Bayesian estimation of undetectable reverberation lags

Tom Mitchell Elliott

Supervised by Dr. Brendon Brewer

A thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Science with Honours in Statistics, The University of Auckland, 2013.

Abstract

Reverberation mapping is a technique used to infer the lag, τ, between two light curves obtained from two different regions of active galactic nuclei (AGN). The lag can later be used in the estimation of the mass of the central black hole. Traditional reverberation mapping methods involve the use of the cross correlation function (CCF), and more recently Bayesian models to estimate the lag, however these require long, expensive observational campaigns to obtain the data necessary to infer the lag accurately. The difficulty in obtaining the data has led to the development of a new technique where the CCFs from multiple poorly-measured AGN are combined, or stacked, to infer a typical lag for the sample. Through simulation, this method was shown to recover the lag, although the population lag variability and uncertainty of the lag estimate due to the sparse data were difficult to differentiate from the stacked CCF. Therefore, we proposed using a Bayesian hierarchical model as an alternative, with the advantage of obtaining individual estimates for population variability and estimation error. After first implementing a simple model to show our method could be used to recover the population mean lag, we constructed a model that would analyse multiple AGN and infer the population distribution of lags. We were able to recover the population parameters, and their associated uncertainties, and show that the population variability was separable from the uncertainty in estimating the mean lag.

Acknowledgements

Firstly, I give my sincerest thanks to my supervisor, Dr. Brendon Brewer. Whether astrophysical concepts, coding up of a model, or computational demand was holding me back, you would always lend your time and help me to find a solution.

To all of the lecturers who have contributed both to my passion and understanding of statistics, thank you; and to all of my fellow students who have helped me to procrastinate and potentially keep me sane, thank you very much.

Finally, my parents. Mum and Dad, without your support—both financial and psychological—I would not have been able to pursue postgraduate study. My success is a testament to your encouragement and compassion throughout my life.

Thank you.

Contents

List of Tables

List of Figures

Glossary

accretion The process by which surrounding matter falls into a compact object (for example, a black hole), emitting huge levels of electromagnetic radiation.

electromagnetic radiation A form of wave-energy, which includes radio waves, infrared heat, visible light, x-rays, and gamma rays.

light curve A time series of light brightness measurements.

List of Acronyms

AGN active galactic nuclei.

AR(1) first order autoregressive process.

BLR broad line region.

CAR(1) first order continuous autoregressive process.

CCF cross correlation function.

DAG directed acyclic graph.

HMC Hamiltonian or Hybrid Monte Carlo.

MCMC Markov chain Monte Carlo.

MDS medium-deep survey.

Chapter 1

Introduction

Supermassive black holes are the powerhouses of most, if not all, large galaxies. Because of this, there is a lot of interest among astronomers in the physical properties of active galactic nuclei (AGN). One such property is the mass of central black holes, which influences properties of both the black holes themselves, and of the galaxies they power (Beckmann & Shrader, 2012). The main issue with studying black holes and how they influence galaxies is that they cannot be observed directly. Because of this, several techniques have been used to indirectly observe black holes and make inferences about properties such as their mass.

Being an integral aspect of astrophysics, there has been a lot of interest in estimating the mass of black holes. One common method of doing this is to observe the electromagnetic output from two different locations in AGN, and use this to make inferences about the black hole's mass. This technique is called reverberation mapping, first discussed by Blandford & McKee (1982), and is used to estimate the size of the broad line region (BLR), which is proportional to the size of the black hole.

1.1 The Structure and Behaviour of AGN

Active galaxies are identified by their AGN, which are prominent due to their high luminosity. The reason for the high luminosity is that they undergo accretion, the process by which surrounding matter falls into a compact object (Beckmann & Shrader, 2012). As a result of this process, half of the gravitational potential energy of the matter, which mainly consists of gas, is converted into electromagnetic radiation, which can be observed by terrestrial and orbital telescopes (Beckmann & Shrader, 2012; Peterson, 2008).

The AGN are therefore of high interest to astronomers, and their properties are largely influ-

enced by the mass of the central black hole which produces the accretion disk, and is the source of the electromagnetic output (Beckmann & Shrader, 2012). A simplified structure of an AGN is shown in Figure 1.1, which examines the relationship between the central black hole, the accretion disk formed by the accretion process, as well as the BLR clouds.

The central black hole, denoted by BH in the diagram, is invisible to direct observation. However, when it is close to other objects, accretion takes place, which produces the accretion disk (AD), and emits immense levels of radiation. It is this radiation that makes the AGN visible to telescopes. Surrounding the black hole and accretion disk is a collection of clouds called the BLR. The size of this region is related to the mass of the black hole (Peterson, 2008; Beckmann & Shrader, 2012), so estimating its size is an indirect method of estimating the black hole's mass.

The mass, m, of a black hole is related to the size of the BLR through the equation

$$m \approx \frac{v^2 c\tau}{G} \tag{1.1}$$

where c is the speed of light, G is Newton's gravitational constant, and v is the approximate orbital velocity of the BLR clouds. τ is related to the distance between the accretion disk and the BLR clouds, or alternatively the light travel time between these two regions. This is discussed in more detail by Peterson (2008) and Beckmann & Shrader (2012). The main focus of this thesis will be on the statistical aspects of estimating τ from the reverberation mapping data, rather than the

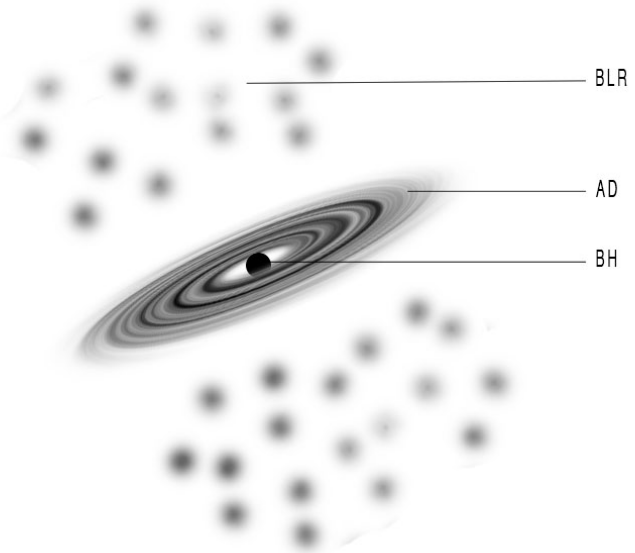

Figure 1.1: Simplified structure of an AGN, showing the central Black Hole (BH), Accretion Disk (AD), and the clouds of the Broad Line Region (BLR). The continuum and BLR light curves come from the accretion disk and BLR respectively, and are shown in Figure 1.2.

physical interpretation of the results.

Once the structure of the AGN has been established, we can observe the different regions using telescopes, and build up data sets of the luminosity, or brightness, at different wavelengths.

1.2 Reverberation Mapping

The light luminosity from the accretion disk is not constant, and instead consists of temporal variations caused by complex processes (Kelly et al., 2009). It is known, however, that the BLR light curve responds to the variations in the accretion disk's output, with a lag between the two that represents the time taken, in days, for the light to travel from the accretion disk to the BLR clouds. The lag is therefore proportional to the size of the BLR. By measuring the luminosity of the continuum and BLR light curves, it is possible to estimate the lag, τ, between them, a technique known as reverberation mapping.

The main goal for any reverberation mapping data analysis is to estimate the lag between two light curves. The continuum light curve is constantly varying, and these variations are echoed by the BLR light curve after a time delay, or lag, denoted by τ. The most common technique used to estimate the lag is calculation of the cross correlation function (CCF) between the two curves (e.g., Barth et al. (2011)).

Reverberation mapping has also been implemented using Bayesian methodology, where instead of finding the peak of the CCF, the posterior probability distribution of the lag, τ, and any other parameters, is calculated given the observed data. There have been several implementations of this, including the contributions by Kelly et al. (2009), Pancoast et al. (2012) and Zu et al. (2011), who have all been able to show that a Bayesian approach can use a first order continuous autoregressive process (CAR(1)) likelihood model for the continuum light curve data.

Figure 1.2 shows a simple, simulated reverberation mapping data set. The points with error bars are pseudo observations, similar to what reverberation mapping experiments actually obtain, while the dotted lines are representative of the true brightness of the continuum and BLR light curves. The errors associated with the measurements are supplied with the data set. From the true light curves, we can see how the second light curve tracks the first, but is delayed by a lag, $\tau = 10$ days. This is the parameter we wish to estimate, and is typically between 3–10 days; however, because of the sparse observations, it would be difficult to estimate τ accurately.

To obtain the desired accuracy, reverberation mapping experiments require small time intervals between observations that are smaller than the time lag between the continuum and BLR light curves. They also require long observational periods to get accurate estimation of the lag, as well

as significant variations in the continuum light curve that can be seen in the lagged response. Unfortunately, many objects emit reasonably flat light curves from which reverberation mapping cannot be done accurately.

1.3 The Problem of Sufficient Data

Only a few AGN have been mapped well enough to get accurate estimates of the black hole masses. The main reason for this is the difficulty in obtaining the data. While the continuum light curve is reasonably simple to observe due to its high luminosity, there are still patches in the data due to weather disruption and the restriction that observations can only be made at night. The BLR light curve, however, is more difficult to observe as it is significantly dimmer that the continuum. Therefore, long observational campaigns are limited by funding and the availability of telescopes capable of measuring the brightness accurately (Peterson, 2008).

However, in a recent paper by Fine et al. (2012), a new method of analysing AGN was implemented. Instead of taking only observational data for a single object, the paper discusses the use of multiple AGN with poorly observed BLR light curves, and combining the results from CCF analysis to estimate the mean lag, τ, for the group of objects studied (Fine et al., 2012, 2013). Their technique, referred to as stacked cross-correlation, was used to successfully recover the lag for simulated data where the BLR light curve had only two observations per object, but there were 30 objects. After repetitive simulations, Fine et al. (2012) demonstrated their technique by

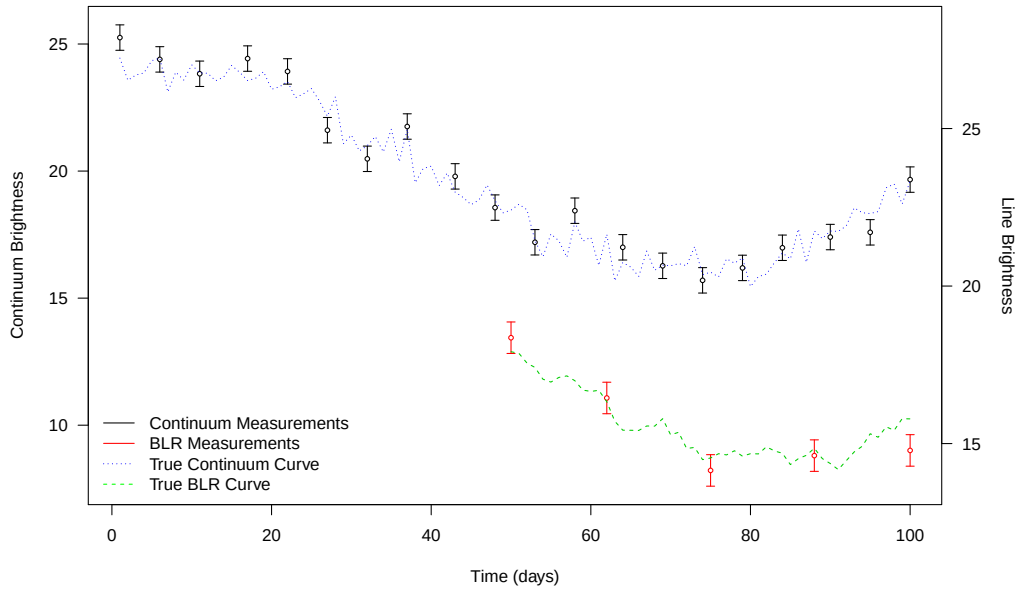

Figure 1.2: A simplified reverberation mapping data set, showing the continuum and BLR light curves for a hypothetical AGN, with a lag, τ, set to 10 days.

analysing real AGN data collected by the Pan-STARRS medium-deep survey (MDS). They were able to find clear peaks in the stacked CCFs, allowing them to estimate the typical value of τ in the sample.

1.4 A Hierarchical Bayesian Alternative

While the stacked cross-covariance method has been shown to work through simulation by Fine et al. (2012), there are still some areas of uncertainty. Firstly, the width of the stacked CCF cannot be separated into the experimental error associated with measuring the light brightness or gaps in the data, and the real variability of lags in the population. This means that it is difficult to give an estimate of the accuracy of the lag estimate and explain the variability of object lags in the population. Therefore, we decided to use a Bayesian approach to implement a hierarchical model to describe the processes occurring and determine the lag.

In a Bayesian approach, rather than finding the maximum likelihood values, or the peak of a CCF, we calculate the probability distribution of the unknown parameters, $\boldsymbol{\theta}$, in light of the observed data, $\boldsymbol{X}$, which is termed the posterior distribution, which is calculated by using Bayes' theorem:

$$\mathbb{P}\left(\boldsymbol{\theta} \,|\, \boldsymbol{X}\right) \propto \mathbb{P}\left(\boldsymbol{X} \,|\, \boldsymbol{\theta}\right) \mathbb{P}\left(\boldsymbol{\theta}\right), \tag{1.2}$$

where $\mathbb{P}\left(\boldsymbol{\theta}\right)$ is the prior probability distribution of the parameters, and $\mathbb{P}\left(\boldsymbol{X} \,|\, \boldsymbol{\theta}\right)$ is the likelihood of the data given some value of $\boldsymbol{\theta}$. The prior probability distribution can be subjective—based on personal belief or from previous experiments and observations—or can be chosen to have minimal effect on the posterior probabilities, often referred to as non-informative priors.

One additional component of using a Bayesian approach to model data is that we need to specify the entire model, including many nuisance parameters, which can later be marginalised out, allowing us to calculate the posterior distributions of the parameters we are interested in.

For the reverberation mapping data, it would make sense to use a hierarchical model if we can assume that there is a relationship between the parameters of the different AGN. This type of model will allow for a population distribution of lags, which has a mean μ_τ and variance σ_τ. Subsequently, each object, i, will have its own lag, τ_i, which is theoretically drawn from the population distribution.

In our model, we will eventually have a large number of parameters because each value of the discrete approximation to the continuum light curve will need to be estimated. Because of this, Markov chain Monte Carlo (MCMC) sampling techniques will be used to draw samples from the posterior distribution and make inferences as required. We will use a software package that uses

Hamiltonian or Hybrid Monte Carlo (HMC), called STAN (Stan Development Team, 2013a), which uses Hamiltonian dynamics to explore the posterior distribution rather than the usual Metropolis Hastings algorithm which uses proposal densities and an acceptance probability (Neal, 1993).

This thesis will first discuss how using a hierarchical model can be used to recover population parameters when individual objects are poorly measured. To do this, we will design a simple simulation experiment similar to the reverberation data we will eventually be using. After this, we will implement a model for a full reverberation mapping data set, as well as run some simulations to test our hypothesis that, even though the individual objects are poorly measured and the lag cannot be recovered, as in Figure 1.2, when we combine the data from multiple objects using a hierarchical model we can recover the population distribution of lags as an alternative to using stacked CCFs as done by Fine et al. (2012, 2013).

Chapter 2

A Simple Hierarchical Model

Before building a full-scale model to deal with multiple reverberation mapping data sets, we decided to trial our hierarchical approach using a simplified, hypothetical situation that would allow us to compare stacked CCFs with hierarchical Bayesian methods. To do this, we imagined that the continuum light curve emitted a single Gaussian pulse at a known time, and the data is of the responding light curve, which echoed the initial pulse after a time lag.

We gave this initial Gaussian pulse a width of 1 and unknown amplitude, α, and applied a positive shift, τ, to the response. The true response is a smooth curve, from which we simulated noisy observations with a known variance, β. Therefore, the output curve can be expressed as

$$Y(t) = \alpha \exp\left\{\frac{(t-\tau)^2}{2(1+\nu^2)}\right\}, \tag{2.1}$$

where ν represents the level of convolution or blurring in the response. The measurements were then assumed to be drawn from a normal distribution,

$$y(t) \sim \mathcal{N}\left(Y(t), \beta^2\right). \tag{2.2}$$

Although this situation is not very illustrative of reverberation mapping for a single AGN where the continuum variations are more complex, our aim was to use the data from multiple objects that each, on their own, could not be modelled accurately, to check that our methodology would work.

To do this, we replicated (2.1) for N different objects, and denoted each object using the subscript $i \in \{1, \ldots, N\}$. Further, each object had M observations at different times, denoted by the subscript $j \in \{1, \ldots, M\}$. Using this notation, we used y_{ij} to represent an observation of

Parameter	Population median	Population variance	Distribution
Positive shift, τ	$\mu_\tau = 11$	$\sigma_\tau = 0.2$	$\log\mathcal{N}\left(\mu_\tau, \sigma_\tau^2\right)$
Amplitude, α	$\mu_\alpha = 1.0$	$\sigma_\alpha = 0.1$	$\log\mathcal{N}\left(\mu_\alpha, \sigma_\alpha^2\right)$
Width, ν	$\mu_\nu = 1.5$	$\sigma_\nu = 0.12$	$\log\mathcal{N}\left(\mu_\nu, \sigma_\nu^2\right)$

Table 2.1: Population parameters and their distributions for the simple model, accompanied by the values used in the simulation discussed in Section 2.1.

object i at time j.

For each object, i, we sampled the parameters, τ_i, α_i and ν_i from their own distributions as a hierarchical model. The distribution used was a log-normal, which ensured that the parameters, τ, α, and ν, were all positive. We also needed a variance for each population distribution, which we treated as unknown when doing the inference in Section 2.3. To differentiate between object parameters and population parameters, we used the notation in (2.1) and (2.2) for the object parameters, and μ and σ, with the necessary subscripts, for the population parameters as shown in Table 2.1. These additional parameters were all sampled in the same manner as τ.

A directed acyclic graph (DAG) of the model is shown in Figure 2.1, which shows the relationship between the population parameters, individual object parameters, and the data. This is the same model we used to analyse the data and estimate the population parameters in Section 2.3.

We used log-normal distributions with the following parameterisation. First, the parameter μ for each is the median lag, which is obtained by exponentiation of the mean values obtained on the log scale. Second, the parameter σ in each distribution is a scale factor rather than variance, however we will call it the variance for means of familiarity and interpretation.

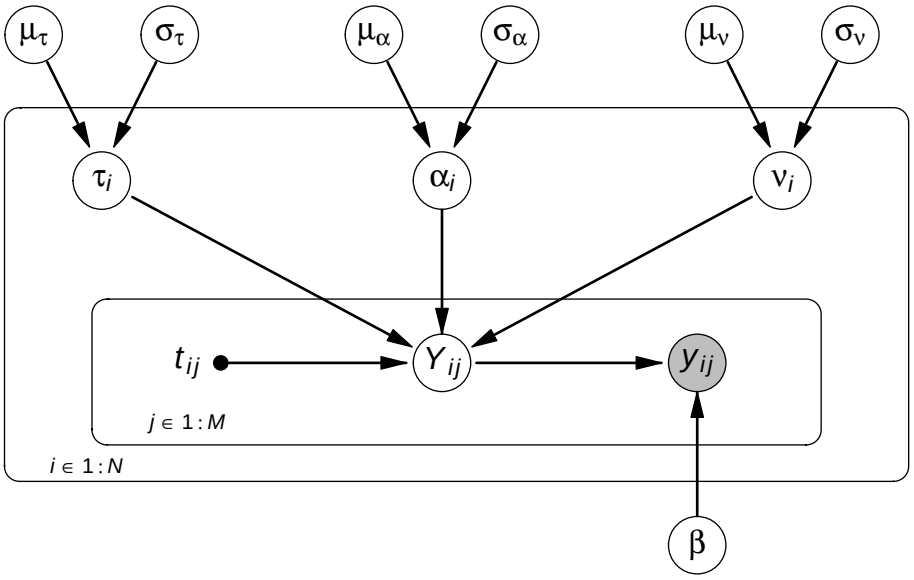

Figure 2.1: A DAG of the simple hierarchical model, demonstrating how the population parameters are related to the observations.

2.1 Data Generation

The data generation process was conducted using **R**, a free software package for statistical computing (R Core Team, 2012). The first step involved randomly generating object parameters from their respective distributions, with some predefined population parameters. We used the equation

$$\theta_i = \mu_\theta \exp\{\sigma_\theta Z\}, \qquad Z \sim \mathcal{N}(0,1) \tag{2.3}$$

to sample the object parameters, where θ is used to represent any of the three parameters (τ, α, ν). The population parameters are listed in Table 2.1. After sampling the parameters for each object, we simulated M measurements, with error to replicate a real life scenario. To do this, we first sampled M time values from $\mathcal{U}[0,\ 20]$, and then used (2.1) and (2.2) to generate noisy observations at the given times.

The **R** script used to generate the data is shown in Appendix A. We simulated 100 objects using the default parameter values (Table 2.1), each with 5 data points. To visualise the data, 8 objects from the sample are plotted in Figure 2.2. This shows that, while some objects have a fairly good representation of points around the peak, others do not, and estimation of the lag for these, represented by the position of the peak, would be impossible.

After having generated a data set, we were able to use different methods to attempt recovery of the initial, known parameters and test the validity of our proposed method. The method used by Fine et al. (2012) was also implemented to give us a baseline to which we could make comparisons.

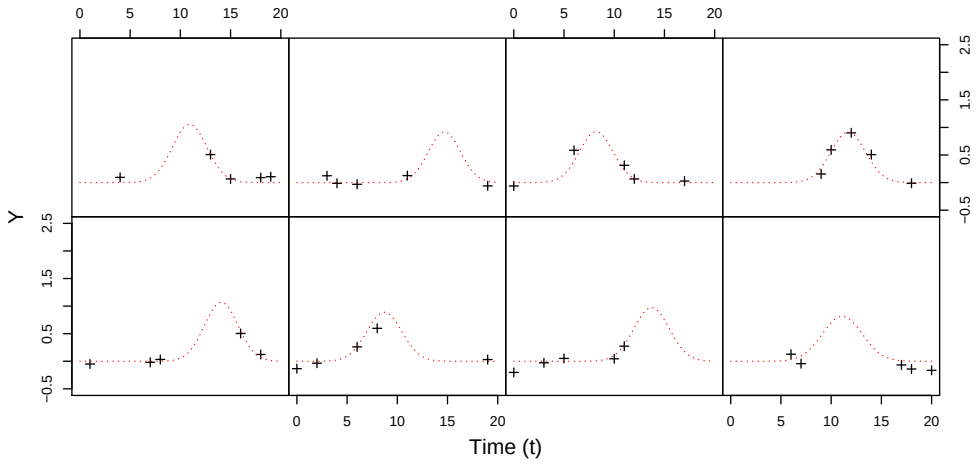

Figure 2.2: Several simulated objects. The red curve shows the actual light curve for that object, and the crosses are the recorded data.

2.2 Cross Correlation Method

To compare to Fine et al. (2012), we used their technique of stacking the results from analysing each object individually to get an overall stacked CCF, and to then use the peak of this to estimate μ_τ. The difference between their method and the method used here is that we were using a simplified data set, so the response, $\boldsymbol{y}$, responded to the fixed function obtained by substituting $\tau = 0$, $\nu = 0$ and $\alpha = 1$ into (2.1).

For each object, we used the following equation to calculate the cross-correlation for the lags:

$$X(\tau) = \frac{1}{n} \sum_{\tau} C(t_i - \tau)\, y_i \tag{2.4}$$

These were then stacked by calculating the mean and standard deviation of the CCF calculated for the range of τ shown in Figure 2.3. These results show a peak at 10.7, which is reasonably close to the input value of 11. However, it is difficult to obtain an estimate of the uncertainty for this estimate separately from the population variability in lags.

2.3 Hierarchical Bayesian Model

A hierarchical model makes the assumption that the individual lags for each object are related. This relationship was described by fitting a population distribution described by μ_τ, the population

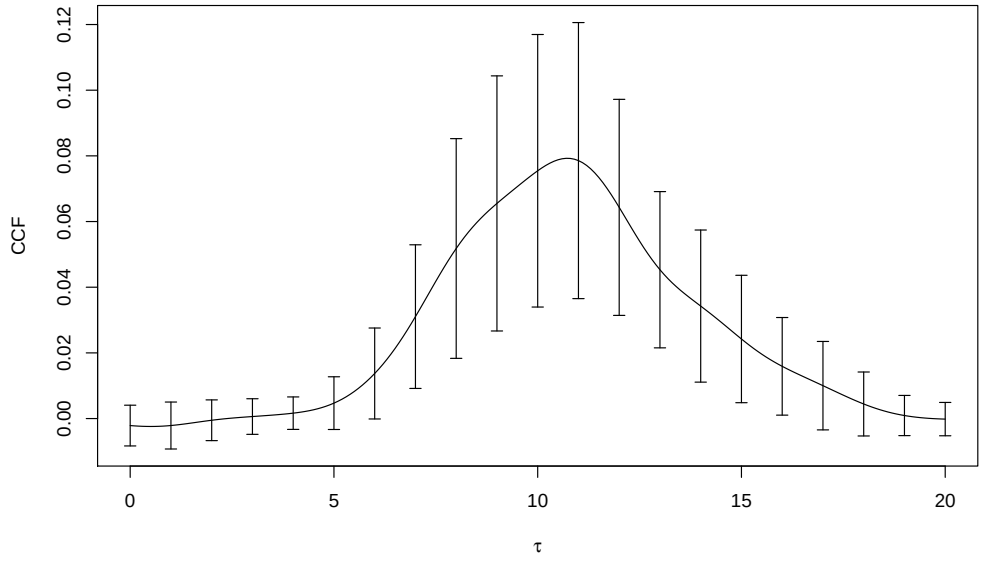

Figure 2.3: Stacked CCF for the simple model, showing the mean as a solid line, and standard deviations represented by error bars.

Parameter	Prior
μ_τ	$\mathcal{U}(0, 20)$
μ_ν	$\mathcal{U}(0, 10)$
μ_α	$\mathcal{U}(0, 10)$

Parameter	Prior
σ_τ	$\log\mathcal{U}(-10, 5)$
σ_ν	$\log\mathcal{U}(-10, 5)$
σ_α	$\log\mathcal{U}(-10, 5)$
β	$\log\mathcal{U}(-10, 5)$

Table 2.2: Parameters and their priors, as used in the simple hierarchical model. $\mathcal{U}$ = Uniform, $\log\mathcal{U}$ = Log-Uniform with parameters given on the log scale. The μ's are the population medians because we used a log-normal distribution.

median lag, and σ_τ, the corresponding standard deviation. Now, the lag of each individual object, τ_i, was assumed to be log-normally distributed as

$$\log \tau_i \sim \mathcal{N}\left(\log \mu_\tau, \sigma_\tau^2\right). \tag{2.5}$$

The remaining two parameters, α and ν, were modelled in the same way as τ. The DAG for this model is shown in Figure 2.1.

2.3.1 The Model

To model the data, we used STAN (Stan Development Team, 2013a), which uses Hamiltonian dynamics to explore the posterior distribution. The first issue was our choice of priors for each of the parameters. The priors used in the model are shown in Table 2.2, although we tried several others which had no visible effect on the final posterior distribution.

For the population medians, we used uniform prior distributions. For μ_τ, we made the assumption that the true lag was within the range of the data. For μ_ν and μ_α, we saw from plotting the data that the amplitudes and widths were all less than 5, and positive by definition. For the variance parameters, we used log-uniform distributions, which approximate Jeffreys' priors. The choice of -10 and 5 as the bounds for the uniform give the parameters a range of approximately 0 – 150, which will undoubtedly contain the true value. We used this same prior for β, the standard deviation of the measurement errors.

For the priors on the parameters, such as that in (2.5), we implemented techniques suggested in the STAN Reference (Stan Development Team, 2013b). One of these was to use latent variables with standard normal distributions, such that

$$\log \tau_i = \log \mu_\tau + \sigma_\tau \bar{\tau}_i \tag{2.6}$$

and the latent parameter, $\bar{\tau}_i$, has a standard normal prior

$$\bar{\tau}_i \sim \mathcal{N}(0, 1) \tag{2.7}$$

However, this is exactly the same as giving τ_i a log-normal distribution, with parameters μ_τ and σ_τ, which is what is defined in Table 2.2. The reason for using this technique was that it makes STAN's sampler more efficient (Stan Development Team, 2013b).

For this hierarchical model, the STAN model used to generate the results is similar to the one shown in Appendix A.2, however slightly modified to analyse all objects simultaneously (see Section 2.3.2). From Table 2.3, we see that the posterior mean and median are very close to the true value of 11 we used when generating the data.

2.3.2 Results and Sampling Issues

The summary statistics for the hierarchical model are shown in Table 2.3. Comparing these results to the true values in Table 2.1, we see that the parameter estimates for all of the parameters are very accurate, and the 95% credible intervals contain the true values.

While sampling, however, some chains would remain unconverged, sampling slightly different values for several parameters, including μ_τ. As a result, when we run more than 3 chains, we were often unable to obtain convergence, even after running the sampler for 100,000 iterations. The main parameters which did not converge were μ_τ and σ_τ, where different chains would converge to different values. We believe that this was because of the limited data for each object, and therefore the estimates for any individual object in the hierarchical model could find multiple modes that fit the data equally well. This would alter the population parameter distribution if the chain was unable to move easily between modes because of unlikely intermediate values.

To overcome this problem, instead of analysing all of the objects simultaneously, we decided to generate posterior samples for each object individually, as we found this computationally easy to achieve. Then we used another method to estimate what the posterior distribution would have been if we had used the hierarchical model.

2.3.3 Combining Individual Posterior Samples

Before we continue, we need to show how we can infer the posterior distributions of the hyperparameters, in this case μ_τ and σ_τ, from the posterior samples obtained from each individual object when doing the full hierarchical analysis in one step is too difficult.

We want to infer the hyperparameters, $\boldsymbol{\Theta} = (\mu_\tau, \sigma_\tau)$, from the data $\boldsymbol{Y} = (Y_1, \ldots, Y_N)$ (for

	Hierarchical (20,000 samples)					Individual (800 samples per object)				
Parameter	Mean	SD	0.025	0.5	0.975	Mean	SD	0.025	0.5	0.975
μ_τ	10.98	0.23	10.52	10.99	11.41	11.04	0.27	10.52	11.04	11.60
μ_ν	1.48	0.055	1.37	1.48	1.59	–	–	–	–	–
μ_α	0.97	0.026	0.92	0.97	1.02	–	–	–	–	–
σ_τ	0.18	0.016	0.15	0.18	0.22	0.20	0.02	0.16	0.20	0.24
σ_ν	0.014	0.025	0.000055	0.0021	0.095	–	–	–	–	–
σ_α	0.13	0.025	0.083	0.13	0.18	–	–	–	–	–
β	0.11	0.0050	0.096	0.10	0.12	–	–	–	–	–

Table 2.3: Results from the two Bayesian models, including means, standard deviations, and the 2.5%, 50%, and 97.5% quantiles.

simplicity, we will let each Y_i represent all data associated with object i). By Bayes' rule, the posterior distribution of $\boldsymbol{\Theta}$ is:

$$f\left(\boldsymbol{\Theta} \,|\, \boldsymbol{Y}\right) \propto f\left(\boldsymbol{\Theta}\right) f\left(\boldsymbol{Y} \,|\, \boldsymbol{\Theta}\right). \tag{2.8}$$

Unfortunately, we do not know the likelihood, $f\left(\boldsymbol{Y} \,|\, \boldsymbol{\Theta}\right)$; however, we know the distribution of the object parameters, $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_N)$, so the joint prior distribution of the object parameters and the hyperparameters can be written as:

$$f\left(\boldsymbol{\Theta}, \boldsymbol{\tau}\right) = f\left(\boldsymbol{\Theta}\right) \prod_{i=1}^{N} f\left(\tau_i \,|\, \boldsymbol{\Theta}\right) \tag{2.9}$$

We can now rewrite (2.8) including the individual τ_i's as

$$\begin{aligned} f\left(\boldsymbol{\Theta}, \boldsymbol{\tau} \,|\, \boldsymbol{Y}\right) &\propto f\left(\boldsymbol{\Theta}\right) f\left(\boldsymbol{\tau} \,|\, \boldsymbol{\Theta}\right) f\left(\boldsymbol{Y} \,|\, \boldsymbol{\Theta}, \boldsymbol{\tau}\right) \\ &\propto f\left(\boldsymbol{\Theta}\right) f\left(\boldsymbol{\tau} \,|\, \boldsymbol{\Theta}\right) f\left(\boldsymbol{Y} \,|\, \boldsymbol{\tau}\right) \\ f\left(\boldsymbol{\Theta}, \boldsymbol{\tau} \,|\, \boldsymbol{Y}\right) &\propto f\left(\boldsymbol{\Theta}\right) \prod_{i=1}^{N} f\left(\tau_i \,|\, \boldsymbol{\Theta}\right) f\left(Y_i \,|\, \tau_i\right) \end{aligned} \tag{2.10}$$

As we are not interested in the individual object parameters, we can marginalise them out:

$$f\left(\boldsymbol{\Theta} \,|\, \boldsymbol{Y}\right) = \int f\left(\boldsymbol{\Theta}, \boldsymbol{\tau} \,|\, \boldsymbol{Y}\right) \, \mathrm{d}^N \boldsymbol{\tau}. \tag{2.11}$$

Substituting the posterior from (2.10) into (2.11), we get the posterior of $\boldsymbol{\Theta}$ as

$$\begin{aligned} f\left(\boldsymbol{\Theta} \,|\, \boldsymbol{Y}\right) &\propto \int f\left(\boldsymbol{\Theta}\right) \prod_{i=1}^{N} f\left(\tau_i \,|\, \boldsymbol{\Theta}\right) f\left(Y_i \,|\, \tau_i\right) \, \mathrm{d}^N \boldsymbol{\tau} \\ &\propto f\left(\boldsymbol{\Theta}\right) \prod_{i=1}^{N} \left(\int f\left(\tau_i \,|\, \boldsymbol{\Theta}\right) f\left(Y_i \,|\, \tau_i\right) \, \mathrm{d}\tau_i \right) \end{aligned} \tag{2.12}$$

Comparing this to (2.8), we can see that the product component of (2.12) is the likelihood required

to find the posterior distribution of the population hyperparameters.

Now we require some way of using the posterior samples generated from analysing the individual objects in the likelihood equation above. These samples represent the distribution

$$f\left(\tau_i \,|\, Y_i\right) \propto \pi\left(\tau_i\right) f\left(Y_i \,|\, \tau_i\right), \tag{2.13}$$

where $\pi\left(\tau_i\right)$ is the prior distribution for the parameters in the individual object models. We can use the posterior samples to approximate the likelihood part of (2.12) by making it into an expectation with respect to the posteriors from (2.13):

$$\begin{aligned}\prod_{i=1}^{N}\left(\int f\left(\tau_i \,|\, \boldsymbol{\Theta}\right) f\left(Y_i \,|\, \tau_i\right)\, \mathrm{d}\tau_i\right) &= \prod_{i=1}^{N}\left(\int f\left(\tau_i \,|\, \boldsymbol{\Theta}\right) \frac{\pi\left(\tau_i\right)}{\pi\left(\tau_i\right)} f\left(Y_i \,|\, \tau_i\right)\, \mathrm{d}\tau_i\right) \\ &\propto \prod_{i=1}^{N}\left(\int \frac{f\left(\tau_i \,|\, \boldsymbol{\Theta}\right)}{\pi\left(\tau_i\right)} f\left(\tau_i \,|\, Y_i\right)\, \mathrm{d}\tau_i\right) \\ &\propto \prod_{i=1}^{N} \mathbb{E}\left[\frac{f\left(\tau_i \,|\, \boldsymbol{\Theta}\right)}{\pi\left(\tau_i\right)}\right]. \end{aligned} \tag{2.14}$$

Finally, we can use (2.14) with the posterior samples as an approximation to the likelihood in (2.8), and, providing we generate a sufficient number of samples for each object, obtain a posterior distribution for the population parameters as though we had performed a hierarchical model.

We first generated the samples using the model in Appendix A.2, using 1000 iterations for each object and discarding the first 200 as burn-in (using more samples did not improve accuracy and only increased computational requirements in the next step).

With these samples, we used StretchR (Brewer, 2012b) to run MCMC simulations to generate posterior samples for μ_τ and σ_τ. StretchR uses the algorithm discussed by Foreman-Mackey et al. (2013), which uses multiple walkers (points in S-dimensional space, where S is the number of parameters being estimated) which are initially dispersed over the prior. At each step, two walkers are selected at random, w_k and w_p. A new point, $w^\star = Zw_k + (1-Z)w_p$, is proposed that lies on the vector connecting w_k and w_p. From here, w_k is moved to $w^\star$ with acceptance probability

$$q = \min\left\{1,\, Z^{S-1}\frac{p\left(w^\star\right)}{p\left(w_k\right)}\right\}, \tag{2.15}$$

where Z is a random variable drawn from a distribution $g\left(z\right)$, where

$$g\left(z\right) \propto \frac{1}{\sqrt{z}}, \qquad z \in \left[0.5, 2\right],$$

and $p(w_i)$ is the probability of the parameter values represented by walker i, given the data.

In this way, the chain itself does not make up an independent posterior sample as in usual MCMC. Instead, the walkers converge onto the posterior distribution, and a sample of walkers at a single iteration can be used as a posterior sample if enough walkers are used (e.g., 1000). Using the code in Appendix A.3, we obtained a posterior distribution for μ_τ and σ_τ, for which the summarised results are shown in Table 2.3.

2.4 Summary of the Simple Model

The two Bayesian methods achieved reasonably similar results, although the large number of parameters in the full hierarchical model meant that it could not be used easily. It required many more iterations of the sampler to achieve the same results as the individual method (described in Section 2.3.3), and the computational requirements at each iteration were significantly greater.

We also had the issue where we were unable to achieve convergence when multiple chains were used. Because this issue remained consistent throughout multiple reparameterisations, different priors, and longer chains, we made the decision to analyse the objects individually and combine the results at the end, as discussed in Section 2.3.3.

One possible drawback to this method is that we only have posterior samples for μ_τ and σ_τ, and while we could include the other population hyperparameters in the StretchR model, this would only increase the computational requirements. Furthermore, these nuisance parameters are not of interest, as we only wish to infer μ_τ and σ_τ.

Our main purpose for implementing this simple model was to show that a Bayesian hierarchical model could be used to estimate the average population lag in a collection of related objects. We also wanted to compare this to the results of stacked CCFs. We used an initial lag[1], $\mu_\tau = 11$, and were able to obtain an estimate of $\widehat{\mu}_\tau = 10.7$ using stacked CCF (Fine et al., 2012), $\widehat{\mu}_\tau = 10.99$ (95% CI: 10.52 – 11.41) using a full hierarchical model, and $\widehat{\mu}_\tau = 11.04$ (10.52 – 11.60) by combining individual object samples.

In the Bayesian models, we were also able to measure the population variability, and separate this from experimental error. Both Bayesian models were able to recover the initial value of $\sigma_\tau = 0.20$ with similar accuracy. All of the credible intervals contained the true values, which gave us confidence that when we scaled up, we should be able to achieve similar results. We need to note that when using the stacked CCF, it was not possible to easily or intuitively estimate the population variability of the lags.

[1]This is the median lag because we used a log-normal distribution.

Chapter 3

Building A Suitable Model

Having shown that a Bayesian hierarchical model could be used to estimate population parameters, given only a few usable data points for a sample of objects, we were ready to scale up and model real reverberation mapping data. This new model had two light curves, which are time series of light brightness, rather than a single response curve that was used in Chapter 2. This time, instead of using a single Gaussian pulse as the signal, we needed to model the first light curve, and use this more complicated signal to model the variation in the responding light curve and infer the lag.

To test the model, we used reverberation mapping data from ARP151, which is shown in Figure 3.1, provided by the LICK AGN Monitoring Project (Bentz et al., 2009). We were then able to compare this to previous results (Brewer, 2012a), and see how well this could be done when the number of data points was reduced, replicating the type of data used by Fine et al. (2012).

For ease of notation, we will drop the subscript i from the parameters, as we will only need to discuss the modelling of a single object.

3.1 The First Light Curve: Continuum

For the first light curve, we needed to make a discrete approximation of a CAR(1). This has been justified in previous studies as a good model for AGN variability (Kelly et al., 2009; Zu et al., 2013). Therefore, for an object which has M_C observations of the continuum light curve, the true, unknown value at each time point, C_t, can be described by a first order autoregressive process (AR(1)) model:

$$C_t = \mu + \alpha\left(C_{t-1} - \mu\right) + \beta^2\gamma_t, \qquad t \in \{1, 2, \ldots, \max\left(\boldsymbol{T}_C, \boldsymbol{T}_L\right)\} \tag{3.1}$$

where μ is the long-run mean brightness of the light curve, α is the autocorrelation coefficient, and β is the random variance component (when this is 0, the time series simply decays exponentially to the mean). The initial value, C_0, will be assumed normally distributed around μ. Each of the $\boldsymbol{\gamma}$ are independent and have a standard normal prior. $\boldsymbol{T}_C$ and $\boldsymbol{T}_L$ are the times at which the data points were observed in both light curves.

The data, $\boldsymbol{c}$, are noisy observations at only some time points, and each has a 'known' measurement error, ε_t, that is provided with the data, so

$$c_t \sim \mathcal{N}\left(C_t, \varepsilon_t^2\right), \qquad t \in \boldsymbol{T}_C, \tag{3.2}$$

where $\boldsymbol{T}_C$ is the vector of M_C time measurements for the continuum light curve.

To model this in STAN, we supplied the three data vectors, $\boldsymbol{c}$, $\boldsymbol{T}_C$, and $\boldsymbol{\varepsilon}$, as well as M_C. The times were in days, however to make the discrete approximation more accurate, we used 10 points per day. Following along from the suggestions in the STAN User Manual (Stan Development Team, 2013b), we used latent variables, with a standard normal distribution, like in (2.6). For priors, we tried a range of different options, but these had negligible differences on the posterior and mainly affected the time it took for the chains to converge.

We used a Beta $(20, 1)$ prior for the autocorrelation coefficient α, because we knew the value would be very close to 1, and when we used less informative priors the sampler would get stuck

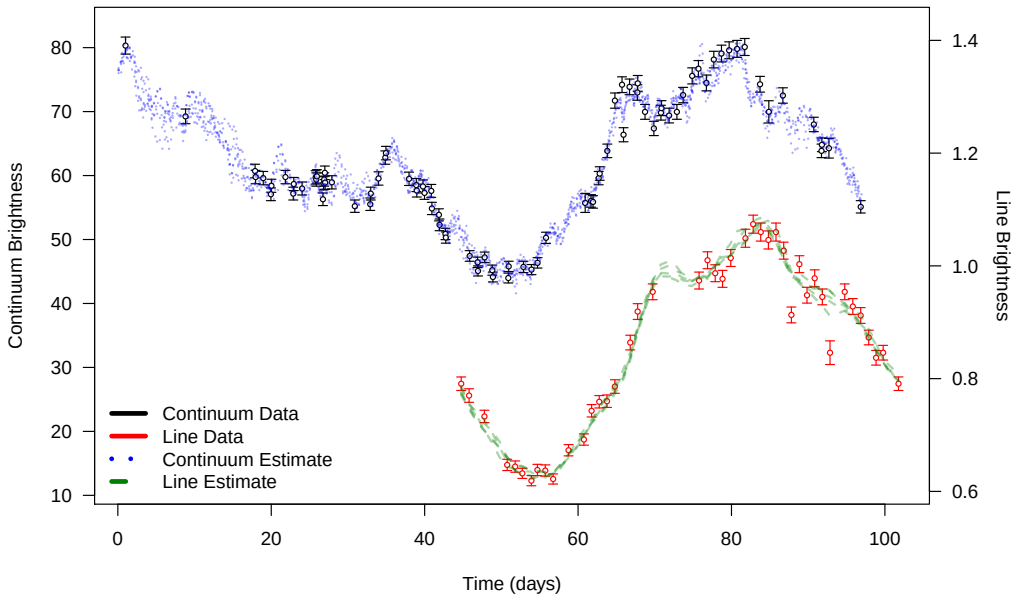

Figure 3.1: Reverberation mapping data for ARP151, an AGN monitored by Bentz et al. (2009). The blue and green lines show posterior estimates of the true continuum and BLR light curves respectively.

Parameter	Prior
μ	$\mathcal{N}\left(0,\, 1000^2\right)$
α	$\mathcal{B}\left(20,\, 1\right)$
β	$\mathcal{E}\left(1\right)$
C_0	$\mathcal{N}\left(\mu,\, 10^2\right)$
γ_t	$\mathcal{N}\left(0,\, 1\right)$

Parameter	Prior
b	$\mathcal{E}\left(0.1\right)$
a	$\mathcal{U}\left[0,\, 0.9b\right]$
η	$\mathcal{U}\left[0,\, 1\right]$
λ	$\mathcal{N}\left(0,\, 100^2\right)$

Table 3.1: Parameters used in the reverberation mapping time series model, along with their respective prior distributions. $\mathcal{E}$ = Exponential, $\mathcal{B}$ = Beta.

sampling lower values of α and increasing the variance component, β. For β, we used an exponential prior which helped to keep the values lower. Again, this only affected the time it took for the sampler to converge, and when we used less informative priors such as a log-normal, the chain could get stuck at high values of β and essentially fit a noisy straight line through the data. For the mean brightness of the light curve, μ, we used a wide normal prior distribution. A complete list of the parameters and their associated priors is shown in Table 3.1.

The final STAN model is shown in Appendix B.1, and implements the priors discussed above, as well as the standardisation discussed in Section 2.3.1, which allows STAN to sample from standard normal distributions. Using this half of the model, we were able to generate posterior samples for the 'true' continuum light curve, which we would need to estimate the second light curve. These are represented by the blue lines in Figure 3.1.

3.2 The Second Light Curve: BLR

To implement the second half of the likelihood function, we needed a transfer function that would use the estimated continuum light curve to produce values for the responding BLR light curve. We used a uniform transfer function,

$$\Psi\left(\tau\right) = \begin{cases} \dfrac{1}{b-a}, & a \leq \tau \leq b, \\ 0, & \text{otherwise.} \end{cases}, \qquad b > a > 0 \tag{3.3}$$

Here, a and b are the minimum and maximum lags respectively, and all points in the continuum light curve between these two lags are averaged over to determine the corresponding value in the BLR light curve.

The response from the BLR, $\boldsymbol{L}$, can therefore be calculated using the following equation:

$$L_t = \eta \int_0^\infty \Psi\left(\tau\right) \left[C_{t-\tau} + \lambda\right] \mathrm{d}\tau, \qquad t \in \left\{1, 2, \ldots, \max\left(\boldsymbol{T}_L\right)\right\}, \tag{3.4}$$

using the definition of Y_t from (3.1) and $\Psi\left(\tau\right)$ from (3.3).

There are two additional response coefficients in (3.4): η and λ. These each adjust for differences in the scale of measurements and the magnitude of fluxes between the two light curves. The second, λ, was introduced because previous modelling of some AGN, such as Mrk 50, found λ to be non-zero (Pancoast et al., 2012). However, there were some implications due to the high posterior correlation between these two parameters, discussed briefly in Section 3.3.

Once we had an initial estimate of the true value at each time point of the BLR curve, we could implement the likelihood, where we assumed the observed M_L data points, $\boldsymbol{l}$, are normally distributed around the mean, $\boldsymbol{L}$, with a known estimate of experimental error, $\boldsymbol{\phi}$:

$$l_t \sim \mathcal{N}\left(L_t,\, \phi_t^2\right), \qquad t \in \boldsymbol{T}_L \tag{3.5}$$

where $\boldsymbol{T}_L$ are the time recordings for the data points in the BLR light curve. Note that in (3.1), we need to estimate the continuum light curve up to the most recent time measurement of both light curves. If we do not include times up until the maximum in $\boldsymbol{T}_L$, then we will get undefined values in (3.4).

For the response coefficients, we used a uniform prior on η, and a wide normal prior on λ. For the maximum and minimum lag, b and a respectively, we tried several different parameterisations. The best, however, used an exponential prior on b, and then implemented a conditional uniform prior on a, $f\left(a \,|\, b\right) \sim \mathcal{U}\left[0,\, 0.9b\right]$, where the 0.9 was required to ensure $\Psi\left(\tau\right)$ was not too narrow, as this lead to the sampler getting stuck at certain values and unable to move around the parameter space.

3.3 Implementation and Sampling Issues

Implementing the transfer function into the STAN model required several different approaches, and the most effective is shown in Appendix B.1. This involved implementing a for-loop over all of the values $i \in \{1, \ldots, M_L\}$, and for each, summing the values of C_t where $t \in [(T_L)_i - b,\, (T_L)_i - a]$. The sum part was done by using an `if()` statement, and building up the sum when the condition above was true. While this was somewhat inefficient, STAN does not allow sub-setting of data vectors. If this had been the case, we would have needed much less computational effort to calculate the required sums.

The posterior correlation between η and λ was strongly negative, especially when less continuum data points were available for a particular BLR value. The marginal distribution of τ did not depend strongly on λ, however, so it was of no great concern. Unfortunately, this high correlation

caused the step sizes in the STAN sampler to decrease, which meant that chains could take a long time to converge, and we were unable to amend this despite various efforts.

We checked our model by using the ARP151 data set, and were successfully able to attain results similar to those from other studies which found $\tau \approx 4$ days (Brewer, 2012a). Surprisingly, even when we reduced the number of observations of the second light curve to three randomly chosen points, we were still able to get a reasonably accurate estimate of τ.

Now that we had a working model for estimation of the lag between the two light curves, we were finally ready to move on to full-scale implementation of the technique by combining data from multiple AGN to infer characteristics of the population.

Chapter 4

Simulated AGN

Once we had shown that our STAN model could be used to estimate τ for a single object, we were ready to use it on multiple objects and attempt to infer the population distribution of lags. To do this, we first simulated 100 AGN based on the data observed from ARP151 in Chapter 3. Afterwards, we performed the analysis of these objects, generating samples of τ for each, and then sampled these in StretchR to get a final posterior distribution for the overall population lag parameters, μ_τ and σ_τ.

4.1 Simulation of AGN

To generate random AGN reverberation mapping data sets, we needed an AR(1) time series, accompanied by a lagged response curve. The **R** code used to simulate this is shown in Appendix C.1. We followed procedures similar to those from Section 2.1 to simulate the data. This involved using some specified population values, which we chose to match those used by Fine et al. (2012). For each simulated AGN, the function `makeAGN()` generated parameter values based on the distributions specified in Table 4.1. Next, we simply used (3.1) to simulate an AR(1) time series of light brightness measurements to simulate the continuum light curve. Using the values from this, we used the transfer function in (3.3) to generate the responding BLR light curve.

Using these simulated light curves, we could then make noisy measurements at randomly selected time points. For each object, we made 80 observations of the continuum and 3 observations of the BLR, again using the errors specified in Table 4.1. We saved the times, brightness and errors of the two curves for each simulated object so we could simply apply the STAN model to them in turn to generate posterior samples. Some of the simulated objects are shown in Figure 4.1, which emphasises that the lag can be estimated easily for some, while for others it is near impossible.

Parameter	Sampling Distribution
AR(1) mean	$\mu \sim \mathcal{N}\left(50, 5^2\right)$
Autocorrelation	$\alpha \sim \mathcal{U}\left[0.990, 0.999\right]$
AR(1) variance	$\beta \sim \mathcal{N}\left(2, 0.1^2\right)$ T $(0, \infty)$
Continuum measurement error	$\epsilon \sim \mathcal{N}\left(0.8, 0.1^2\right)$ T $(0, \infty)$
Lag (days)	$\tau \sim \mathcal{N}\left(4, 0.5^2\right)$ T $(0, \infty)$
Broad line measurement error	$\phi \sim \mathcal{N}\left(0.3, 0.05^2\right)$ T $(0, \infty)$

Table 4.1: The population parameter values used in the simulation, showing their mean and standard deviation. In the case of parameters that are strictly positive, we used a truncated normal distribution, otherwise a normal distribution.

4.2 Sampling AGN and Lag Estimation

The next step was simply to use the pre-existing model in Appendix B.1 (developed in Chapter 3) and loop over the 100 simulated AGN objects, saving the posterior samples of τ for each of them. While this process was slow, most of the objects were found to be converged using the Gelman & Rubin diagnostic check (Gelman & Rubin, 1992) after using 50,000 iterations. Some objects, however, did not converge after this time, so we rerun these for more iterations and a thinning interval, and most of these converged. We continued this process until we attained convergence of the chains for all 100 objects.

Having sampled all of the objects individually, we repeated the methods from Section 2.3.3 for the AGN data, which required that the prior on τ used on the individual objects was known.

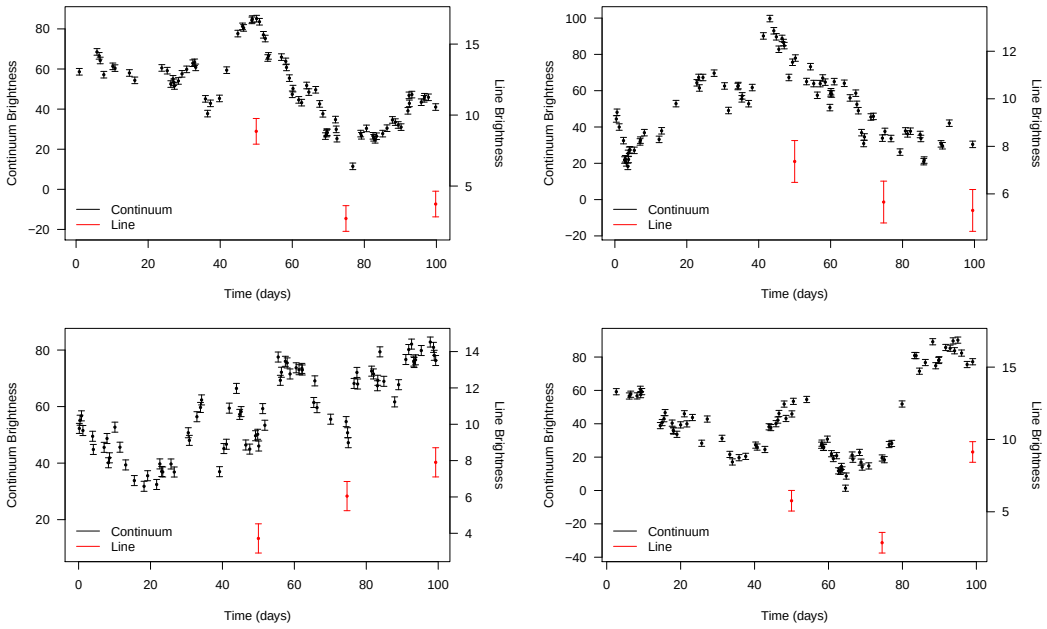

Figure 4.1: A sample of the simulated AGN, which show that some have more easily observed lags that others.

Parameter	Mean	SD	0.025	0.5	0.975
μ_τ	4.26	0.28	3.71	4.29	4.80
σ_τ	0.53	0.30	0.05	0.49	1.22

Table 4.2: Summary statistics for the posterior distribution of the population lag mean and variance obtained from the individual Bayesian method. The true values used in the simulation were 4 and 0.5 respectively.

However, because in the model we specified τ in terms of a and b, it was complicated to analytically calculate the prior on τ.

Despite this, because b had an exponential prior, which is simply a Gamma with $\alpha = 1$, and a was a uniform between 0 and $0.9b$, it seemed reasonable enough to assume that τ would have a Gamma prior with slightly different parameters. Therefore, we used numerical methods to calculate the implied prior for τ by simulating values of a and b from their respective priors, and calculating τ. We computed the prior distribution for τ used in each object to be $\pi(\tau_i) \sim \mathcal{G}(0.982, 0.138)$.

Now we could implement the StretchR program to analyse the posterior samples for the objects and sample the posterior distribution of μ_τ and σ_τ which describe the population. This is implemented using the StretchR likelihood functions shown in Appendix C.2. As before, we used 1000 walkers and let the StretchR script iterate until there was no significant change in the distribution of the population parameters. A time lapse of the walker distribution is shown in Figure 4.2. The walkers started spread out over the prior, and finished converged on the posterior distribution. The summary statistics are shown in Table 4.2.

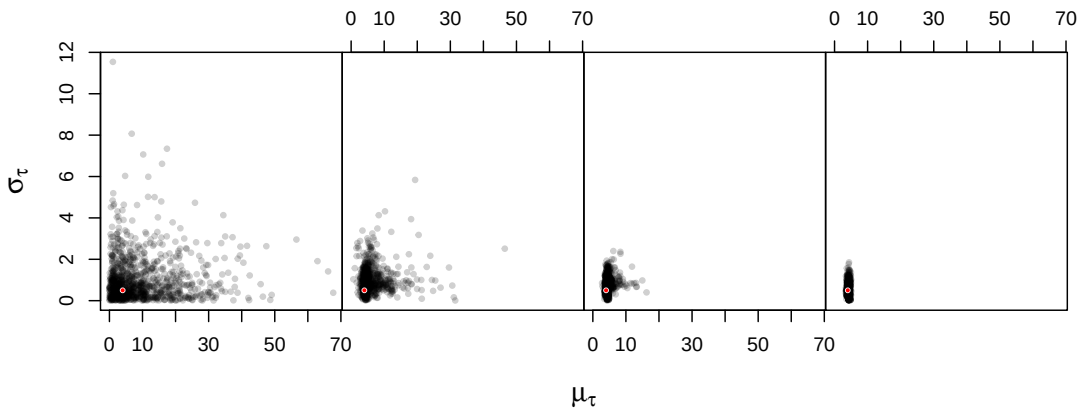

Figure 4.2: Sequential distribution of the StretchR walkers as they move from the prior distribution (left) to the posterior distribution (right). The red dot is the true value used in the simulation, $(\mu_\tau, \sigma_\tau) = (4, 0.5)$.

4.3 Comparison to Stacked CCF

We wanted to compare our results to the methods of Fine et. al, so we used their CCF formula

$$X(\tau) = \sum_{i,j}^{t_j - t_l \in [\tau, \tau+\delta]} \frac{\frac{n_L - 1}{n_L}\left(C_i - \overline{C}\right)\left(L_j = \overline{L}_{k \neq j}\right)}{n_{\text{pair}}}, \tag{4.1}$$

where n_C and n_L are the number of data points in the corresponding light curves, and n_{pair} is the number of points in each component of the sum. $\overline{C}$ is the sample mean of all of the observations of the first light curve, while $\overline{L}_{k \neq j}$ is the mean of the second light curve excluding the jth point. The reasons for this are discussed by Fine et al. (2012).

The extra parameter δ denotes the bin size. We tried variable bin sizes, and produced the stacked CCF shown in Figure 4.3 using $\delta = 2$. Additionally, due to sparsity in the data, we used a loess smoothing function in **R** to smooth the stacked CCF. Without doing so, the peak is still the same as in Figure 4.3, however it is more difficult to make out due to the stacked CCF being noisy. From this plot, we can see a peak in the stacked CCF at about 3.5 days, which is within a plausible range for the true value of 4 days.

Comparing the stacked CCF and Bayesian approaches, we can see that neither are very accurate in estimating μ_τ. The Bayesian method yielded an overestimate of 4.26, while the stacked CCF underestimated it to be approximately 3.5. However, in the Bayesian result, we get a 95% credible

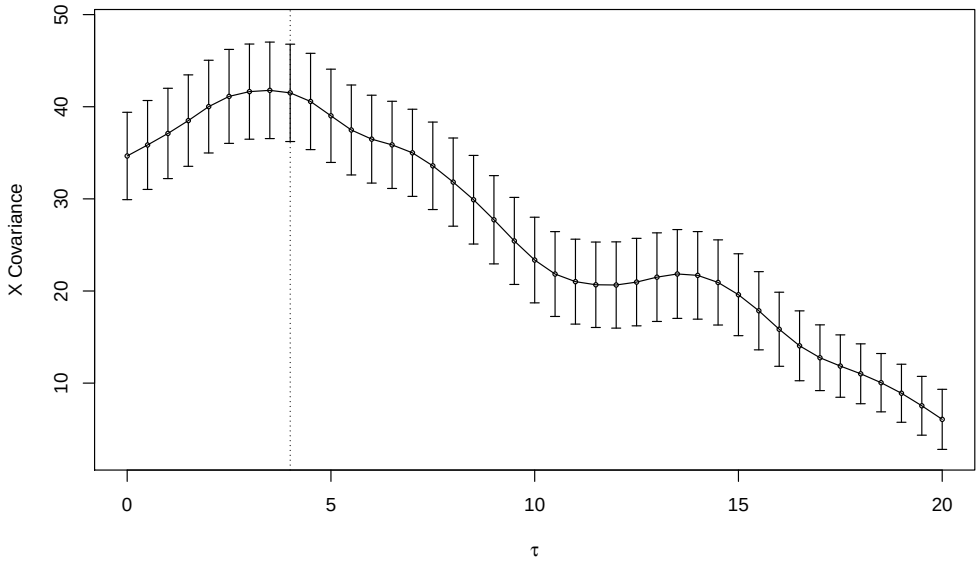

Figure 4.3: Loess smoothed stacked CCF for the simulated AGN data, with error bars representing the standard deviation. The vertical dotted line indicates the true value of $\mu_\tau = 4$.

interval of $(3.7, 4.8)$, which contains the true value. We can also estimate the population variance using the posterior median (due to right skew in the distribution of σ_τ) to be 0.49, although the 95% credible interval is quite wide. The stacked CCF method does not allow us to break the uncertainty in the wide peak into these two components, so it is difficult to determine how accurate the estimate is.

Overall, our model has successfully enabled us to estimate the distribution of lags in a population of AGN, with the advantage that our estimates have an intuitive estimation of error that is separate from our estimate of population variability.

Chapter 5

Discussion and Conclusion

The main aims of this thesis were to implement a Bayesian alternative to the stacked CCF used by Fine et al. (2012, 2013), in the hopes to provide a more intuitive estimate of the mean and variability of reverberation lags in a population of AGN.

Reverberation mapping data is reasonably hard to obtain, especially for accurate inference of the lag, which is used to provide an estimate of the mass of the central black hole. While some objects have well-measured continuum and broad line light curves, many only have a few measurements of the BLR. It is for this reason that Fine et al. (2012) initially proposed the method of stacking the results of reverberation mapping for multiple poorly-measured objects. In this way, previously unusable data could be used to make generalised inferences about a population of AGN.

As noted by Fine et al. (2012), the objects that would be analysed collectively were from a similar system, and so could be thought of as a sample from a related population. In this way, it makes sense to use a hierarchical model, where the reverberation lag for all of the individual objects are related by one overall population distribution of lags. Additionally, by using a hierarchical Bayesian model, we are able to attain an estimate of the error for the mean, and the overall variability between objects in the population separately from one another.

To trial our approach, we made up a simplified version of the data where instead of the continuum being a CAR(1) process, it emitted a single Gaussian pulse at time 0. The response was therefore a Gaussian function with unknown width and amplitude, and a peak that has been delayed by a lag, τ.

By simulating data sets, we were able to obtain an accurate estimate of the mean and variance by different methods. However, due to issues with convergence, we decided to use the method where we individually analyse each object, and then use these samples to obtain a posterior distribution of μ_τ and σ_τ, the median and variance of the population lags.

After showing that this method could be used to estimate the distribution of lags in a sample of objects that can not individually be analysed accurately, we were able to move on to a full reverberation mapping data set.

We created a model in STAN that used a discrete AR(1) approximation to the CAR(1) for the likelihood of the continuum, and a uniform transfer function for the lagged response. Two response coefficients, λ and η, were used to allow for changes in scale and magnitude of the light curves. We used the ARP151 data from Bentz et al. (2009) to test our model, and were able to obtain an estimate of τ that was comparable to previous studies (Brewer, 2012a).

Scaling up, we simulated 100 AGN objects, each with only three observations of the BLR, and implemented the model on each of these. Afterwards, we used the likelihood determined in Chapter 2 to get a posterior sample for the mean and variance of the population lag, μ_τ and σ_τ respectively. We were able to recover the true values used in the simulation with reasonable accuracy. We also implemented the stacked CCF method for comparison. While this method gave us an estimate of μ_τ with similar accuracy, we could not intuitively separate out the estimation error of μ_τ from the population variability, σ_τ, that was seen in the wide peak. Comparison with the results from our Bayesian model would suggest that most of the width in the stacked CCF could be attributed to the variability of lags in the population, σ_τ, rather than uncertainty about μ_τ.

Although the two methods yield similar point estimates of the mean lag, our Bayesian model allows us to estimate the variability in the population, and therefore show how precise our estimate of the mean population lag is. Unfortunately, we were unable to obtain the real data used by Fine et al. (2012, 2013), so we could not do an exact comparison of our method with theirs.

Perhaps some downfalls of our method are the time it takes to run the sampler, whereas CCF is relatively immediate. There may be a way to increase the efficiency of STAN, or perhaps a different sampler will be needed that has more flexibility. Yet, despite this inefficiency, our model was able to give us intuitive estimates of the lag in the population of AGN. Furthermore, on comparison of the time taken to analyse the data with the time taken to physically obtain it (i.e., several months), our method is still quite efficient.

In future, this may allow astronomers to infer characteristics about collections of galaxies in certain parts of the universe, such as the distribution of their sizes. Therefore, if general information about a population of AGN is desired, they will not need to carry out detailed and expensive reverberation mapping campaigns, rather they will only require a small amount of information from a collection of objects.

Appendix A

Scripts Used in Simple Model

A.1 Data Generation

```
GenerateData <-
  function(x, tau, nu, alpha, sigma,
           sig.tau = 0.2, sig.nu = 0.12, sig.alpha = 0.1) {

  # Take tau, nu, alpha, and return random value for each object
    tau.rand   <- exp(log(tau) + sig.tau * rnorm(1))
    nu.rand    <- exp(log(nu) + sig.nu * rnorm(1))
    alpha.rand <- exp(log(alpha) + sig.alpha * rnorm(1))

  # Simulate measurement error using normal
    y <- rnorm(length(x), L(x, tau.rand, nu.rand, alpha.rand), sigma)
    list(y = y, tau = tau.rand, nu = nu.rand, alpha = alpha.rand)
}

L <- function(t, tau, nu = 1, alpha = 1)
    alpha * exp(-((t - tau)^2) / (2 * (1 + nu^2)))
```

A.2 STAN Model

```
data {
  int<lower=1> n;
  real t[n];
  real y[n];
}

parameters {
  real<lower=0,  upper=20> tau;       // value we are estimating
  real<lower=0,  upper=10> nu;        // width
  real<lower=0,  upper=10> alpha;     // height
  real<lower=-10,upper=10> log_beta;  // variance of each data point
}

transformed parameters {
  real beta;
  beta <- exp(log_beta);
}

model {
  tau      ~ uniform(0, 20);
  nu       ~ uniform(0, 10);
  alpha    ~ uniform(0, 10);
  log_beta ~ uniform(-10, 10);

  for(i in 1:n) {
    real Y;
    Y <- alpha * exp(- pow(t[i] - tau, 2) / (2 * (1 + pow(nu, 2))));
    y[i] ~ normal(Y, beta);
  }
}
```

A.3 StretchR Model Functions

```
numDimensions <- as.integer(2)

logPrior <- function(params) {
  # Given a vector of parameters, evaluate the natural logarithm of the prior
  # probability density.

    result <- 0
    if (params[1] < 0 | params[1] > 20   # mu_tau    ~ U[0, 20]
      | params[2] < 0 | params[2] > 2)   # sig_tau   ~ U[0, 2]
        result <- -Inf
    result
}


startingPoint <- function() {
  # Call this function to generate a single point in parameter space
  # for starting an individual walker.

    params    <- rep(NA, numDimensions)
    params[1] <- runif(1, 0, 20)
    params[2] <- runif(1, 0, 2)
    params
}


logLikelihood <- function(params) {
  # Given a vector of parameters, evaluate the natural logarithm of the
  # likelihood function.

    sums <- sapply(allsamples, function(x) {  # **
        f1  <- dlnorm(x[, "tau"], log(params[1]), params[2])
        pi1 <- dunif(x[, "tau"], 0, 20)
        log(mean((f1) / (pi1)))
    })         # returns a vector of sums
    sum(sums)  # returns the complete log likelihood
}
```

** Here, `allsamples` is a list containing the posterior samples (as a matrix) for all of the N objects.

Appendix B

Time Series Model Scripts

B.1 STAN Model for a Single AGN

```
data {
  // --- Data for model implementation
  int N;                                    // total number of Ys
  int f;                                    // number of points per day

  // --- Data for the first time series
  int n_y;                                  // # obs of first TS
  vector[n_y] y;                            // obs values
  vector[n_y] eps;                          // obs errors
  int t_y[n_y];                             // obs times

  // --- Data for the second time series
  int n_l;                                  // # obs of second TS
  vector[n_l] l;                            // obs values
  vector[n_l] phi;                          // obs errors
  int t_l[n_l];                             // obs times
}

parameters {
  // --- Parameters for first time series
  vector[N] Y_raw;                          // standardised Y values
  real<lower=0,upper=1> alpha;              // autocorrelation
  real<lower=0> beta;                       // randomness
  real mu;                                  // mean

  // --- Parameters for second time series
  real<lower=0,upper=1> eta;                // vertical shift
  real lambda;                              // difference in variance

  // --- Transfer function:
  real<lower=0.2> b;                        // max lag
  real<lower=0, upper=0.9*b> a;             // min lag
}
```

```
transformed parameters {
  vector[N] Y;                              // the Y values
  vector[n_y] y_std;                        // standardised obs. for first ts
  vector[n_l] L;                            // expected values for second ts.
  vector[n_l] l_std;                        // standardised obs. for second ts

  // --- Trans. pars for first time series
  Y[1] <- mu + 10 * Y_raw[1];               // Y[1] ~ normal(mu, 10)
  for(t in 2:N)
    Y[t] <- mu + alpha * (Y[t - 1] - mu) + beta * Y_raw[t];

  // --- Standardise observations for first time series
  for(i in 1:n_y)
    y_std[i] <- (y[i] - Y[t_y[i]]) / eps[i];

  // --- Expected values for second time series
  for(i in 1:n_l) {
    int total;                              // number of points in sum
    real sum;                               // the sum of points
    total <- 0;
    sum <- 0;

    // -- Computing the sum, using f to convert from days to ns
    for(j in 1:N) {
      if(j > (t_l[i] - b * f) && (j < t_l[i] - a * f)) {
        sum   <- sum + Y[j];
        total <- total + 1;
      }
    }

    // -- Calculate values for second time series
    L[i] <- eta * sum / max(1, total) + lambda;

    // -- Standardise observations for second time series
    l_std[i] <- (l[i] - L[i]) / phi[i];
  }
}

model {
  // --- Priors for first time series
  Y_raw ~ normal(0, 1);
  alpha ~ beta(20, 1);
  beta  ~ gamma(1, 1);
  mu    ~ normal(0, 1000);

  // --- Priors for second time series
  eta    ~ beta(1, 1);
  lambda ~ normal(0, 100);
  b      ~ gamma(1, 0.1);
  a      ~ uniform(0, 0.9*b);

  // --- Likelihood
  y_std ~ normal(0, 1);     // first time series
  l_std ~ normal(0, 1);     // second time series
}

generated quantities {
  real tau;
  tau <- (a + b) / 2;
}
```

Appendix C

AGN Simulation and Estimation Scripts

C.1 Data Generation

```
makeAGN <-
    function(mean = NULL, rms = NULL,
             N = 100, ny = 30, nl = 3) {
        require(msm)

        if (is.null(mean))
            mean = c(2, 40, 0.8, 0.3, 50)
        if (is.null(rms))
            rms = c(0.1, 5, 0.1, 0.05, 5)

      # Generate the parameters
        parameters <- numeric(6)
        parameters[1:4] <- rtnorm(4, mean[1:4], rms[1:4], lower = 0.01)
        parameters[5] <- rnorm(1, mean[5], rms[5])
        parameters[6] <- 1 - runif(1, 0.001, 0.01)
        names(parameters) <- c("beta", "tau", "epsilon",
                               "phi", "mu", "alpha")

      # Because we use uniform transfer function, need to infer
      # a and b from tau:
        a <- round(parameters["tau"] - 2)
        b <- round(parameters["tau"] + 2)

      # Generate a continuum time series
        Y <- numeric(N)
        Y[1] <- rnorm(1, parameters["mu"], parameters["beta"])
        for (i in 2:N)
            Y[i] <- parameters["mu"] +
                parameters["alpha"] * (Y[i - 1] - parameters["mu"]) +
                parameters["beta"] * rnorm(1)
```

```
      # Observations of continuum
        y.index <- round(sample(1:N, ny))
        y <- rnorm(ny, Y[y.index], parameters["epsilon"])

      # Observations of second series
        l.index <- seq(N / 2, max(y.index), length = nl)
        L <- transfer(l.index, Y, b, a, eta = 0.1, c = 0)
        l <- rnorm(nl, L, parameters["phi"])

      # Output:
        agn(y, y.index / 10, rep(parameters["epsilon"], length(y)),
            l, l.index / 10, rep(parameters["phi"], length(l)))
    }


set.seed(1234)  # to create the same data each time

n.sim = 0  # going to ensure everything is positive **
while (n.sim < 100) {
    x <- makeAGN(N = 1000, ny = 80, nl = 3)
    m <- numeric(2)
    m[1] <- min(contBrightness(x))
    m[2] <- min(lineBrightness(x))

    if (all(m > 0)) {
        n.sim = n.sim + 1
        saveAGN(x, paste("sim", sprintf("%03d", n.sim), sep = ""),
                dir = "./objects2/"))
    }
}
```

** The reason for this was to make sure all of the values were positive, otherwise we could get errors in our model which, for now, assumes that the brightness always non-negative.

C.2 StretchR Model Functions

```
numDimensions <- 2L

logPrior <- function(params) {
  # Given a vector of parameters, evaluate the natural logarithm of the prior
  # probability density.

    result <- 0
    if (params[1] < 0   # mu_tau    ~ Normal(40, 5)
      | params[2] < 0)  # sig_tau   ~ Gamma(0.001, 0.001)
        result <- -Inf
    else {
        result <- dnormt(params[1], 40, 5, log = TRUE) +
                  dgamma(params[2], 1, 1, log = TRUE)
    }
    result
}


startingPoint <- function() {
  # Call this function to generate a single point in parameter space
  # for starting an individual walker.

    params <- rep(NA, numDimensions)
    params[1] <- rgamma(1, 0.9219330, 0.1381571)
    params[2] <- rgamma(1, 1, 1)
    params
}


logLikelihood <- function(params) {
  # Given a vector of parameters, evaluate the natural logarithm of the
  # likelihood function.

    sums <- sapply(allsamples, function(x) {
        f  <- dnorm(x[, "tau"],  params[1], params[2])
        pi <- dgamma(x[, "tau"], 0.9819330, 0.1381571)
        log(mean(f / pi))
    })  # returns a vector of sums
    sum(sums)  # returns the complete log likelihood
}
```

NOTE: the code used to generate the samples can be found on the StretchR repository at `https://github.com/eggplantbren/StretchR`.

Bibliography

Barth A. J., et al. (2011). Broad-line reverberation in the Kepler-field Seyfert galaxy Zw 229-015. *The Astrophysical Journal*, 732(2):121.

Beckmann V., Shrader C. (2012). *Active Galactic Nuclei*. doi:10.1002/9783527666829.

Bentz M. C., et al. (2009). The lick AGN monitoring project: Broad-line region radii and black hole masses from reverberation mapping of Hβ. *The Astrophysical Journal*, 705(1):199.

Blandford R. D., McKee C. F. (1982). Reverberation mapping of the emission line regions of Seyfert galaxies and quasars. *The Astrophysical Journal*, 255:419–439.

Brewer B. (2012a). Bayesian analysis of reverberation mapping data. In Feigelson E. D., Babu G. J., editors, *Statistical Challenges in Modern Astronomy V*, Lecture Notes in Statistics, pages 189–195. Springer New York.

Brewer B. J. (2012b). StretchR: Affine Invariant MCMC Sampling in R. `https://github.com/eggplantbren/StretchR`.

Fine S., et al. (2012). Composite reverberation mapping. *Monthly Notices of the Royal Astronomical Society*, 427(4):2701–2710, doi:10.1111/j.1365-2966.2012.21248.x.

Fine S., et al. (2013). Stacked reverberation mapping. *Monthly Notices of the Royal Astronomical Society: Letters*, 434(1):L16–L20, doi:10.1093/mnrasl/slt069.

Foreman-Mackey D., Hogg D. W., Lang D., Goodman J. (2013). `emcee`: The MCMC hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306–312.

Gelman A., Rubin D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472.

Kelly B. C., Bechtold J., Siemiginowska A. (2009). Are the variations in quasar optical flux driven by thermal fluctuations? *The Astrophysical Journal*, 698(1):895.

Neal R. M. (1993). *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto.

Pancoast A., et al. (2012). The lick AGN monitoring project 2011: Dynamical modeling of the broad-line region in Mrk 50. *The Astrophysical Journal*, 754(1):49.

Peterson B. M. (2008). The central black hole and relationships with the host galaxy. *New Astronomy Reviews*, 52(6):240–252.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, `http://www.R-project.org/`. ISBN 3-900051-07-0.

Stan Development Team (2013a). Stan: A `c++` library for probability and sampling, version 1.3. `http://mc-stan.org/`.

Stan Development Team (2013b). *Stan Modeling Language User's Guide and Reference Manual, Version 1.3*, `http://mc-stan.org/`.

Zu Y., Kochanek C. S., Peterson B. M. (2011). An alternative approach to measuring reverberation lags in active galactic nuclei. *The Astrophysical Journal*, 735(2):80.

Zu Y., Kochanek C. S., Kozłowski S., Udalski A. (2013). Is quasar optical variability a damped random walk? *The Astrophysical Journal*, 765(2):106.