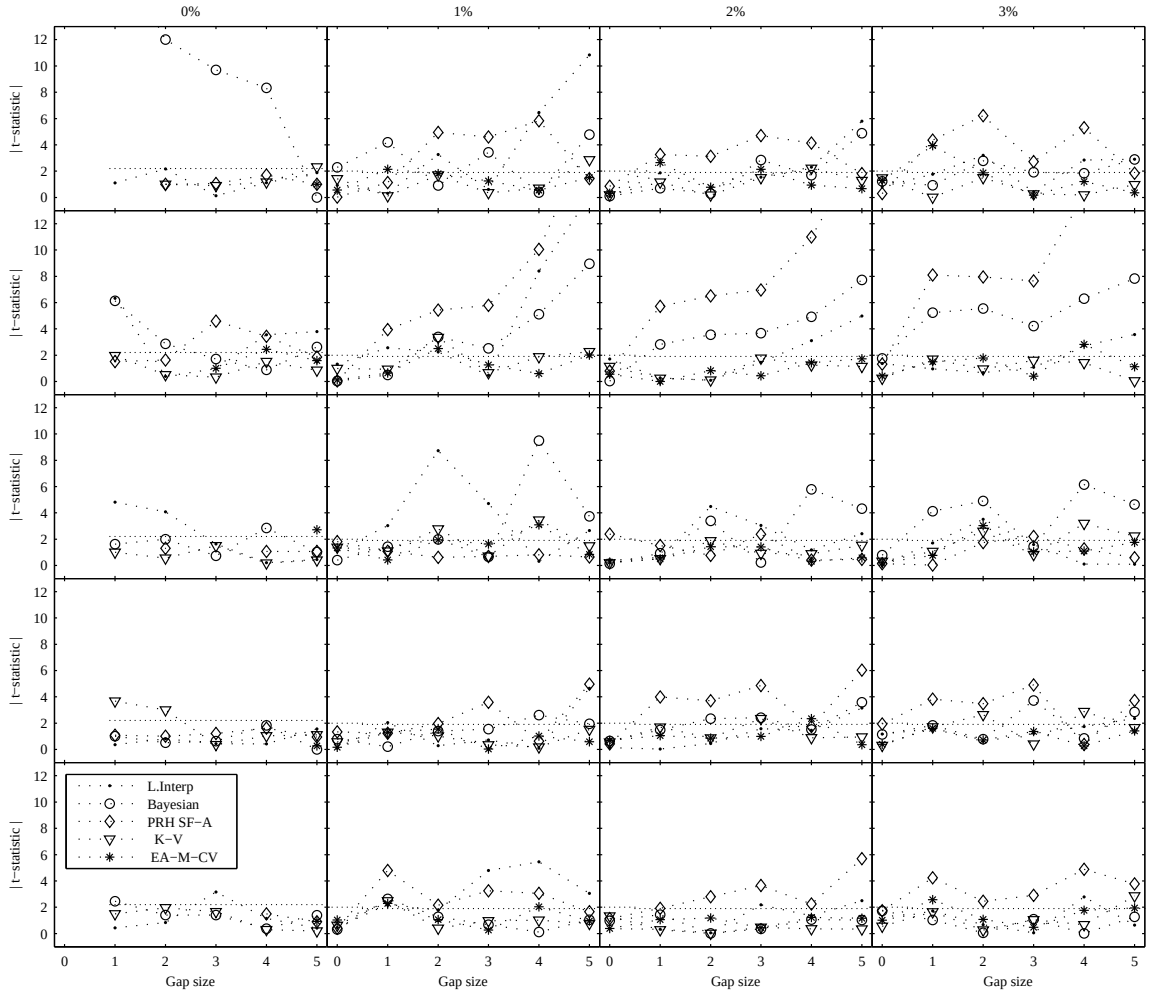


Figure 6.4: t-test on DS-500. Each row corresponds to a different underlying function (DS-500-1, DS-500-2,..., DS-500-5), and each column corresponds to a different level of noise (0%, 1%, 2% and 3%). Every plot shows the results of $|\mathcal{T}|$ from five methods; i.e., Linear interpolation, Bayesian estimation, PRH, K-V and EA-M-CV; shaded point, circle, diamond, triangle and asterisk respectively. Note that all the plots have the same scale at y-axis. The horizontal dotted lines show the threshold for 95% confidence level; see §6.1.1 for details.

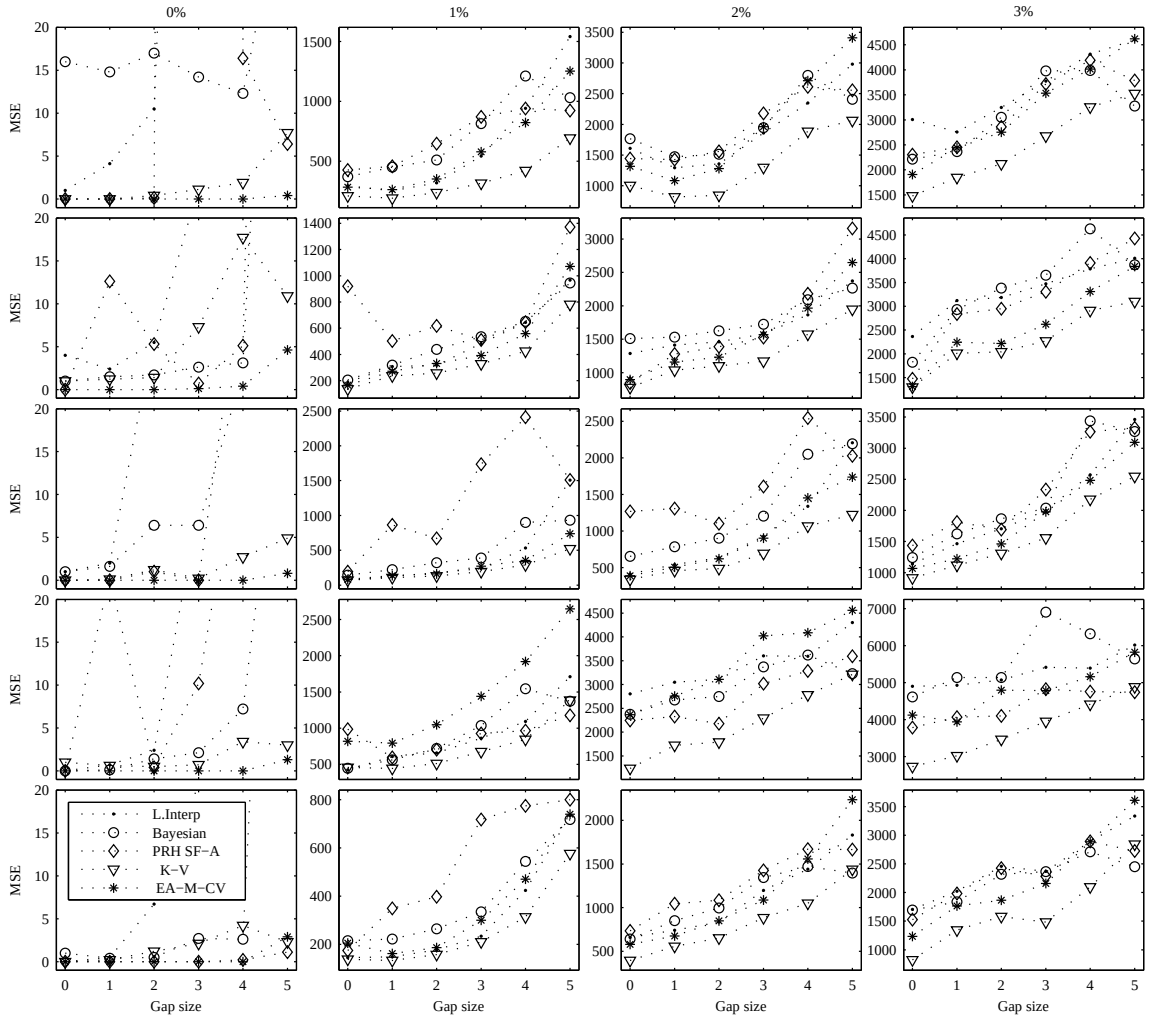


Figure 6.5: MSE on DS-500. Each row corresponds to a different underlying function (DS-500-1, DS-500-2,..., DS-500-5), and each column corresponds to a different level of noise (0%, 1%, 2% and 3%). Every plot shows the results of MSE statistic from five methods; i.e., Linear interpolation, Bayesian estimation, PRH, K-V and EA-M-CV; shaded point, circle, diamond, triangle and asterisk respectively. For the 0% noise column, these plots show values of MSE in the range of 0–20 only.

6.1.2 DS-5

On these data, we show only results from the following methods: Linear Interpolation, D_1^2, $D_{4,2}^2$, PRH method (SF from image A), K-V and EA-M-CV. We choose the two versions of dispersion spectra and PRH method because they are the most popular

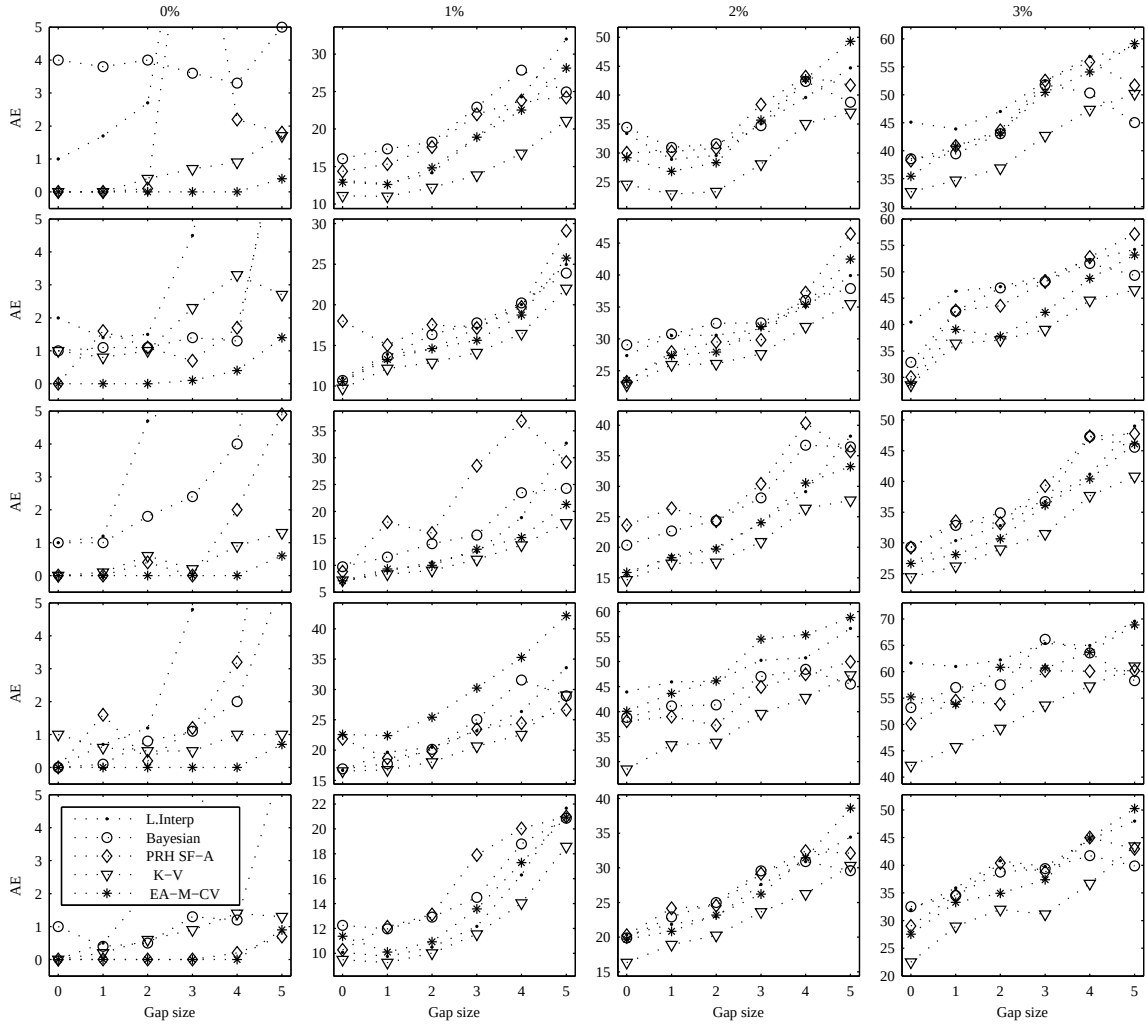


Figure 6.6: AE on DS-500. Each row corresponds to a different underlying function (DS-500-1, DS-500-2,..., DS-500-5), and each column corresponds to a different level of noise (0%, 1%, 2% and 3%). Every plot shows the results of AE statistic from five methods; i.e., Linear interpolation, Bayesian estimation, PRH, K-V and EA-M-CV; shaded point, circle, diamond, triangle and asterisk respectively. For the 0% noise column, these plots show values of AE in the range of 0–5 only.

(see §3.1), moreover, these methods have been used to analyse optical data (low noise) [59, 13, 21, 70], and DS-5 data simulate optical data. As our methods, we select K-V and EA-M-CV only.

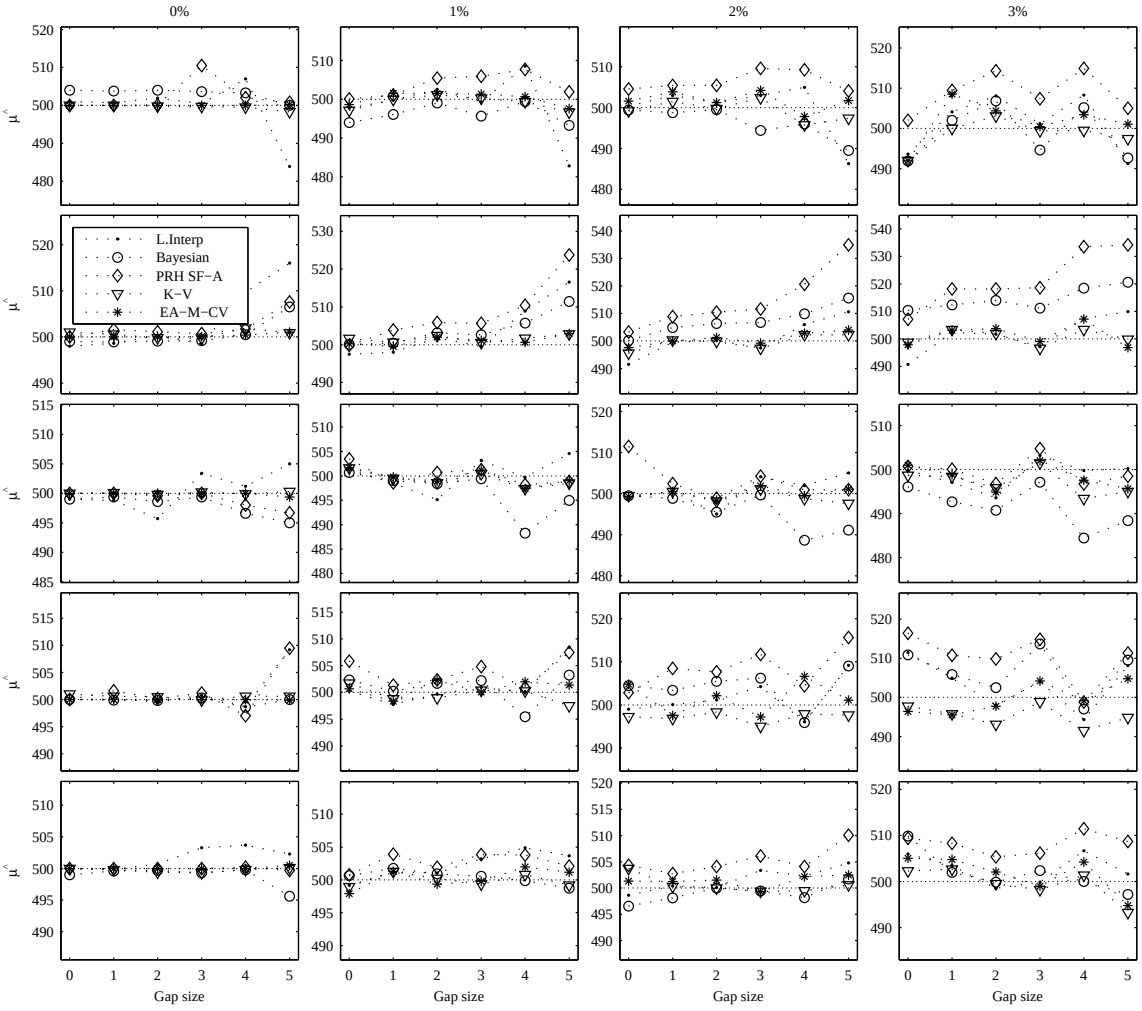


Figure 6.7: $\hat{\mu}$ on DS-500. Each row corresponds to a different underlying function (DS-500-1, DS-500-2,..., DS-500-5), and each column corresponds to a different level of noise (0%, 1%, 2% and 3%). Every plot shows the results of $\hat{\mu}$ statistic from five methods; i.e., Linear interpolation, Bayesian estimation, PRH, K-V and EA-M-CV; shaded point, circle, diamond, triangle and asterisk respectively.

Experimental Set Up

For all methods, time delay trials Δ_t are generated between $\Delta_{min} = 0$ and $\Delta_{max} = 10$ with increments of 0.1, and the offset M is fixed to its true value 0.1. For linear interpolation, we use a time resolution of 0.01 days to interpolate the light curves. A decorrelation length of $\delta = 5$ is used for Dispersion spectra method, $D^2_{4,2}$. To

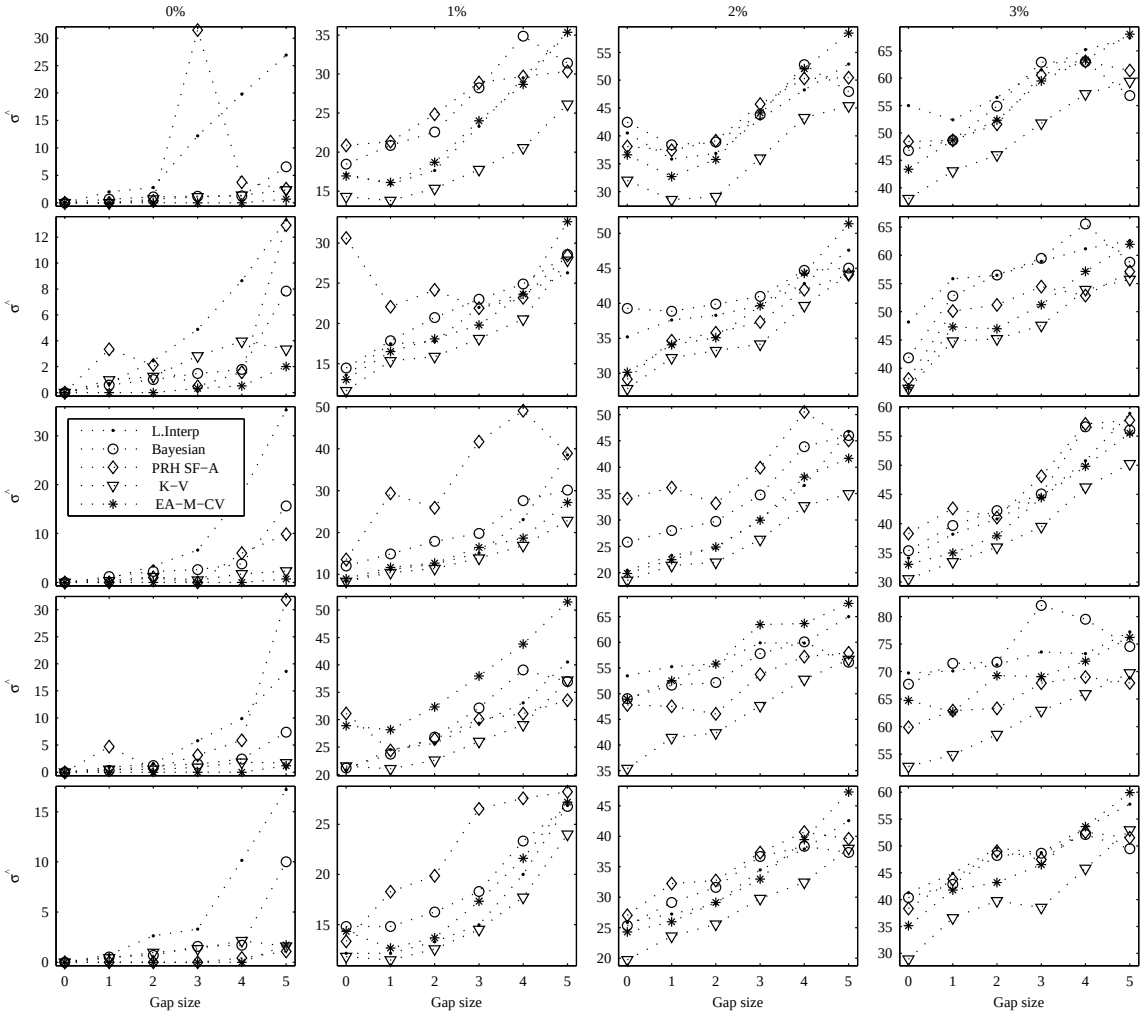


Figure 6.8: $\hat{\sigma}$ on DS-500. Each row corresponds to a different underlying function (DS-500-1, DS-500-2,..., DS-500-5), and each column corresponds to a different level of noise (0%, 1%, 2% and 3%). Every plot shows the results of $\hat{\sigma}$ statistic from five methods; i.e., Linear interpolation, Bayesian estimation, PRH, K-V and EA-M-CV; shaded point, circle, diamond, triangle and asterisk respectively. For the 0% noise column, these plots show values of $\hat{\sigma}$ in the range of 0–5 only.

estimate the structure function of PRH method, we use bins in the range 0–10 days. Linear regression is also used to estimate the structure function given by $\mathcal{A}$ and $\mathcal{B}$ in (3.8). The image A (data) is only used to estimate the structure function, and $\vec{A}$ in (3.4) is obtained as in previous section. For K-V, we set $LowerBound = 1$ and

Table 6.4: DS-5: Statistical Analysis of all the time delay estimates $\eta = 38,505$.

Statistic	Interp[a]	D_1^2	$D_{4,2}^2$	PRH	K-V	EA-M-CV
95% CI	[5.06, 5.07]	**[5.00, 5.02]**	[5.58, 5.59]	[2.67, 2.73]	[4.94, 4.95]	**[5.00, 5.02]**
CI range	**0.01**	0.02	**0.01**	0.06	**0.01**	0.02
MSE	0.49	0.74	0.99	13.46	**0.47**	0.63
AE	**0.39**	0.52	0.59	3.01	**0.39**	0.41
$\hat{\mu}$	5.068	**5.013**	5.589	2.704	4.946	5.015
$\hat{\sigma}$	0.70	0.86	0.80	2.86	**0.68**	0.79

[a] linear interpolation method.

$UpperBound = 15$ with increments of 1. The bounds for EA-M-CV are $\Delta = [0, 10]$, $k = [1, 15]$ and $\theta = [1, n]$.

Results

The results are shown in Table 6.4 and Fig. 6.9. The statistics are given in Appendix A, where $\mu_0 = 5$ and $\eta = 38,505$. Again, the best results are highlighted with bold fonts. In Fig. 6.9 are only the results from linear interpolation, D_1^2, K-V and EA-M-CV. Note that the y-axis scale is the same on all plots.

Table 6.5 shows the quantity of cases where the true delay ($\Delta = 5$) falls within the 95% confidence interval (CI); see §6.1.1 and Fig. 6.3 for more details.

Table 6.6 shows the results from t-test. As above, in Table 6.6 are the number of cases that satisfy the 95% confidence threshold on $|\mathcal{T}|$ values; see Fig. 6.10 and §6.1.1.

From Table 6.4 and Fig. 6.9, the best results are for K-V, EA-M-CV, Linear Interpolation and D_1^2. Since the noise is about 0.01 mag ($< 0.106\%$) in real optical data, one is interested on results by level of noise. Furthermore, in Table 6.7 are the results of MSE, AE, $\hat{\mu}$ and $\hat{\sigma}$ on all estimates, which are grouped by noise level regardless the gap size and underlying function. The best results are also in bold fonts.

Figures 6.11-6.14 show the results from MSE, AE, $\hat{\mu}$ and $\hat{\sigma}$ statistics on DS-5

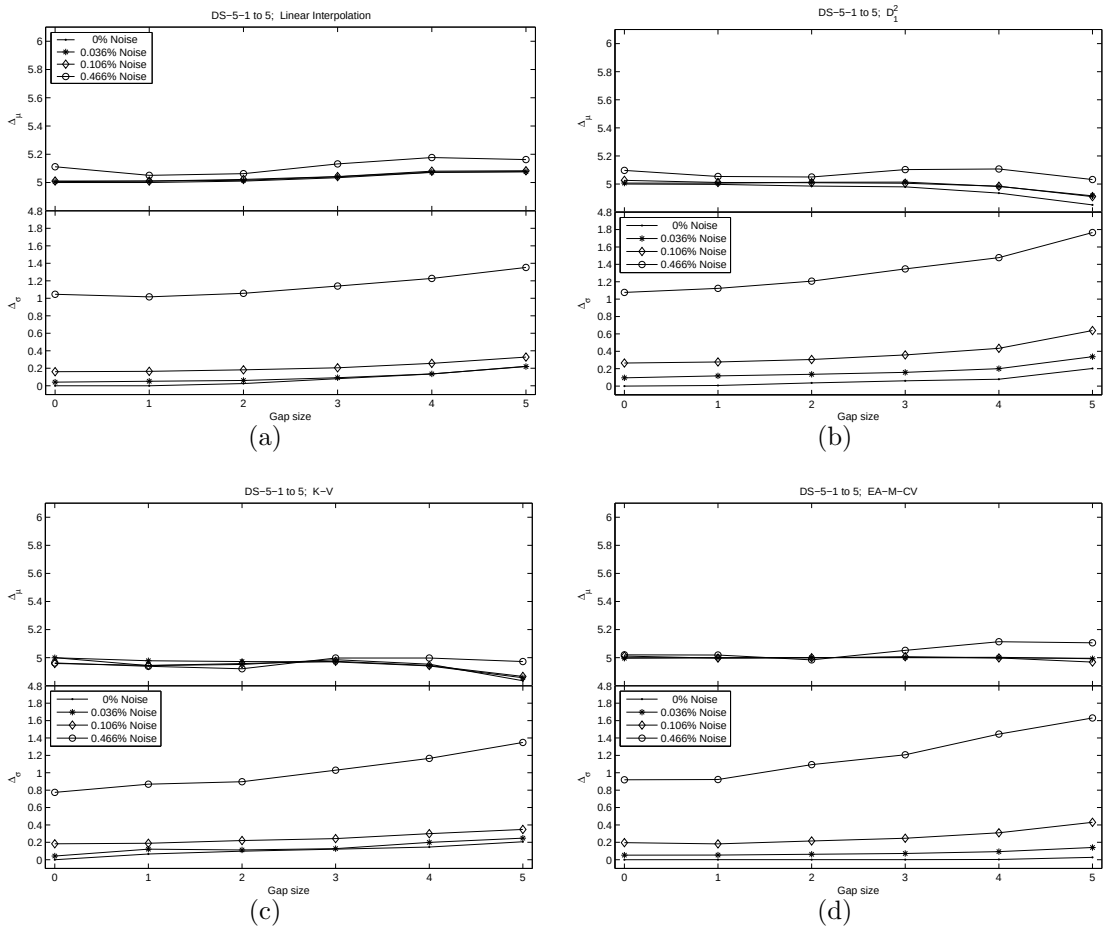


Figure 6.9: Results on DS-5 from: **(a)** Linear Interpolation, the range for Δ_μ is 0.176, and the maximum Δ_σ is 1.35; **(b)** D_1^2, where the range for Δ_μ is 0.255, and the maximum Δ_σ is 1.76; **(c)** K-V, where the range for Δ_μ is 0.164, and the maximum Δ_σ is 1.34; and **(d)** EA-M-CV, where the range for Δ_μ is 0.144, and the maximum Δ_σ is 1.62. See §6.1.2 for details.

respectively. The results are also grouped by underlying function and noise level.

6.1.3 PRH Data

Here, we compare only PRH method against EA-M-CV since the data is generated by PRH methodology (see §3.2.3). Therefore, the method to beat is PRH. In fact, we do the comparison with the PRH method by fixing the PRH parameters to those

Table 6.5: 95% CI on DS-5. Quantity of cases that are within the 95% Confidence Intervals.

	Noise Level			
Method	0%	0.036%	0.106%	0.466%
Linear Interpolation	24	6	13	17
D_1^2	23	14	22	20
$D_{4,2}^2$	12	4	0	0
PRH	0	0	0	6
K-V	19	6	6	13
EA-M-CV	**27**	**23**	**25**	**22**

See §6.1.2 for more details.

Table 6.6: t-test on DS-5. Quantities mean the number of cases that are significant, at 95% level. The results are grouped by noise level.

	Noise Level			
Method	0%	0.036%	0.106%	0.466%
Linear Interpolation	12	6	13	17
D_1^2	10	13	21	20
$D_{4,2}^2$	6	1	0	0
PRH	0	2	14	16
K-V	11	5	6	13
EA-M-CV	**22**	**23**	**24**	**22**

See §6.1.2 for more details.

Table 6.7: DS-5: Results Grouped by Noise Level

	Noise Level			
Statistic	0%	0.036%	0.106%	0.466%
Method: Linear Interpolation				
MSE	0.020	0.023	**0.063**	1.417
AE	0.076	0.093	**0.186**	0.915
$\hat{\mu}$	5.03	5.04	5.04	5.11
$\hat{\sigma}$	0.13	0.14	0.24	1.18
Method: D_1^2				
MSE	0.017	0.044	0.182	2.014
AE	0.060	0.147	0.321	1.121
$\hat{\mu}$	4.95	4.98	4.98	5.07
$\hat{\sigma}$	0.12	0.20	0.42	1.41
Method: K-V				
MSE	0.029	0.041	0.084	**1.312**
AE	0.117	0.139	0.219	**0.833**
$\hat{\mu}$	4.93	4.94	4.93	**4.96**
$\hat{\sigma}$	0.11	0.13	**0.21**	**0.83**
Method: EA-M-CV				
MSE	$\mathbf{1.9 \times 10^{-4}}$	**0.008**	0.090	1.831
AE	$\mathbf{4.7 \times 10^{-3}}$	**0.066**	0.216	0.984
$\hat{\mu}$	**4.99**	**4.99**	**4.99**	5.05
$\hat{\sigma}$	**0.01**	**0.09**	0.30	1.35
η	255	12,750	12,750	12,750

values used to generate the data (idealised scenario). That is, the structure function (SF) to define the covariance matrix in PRH method is used in two ways: SF is fixed to its true value (SF*) and estimated following the PRH method (SF+). From our methods, we choose EA-M-CV because it gives good results on data with low noise (DS-5).

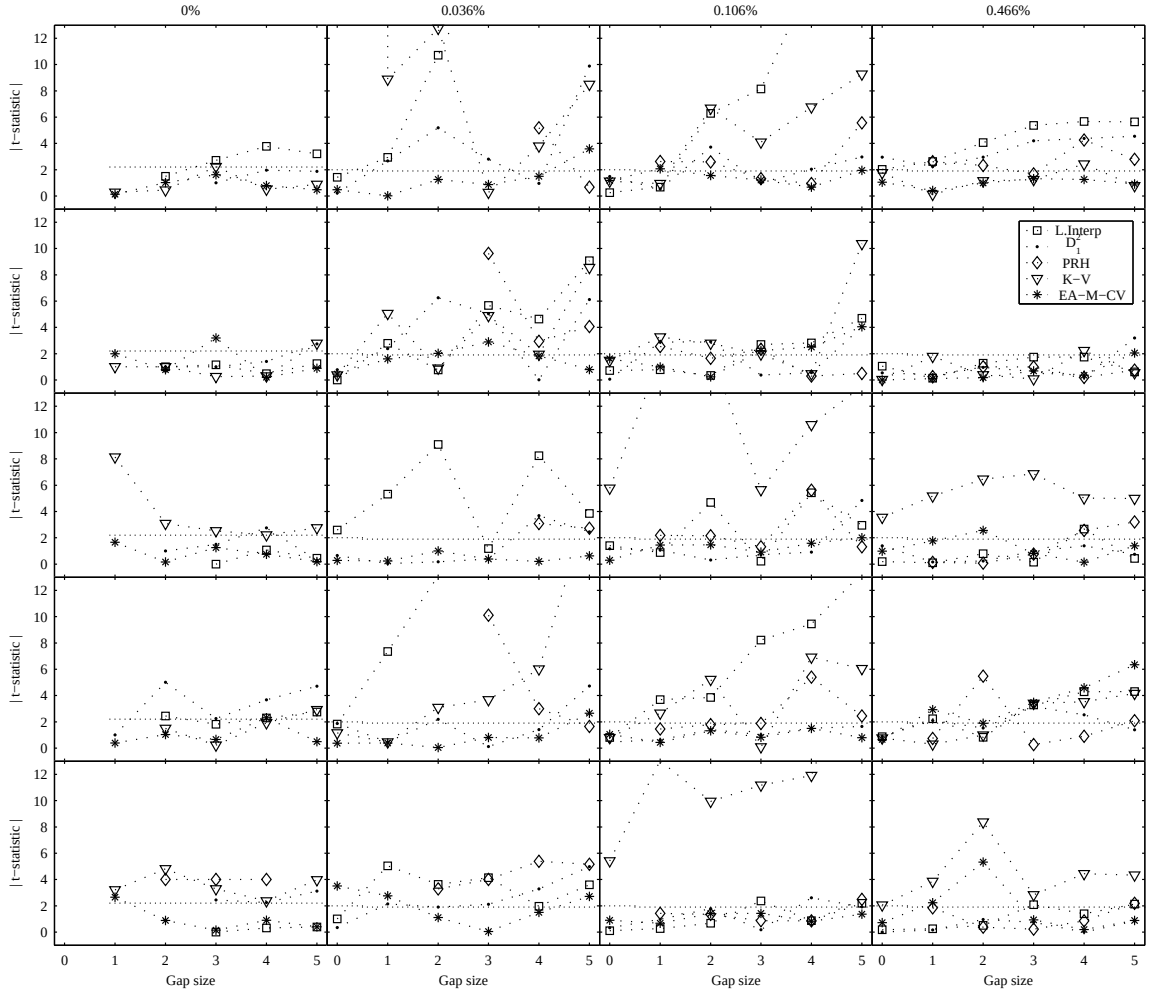


Figure 6.10: t-test on DS-5. Each row corresponds to a different underlying function (DS-5-1, DS-5-2,..., DS-5-5), and each column corresponds to a different level of noise (0%, 0.036%, 0.106% and 0.466%). Every plot shows the results of $|\mathcal{T}|$ from five methods; i.e., Linear Interpolation D_1^2, PRH, K-V and EA-M-CV; square, shaded point, diamond, triangle and asterisk respectively. See §6.1.2 for details.

Experimental Set Up

In all cases, we use bounds[1] of $\mu_0 \pm 30$ days with unitary increments during the time delay analysis. The measurement error is also fixed to its true value (variance of 1×10^{-7}) for all methods. The number of estimates per each true delay (μ_0) is

[1]These bounds are also used to estimate the structure function SF+ by PRH method.

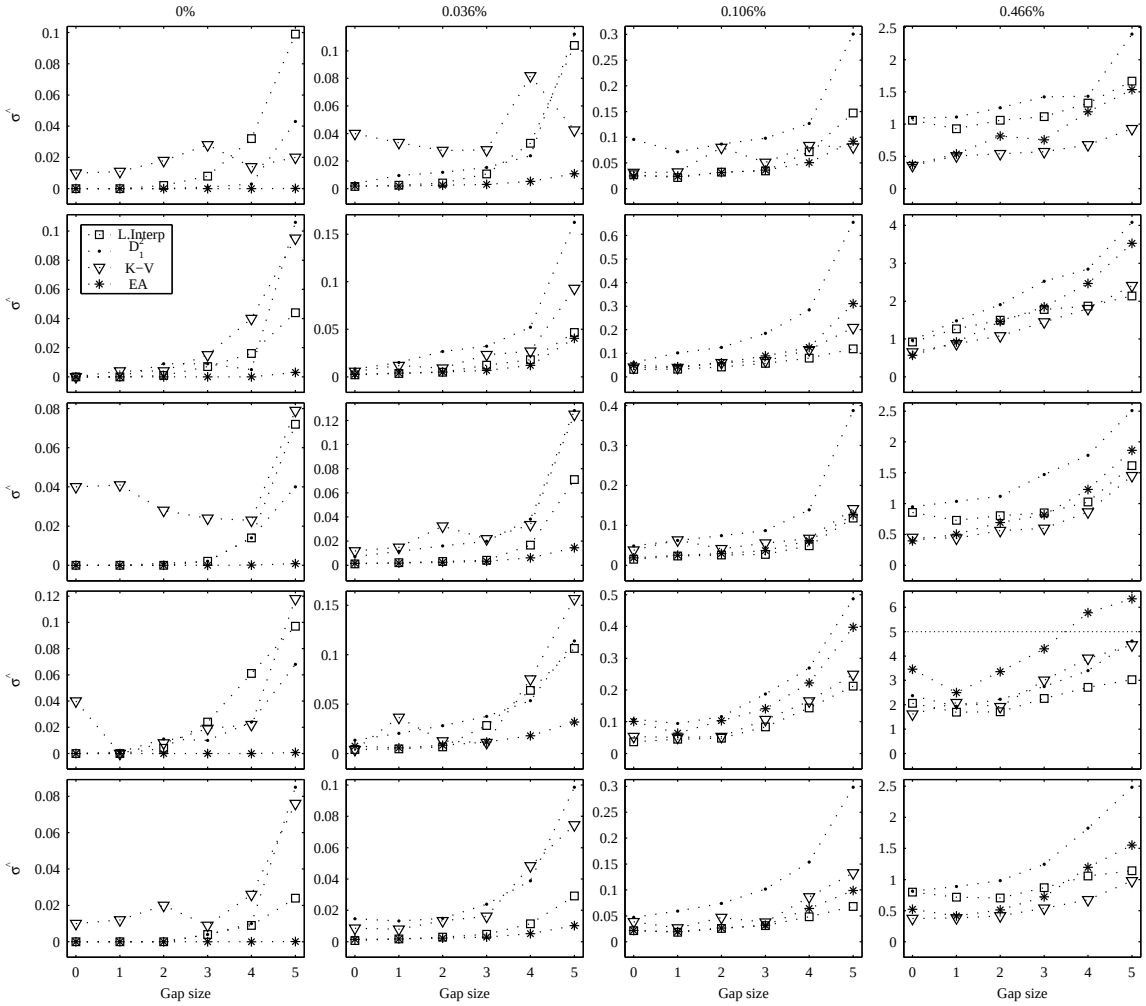


Figure 6.11: MSE on DS-5. Each row corresponds to a different underlying function (DS-5-1, DS-5-2,..., DS-5-5), and each column corresponds to a different level of noise (0%, 0.036%, 0.106% and 0.466%). Every plot shows the results of MSE statistic from four methods; i.e., Linear Interpolation, D_1^2, K-V and EA-M-CV; square, shaded point, triangle and asterisk respectively.

$\eta = 100$. The bounds for EA-M-CV are $\Delta = [\mu_0 - 30, \mu_0 + 30]$, $k = [1, 15]$ and $\theta = [1, n]$.

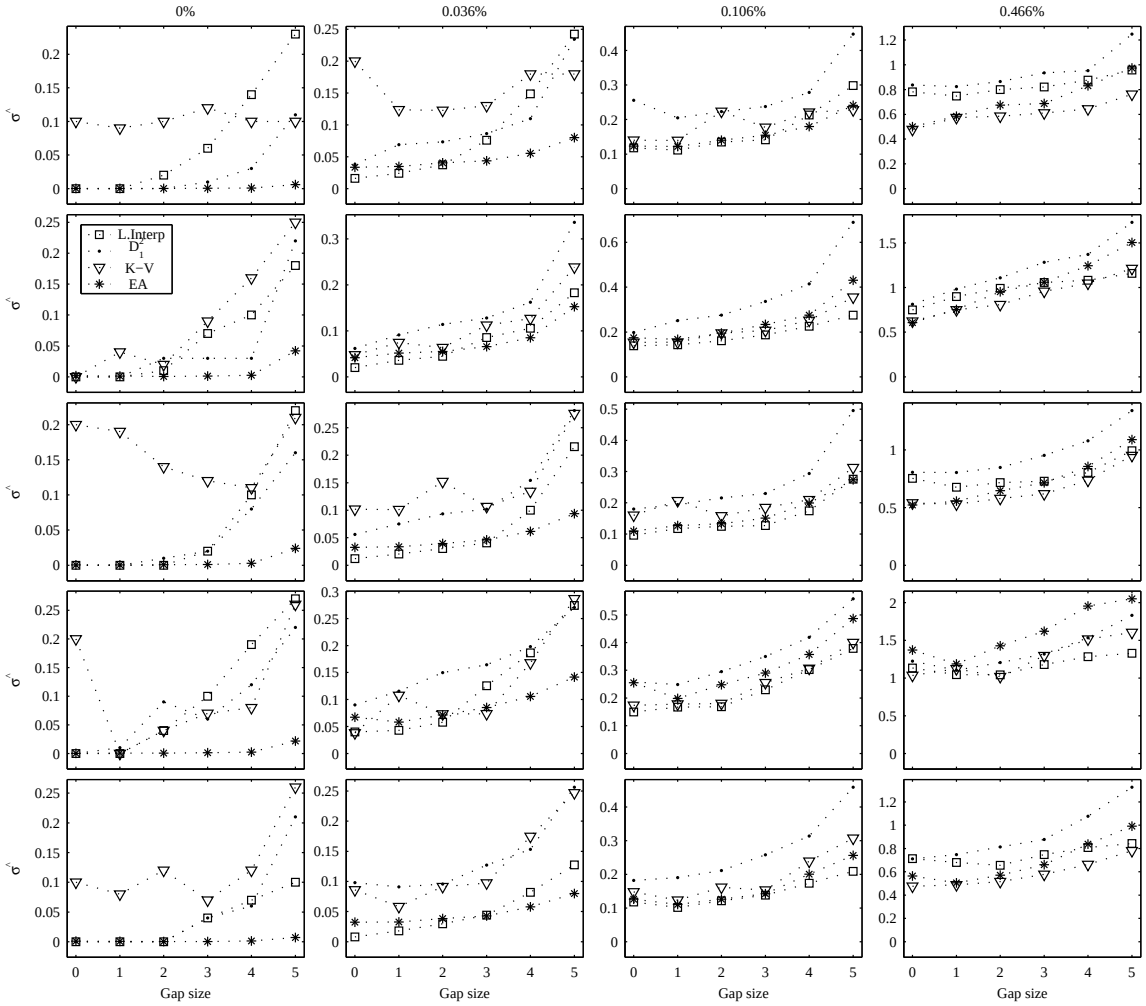


Figure 6.12: AE on DS-5. Each row corresponds to a different underlying function (DS-5-1, DS-5-2,..., DS-5-5), and each column corresponds to a different level of noise (0%, 0.036%, 0.106% and 0.466%). Every plot shows the results of AE statistic from four methods; i.e., Linear Interpolation, D_1^2, K-V and EA-M-CV; square, shaded point, triangle and asterisk respectively.

Results

The results from the PRH method, SF* case, are in Table 6.8. The column μ_0 denotes the true time delay, which is also our hypothesis in the t-test. The following columns are the statistics used in this analysis, where CI range is the range between the 95% confidence interval (CI) of $\hat{\mu}$; see the Appendix A for details. The last row is the

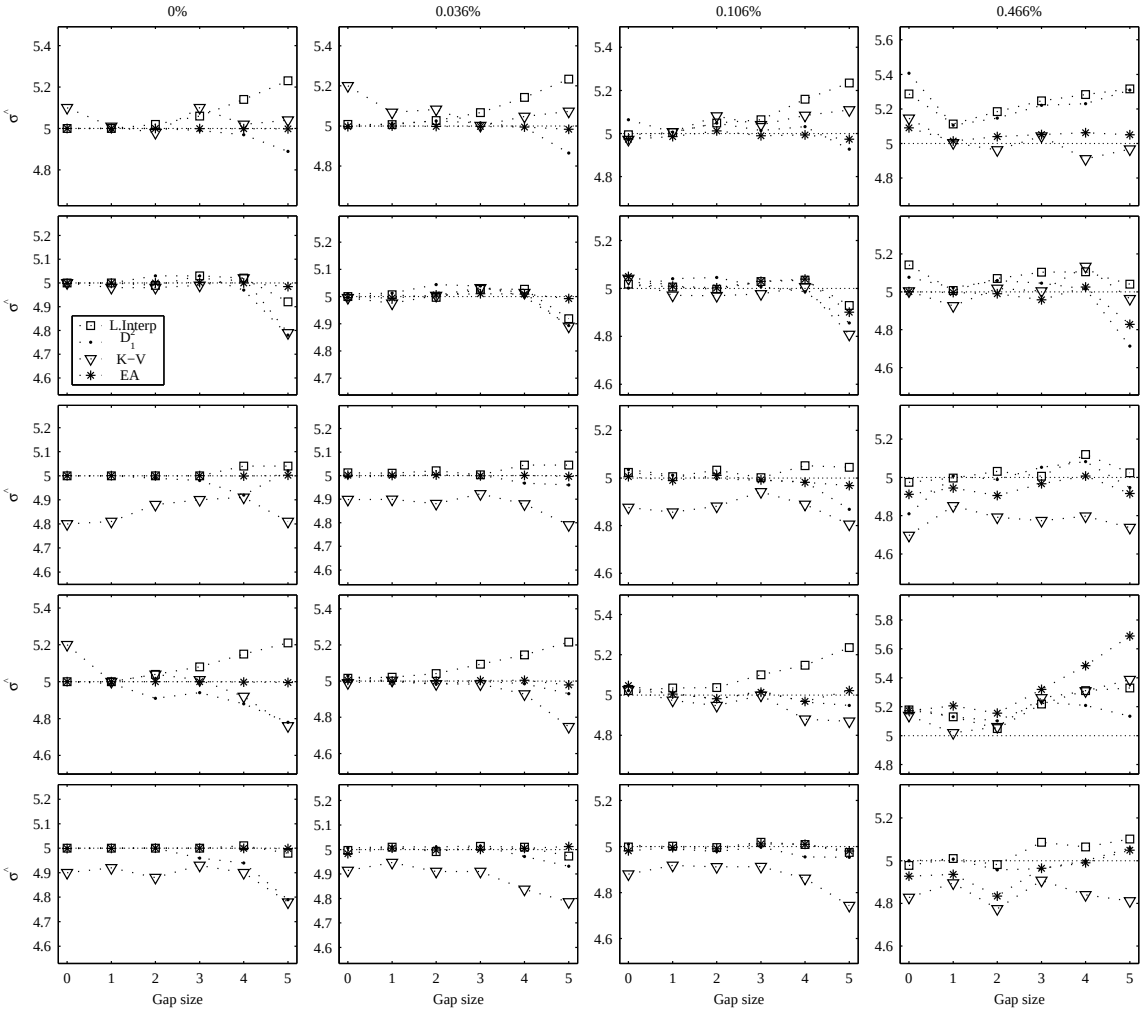


Figure 6.13: $\hat{\mu}$ on DS-5. Each row corresponds to a different underlying function (DS-5-1, DS-5-2,..., DS-5-5), and each column corresponds to a different level of noise (0%, 0.036%, 0.106% and 0.466%). Every plot shows the results of $\hat{\mu}$ statistic from four methods; i.e., Linear Interpolation, D_1^2, K-V and EA; square, shaded point, triangle and asterisk respectively.

average (Avg). The results from the SF+ case are in Table 6.9. Finally, the results from EA are in Table 6.10.

Strictly speaking, one should compare Table 6.9 with Table 6.10 because Table 6.8 has the results from SF*.

In Table 6.11 are the results from $\hat{\mu}$ and $\hat{\sigma}$ where one can measure the bias and

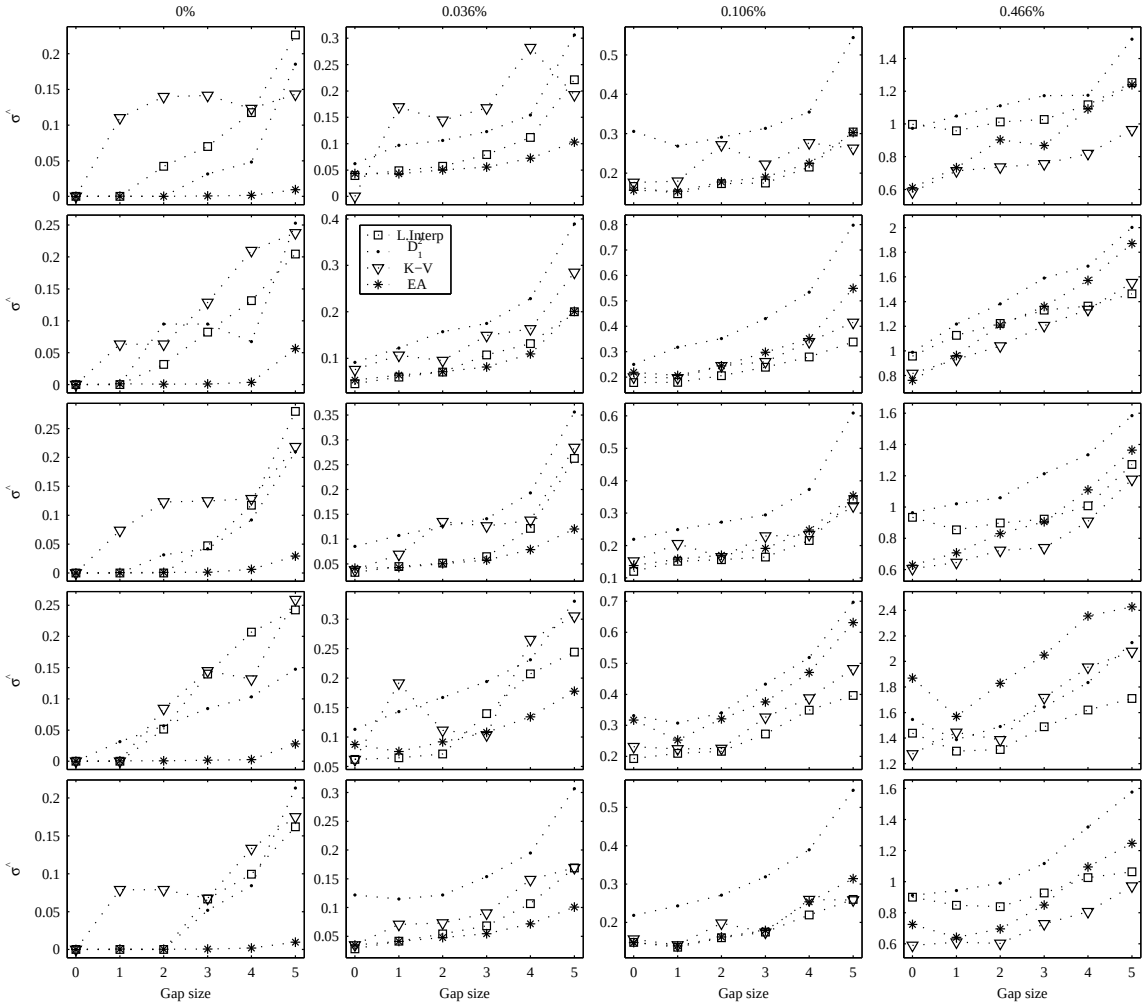


Figure 6.14: $\hat{\sigma}$ on DS-5. Each row corresponds to a different underlying function (DS-5-1, DS-5-2,..., DS-5-5), and each column corresponds to a different level of noise (0%, 0.036%, 0.106% and 0.466%). Every plot shows the results of $\hat{\sigma}$ statistic from four methods; i.e., Linear Interpolation, D_1^2, K-V and EA; square, shaded point, triangle and asterisk respectively.

variance of estimates. As one can see, EA-M-CV is competitive even with the idealised case (SF*).

Table 6.8: PRH Data Results from PRH Method with SF* (idealised scenario). The columns show the results from the statistical analysis (details in §6.1.1 and Appendix A). The rows correspond to the true delays (μ_0).

μ_0	$\mathcal{P}$	$\mathcal{T}$	95% CI	CI range	AE	MSE
34	1.000	0.000	33.5 - 34.4	0.92	0.44	5.28
43	0.702	0.382	42.2 - 44.1	1.87	1.54	21.96
49	0.839	2.375	49.2 - 52.0	2.81	2.36	52.32
59	0.447	1.899	58.9 - 61.1	2.21	1.78	31.96
66	0.465	0.272	65.5 - 66.5	1.02	0.71	6.55
76	0.671	2.026	76.0 - 77.5	1.55	0.95	15.67
99	0.001	2.701	99.5 - 102.3	2.86	2.79	55.39
Avg	0.374			1.89	1.51	27.02

Table 6.9: PRH Data Results from PRH Method with SF+. Each row shows the results from the statistical analysis for the true delay μ_0. For more details see §6.1.3

μ_0	$\mathcal{P}$	$\mathcal{T}$	95% CI	CI range	AE	MSE
34	0.000	-4.881	18.9 -27.6	8.72	22.79	593.45
43	0.210	1.260	81.8 - 48.2	6.46	13.51	266.19
49	0.315	-1.008	43.7 - 50.7	6.96	15.15	307.95
59	0.977	-0.028	54.6 - 63.1	8.51	19.66	455.20
66	0.257	-1.138	58.7 - 67.9	9.23	22.21	542.99
76	0.031	-2.188	66.7 - 75.5	8.83	20.95	513.97
99	0.407	-0.832	93.0 - 101.4	8.39	18.84	446.00
Avg	0.314			8.16	19.02	446.54

Table 6.10: PRH Data: Results from EA-M-CV. Every row corresponds to the results from the statistical analysis for each true delay μ_0; see §6.1.3 for details.

μ_0	$\mathcal{P}$	$\mathcal{T}$	95% CI	CI range	AE	MSE
34	0.600	0.525	32.3 - 36.8	4.58	7.68	132.27
43	0.330	0.977	42.5 - 44.4	1.96	2.28	24.34
49	0.389	-0.864	46.8 - 49.8	2.94	3.99	54.69
59	0.684	0.407	57.2 - 61.9	4.35	7.28	119.00
66	0.957	-0.052	63.9 - 67.9	3.98	7.16	99.53
76	0.301	-1.039	73.0 - 76.9	3.83	6.89	93.42
99	0.830	0.214	96.7 - 101.7	4.94	8.61	153.43
Avg	0.585			3.80	6.27	96.67

Table 6.11: PRH Data Results: Variance ($\hat{\sigma}$) and Bias ($|\mu_0 - \hat{\mu}|$). The true delay is denoted by μ_0, and the estimated time delay by $\hat{\mu}$. Every row corresponds to a different true delay; see §6.1.3 for details.

	PRH with SF*			PRH with SF+			EA-M-CV		
μ_0	$\hat{\mu}$	$\hat{\sigma}$	$\lvert\mu_0 - \hat{\mu}\rvert$	$\hat{\mu}$	$\hat{\sigma}$	$\lvert\mu_0 - \hat{\mu}\rvert$	$\hat{\mu}$	$\hat{\sigma}$	$\lvert\mu_0 - \hat{\mu}\rvert$
34	34.0	**2.3**	**0.00**	23.2	21.9	10.73	34.4	11.8	0.46
43	43.1	**4.7**	**0.18**	45.0	16.2	2.05	43.7	4.8	0.79
49	50.6	**7.0**	1.68	47.2	17.5	1.77	48.4	7.7	**0.57**
59	60.0	**5.5**	**0.07**	58.9	21.4	0.06	59.9	9.7	0.96
66	66.0	**2.5**	**0.07**	63.3	23.2	2.65	65.7	9.2	0.21
76	76.7	**3.8**	**0.79**	71.1	22.2	4.87	75.1	10.2	0.86
99	100.9	**7.2**	1.95	97.2	21.1	1.76	99.8	12.2	**0.89**
Avg		**4.7**	0.81		20.5	3.41		9.4	**0.68**

Table 6.12: Harva Data Results from Bayesian Estimation Method against EA-M-CV. The last three columns correspond to three different data sets. The second column denotes the statistic. For more details see §6.1.4 and Appendix A.

		Data Set		
Method	Statistic	0.1	0.2	0.4
Bayesian method	$\mathcal{P}$	0.59	0.63	**0.06**
	$\mathcal{T}$	0.52	0.48	**1.87**
	MSE	32.18	**9.43**	**41.89**
	AE	1.84	**1.94**	**3.7**
	$\hat{\mu}$	35.2	35.1	**35.8**
	$\hat{\sigma}$	5.7	**3.1**	**6.4**
EA-M-CV	$\mathcal{P}$	**0.92**	**0.76**	0.007
	$\mathcal{T}$	**0.09**	**0.30**	2.70
	MSE	**10.06**	23.25	66.99
	AE	**1.76**	3.28	5.72
	$\hat{\mu}$	**35.0**	35.1	36.4
	$\hat{\sigma}$	**3.1**	4.8	8.0

6.1.4 Harva Data

For these data, we use a statistical analysis as above, where the true delay is $\mu_0 = 35$ and $\eta = 225$. We only compare the Bayesian method with EA-M-CV.

The configuration of Bayesian method is in Harva and Raychaudhury [34]. The parameter setting for EA-M-CV is $\Delta = [0, 70]$, $k = [1, 15]$ and $\theta = [1, n]$. The offset and radio parameters $a_{k(a)}$ and $b_{k(a)}$ were fixed as in §3.2.5, respectively.

The results[2] are in Table 6.12. The best results are in bold fonts. In Table 6.12, each data set 0.1, 0.2 and 0.4 corresponds to a different level of noise (see 3.2.5).

[2] $\hat{\mu}$ and $\hat{\sigma}$ from Bayesian method are not reported in [34], privately communication.

6.2 Real Data: Q0957+561

In this section are the results of the data introduced in §2.3.1 and §2.3.2. The quasar Q0957+561 is described in §2.3. These data sets are shown in Fig. 2.3.

6.2.1 Radio Data

The radio data sets are plotted at the left-hand side in Fig. 2.3.

On these data, we use K-F, K-V and EA-M-CV and EA-R-CV. We employ flux ratios $M = 1/1.44$, $M = 1/1.43$ and $M = 1/1.42$ for the 4 cm, 6 cm and 6*cm data, respectively (the most likely values given our models). We tested time delay trials between $\Delta_{min} = 300$ and $\Delta_{max} = 500$, with increments of 1 day. The noise model is assumed to be zero mean i.i.d. Gaussian with standard deviation of 2% of the observed flux value.

For K-F (in §4.2), we used Algorithm 4.1 with bounds $\omega = [100, 1200]$ and unitary increments with the threshold λ set to 0.001. The selected kernel widths (ω) were 481, 488 and 528 days, and the estimated time delays were 409 days, 459 days and 405 days for the 4 cm, 6 cm and 6*cm data, respectively.

To calculate confidence intervals on the time delay estimates, we performed 500 Monte Carlo simulations by adding noise realisations to the observed data. The parameters ω and λ were fixed as above. Confidence intervals were determined as standard deviations of time delay estimates across the Monte Carlo samples, i.e., $\hat{\sigma}$ with $\eta = 500$. The results are shown in Table 6.13. Flux reconstructions with these time delays are shown in Fig. 6.15. Within each plot, at the top is the image A and at the bottom is image B. The continuous lines are our reconstructed underlying light curves, $h_A(t_u)$ and $h_B(t_v)$ in Eq. 4.12.

For K-V (also see §4.2), the parameter k was estimated by Algorithm 4.1 (ω is replaced by k) with $LowerBound = 1$ and $UpperBound = 15$ (increments of 1). We obtained $k = 3$ and $\lambda = 10^{-6}$ for 4 cm, and the estimated time delay was 409 days. For 6 cm data, we found $k = 3$ and $\lambda = 10^{-3}$, and the delay of 449 days. On 6*cm data, we found $k = 5$ and $\lambda = 10^{-3}$, and the time delay is 427 days. The results from 500 Monte Carlo samples (MC data) are in Table 6.13. Flux reconstructions are

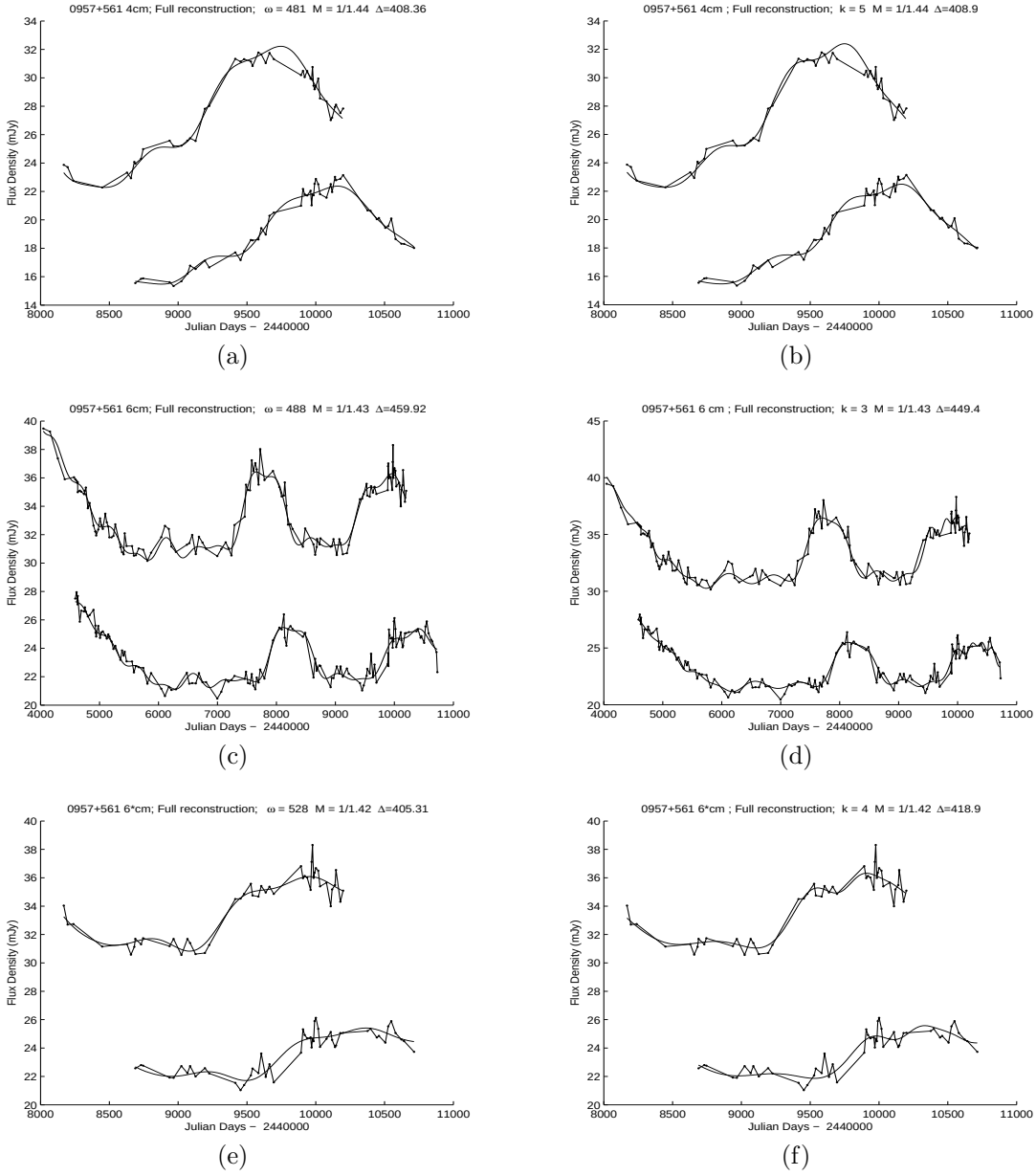


Figure 6.15: Reconstructions on Radio Data: **a)** K-F on 4 cm, **b)** K-V on 4 cm, **c)** K-F on 6 cm, **d)** K-V on 6 cm, **e)** K-F on 6*cm and **f)** K-V on 6*cm. For more details, see §6.2.1.

shown in Fig. 6.15.

The results of our EA-M-CV are in Tables 6.14, 6.15 and 6.16, where Δ is given in days. The parameter setting is $\Delta = [400, 600]$, $k = [1, 15]$ and $\theta = [1, n]$; the ratio M was set as above. Since our EA is stochastic, we perform ten realisations on each

Table 6.13: Results on Radio Data (MC data) from K-F and K-V. Here $\hat{\mu}$ and $\hat{\sigma}$ are obtained from 500 Monte Carlo simulations for each data set.

	K-F	K-V
Data	$\hat{\mu} \pm \hat{\sigma}$	$\hat{\mu} \pm \hat{\sigma}$
4 cm	408.3±10	408.9±11
6 cm	459.9±18	449.4±27
6*cm	405.3±29	418.9±40

Note that quantities are in days.

data set.

In theory, the results from Tables 6.14 to 6.16 should give the same time delay estimation. We have seen that the difference lies in sampling, that is the gaps or missing data. Hence, the two data sets, 4 cm and 6*cm, should give close estimates because they are sampled at the same observational times. In Table 6.14 (4 cm), there are some time delay estimates above 400 days, where in Table 6.16 (6*cm) all estimates are below 400 days. To investigate this, we run EA-M-CV and EA-R-CV on 4 cm until 100 generations rather than 50 generations; this allows us to perform a better search since the algorithm may be trapped in local minima. The M parameter is also evolved when using EA-R-CV. Besides, 100 realisations are under analysis. The results are in Fig. 6.16. We obtain the mode of estimates by taking the integer part only (flooring). For EA-M-CV, the mode is 398 days with a frequency of 91. When evolving M (EA-R-CV), the mode is 396 days appearing 65 times. Therefore, estimates outside the modes might be considered as outliers.

Moreover, if one evolves M then the estimates are distinct on the same data set, 4 cm (see Fig. 6.16). Therefore, we also tested radio data by evolving M and using real representation; flooring at fitness function. The results are in Tables 6.17 to 6.19.

Results from 500 Monte Carlo simulations are in Table 6.30. The parameters k and θ were set to their best values; i.e., from the above results, we obtain the more frequent values.

In summary, our estimates on observed data are in Table 6.20, where EA-CV

Table 6.14: EA-M-CV: Results on Radio Data (4 cm), ten realisations. Each row shows the best parameter combination after 50 generations; see §6.2.1 for details.

Realisation	Δ	θ	k	f_x
1	412.10	35	4	1.9261528
2	396.65	28	3	1.9414268
3	396.70	28	3	1.9414273
4	396.70	28	3	1.9414274
5	415.16	34	4	1.8753556
6	396.66	28	3	1.9414268
7	421.02	34	4	1.9226421
8	422.27	35	4	1.9250800
9	396.65	28	3	1.9414268
10	396.66	28	3	1.9414268

Table 6.15: EA-M-CV: Results on Radio Data (6 cm), ten realisations. Each row shows the best parameter combination after 50 generations; see 6.2.1 for details.

Realisation	Δ	θ	k	f_x
1	449.42	40	6	4.2529894
2	461.70	40	6	4.2146423
3	455.32	40	6	4.2286100
4	459.56	40	6	4.2254048
5	456.84	41	6	4.2460969
6	467.43	40	6	4.2375457
7	449.42	40	6	4.2529894
8	461.77	40	6	4.2146423
9	455.32	40	6	4.2286100
10	459.56	40	6	4.2254048

Table 6.16: EA-M-CV: Results on Radio Data (6*cm), ten realisations. Each row shows the best parameter combination after 50 generations; see 6.2.1 for details.

Realisation	Δ	θ	k	f_x
1	397.80	12	5	3.9957233
2	389.53	12	5	3.9983381
3	399.31	12	5	3.9960528
4	396.63	12	5	3.9956380
5	371.99	22	7	3.8485912
6	391.27	12	5	3.9971660
7	393.66	12	5	3.9960962
8	396.97	12	5	3.9956473
9	396.84	12	5	3.9956422
10	371.84	15	3	4.0146605

Table 6.17: EA-R-CV: Results on Radio Data (4 cm), ten realisations. Each row shows the best parameter combination after 50 generations; see 6.2.1 for details.

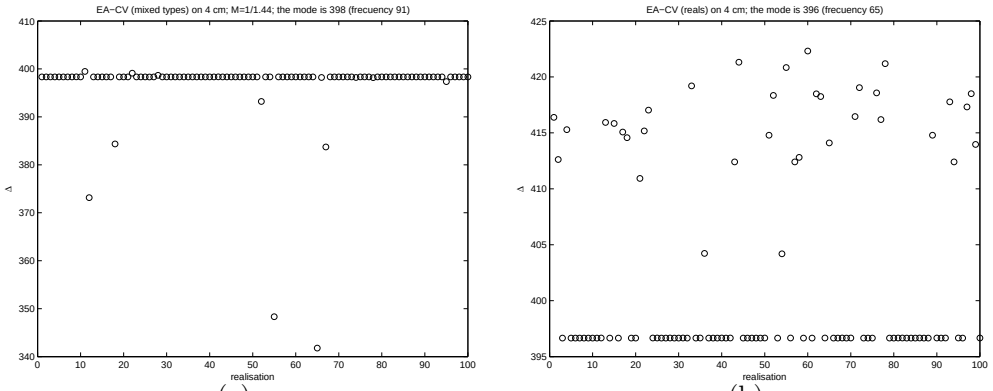
Realisation	Δ	M	θ	k	f_x
1	396.66	1.4400	28	3	1.9414
2	396.65	1.4400	28	3	1.9414
3	414.10	1.4358	34	4	1.8544
4	396.65	1.4400	28	3	1.9414
5	396.65	1.4400	28	3	1.9414
6	397.22	1.4413	28	3	1.9429
7	412.86	1.4350	34	4	1.8600
8	396.66	1.4400	28	3	1.9414
9	396.61	1.4399	28	3	1.9414
10	419.72	1.4361	35	4	1.8623

Table 6.18: EA-R-CV: Results on Radio Data (6 cm), ten realisations. Each row shows the best parameter combination after 50 generations; see 6.2.1 for details.

Realisation	Δ	M	θ	k	f_x
1	464.60	1.4299	40	6	4.1829
2	451.40	1.4294	40	3	4.3191
3	455.92	1.4297	40	6	4.1953
4	453.72	1.4277	40	6	4.1947
5	466.34	1.4313	40	6	4.2588
6	480.95	1.4317	32	5	4.3032
7	457.13	1.4295	40	6	4.2093
8	476.49	1.4314	34	6	4.2663
9	454.86	1.4314	40	6	4.2127
10	456.30	1.4307	40	6	4.1608

Table 6.19: EA-R-CV: Results on Radio Data (6*cm), ten realisations. Each row shows the best parameter combination after 50 generations; see 6.2.1 for details.

Realisation	Δ	M	θ	k	f_x
1	397.22	1.4220	12	5	3.9916
2	397.21	1.4220	12	5	3.9916
3	397.22	1.4220	12	5	3.9916
4	397.22	1.4220	12	5	3.9916
5	397.22	1.4220	12	5	3.9916
6	397.22	1.4220	12	5	3.9916
7	397.22	1.4220	12	5	3.9916
8	397.22	1.4220	12	5	3.9916
9	397.22	1.4220	12	5	3.9916
10	397.22	1.4220	12	5	3.9916



(a) (b)

Figure 6.16: Results on 4cm Data: **a)** EA-M-CV with 100 generations. **b)** EA-R-CV evolving M and 100 generations.

Table 6.20: Results on Radio Data (observed data). Three data sets are involved: 4 cm, 6 cm and 6*cm. For details see §6.2.1.

	K-F	K-V	EA-CV	Reference [32]
4 cm	409	409	397	397
6 cm	459	449	450-460	452
6*cm	405	427	397	–

Note that the time delays are in days.

denotes the best results from both EA-M-CV and EA-R-CV. As reference, Haarsma *et al.* [32] report time delays of 397±12 and 452^{+14}_{-15} days for the 4 cm and 6 cm data, respectively, and 409±30 on the combined 4+6 cm data set by using the PRH method. They also report results of the Dispersion spectra method ($D^2_{4,2}$): 383^{+15}_{-19} and 416^{+22}_{-24} days for the 4 cm and 6 cm data, respectively, and 395^{+13}_{-15} days on the combined 4+6 cm data set.

6.2.2 Optical Data

On these data, we use D^2_1, $D^2_{4,2}$, K-V, (1+1)ES, EA-M-CV and EA-R-CV methods.

For all methods, bounds are set to $\Delta_{min} = 400$ and $\Delta_{max} = 450$ days given that

our prior knowledge is that the best time delay is around 417 days [49, 59]. We evaluate D_1^2, $D_{4,2}^2$ and K-V in this range with unitary increments. The results are shown in Table 6.21; for DS1, DS2 and DS3 data sets (in §2.3.2). The decorrelation length δ in Table 6.21 is the same adopted by Kundic *et al.* [49] and Ovaldsen *et al.* [59]. Hence, Δ and M are those with the minimum dispersion spectra D_1^2 and $D_{4,2}^2$ respectively, for a given δ. Figure 6.17 depicts the best time delay versus decorrelation length δ such as M gives the minimum $D_{4,2}^2$.

The confidence intervals[3] were estimated through 500 Monte Carlo simulations ($\eta = 500$) over the noise processes (4.5) by fixing the parameters M and δ to the best values, as in Table 6.21. The results are shown in Table 6.22.

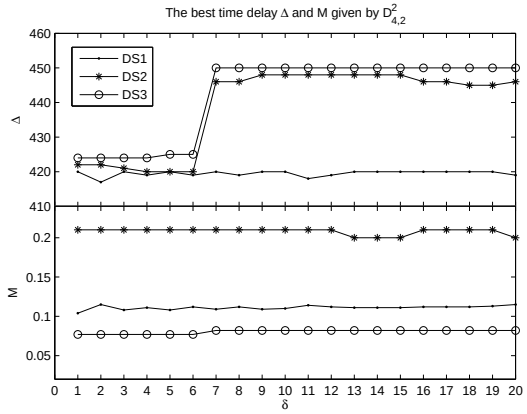


Figure 6.17: Dispersion spectra $D_{4,2}^2$ for DS1, DS2 and DS3 . The top panel shows Δ vs. δ. At the bottom panel is M vs. δ. The best Δ is found when $\min_M D_{4,2}^2$.

For K-V, we have fixed M to 0.117, 0.21 and 0.076 for DS1, DS2 and DS3 respectively [49, 59]. The regularisation parameter λ and the smoothing parameter k were chosen through Algorithm 4.1. The results are in Table 6.21. The reconstructions are shown in Fig. 6.18. The confidence intervals are also estimated through 500 Monte Carlo simulations fixing M, k and λ to the best values. The results are also in Table 6.22.

In Fig. 6.19 is shown the estimated time delay versus θ (patterns). There is a

[3]Typically, the confidence intervals for these methods are estimate via bootstraps with replacement and median filter [65, 63].

Table 6.21: Results on Observed Optical Data

	Dispersion spectra		K-V	
Data set	D_1^2: Δ (M)	$D_{4,2}^2$: Δ $(M;\delta)$	Q : Δ $(k;\lambda)$	Reconstruction
DS1	417 (0.119)	420 (0.109;7)	420 (3;10^{-4})	Fig. 6.18 (a)
DS2	429 (0.210)	446 (0.210;7)	420 (3;10^{-5})	Fig. 6.18 (b)
DS3	425 (0.077)	424 (0.077;4)	435 (7;10^{-6})	Fig. 6.18 (c)

Δ is given in days.

Table 6.22: Confidence Intervals: 500 Monte Carlo simulations

	Dispersion spectra		K-V
Data set	$D_1^2 : \hat{\mu} \pm \hat{\sigma}$	$D_{4,2}^2 : \hat{\mu} \pm \hat{\sigma}$	$\hat{\mu} \pm \hat{\sigma}$
DS1	416.7±0.9	419.9±1.3	419.5±0.7
DS2	421.6±2.8	443.5±8.2	420.9±4.0
DS3	426.7±2.3	438.5±12.7	436.6±6.1

μ_Δ and σ_Δ are given in days

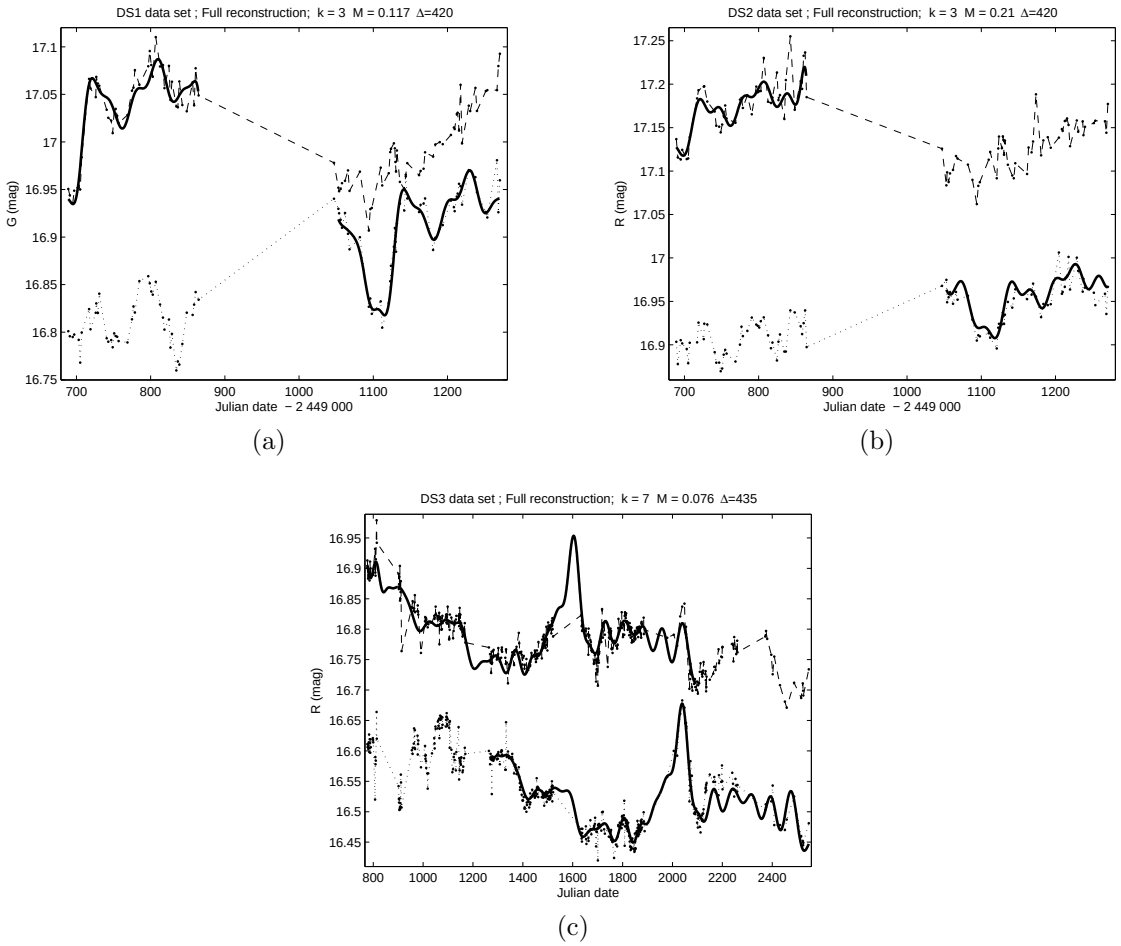


Figure 6.18: Reconstructions on Optical Data. (**a**) DS1: $\Delta = 420$, $M = 0.117$, $k = 3$ and $\lambda = 0.001$. (**b**) DS2: $\Delta = 420$, $M = 0.21$, $k = 3$ and $\lambda = 0.001$. (**c**) DS3: $\Delta = 435$, $M = 0.076$, $k = 7$ and $\lambda = 10^{-6}$. In each plot, at the top is image A (dashed) and at the bottom image B (dotted). Shaded circles are at observations. Continuous lines are our curves modelling the underlying source. Note that the image A for DS3 has been shifted upwards by 0.20 mag for visualisation.

tendency of time delay estimates levelling at 419 days for the range $\theta = [49, 72]$ for DS1. For DS2, $\Delta = 420$ in the range $\theta = [50, 61]$. On DS3, there is not a well defined pattern when $k = 7$, which is given by K-V. But, there are two well defined patterns when $k = 5$, which is suggested by EA-CV (see Tables 6.27 and 6.28), where the patterns are at $\theta = [83, 110]$ ($\Delta = 428$–429) and $\theta = [123, 154]$ ($\Delta = 426$).

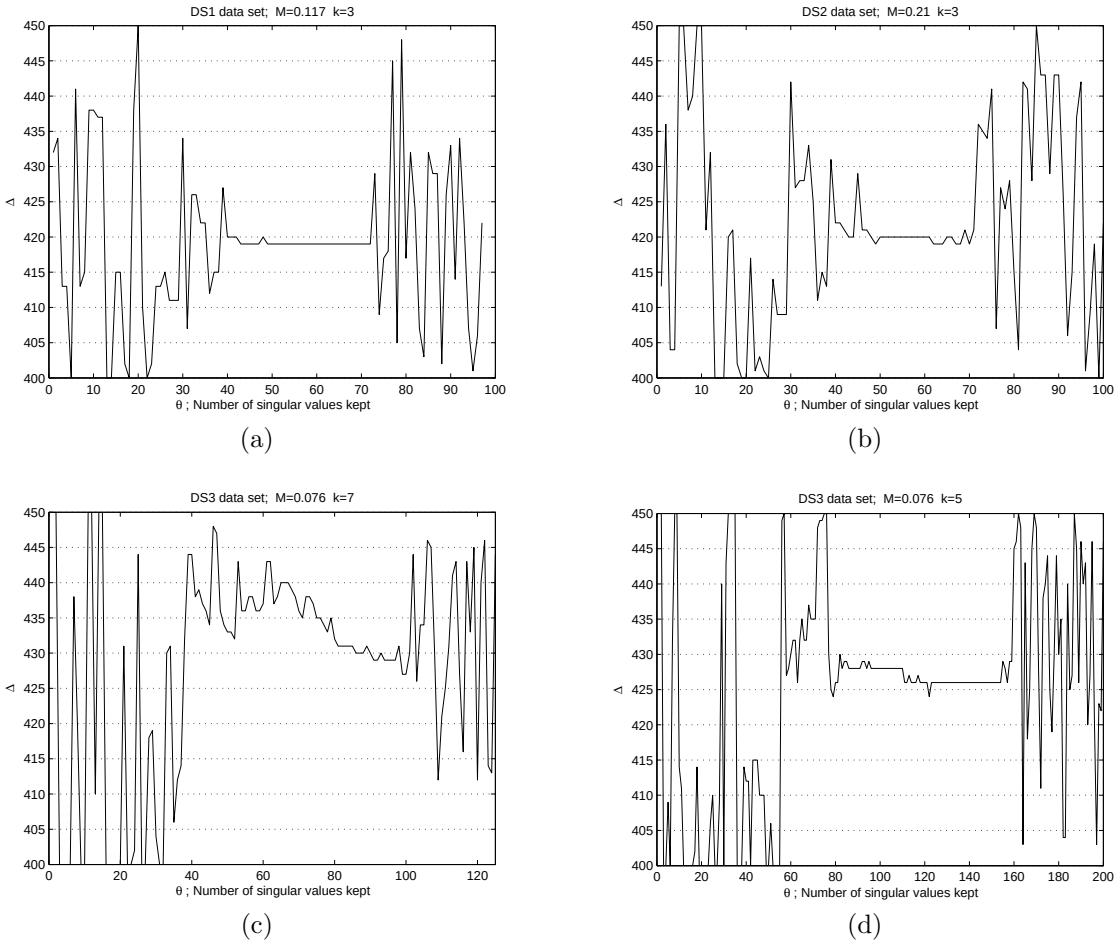


Figure 6.19: Patterns on Optical Data. The estimated Δ versus θ. For each θ, Q is evaluated in the range $\Delta_{min} = 400$ to $\Delta_{max} = 450$. **(a)** DS1: $M = 0.117$ and $k = 3$. The delay estimates are stable in the range $\theta = [49, 72]$ ($\Delta = 419$). **(b)** DS2: $M = 0.21$ and $k = 3$. The delay estimates are stable in the range $\theta = [50, 61]$ ($\Delta = 420$). **(c)** DS3: $M = 0.076$ and $k = 7$ (K-V). Here, there is not a well defined pattern, but stability can be found in the ranges $\theta = [81, 85]$ ($\Delta = 431$) and $\theta = [94, 97]$ ($\Delta = 429$), and $\Delta = 429$–431 when $\theta = [81, 98]$. **(d)** DS3: $M = 0.076$ and $k = 5$ (EA-CV). The delay estimates are stable in the ranges $\theta = [83, 110]$ ($\Delta = 428$–429) and $\theta = [123, 154]$ ($\Delta = 426$).

By using the (1+1)ES in §5.2, the precision is set to 200 and variable bounds set as above and allowing 15,000 iterations. The convergence is reached after 14,410

Table 6.23: EA-R-CV: Results on DS1. There are ten realisations. Each row shows the best parameter combination after 50 generations. See §6.2.2 for details.

Realisation	Δ	M	θ	k	f_x
1	419.67	0.1495	58	3	0.0019249601
2	419.67	0.1462	58	3	0.0019249602
3	419.67	0.1923	58	3	0.0019249605
4	419.68	0.1398	58	3	0.0019249620
5	419.68	0.1217	58	3	0.0019249577
6	419.68	0.1197	58	3	0.0019249593
7	419.68	0.1733	58	3	0.0019249592
8	419.68	0.1516	58	3	0.0019249615
9	419.67	0.1482	58	3	0.0019249588
10	419.68	0.1656	58	3	0.0019249586

iterations by using the same fitness function (Algorithm 5.1 in §4.2), so we also floor at fitness function for integer variables. This ES yields on DS1 $\Delta = 419.6$, $M = 0.1732$, $\theta = 58$, $k = 3$, and MSE$= 1.9249617 \times 10^{-3}$.

Regarding our EAs, we use the following general bounds: $\Delta = [400, 450]$, $k = [1, 15]$, $\theta = [1, n]$, and $M = [0, 0.30]$. We start showing results of EA-R-CV on DS1 by evolving M also. Hence, the parameters that are integers are floored at fitness function. The results on DS1 are in Table 6.23.

In Table 6.24 are ten realisations, the result from EA-M-CV on DS1. Here, M is not evolved and fixed to 0.117. The variable bounds are also set as above. Table 6.23 shows that regardless of the value of M the time delay Δ is consistent, which justifies that M does not need to be evolved. Rather, we use the reported value $M = 0.117$ [49].

Results from EA-R-CV on DS2 by evolving M are in Table 6.25 (flooring at fitness function).

Table 6.26 shows ten realisations from EA-M-CV on DS2, where M is set to 0.21 [49].

For DS3, Table 6.27 has the results from EA-R-CV, where M is also evolved;

Table 6.24: EA-M-CV: Results on DS1. There are ten realisations. Each row shows the best parameter combination after 50 generations. See §6.2.2 for details.

Realisation	Δ	θ	k	f_x
1	419.68	58	3	0.0019249744
2	419.67	58	3	0.0019249722
3	419.69	58	3	0.0019249722
4	419.67	58	3	0.0019249719
5	419.66	58	3	0.0019249691
6	419.66	58	3	0.0019249670
7	419.66	58	3	0.0019249753
8	419.67	58	3	0.0019249724
9	419.47	71	3	0.0018908716
10	419.67	58	3	0.0019249711

Table 6.25: EA-R-CV: Results on DS2. There are ten realisations. Each row shows the best parameter combination after 50 generations. See §6.2.2 for details.

Realisation	Δ	M	θ	k	f_x
1	418.53	0.2016	55	4	0.00185386
2	418.90	0.2244	56	4	0.001839987
3	418.73	0.1535	56	4	0.001832136
4	419.85	0.1769	55	4	0.001907173
5	419.09	0.1831	56	4	0.001849627
6	418.11	0.1595	55	4	0.001865103
7	418.80	0.1999	55	4	0.001853981
8	420.38	0.1861	55	4	0.001893737
9	418.81	0.1500	55	4	0.001870789
10	419.31	0.1985	56	4	0.001868472

Table 6.26: EA-M-CV: Results on DS2. There are ten realisations. Each row shows the best parameter combination after 50 generations. See §6.2.2 for details.

Realisation	Δ	θ	k	f_x
1	419.05	54	4	0.0019407480
2	418.85	56	4	0.0018662351
3	419.59	55	4	0.0018971073
4	420.03	55	4	0.0019056176
5	418.78	55	4	0.0018865383
6	420.56	56	4	0.0019137956
7	421.35	56	4	0.0019009342
8	419.51	56	4	0.0018848796
9	420.52	56	4	0.0019022225
10	418.69	55	4	0.0018722574

Table 6.28 shows the results from EA-M-CV, whereas M is set to 0.076 [59].

Results from 500 Monte Carlo simulations are in Table 6.30. The parameters k and θ were set to their best values. That is, we obtain the more frequent values for θ and k from the results in Tables 6.27 and 6.28.

In summary, the results on observed optical data are in Table 6.29, which one should compare with Table 6.21. On DS3, we obtained six realisations only. DS3 is the largest data set under analysis ($n = 422$). In 6.29, the column Reference provides the reported time delays. Nevertheless, those delays are the assumed time delays. That is, distinct estimates are obtained from different methods on the same data.

6.2.3 Q0957+561 Summary

The results on radio and optical data are in Table 6.30. There, we concentrate all results including those from 500 Monte Carlo simulations (MC), where the parameters were fixed to their more likely values; for results from EA, we fixed k and θ according to their frequency of appearance in the above results – in both EA-M and EA-R. In Table 6.30, $\hat{\mu}$ and $\hat{\sigma}$ denote the mean and standard deviation of estimates from the

Table 6.27: EA-R-CV: Results on DS3. There are six realisations. Each row shows the best parameter combination after 50 generations. See §6.2.2 for details.

Realisation	Δ	M	θ	k	f_x
1	428.89	0.0947	98	5	0.00286733794976
2	427.20	0.0864	109	6	0.00286423323078
3	427.58	0.0732	108	6	0.00286220282065
4	428.90	0.0982	95	5	0.00286734138156
5	428.89	0.0947	98	5	0.00286733794976
6	427.28	0.0667	109	6	0.00286438534341

Table 6.28: EA-M-CV: Results on DS3. There are six realisations. Each row shows the best parameter combination after 50 generations. See §6.2.2 for details.

Realisation	Δ	θ	k	f_x
1	429.26	98	5	0.00286818213046
2	428.95	98	5	0.00286737380445
3	431.43	101	7	0.00247621668255
4	429.29	98	6	0.00288105281463
5	427.08	109	6	0.00286475222488
6	427.72	108	6	0.00286323588324

Table 6.29: Results from EAs on Observed Data

Data set	EA-R-CV	EA-M-CV (M)	Reference
DS1	419.6	419.6 (0.117)	417 [49]
DS2	418.1–420.3	418.6–420.5 (0.210)	417 [49]
DS3	427.2–428.8	427.0–431.4 (0.076)	424.9 [59]

MC data, where $\eta = 500$ (see Appendix A).

Table 6.30: Q0957+561 Summary of Results; see §6.2.3 for details.

	K-V		EA		
Data	$\Delta(k;\lambda)$	$\hat{\mu} \pm \hat{\sigma}$	$\Delta(k;\theta)$	$\hat{\mu} \pm \hat{\sigma}$	M
4 cm	409 (5;10^{-6})	408.9±11	396.6–397.2 (3;28)	393.8±12	1/1.44
6 cm	449 (3;10^{-3})	449.4±27	449.4–476.4 (6;40)	451.5±25	1/1.43
6*cm	427 (4;10^{-3})	418.9±40	393.6–399.3 (5;12)	414.0±59	1/1.42
DS1	420 (3;10^{-4})	419.5±0.7	419.6 (3;58)	422.3±4	0.117
DS2	420 (3;10^{-5})	420.9±4.0	418.1–420.3 (4;55)	420.5±4	0.210
DS3	435 (7;10^{-6})	436.6±6.1	428.8–429.2 (5;98)	432.4±8	0.076

6.3 Chapter Summary

We presented results from the methods introduced in §3 to §5. We used several types of data: real and artificial (see §2 and §3). On real data, we used optical and radio data from quasar Q0957+561, which is a complicated system. Regarding artificial data, we also used several types of these data: optical-like data (DS-5), radio-like data (DS-500), PRH data and Harva data. These artificial data simulate distinct noise levels and various gaps. Moreover, they simulate short and long time delays, and irregularly sampled time series always were under analysis. The discussion of results and conclusions (for real and artificial data) are in the following chapter.

Chapter 7

Conclusions and Future Work

IN this chapter (§7.1), first, we present the general conclusions. Then we will discuss the results obtained in §6 and draw some particular conclusions, which are presented per each class of data, artificial and real data. Answers to the questions formulated in §1.3 are given. Furthermore, advantages and disadvantages of the methods studied in this thesis are also discussed. Finally, we will introduce our further research directions in §7.2 where ideas are presented within different sections separately; since we started to explore some of them, we also present some hints.

7.1 Conclusions

We have introduced a new approach for measuring the time delay between light curves of two images of a gravitationally lensed system, based on kernel linear regression and evolutionary algorithms, in particular K-V and EA-M-CV (see §4.2 and §5.3, respectively). Using a large set of controlled experiments using artificially generated data (DS-500 and DS-5), PRH data and Harva data, we compared the accuracy of our methods with that of other methods used in the literature for time delay estimation, notably the DCF, LNDCF, PRH, Dispersion spectra and Bayesian method; see §3.2 for details on these methods.

Artificial Data

Running a controlled set of experiments is essential for a well-grounded comparison of competing models. For the artificial data, unlike in the case of observed fluxes, one has the luxury of knowing exactly the magnification ratio or offset M and the true time delay Δ; the noise process is also known. Therefore, we can reliably measure the bias and variance of the time delay estimates given by the studied methods. Obviously, one cannot fully measure the bias when estimating the time delay from real observations. On the artificial data, we conclude that our kernel-based methods presented in this thesis came across as the most accurate and stable methodologies for estimating the time delays between multiple images of a gravitationally lensed quasar (see Figs. 6.1 to 6.14, and Tables 6.1 to 6.12).

On the one hand, previous attempts for generating artificial data have been tried (e.g., see [69, 9, 21, 70]). However, in general, they only focus on a few types of data sets by either simulating specific features in the light curves or dealing with sampling issues, and the performance is usually compared with a single method. On the other hand, our artificial data sets (DS-500 and DS-5) contain simulated light curves of widely varying (but still realistic) shapes, observational gaps and noise levels.

On DS-500, we tested all methods introduced in §3 to §5. In Fig. 6.1, one can see that the best performance is for the linear interpolation method on these data, where this simple method performs better than PRH method, D_1^2 and $D_{4,2}^2$ – the most used ones. The disadvantage is that it is slower because a resolution of 0.01 in time is needed to evaluate the error for trial time delays; this yields too many points in the fitting. Methods based on cross correlation also show a bad performance, i.e., accuracy. Therefore, we avoid DCF and LNDCF on DS-5, PRH Data and Bayesian Data. In Fig. 6.2, the best results are from our kernel-based methods – including those in Fig. 6.1 – apart from EA-R-LL. Hence, EA-R-LL suggests that the negative log-likelihood (LL) is not good in estimating the kernels parameters and the time delay, where EA-R-CV performs much better in terms of bias and variance. The results from Bayesian method are competitive with our methods despite it showing more bias on these data. Our methods introduced in sections §4.2.2 and §4.2.2 (K-F and K-V) give similar results (see Figs. 6.2b & 6.2c), although K-V tends to require

less computational time (see §4.3). From Table 6.1, we conclude that the best results are from K-V and EA-M-CV. Additionally, Tables 6.2 and 6.3 also show that the best results are from K-V and EA-M-CV through 95% confidence intervals and t-test at a 95% significance level (see Figs. 6.3 and 6.4). Similar results are obtained from the MSE, AE, $\hat{\mu}$ and $\hat{\sigma}$ estimators (see Figs. 6.5 to 6.8 respectively).

Regarding DS-5, the best methods are K-V, Linear Interpolation and EA-M-CV according to our statistical analysis over all estimates; see Table 6.4. However, the 95% CI and t-test suggest that the results from EA-M-CV are more statistically significant than others (see Table 6.5 and 6.6). DS-5 data are important because they simulate optical data, which currently are widely used. As mentioned above, error bars on these data are about 0.01 mag (or less); i.e., $< 0.106\%$. Hence, in Table 6.7 and Figs. 6.10–6.14, the best results for low noise are also from EA-M-CV. From Tables 6.4 to 6.7 and Figs. 6.9 to 6.14, we conclude that the best results are from EA-CV-M.

We stress that the results from t-test in Table 6.3 on DS-500 with 0% of noise show a bad performance for EA-M-CV. This is due to Eq. (A.6) because if $\hat{\sigma} \to 0$ then $\mathcal{T} \to \infty$ regardless of the bias $\hat{\mu} - \mu_0$. Here, it is clear that the assumption of normality is not unique. In fact the derivation of Eq. (A.6) is given by assuming a standard normal distribution, i.e., $N(0, 1)$ (see [18, 40]). In Fig. 6.8 is shown $\hat{\sigma}$; there one can see that its value is close to zero. Therefore, $\mathcal{T}$ and $\mathcal{P}$ are robust when $\hat{\sigma} \geq 1$. However, the estimator MSE takes into account $\hat{\mu}$ and $\hat{\sigma}$ at the same time, i.e., bias and variance. Because what we want is a method that shows accuracy, the estimators $\hat{\mu}$, $\hat{\sigma}$, MSE and AE are robust with or without the assumption of normality.

We also performed some nonparametric tests such as sign test and signed-rank test which are distribution-free. These tests, in a similar analysis as in Tables 6.3 and 6.6, suggest that the results from EA-M-CV on DS-500 and DS-5 are the most statistically significant. Therefore, regarding the normality on results, the use of t-test in the statistical analysis is not a concern.

Recalling the 95% CI, in Fig. 6.3, the greater $\hat{\sigma}$, the larger the interval. Therefore, the probability that the true μ_0 falls within the interval is high. This does not make the 95% CI robust because we require low variance for accuracy. The 95% CI is

proportional to $\hat{\sigma}$.

We compared K-V with EA on DS-500 and DS-5. On the one hand, K-V employs the log-likelihood (LL) as loss function where the parameters k and λ are estimated via cross-validation (see Algorithm 4.1). On the other hand, in Algorithm 5.1, the fitness function of EA is the MSE_{CV} given by cross-validation (CV). Instead, we compared K-V with EA by using the mean squared error as the measurement of goodness of fit for K-V rather than LL. Thus, the observational error is considered constant (see Eq. 4.12 in §4.2). Consequently, we found that the results from LL are more accurate than simple mean squared error. Moreover, we tested another fitness function. That is, instead of MSE_{CV}, the fitness is given by LL where no cross-validation is performed because k and θ are also evolved. The results are that the fitness given by CV performs better than LL, where LL is less time-consuming (e.g., compare Fig. 6.2d with 6.2e–f). Similar results are obtained on DS-5.

On PRH Data, there are seven different true delays. Here, we only compared PRH method with EA-M-CV. Two versions of PRH method were used, SF* and SF+. On the one hand, in SF*, one fixes the parameters to those values used to generate the data. That is why we call it the idealised scenario. As expected, in Tables 6.8 to 6.11, the best results are from SF*, but EA-M-CV is competitive with it. In fact, the average bias is better for EA-M-CV. On the other hand, with SF+, where one finds the structure function from data (as in practice), we conclude that the best performance is for EA-M-CV (see Tables 6.8 to 6.11).

Harva Data are generated through the model formulation of the Bayesian method (see §3.2.5). Therefore, we only compared Bayesian method with EA-M-CV. For 0.1 data set, the best results are from EA-M-CV. For 0.2 data set, the best results are from either Bayesian method or EA-M-CV depending on the statistic. Regarding 0.4 data set, the best results are from Bayesian method. Note that 0.1, 0.2 and 0.4 are standard deviations (error bars). Recalling optical data, the standard deviations are about 0.01 mag, so the levels of noise of Harva data are too high compared with real optical data. Another criticism of these data is that they are unrealistic since they have ratio and offset at same time, i.e., $a_{k(a)}$ and $b_{k(a)}$ in (3.14). As we have seen, in practice, the data is either optical or radio (see §2.3). Moreover, the error bars are

not constant in practice.

Q0957+561: Radio and Optical Data

There has been a great deal of concern about the difference in time delay estimates from the two different wavelengths on radio data, 4 cm and 6 cm, since gravitational lensing is achromatic. Inspired by the results from our experimentation with artificial data, where the uncertainty of time delay estimates increases as the gap size increases, we have generated the data set 6*cm in order to avoid the effect of the different gap sizes for different wavelengths. We conclude that such systematic differences between results obtained from observations at various wavelengths are due to the irregular sampling, and in particular, due to the presence of large gaps in the monitoring data. Experiments with simulated data sets like ours help in the understanding of how the results depends on the sampling, and in assessing the reliability of the time delays obtained by various methods. Such gaps are unavoidable in realistic long-term observing programmes, often leading to unacceptably deviant time delays (in this case, too large by more than 10%). Several recent analyses have come to this conclusion in various ways [25, 70, 21].

We have estimated the time delay between of two images of the quasar Q0957+561 from radio observations at 4 cm and 6 cm, where no agreement has been found. However, the 6*cm data set yields essentially the same value for the time delay at that obtained from the 4 cm data set (see EA-CV in Table 6.20), as opposed to a value of $\sim$450 days as obtained from the full 6 cm data set, which covers a longer monitoring period.

Even though, on optical data, DS1 and DS2 have the same sampling, but the time delay estimates yield different results (see Tables 6.21, 6.22 and 6.29). Moreover, the observational errors are comparable so the difference cannot be regarded to the noise. Here, we attribute the difference to the features in the light curves due to distinct filters, g- and r-band; see §2.3.2. Consequently, we conclude that the features in the light curve play an important role, specially on optical data where high variability is present.

As a result of our analysis, we conclude that DS1 is the best data set for Q0957+561

including optical and radio data in the comparison. In Tables 6.21 and 6.29, one can see that the results are consistent with all our methods. A time delay of 420 days was found by K-V because we only explore delays in the range 400–450 days with unitary increments; i.e., $\Delta \in \aleph$ (see in Table 6.21). But, the MC simulations give us 419.5 ± 0.7 days as a time delay (see Table 6.22). Our EAs suggest a time delay of 419.6 days regardless of representation (see Table 6.29), where $\Delta \in \Re$. In fact, we also tested EA-R-LL and gave us the same results. Moreover, the (1+1)ES introduced in §5.2 yields the same result (see §6.2.2). Therefore, we conclude that one can safely claim a time delay of 419.6 days for this quasar, where a time delay of 417 days was reported on DS1 [49], which is underestimated.

We did not use (1+1)ES neither on artificial data nor on radio data because it is costlier than any variant of our EA; (1+1)ES requires more iterations. On the one hand, if $g = 50$ (the maximum number of generations), then we perform 7,800 evaluations to the fitness function with our EAs because of our elitist strategy. On the other hand, (1+1)ES converges around 14,000 iterations for different initialisations. Every iteration corresponds to a fitness evaluation. Therefore, (1+1)ES demands more computational time (about twice). Since we use the same fitness functions, one expects to obtain a similar performance to EA.

Nowadays, researchers concentrate on optical data because the data can be gathered with high precision [13, 59, 70, 21]. Therefore, we have empirically proved through results on DS-5 that EA-M-CV is a promising approach because it is the most accurate method when the noise is less than 0.106%; see §6.1.2.

Thesis Questions

Following the above conclusions, we recall the first two questions in §1.3: What is the effect of noise in the time delay estimation? What is the influence of gaps? At the bottom of Figs. 6.1 and 6.2, one can observe a general trend of increased uncertainty as the gap size increases. The uncertainty is also proportional to the noise level. Besides, in Table 6.7 one can see quantitatively a clear increasing tendency on MSE, AE and $\hat{\sigma}$, as the noise increases. Therefore, we conclude that the better the sampling (no gaps) and the cleaner the data (no noise), the more accurate the time delay estimates.

Finally, the last question in §1.3: What is the effect of features? This can be explained by 4 cm and 6*cm data, where the sampling is the same and the error bars are roughly the same (2% of its flux in both cases). On observed data, the estimates on these data are similar from all methods and the same from EA-CV (see Table 6.20). Nevertheless, on MC data, the results are much more different, especially the variance ($\hat{\sigma}$); see Table 6.30. On 6*cm data, the variance is high because the features in the light curves are smooth; compare Figs. 2.3c and 2.3e. Something similar happens on optical data because DS1 and DS2 have the same sampling and comparable error bars, but the variance is high on DS2 regardless of the method (see Table 6.22). This is due to the features filtered by the r-band filter; also compare Figs. 2.3d and 2.3f. Moreover, the same phenomenon occurs on artificial data (DS-500 and DS-5). If the underlying function is smooth[1], the variance is high.

7.2 Future Work

Several research directions arise. Here we will introduce some ideas, where in some cases we have started to explore. Therefore, some of them are more detailed than others.

7.2.1 Speedup

One of the main concerns of our approach is speedup because data sets of thousands of samples become intractable. With our approach, we have obtained accuracy though the time complexity is high; see §4.3 and §5.5.

Since the SVD inversion is $O(n^3)$ in (4.17), the straightforward approach is to speed up this. We use Moore-Penrose inverse through SVD (see §4.2.1), but there are several methods for matrix inversion [72]. Unfortunately, most of them deal with square matrices and satisfy certain conditions so few approaches deal with matrices of the type of $\vec{K}$ in (4.14). Courrieu [16] claims that his method, *geninv*, can compute the Moore-Penrose inverse faster than typical methods when n is large. This method is introduced in the context of RBF networks (see §4.1.3). We tested such a method,

[1]Here, we mean that the light curve has not sharp events (peaks) with high amplitude.

but the performance is not good because with this method one cannot deal with ill-conditioning. In fact, its parameter 'tol' involves another parameter [16], which is not straightforward to set.

Inspired by wavelets theory [72](§13.10), the fast solution of linear systems is another research direction because Press *et al.* [72] claims that some linear systems become a sparse system in the wavelet basis. Moreover, Beylkin *et al.* [4] argue that separated representation can reduce the cost of numerical computations for solving high dimensional linear systems, maintaining the accuracy. They also argue that the curse of dimensionality is the greatest impediment to computing in higher dimensions; also see [2].

The Kernel Recursive Least-Squares Algorithm (KRLS), a nonlinear version of recursive least squares algorithm, is designed for on-line signal processing applications [22]. KRLS aims to obtain sparsity so it can be used on real-time problems. Since this algorithm is designed for a single signal, we tested it on a single light curve; image A. On our artificial data (DS-500), one can obtain sparsity, but the accuracy decreases when fitting the underlying curve. Furthermore, the time delay estimation involves two time series, and we fit two curves for a given trial time delay. Therefore, the on-line approach is not useful because the matrix $\vec{K}$ (4.14) is regenerated with each trial time delay. We also explored k-means algorithm [36, 35, 55] to select the quantity of Gaussian kernels in our model (4.6)-(4.7), and for positioning them. But, the quantity of clusters must be fixed, and when one obtains sparsity, one loses accuracy in the fitting. This leads to a biased time delay estimation. Nevertheless, sparsity is a promising topic to speed up our approach, but further research is needed.

Speedup can be also achieved by parallelisation of K-V and EA-M-CV. Cross-validation can be computed in parallel (see Algorithm 4.1), for instance, distinct values of variables $\{\lambda, \omega, l, \Delta_t\}$ can be computed separately, and then the results brought together. The same occurs for Algorithm 4.2, where distinct values of Δ_t can be computed separately. Moreover, regarding EA-M-CV, the natural parallel property of evolutionary algorithms makes it a candidate for parallelisation (e.g., see [10, 53, 80]).

Since EA-M-CV involves a costly fitness function (see §5.5), one can approximate

the fitness function by k-nearest neighbour local function approximation [76]. Regis *et al.* [76] claims that the number of fitness evaluations can be reduced, consequently an EA is speeded up. They compared their results with several variants of (γ,ρ)ES on synthetic and real data – an optimisation problem of groundwater bioremediation – and they show significant results on their test data (with less than 12 dimensions).

We also tried to evolve all parameters, i.e., the weights $\vec{\alpha}$ in (4.6)-(4.7) and Δ. This allows us to avoid SVD in (4.17). However, the performance is poor because the amount of parameters to evolve increases as the number of samples n do. In fact, we tested this approach with artificial data without noise, and even in this situations the performance is unsatisfactory. This is due to so many variables because one is only able to optimise up to 30 dimensions through typical evolutionary approaches [88]. Perhaps, new frameworks would overcome this problem.

7.2.2 Theoretical Analysis

We have shown that our methods perform well on radio and artificial data. However, we are interested on theoretical proofs so one can compare the performance of methods, specifically for the time delay estimation, in a theoretical manner regardless of the type of data (artificial, optical or radio data).

Ongoing research includes asymptotic normality of the time delay estimates from K-F method. This is inspired by the theory of empirical processes [89], area of mathematical statistics. Here, the aim is to find the bounds for $\vec{\alpha}$ and Δ in (4.6)-(4.7). This work is been carried out in collaboration with Dr. Leila Mohammadi.

7.2.3 Superimposed Light Curves

Imagine that an observer obtain a single light curve rather than two. But, there is some knowledge that this light curve is gravitational lensed. Therefore, the components are two light curves coming from a single source where one is a delayed version of the other. This problem is called unresolved photometry [62]. The main motivation is that new gravitational lens systems may be discovered through their time delays, since the time delay estimation is considered the most unambiguous confirmation of

the lensing hypothesis [70]. In other words, there are gravitational lens undiscovered because of the unresolved photometry.

We started to explore this problem by following our model formulation in §4.2. We model a single light curve (observed data) as

$$x_C(t_i) = h_C(t_i) + \epsilon_C(t_i), \tag{7.1}$$

where observation errors $\epsilon_C(t_i)$ are modelled as a zero-mean Normal distribution

$$N(0, \sigma_C(t_i)), \tag{7.2}$$

and assumed known, whereas the superimposition of two light curves is given by

$$h_C(t_i) = h_A(t_i) + M \ominus h_B(t_i) \tag{7.3}$$

where $h_A(t_i)$ and $h_B(t_i)$ are the underlying curve modelling the source and the delayed version of the source, respectively (see §4.2), and $\ominus = \{\times, -\}$ denotes either multiplication or subtraction – radio or optical data – respectively.

Given the observed data, $\vec{x_C}$, now the likelihood of our model reads

$$P(Data|Model) = \prod_{t_i=1}^{n} p(x_C(t_i) \mid \Delta, \{\alpha_j\}), \tag{7.4}$$

where

$$p(x_C(t_i) \mid \Delta, \{\alpha_j\}) \;\; = \;\; \frac{1}{2\pi\sigma_C^2(t_i)} \exp\left\{\frac{-(x_C(t_i) - h_C(t_i))^2}{2\sigma_C^2(t_i)}\right\}. \tag{7.5}$$

The negative log-likelihood (without constant terms) simplifies to

$$Q = \sum_{i=1}^{n} \frac{(x_C(t_i) - h_C(t_i))^2}{\sigma_C^2(t_i)}. \tag{7.6}$$

To avoid extrapolation when we apply a time delay to our underlying curve (4.6), we do not evaluate the goodness of fit over all observations:

$$Q = \sum_{u=b_1}^{n} \frac{(x_C(t_u) - h_C(t_u))^2}{\sigma_C^2(t_u)} \tag{7.7}$$

where b_1 is the index satisfying $t_{b_1} \geq t_1 + \Delta_{max}$.

In order to test the above formulation, we used Gaussian kernels and variable width as in K-V on DS-500 data. That is, we combine image A and B to obtain $\vec{x_C} = \vec{x_A} + M \times \vec{x_B}$. Then, we apply the above method (7.1)-(7.7) on $\vec{x_C}$, obviously hiding $\vec{x_A}$ and $\vec{x_B}$. We find that our method shows high bias, for instance on DS-500-5-G-0-N-0 data set, which has no gaps and no noise, we estimate a time delay of 616 days, when the true delay is 500 days (see Fig. 7.1); the range of analysis was $\Delta = [400, 700]$ with $k = 3$. Because we use a fixed tolerance $\lambda = 0.001$, we can see in Fig. 7.1 discontinuities on the Q curve as we discussed it in §5.3.1.

To find out why there appears high bias, we depict the full reconstructions for a combined curve $\vec{x_C}$, DS-500-5-G-0-N-0 data set. The parameters are set to $\lambda = 0.001$ and $k = 3$. First, in Fig 7.2a, M is set to its true value $1/1.44$ varying Δ; second, in Fig. 7.2b, Δ is set to its true value ($\Delta = 500$) varying M. Thus, we found that our method comes out with good reconstructions for $\vec{x_C}$ regardless of the time delay value and ratio. One expects that the best reconstruction is only when $\Delta = 500$, but it does not. In other words, Figure 7.2 suggests that we are not able to recover $\vec{x_A}$ and $\vec{x_B}$ with a given $\vec{x_C}$ just by measuring the goodness of fit (7.7) between $\vec{x_C}$ and $\vec{h_C}$ for given ranges of Δ and M. Consequently, one cannot estimate[2] the time delay accurately.

Therefore, more knowledge needs to be incorporated into the model since a single time series with observational errors is not enough. This problem remains a matter open to further research.

7.2.4 Multiple Time Delays

While working with optical data, we found that Goicoechea [26] claims the existence of multiple time delays in DS1 due to possible flares (supernova events). Therefore, this is another research line to follow to estimate time delays. It may help to clarify why one may obtain different estimates for the same quasar as we discussed it in §7.1.

[2]We did not test those methods described in §3, such as DCF, LNDCF, PRH and Dispersion spectra, because it is not straightforward to apply them to combined light curves.

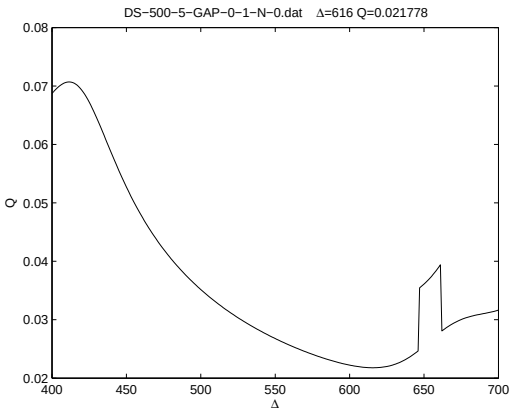


Figure 7.1: Q Curve on Artificial Data Set DS-500-5-G-0-N-0. We use Gaussian kernels with variable width $k = 3$, $M = 1/1.44$ and $\lambda = 0.001$. We explore time delay trials in the range [400, 700] with unitary increments.

7.2.5 Other Applications

The proposed methodology in this thesis (see §4 and §5) can be also applied to other problems, not only for time series from gravitational lensing. In fact, we have not applied our approach to other quasars – only to Q0957+561.

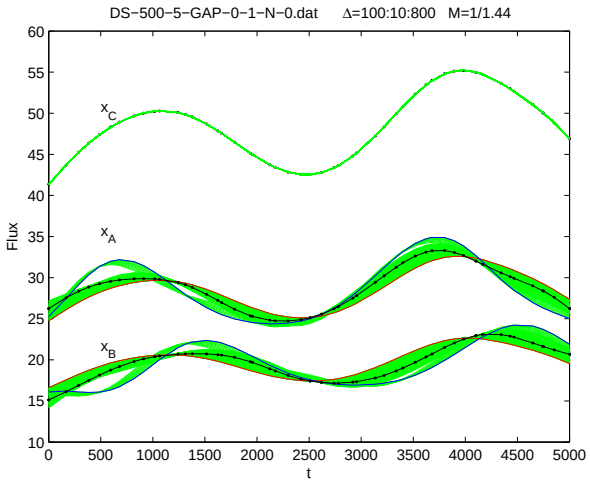
The missing data problems cover those cases where instrumental equipment fails, the observations are incorrectly recorded, weather conditions, sociological factors, etc. Therefore, the observed data are unevenly sampled. This restricts the use of Fourier analysis [72](§13.8). When there are missing data, the straightforward approach is interpolation. However, we have shown in §6.1.1 that interpolation is inaccurate on irregularly sampled time series since it adds false information about the source; see also [72](§13.8).

Problems with missing data are in almost all sciences, where the data availability is influenced by what is easy or feasible to collect; e.g., see [69, 14, 7, 81].

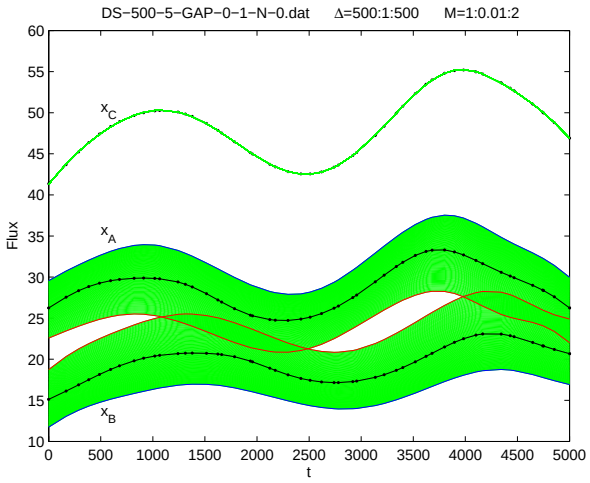
7.3 Chapter Summary

We have drawn the conclusions in §7.1. Comments and conclusions on artificial and real data are also given separately. Advantages and disadvantages of methods

were discussed. Several research directions were given in §7.2, where in some cases preliminary work has been presented. Furthermore, some hints were given with their respective references.



(a)



(b)

Figure 7.2: Reconstructions of Superimposed Light Curves and Their Components for DS-500-5-G-0-N-0. **(a)** M is fixed to its true value 1/1.44. The green curves, for $\mathbf{x_A}$, $\mathbf{x_B}$ and $\mathbf{x_C}$, are reconstructions for time delays in the range of 100 to 800 days with increments of 10 days; the red curves are when $\Delta = 100$, and the blue curves if $\Delta = 800$. The black ones when $\Delta = 500$ (true delay). **(b)** Here, Δ is fixed to 500, and M varies from 0.50 to 1.00 with increments of 0.01; from red curves to blue curves through green curves passing by the black curves that show the true value $M = 1/1.44 = 0.694$. As one can see, in both cases the reconstructions on the superimposed curve $\vec{x_C}$, at the top, are good reconstructions regardless of the time delay Δ and ratio M between $\vec{x_A}$ and $\vec{x_B}$.

Appendix A

Statistical Analysis

Let $\hat{\Delta}_j$ be a estimated time delay between a pair of light curves, where $j = 1, 2, ..., \eta$, and η is the number of time delay estimates.

The empirical mean is

$$\hat{\mu} = \frac{1}{\eta} \sum_{j=1}^{\eta} \hat{\Delta}_j, \tag{A.1}$$

and the empirical standard deviation is

$$\hat{\sigma} = \sqrt{\frac{1}{\eta - 1} \sum_{j=1}^{\eta} (\hat{\Delta}_j - \hat{\mu})^2}. \tag{A.2}$$

The mean squared error is given by

$$\text{MSE} = \frac{1}{\eta} \sum_{j=1}^{\eta} (\hat{\Delta}_j - \mu_0)^2, \tag{A.3}$$

where μ_0 is the true time delay.

The estimators $\hat{\mu}$ and $\hat{\sigma}$ are used to measure bias and variance of estimates respectively. Because the MSE is a squared loss function, the MSE can be decomposed as [3]

$$\begin{aligned} \text{MSE} &= E[(\hat{\Delta}_j - \mu_0)^2 | j] \\ &= [\hat{\mu} - \mu_0]^2 + \hat{\sigma}^2 \\ &= Bias^2 + Variance, \end{aligned} \tag{A.4}$$

where $E[\cdot]$ denotes the expected value so $E[\hat{\Delta}_j|j] = \hat{\mu}$. Equation (A.4) is known as the bias-variance decomposition [45, 35]. Therefore, the single estimator MSE deals with the bias-variance tradeoff.

The average of absolute error is

$$\text{AE} = \frac{1}{\eta}\sum_{j=1}^{\eta} |\hat{\Delta}_j - \mu_0|. \tag{A.5}$$

To measure the significance of estimates, we mainly use the t-test on a single population with unknown variance [18, 40], where the t-statistic is given by

$$\mathcal{T} = \frac{\hat{\mu} - \mu_0}{\hat{\sigma}/\sqrt{\eta}}. \tag{A.6}$$

Thus, we test the hypothesis H_0: $\hat{\mu} = \mu_0$ in (A.6). This is a two-tailed t-test with unknown variance. Since $\mathcal{T}$ has the Student's t-distribution with $\nu = \eta - 1$ degrees of freedom [18], the p-value is the cumulative probability given by its probability density function so

$$\mathcal{P}(x) = 2\int_{-\infty}^{x} \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} \frac{1}{\sqrt{\nu\pi}} \frac{1}{(1+\frac{t^2}{\nu})^{\frac{\nu+1}{2}}} dt, \tag{A.7}$$

where $\Gamma(\cdot)$ is the gamma function[1] [17], and if $\mathcal{T} < 0$ then $x = \mathcal{T}$ else $x = -\mathcal{T}$. Therefore, the larger $\mathcal{P}$, the higher the significance; i.e., the confidence in asserting H_0.

Nevertheless, $\mathcal{T}$ does not have a t-distribution when the population $\{\hat{\Delta}_j|j\}$ is not normal [3]. Since one can never be certain that is exactly normal, the above formulation is nearly correct even if the population is not far from normal, i.e., it is robust with the assumption of normality [3]. Moreover, by the Central Limit Theorem the sample mean is approximately normally distributed if the sample size is not small [1].

The 95% confidence intervals (CI) for $\hat{\mu}$ are given by $\hat{\mu} \pm \mathcal{C} \times \hat{\sigma}/\sqrt{\eta}$, where the constant $\mathcal{C}$ depends on the desired confidence level and the sample size [40]; e.g., see Table IIIa in [3]. For $n = 10$, we use $\mathcal{C} = 2.26$, and $\mathcal{C} = 1.96$ for $n > 25$.

[1] $\Gamma(n) = (n-1!)$ for an integer n, $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(1/2) = \sqrt{\pi}$

We use both $\mathcal{T}$ and $\mathcal{P}$ values, even when they are proportional to each other, because many researchers use only $\mathcal{T}$ to measure the statistical significance.

Typically, $\mathcal{P}$ is used to test whether or not the hypothesis H_0 can be accepted or rejected under the assumption of normality. Thus, if $\mathcal{P} < \alpha$ then the hypothesis is rejected otherwise accepted, where α is the confidence threshold[2]. Typical values of α are 0.05 and 0.01, that is, 95% and 99% of confidence respectively. For our purposes, we are also interested in $\mathcal{P}$ as a measurement of statistical significance; not only for accepting or rejecting the hypothesis with some significance level.

Some nonparametric tests include sign test and signed-rank test [3, 1]. These tests are distribution-free; they do not depend on the distribution of estimates $\{\hat{\Delta}_j|j\}$ as the t-test does. They are based on the median rather than the mean. Hence, if the distribution is symmetric then one may obtain similar results to the t-test when the sample size is not small [3].

[2]We point out that some authors refer to this threshold as the p-value, e.g., see [93]

Appendix B

Notation

Δ	time delay	8
n	quantity of samples in a data set	12
t	observational time	12
f	flux	13
M	ratio or offset between light curves	14
T_S	defines the size of the observational season $T_S \times \Delta$	14
s_1	samples per time interval Δ	14
z	separation between samples if regular sampling	14
$\vec{\sigma_A}$, $\vec{\sigma_B}$	observational errors (std)	15
$\vec{x_A}, \vec{x_B}$	observed image A and image B, respectively	21
Δt_{ij}	time differences (lags)	22
$\Delta\tau$	bin size	22
τ	bin centre	22
$P(\tau)$	quantity of observational pairs per bin	22
$\vec{y}$	combined light curve (single time series)	23
Δ_t	trial time delay	23

Δ_{min}	lower bound for Δ_t	23
Δ_{max}	upper bound for Δ_t	23
$\chi^2(\Delta_t)$	goodness of fit of PRH method	23
C_{ab}	covariance model	24
$V(\tau_{ab})$	structure function	24
$\mathcal{A}$, $\mathcal{B}$	parameters of structure function	24
D_1^2, $D_{4,2}^2$	dispersion spectra methods	25
$\mathbf{\Theta}$	Bayesian method parameters	27
ω	width of Gaussian kernel	32
$K(\cdot,\cdot)$	kernel function	32
T	training set	32
$\vec{\alpha}$	kernel weights	33
λ	SVD tolerance	39
k	parameter for variable width	40
θ	singular values to keep	46
n_p	number of individuals (hypotheses)	47
n_g	maximum number of generations	47
η	quantity of time delay estimates	109
$\hat{\mu}$	estimated mean over time delay estimates	109
$\hat{\sigma}$	estimated standard deviation over time delay estimates..	109
MSE	mean squared error over time delay estimates	109
μ_0	true time delay Δ	109
AE	mean of absolute error over time delay estimates	110
$\mathcal{T}$	t-statistic given by t-test	110

$\mathcal{P}$	p-value, cumulative probability	110

Bibliography

[1] T.W. Anderson and S.L. Sclove. *An Introduction to the Statistical Analysis of Data*. Houghton Mifflin Company, 1978.

[2] J. Bengio, o. Delalleau, and N. Le Roux. The curse of dimensionality for local kernel machines. Technical Report 1258, Département d'Informatique et Recherche Opérationnelle, Université de Montréal, May 2005.

[3] D.A. Berry and B.W. Lindgren. *Statistical Theory and Methods*. Brooks/Cole Publishing Company, 1998.

[4] G Beylkin and M. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *The SIAM Journal on Scientific Computing*, 26(6):2133–2159, 2005.

[5] J. Biethahn and V. Nissen. *Evolutionary Algorithms in Management Applications*. Springer-Verlag, Berlin, 1995.

[6] E. Bonilla-Huerta, B. Duval, and Jin-Kao Hao. A hybrid ga/svm approach for gene selection and classification of microarray data. In Franz Rothlauf et al., editor, *Applications of Evolutionary Computing*, EvoWorkshops 2006, LNCS 3907, pages 34–44. Springer-Verlag, 2006.

[7] W. Bridewell, P. Langley, S. Racunas, and S Borrett. Learning process models with missing data. In *Machine Learning: ECML 2006*, Lecture Notes in Artificial Intelligence (LNAI 4212), pages 557–565. Springer-Verlag, September 2006.

[8] K. Brown. *Diversity in Neural Networks Ensembles*. PhD thesis, School of Computer Science, University of Birmingham, UK, 2004.

[9] I. Burud, P. Magain, S. Sohy, and J. Hjorth. A novel approach for extracting time-delays from light curves of lensed quasar images. *Astronomy and Astrophysics*, 380(2):805–810, 2001.

[10] E. Cant-Paz and D.E. Goldberg. Efficient parallel genetic algorithms: theory and practice. *Computer Methods in Applied Mechanics and Engineering*, 186(1):221–238, 2000.

[11] A. J. Chipperfield, P. J. Fleming, and C. M. Fonseca. Genetic algorithm tools for control systems engineering. In *Proc.1st Int. Conf. Adaptive Computing in Engineering Design and Control*, pages 128–133. Plymouth Engineering Design Centre,UK, 1994. http://www.shef.ac.uk/acse/research/ecrg/getgat.html.

[12] A. J. Chipperfield, P. J. Fleming, H Pohlheim, and C. M. Fonseca. *Genetic Algorithm Toolbox for use with MATLAB*. Automatic Control and Systems Engineering, University of Sheffield, 1.2 edition, 1996. http://www.shef.ac.uk/acse/research/ecrg/getgat.html.

[13] W.N. Colley, R.E. Schild, C. Abajas, D. Alcalde, Z. Aslan, I. Bikmaev, V. Chavushyan, L. Chinarro, J.P. Cournoyer, R. Crowe, V. Dudinov, A.K.D. Evans, Y.B. Jeon, L.J. Goicoechea, O. Golbasi, I. Khamitov, K. Kjernsmo, H.J. Lee, J. Lee, K.W. Lee, M.G. Lee, O. Lopez-Cruz, E. Mediavilla, A.F.J. Moffat, R.Mujica, A. Ullan, J. Munoz, A. Oscoz, M.G. Park, N. Purves, O. Saanum, N. Sakhibullin, M. Serra-Ricart, I. Sinelnikov, R. Stabell, A. Stockton, J. Teuber, R. Thompson, H.S. Woo, and A. Zheleznyak. Around-the-clock observations of the Q0957+561 A,B gravitationally lensed quasar. II. results for the second observing season. *Astronomy and Astrophysics*, 587(1):71–79, 2003.

[14] R. Cook and D. McLeish, editors. *Workshop on Missing Data Problems*. Fields Institute, Toronto, Canada, 2004. http://www.fields.utoronto.ca/programs/scientific/04-05/missing-data/.

[15] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill, second edition, 2002.

[16] P. Courrieu. Fast computation of moore-penrose inverse matrices. *Neural Information Processing – Letters and Reviews*, 8(2):25–29, 2005.

[17] G. Cowan. *Statistical Data Analysis*. Oxford University Press, 1998.

[18] E.J. Dudewicz and S.N. Mishra. *Modern Mathematical Statistics*. John Wiley and Sons, 1988.

[19] R.A. Edelson and J.H. Krolik. The discrete correlation-function - a new method for analyzing unevenly sampled variability data. *Astrophysical Journal*, 333(2):646–659, 1988.

[20] A. Eigenbrod, F. Courbin, S. Dye, G. Meylan, D. Sluse, C. Vuissoz, and P. Magain. COSMOGRAIL: the COSmological MOnitoring of GRAvItational Lenses. II. SDSS J0924+0219: the redshift of the lensing galaxy, the quasar spectral variability and the Einstein rings. *Astronomy and Astrophysics*, 451:747–757, June 2006.

[21] A. Eigenbrod, F. Courbin, C. Vuissoz, G. Meylan, P. Saha, and S. Dye. COSMOGRAIL: The COSmological MOnitoring of GRAvItational Lenses. I. How to sample the light curves of gravitationally lensed quasars to measure accurate time delays. *Astronomy and Astrophysics*, 436:25–35, June 2005.

[22] Y. Engel and S. Mannor. The Kernel Recursive Least-Squares Algorithm. *IEEE Transactions on Signal Processing*, 52(8):2275–2285, Aug 2004.

[23] L.J. Fogel and W. Atmar, editors. *Proceedings of the Second Annual Conference on Evolutionary Programming*. Evolutionary Programming Society, 1992.

[24] L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial intelligence through simulated evolution*. John Wiley & Sons, 1966.

[25] R. Gil-Merino, L. Wisotzki, and J. Wambsganss. The double quasar HE 1104-1805: A case study for time delay determination with poorly sampled light curves. *Astronomy and Astrophysics*, 381(2):428–439, 2002.

[26] L.J. Goicoechea. Multiple delays in QSO 0957+561: observational evidence and interpretation. *Monthly Notices of the RAS*, 334(1):905–911, 2002.

[27] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[28] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, second edition, 1989.

[29] M. V. Gorenstein, I. I. Shapiro, and E. E. Falco. Degeneracies in parameter estimates for models of gravitational lens systems. *Astrophysical Journal*, 327:693–711, April 1988.

[30] A.L. Graps. An Introduction to Wavelets. *IEEE Computational Sciences and Engineering*, 2(2):50–61, 1995. Available online at http://www.amara.com/IEEEwave/IEEEwavelet.html.

[31] D.B. Haarsma, J.N. Hewitt, J. Lehar, and B.F. Burke. The 6 centimeter light curves of B0957+561, 1979-1994: New features and implications for the time delay. *Astrophysical Journal*, 479(1):102–118, 1997.

[32] D.B. Haarsma, J.N. Hewitt, J. Lehar, and B.F. Burke. The radio wavelength time delay of gravitational lens 0957+561. *Astrophysical Journal*, 510(1):64–70, 1999.

[33] M. Harva and S. Raychaudhury. A new Bayesian look at estimating gravitational lens time delays. In the School of Physics and University of Birmingham Astronomy, editors, *RAS National Astronomy Meeting: Bayesian techniques in astronomy*, UK, April 2005. Royal Astronomical Society.

[34] M. Harva and S. Raychaudhury. Bayesian estimation of time delays between unevenly sampled signals. In *IEEE International Workshop on Machine Learning for Signal Processing*. IEEE, 2006. Accepted.

[35] T. Hastie, R Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[36] S. Haykin. *Neural Networks: a Comprehensive Foundation*. Prentice Hall, 1999.

[37] J.H. Holland. *Adaption in natural and artificial systems*. University of Michigan Press, 1975. reprinted in 1992 by MIT Press.

[38] T. Howley and M.G. Madden. The genetic kernel support vector machine: Description and evaluation. *Artificial Intelligence Review*, 24(3):379–395, 2005.

[39] S. Janson and E. Alba. Hierarchical cellular genetic algorithm. In Jens Gottlieb and Gunther R. Raidl, editors, *Evolutionary Computation in Combinatorial Optimization*, EvoCOP 2006, LNCS 3906, pages 111–122. Springer-Verlag, 2006.

[40] R.A. Johnson and G.K. Bhattacharyya. *Statistics: Principles and methods*. John Wiley & Sons, Inc., fifth edition, 2006.

[41] S. S. Keerthi, O. Chapelle, and D. DeCoste. Building Support Vector Machines with Reduced Classifier Complexity. *Journal of Machine Learning Research*, 7:1493–1515, 2006.

[42] C. S. Kochanek, C.R. Keeton, and B.A. McLeod. The importance of Einstein Rings. *Astrophysical Journal*, 547:50–59, 2001.

[43] C. S. Kochanek and P. L. Schechter. The Hubble Constant from Gravitational Lens Time Delays. In W. L. Freedman, editor, *Measuring and Modeling the Universe*, pages 117–+, 2004.

[44] C. S. Kochanek, P. Schneider, and J. Wambsganss. Gravitational Lensing: Strong, Weak & Micro. In G. Meylan, P. Jetzer, and P. North, editors, *Proceedings of the 33rd Saas-Fee Advanced Course*. Springer-Verlag, 2004.

[45] R. Kohavi and D.H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In *Thirteenth International Conference on Machine Learning*, pages 275–283. Morgan Kaufmann, 1996.

[46] E. A. Koptelova, V. L. Oknyanskij, and E. V. Shimanovskaya. Determining time delay in the gravitationally lensed system QSO2237+0305. *Astronomy and Astrophysics*, 452:37–46, June 2006.

[47] J.R. Koza. *Genetic Programming: On the programming of computers by means of natural selection*. The MIT Press, 1992.

[48] J. Kubalik and J. Faigl. Iterative prototype optimisation with evolved improvement steps. In P. Collet et al., editor, *Genetic Programming*, EuroGP 2006, LNCS3905, pages 154–165. Springer-Verlag, 2006.

[49] T. Kundic, E.L. Turner, W.N. Colley, J.R. Gott-III, J.E. Rhoads, Y. Wang, L.E. Bergeron, K.A. Gloria, D.C. Long, S. Malhorta, and J. Wambsganss. A robust determination of the time delay in 0957+561A,B and a measurement of the global value of Hubble's constant. *Astrophysical Journal*, 482(1):75–82, 1997.

[50] H. Kwon and N.M. Nasrabadi. Kernel matched subspace detectors for hyperspectral target detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):178–194, 2006.

[51] R.M. Larsen and P.C. Hansen. Efficient implementations of the sola mollifier method. *Astronomy and Astrophysics*, 121:587–598, 1997.

[52] J. Lehar, J.N. Hewitt, D.H. Roberts, and B.F. Burke. The radio time delay in the double quasar 0957+561. *Astrophysical Journal*, 384:453–466, 1992.

[53] R.E. Lenski, C. Ofria, R.T. Pennock, and C. Adami. The evolutionary origin of complex features. *Nature*, 423:139–144, 2003.

[54] T.M. Mitchell. *Machine Learning*. Computer Science. McGraw-Hill, 1997.

[55] I.T. Nabney. *NETLAB: Algorithms for Pattern Recognition*. Advances in Pattern Recognition. Springer, 2002.

[56] P.R. Newbury and R.J. Spiteri. Inverting gravitational lenses. *SIAM Review*, 44(1):111–130, 2002. Society for Industrial and Applied Mathematics.

[57] A. Oscoz, D. Alcalde, M. Serra-Ricart, E. Mediavilla, C. Abajas, R. Barrena, J. Licandro, V. Motta, and J.A. Munoz. Time delay in QSO 0957+561 from 1984-1999 optical data. *Astrophysical Journal*, 552(1):81–90, 2001.

[58] A. Oscoz, E. Mediavilla, L.J. Goicoechea, M. Serra-Ricart, and J. Buitrago. Time delay of qso 0957+561 and cosmological implications. *Astrophysical Journal*, 479(2):L89–L82, 1997.

[59] J.E. Ovaldsen, J. Teuber, R.E. Schild, and R. Stabell. New aperture photometry of QSO 0957+561; application to time delay and microlensing. *Astronomy and Astrophysics*, 402(3):891–904, 2003.

[60] D. Paraficz, J. Hjorth, I. Burud, P. Jakobsson, and Á. Elíasdóttir. Microlensing variability in time-delay quasars. *Astronomy and Astrophysics*, 455:L1–L4, August 2006.

[61] R.L. Parker. Understanding Inverse Theory. *Annual Review of Earth and Planetary Science*, 5:35–64, 1977.

[62] J. Pelt. Estimation of time delays from unresolved photometry. In L. J. Goicoechea, editor, *25 Years After the Discovery: Some Current Topics on Lensed QSOs*, February 2005.

[63] J. Pelt, J. Hjorth, S. Refsdal, R. Schild, and R. Stabell. Estimation of multiple time delays in complex gravitational lens systems. *Astronomy and Astrophysics*, 337(3):681–684, 1998.

[64] J. Pelt, R. Kayser, S. Refsdal, and T. Schramm. Time delay controversy on QSO 0957+561 not yet decided. *Astronomy and Astrophysics*, 286(1):775–785, 1994.

[65] J. Pelt, R. Kayser, S. Refsdal, and T. Schramm. The light curve and the time delay of QSO 0957+561. *Astronomy and Astrophysics*, 305(1):97–106, 1996.

[66] J. Pelt, R. Schild, S. Refsdal, and R. Stabell. Microlensing on different timescales in the light curves of QSO 0957+561 A,B. *Astronomy and Astrophysics*, 336(3):829–839, 1998.

[67] K.B. Petersen and M.S. Pedersen. *The Matrix Cookbook*. http://matrixcookbook.com, Feb 2006.

[68] T. Phienthrakul and B. Kijsirikul. Evolutionary strategies for multi-scale radial basis function kernels in support vector machines. In Hans-Georg Beyer et al., editor, *Genetic and Evolutionary Computation Conference (GECCO)*, volume 1, pages 905–911, 2005.

[69] F.P. Pijpers. The determination of time delays as an inverse problem - the case of the double quasar 0957+561. *Monthly Notices of the RAS*, 289(4):933–944, 1997.

[70] B. Pindor. Discovering Gravitational Lenses through Measurements of Their Time Delays. *Astrophysical Journal*, 626:649–656, June 2005.

[71] W.H. Press, G.B. Rybicki, and J.N. Hewitt. The time delay of gravitational lens 0957+561, I. Methodology and analysis of optical photometric data. *Astrophysical Journal*, 385(1):404–415, 1992.

[72] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, second edition, 2002.

[73] M.C. Rabello-Soares, S. Basu, and J. Christensen-Dalsgaard. On the choice of parameters in solar-structure inversion. *Monthly Notices of the RAS*, 309:35–47, 1999.

[74] S. Refsdal. On the possibility of determining Hubble's parameter and the masses of galaxies from the gravitational lens effect. *Monthly Notices of the RAS*, 128:307–310, 1964.

[75] S. Refsdal. On the possibility of determining the distances and masses of stars from the gravitational lens effect. *Monthly Notices of the RAS*, 134:315–319, 1966.

[76] R.G. Regis and C.A. Shoemaker. Local function approximation in evolutionary algorithms for the optimization of costly functions. *IEEE Transactions on Evolutionary Computation*, 8(5):490–505, 2004.

[77] J.E. Rowe and D. Hidovic. An evolution strategy using a continuous version of the gray-code neighbourhood distribution. In K. Deb et al., editor, *Genetic and Evolutionary Computation Conference (GECCO)*, volume 1, pages 725–736. Springer-Verlag, 2004.

[78] G.B. Rybicki and W.H. Press. Class of fast methods for processing irregularly sampled or otherwise inhomogeneous one-dimensional data. *Physical Review Letters*, 74(1):1060–1063, 1995.

[79] P. Saha. Gravitational Lensing. *Encyclopedia of Astronomy and Astrophysics*, 2000.

[80] J.G. Sánchez-Velazco. *Gendered Selection Strategies for Genetic Algorithms.* PhD thesis, School of Computer Science, University of Birmingham, UK, 2006.

[81] G. Sanguinetti and N. Lawrence. Missing data in kernel pca. In *Machine Learning: ECML 2006*, Lecture Notes in Artificial Intelligence (LNAI 4212), pages 751–758. Springer-Verlag, September 2006.

[82] J.A. Scales, M.L. Smith, and S. Treitel. *Introductory Geophysical Inverse Theory.* Samizdat Press, 1997. Availabe on line at http://samizdat.mines.edu.

[83] R.E. Schild and D.J. Thomson. The Q0957+561 time delay from optical data. *Astronomical Journal*, 113(1):130–135, 1997.

[84] B. Schölkopf. Introduction to Kernel Methods. In Raffaele Cerulli Nello Cristianini and John Shawe-Taylor, editors, *The Analysis of Patterns*, Erice, Italy, Oct-Nov 2005. Centre "Ettore Majorana" for Scientific Culture. http://www.analysis-of-patterns.net/pdf_slides/Bernhard_Scholkopf.pdf.

[85] H.P. Schwefel. *Evolution and optimum seeking.* John Wiley & Sons, 2004.

[86] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[87] W.M. Spears. *Evolutionary Algorithms: The role of Mutation and Recombination.* Natural computing. Springer-Verlag, 2000.

[88] Zhenguo Tu and Yong Lu. A Robust Stochastic genetic Algorithm (StGA) for Global Numerical Optimization . *IEEE Transactions on Evolutionary Computation*, 8:456–470, October 2004.

[89] S. A. van de Geer. *Empirical Processes in M-Estimation*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge Press, 2000.

[90] D. E. Vanden Berk, B. C. Wilhite, R. G. Kron, S. F. Anderson, R. J. Brunner, P. B. Hall, Ž. Ivezić, G. T. Richards, D. P. Schneider, D. G. York, J. V. Brinkmann, D. Q. Lamb, R. C. Nichol, and D. J. Schlegel. The Ensemble Photometric Variability of ˜25,000 Quasars in the Sloan Digital Sky Survey. *Astrophysical Journal*, 601:692–714, February 2004.

[91] S.V.N. Vishwanathan, N.N. Schraudolph, and A.J. Smola. Step Size Adaptation in Reproducing Kernel Hilbert Space. *Journal of Machine Learning Research*, 7:1107–1133, June 2006.

[92] D. Walsh, R. F. Carswell, and R. J. Weymann. 0957 + 561 A, B - Twin quasistellar objects or gravitational lens. *Nature*, 279:381–384, May 1979.

[93] L. Wasserman. *All of Statistics: A concise course in statistical inference*. Statistics. Springer, 2003.

[94] X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102, Jul 1999.

[95] X. Yao, H.P. Schwefel, B.T. Zhang, and M. Amos, editors. *Advances in Natural Computation*, volume 1-5. World Scientific Publishing, 2006.

Index

Artificial data, 14, 54, 96
  DS-5, 15, 63
  DS-500, 14, 54
  Harva data, 28, 78
  PRH data, 25, 68

Bayesian method, 27
  MCMC, 27

Confidence intervals, 19
  CI, 19
Cross correlation, 22
  DCF, 22
  LNDCF, 22
Cross-validation, 40, 48
  five-fold-CV, 40

Data
  artificial, 14, 25, 28
  real, 12
Data-driven, 4
Dispersion spectra, 25

Einstein ring, 8
Evolution, 44
  operators, 44
    mutation, 44, 50
    recombination, 44, 50
    selection, 44, 50
Evolutionary algorithm, 45
  EA, 44, 47
  EA-CV, 48, 52
  EA-LL, 48, 52
  EA-M, 47
  EA-R, 47
  evolution operators, 50
  fitness function, 48
  fitness landscape, 49
  representation, 47
Evolutionary computation, 44
  evolution strategies (ES), 44
    (1+1)ES, 45, 100
  evolutionary programming (EP), 44
  genetic algorithms (GAs), 44
  genetic programming (GP), 44

Flux, 13
  mag, 13
  millijanskys, 12

General linear least squares, 35
General theory of relativity, 1, 7
Gravitational lensing, 6
  gravitational lens, 1, 8, 10

Imaging device

CCD, 13
Interpolation, 21
linear, 21, 22
splines, 22
Inverse theory, 34
Backus-Gilbert method, 34

Kernel-based method, 36
fixed width (K-F), 40, 42, 52
variable width (K-V), 40, 42, 52
weights, 38
Kernels, 32
basis functions, 33
convex optimisation, 34
dot product, 32
eigen-decomposition, 34
feature space, 32
Gaussian kernels, 32
Gram matrix, 32
kernel machines, 34
kernel trick, 32
KRLS, 102
methods, 32
RBF kernel, 34

Learning
batch learning, 34
learning machines, 34
on-line learning, 33
supervised learning, 31
Log-likelihood, 37, 48
LL, 48

Machine Learning, 30
ML, 30
MACHOs, 1
mean squared error, 109
Microlensing, 1, 13
Multiple time delays, 105

PRH method, 23

Quasar, 1
Q0957+561, 2, 10, 99
Optical data, 12, 85
Radio data, 12, 79

Regression, 33
Regularisation, 39, 45

Singular value decomposition, 38
Moore-Penrose inverse, 38
SVD, 38, 52
tolerance λ, 39
Speedup, 101
Statistical analysis, 109
absolute error, 110
empirical mean, 109
empirical standard deviation, 109
p-value, 110
t-statistic, 110
Superimposed light curves, 103
Support vector machines (SVM), 34

t-test, 61, 69, 110
Theoretical analysis, 103
Time complexity, 42, 52

algorithm efficiency, 42
running time, 42
Time delay, 2, 7
cosmological significance, 8

Wavelets, 35