Bayesian Inference with Applications to Censored Data

Hui Chiang Tay (CID: 00933219)

M3R Project

Department of Mathematics, Imperial College London

Supervisor: Dr. Daniel Mortlock

November 14, 2018

Abstract

This report explores the use of Bayesian inference, with specific extensions in both the theoretical approach and computational methods detailed to handle censored data. Both a mark and recapture problem, and a previously developed linear regression model with the independent variables modelled using a mixture of Gaussian functions are looked at to illustrate these methods. We focused on the simple case with one covariate, and detailed the implementation of a Gibbs sampler. The samples were then examined, and important trends highlighted.

This report is my own unaided work unless otherwise stated.

Contents

1 Introduction

Censoring of data is a common feature of data collected in many fields. For example, in medical studies, trials are often held within a limited time, and therefore the event times are not observed at the end of the trial, leading to the presence of 'right-censored' data. In such survival studies, the way the experiment is conducted may also result in censoring. For example, suppose a factory wished to determine how long a certain mechanical part can last before failure by measuring the event times. It can choose to observe a number of parts until a predetermined time, after which all remaining event times are right-censored, which is a form of Type 1 censoring. It can also choose to observe the parts until a predetermined number have failed, again right-censoring the remaining event times. This is called Type 2 censoring.

In addition, astronomical surveys usually involve observation of astronomical objects at various wavebands. However, due to limitations of measurement instruments, some objects may go undetected, and only knowledge of its upper limit may be possible. Such a phenomenon is known as 'left-censoring' [1]. Applying Bayesian methods in contexts where measurement effects are non-negligible is a well-developed field [2]

This report first outlines the theory behind Bayesian inference, as well as several sampling methods and how they can be extended to cope with censored data. It then applies the theory to a simple mark and recapture problem involving missing data, and a previously developed linear regression model involving interval-censored data that arises from measurement error. We focused on the simple case with one covariate and detailed the implementation of a Gibbs sampler, which we then used to confirm some results in the original paper. We aim to show that measurement effects contain significant information and should not be ignored as part of a statistical analysis, and demonstrate how to factor them in through Bayesian inference.

Throughout this report, we will most often denote y as the vector of observations, y_i, and θ as parameters of a distribution. $\mathsf{P}(y \mid \theta)$ is thus the probability density function of y, with known or unknown parameters θ.

1.1 Bayesian Inference

We first begin by developing a foundation of Bayesian inference, which can be summarised in 3 general steps [3]:

1. Formation of a full probability model involving all observable and unobservable quantities in the problem. Knowledge of the physical process should be incorporated via the prior distribution.

2. Calculation of the posterior distribution using both the prior distribution and

the collected data through the likelihood function, which represents the probability of obtaining a specific set of data given the model parameter(s).

3. Reaching conclusions about the parameter(s) of interest.

If we were interested in an unknown parameter, say θ, we first sum up our knowledge about this parameter through a prior distribution, $\mathsf{P}(\theta)$. We then collect data, say y_i where $i = 1, \ldots, n$, and incorporate it into our model as the likelihood, $\mathsf{P}(y \mid \theta)$, where where y is a row vector of length n, $(y_1, \ldots, y_n)$. This results in new probabilities that describe our knowledge of θ through the posterior distribution, $\mathsf{P}(\theta \mid y)$. The relationship between these distributions is given by Bayes' Theorem as[4],

$$\mathsf{P}(\theta \mid y) = \frac{\mathsf{P}(y \mid \theta)\mathsf{P}(\theta)}{\int \mathsf{P}(y \mid \theta)\mathsf{P}(\theta)\, d\theta}, \tag{1}$$

where we have used the Law of Total Probability to establish the integral in the denominator. This integral represents the marginal probability of the data, and since it does not depend on the parameters θ, we can treat it as a normalising constant, and thus obtain

$$\mathsf{P}(\theta \mid y) \propto \mathsf{P}(y \mid \theta)\mathsf{P}(\theta). \tag{2}$$

With a chosen probability model, the data affects the posterior distribution only through the likelihood function, $\mathsf{P}(y \mid \theta)$. This relationship obeys the likelihood principle, which states that all the information obtainable from an experiment is contained in the likelihood function, and that two likelihood functions of θ contain the same information about θ if they are proportional to each other [5]. The likelihood function is often represented through sampling distributions, and we can check the validity of our model using these sampling distributions.

Suppose that after forming the posterior distribution, we would like to estimate a new unknown but observable quantity, say $\tilde{y}$. We do this by making a predictive inference on $\tilde{y}$, by forming the posterior predictive distribution [3],

$$\begin{aligned} \mathsf{P}(\tilde{y} \mid y) &= \int \mathsf{P}(\tilde{y}, \theta \mid y)\, d\theta && (3) \\ &= \int \mathsf{P}(\tilde{y} \mid \theta, y)\mathsf{P}(\theta \mid y)\, d\theta && (4) \\ &= \int \mathsf{P}(\tilde{y} \mid \theta)\mathsf{P}(\theta \mid y)\, d\theta. && (5) \end{aligned}$$

where we have again applied Bayes' Theorem in Equation 4, and used the conditional independence of y and $\tilde{y}$ given θ in Equation 5. The predictive distribution not only allows us to provide estimates of $\tilde{y}$, but also allows us to form credible intervals for which our prediction may fall in.

1.2 Prior Distributions

A prominent feature of Bayesian inference that distinguishes it from the frequentist approach is the ability to incorporate prior knowledge of the parameter through the prior distribution. This may introduce subjectivity into the statistical model, in the sense that the the statistician is incorporating their individual beliefs into the model, instead of it being reliant only on observations from the data [6]. However, we should also recognise that any study will require scientific judgment in specifying the statistical model, which in itself will also introduce subjectivity [3]. For example, specifying a linear regression model makes assumptions about the relationship between the observed response variable, covariates and regression parameters. Discussed below are several ways to select a prior distribution for Bayesian inference.

1.2.1 Data-Based Priors

In an ideal scenario, prior knowledge about the underlying distribution will be available, which will make for a suitable choice of a prior. However, in the absence of such knowledge, the use of data from previous experiments could be used as a prior distribution for the current experiment, described by Lawson and Lesaffre as a 'sequential use of Bayes Theorem', outlined below [7].

Suppose y_k is the data collected in the k^{th} experiment, where in this case $k = 1, 2$. We have,

$$\begin{aligned} \mathsf{P}(\theta \,|\, y_1, y_2) &\propto \mathsf{P}(y_1, y_2 \,|\, \theta)\mathsf{P}(\theta) & (6)\\ &= \mathsf{P}(y_1 \,|\, \theta)\mathsf{P}(y_2 \,|\, \theta)\mathsf{P}(\theta) & (7)\\ &\propto \mathsf{P}(y_2 \,|\, \theta)\mathsf{P}(\theta \,|\, y_1), & (8)\end{aligned}$$

where we applied Bayes' Theorem in Equations 6 and 8, and assumed in Equation 7 that the experiments are independent of each other. In this manner, we are able to use the posterior distribution from the previous experiment as the prior for the current experiment, and extending it further, we have

$$\mathsf{P}(\theta \,|\, y_1, \ldots, y_k) \propto \prod_{i=1}^{k} \mathsf{P}(y_i \,|\, \theta)\mathsf{P}(\theta). \tag{9}$$

We see that as we do more experiments, the posterior distribution is dominated by the data, and the initial prior distribution becomes less important. However, it is important to note that here we have assumed that the results of the experiments are all independent of each other.

1.2.2 Non-informative Priors

However, it is often the case that we are ignorant about the underlying process, and wish to represent this ignorance through a non-informative prior in such a way that it has minimal effect on the posterior distribution. This leads us to an improper prior distribution, since the integral over the support of the prior is infinity. We highlight two ways to cope with this issue.

Firstly, as argued by Bayes and Laplace, we can adopt the principle of insufficient reason, which advocates the adoption of equal prior probabilities for a discrete probability density function, and a flat prior for the continuous case [7]. This appears to be a logical choice to represent ignorance, since we are indifferent to each member within the support of the density function.

However, we see that the choice of such a prior might bear some information because it might not be scale invariant. For example, suppose we used a uniform prior for θ, the probability parameter of a Binomial distribution. In this case, the likelihood function and the posterior distribution coincides, which might imply that this was a truly non-informative prior. However, if we were to now reparemeterise θ using $\psi = \log(\theta)$, and we used the flat prior on ψ, this would result in a non-flat prior for θ, and in turn a different posterior distribution for θ, thus implying that some additional information has been introduced [7].

To cope with the above problem, we can introduce the Jeffreys Prior, $\mathsf{P}(\theta) = \sqrt{|\,(I(\theta))\,|}$ where $I(\theta) = -\mathbb{E}\left(\frac{d^2 \log \mathsf{P}(y\,|\,\theta)}{d\theta^2}\right)$ is the Fisher information matrix. This is advantageous to the flat prior because it is invariant under parameterisation of θ [8]. Given a differentiable transformation, say $\phi = h(\theta)$, we see this property as follows [7],

$$\begin{aligned} I(\phi) &= -\mathbb{E}\left(\frac{d^2 \log \mathsf{P}(y\,|\,\phi)}{d\phi^2}\right) && (10)\\ &= -\mathbb{E}\left(\frac{d^2 \log \mathsf{P}(y\,|\,\theta)}{d\theta^2}\right)\left|\frac{d\theta}{d\phi}\right|^2 && (11)\\ &= I(\theta)\left|\frac{d\theta}{d\phi}\right|^2, && (12) \end{aligned}$$

where Equation 10 arises from the change of variables theorem.

When we adopt either of these non-informative priors, we often end up with an improper probability distribution for $\mathsf{P}(\theta)$ that does not integrate to a finite constant. We can interpret such a prior as the 'well-defined limit of a sequence of proper priors' [9], or the limit of data dependent prior distributions [10]. Generally, the posterior distribution will be more well-behaved due to the presence of the likelihood function, and we can ensure this by taking the limit of the distribution, which is sufficient for practical purposes [9].

1.2.3 Conjugate Priors

We say that a prior distribution is conjugate to the specified distribution of a likelihood function if the posterior distribution follows the same parametric form as the prior distribution. This results in a mathematically convenient form for the posterior distribution. The formal definition for the property of conjugacy as given by Gelman et al is as follows. If $\mathcal{F}$ is a class of sampling distributions $\mathsf{P}(y \mid \theta)$, and $\mathcal{P}$ is a class of prior distributions for θ, then the class $\mathcal{P}$ is conjugate for $\mathcal{F}$ if $\mathsf{P}(\theta \mid y) \in \mathcal{P}$ for all $\mathsf{P}(\cdot \mid \theta) \in \mathcal{F}$ and $\mathsf{P}(\cdot) \in \mathcal{P}$[3].

For example, suppose we have the likelihood taking the form of an Exponential distribution with rate parameter θ. We assume a Gamma distribution with hyperparameters α and β for the prior. The posterior distribution is thus,

$$\begin{aligned}\mathsf{P}(\theta \mid y) &\propto \mathsf{P}(\theta)\mathsf{P}(y \mid \theta)\\ &= \frac{\beta^{\alpha}}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta}\prod_{i=1}^{n}\theta e^{-\theta y_i}\\ &\propto \theta^{\alpha+n-1}e^{-\theta\left(\beta+\sum_{i=1}^{n} y_i\right)},\end{aligned} \tag{13}$$

which is a Gamma$(\alpha + n, \beta + \sum_{i=1}^{n} y_i)$ distribution.

Another example which we will make use of in the later sections is the scaled-inverse χ^2 distribution with degrees of freedom ν_0 and scale parameter τ^2. This takes the form,

$$\mathsf{P}\left(y \mid \nu_0, \tau^2\right) = \frac{\left(\tau^2\nu_0/2\right)^{\nu_0/2}}{\Gamma(\nu_0/2)}\frac{\exp\left(-\frac{\nu_0\tau^2}{2y}\right)}{y^{1+\frac{\nu_0}{2}}}. \tag{14}$$

If we form a likelihood function using a Normal distribution with known mean μ and unknown variance σ^2,

$$\begin{aligned}\mathsf{P}\left(y \mid \mu, \sigma^2\right) &= \prod_{i=1}^{n}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y_i-\mu)^2}{2\sigma^2}\right)\\ &\propto \left(\sigma^2\right)^{-\frac{n}{2}}\exp\left(-\frac{n}{2}\frac{\nu}{\sigma^2}\right),\end{aligned} \tag{15}$$

where $\nu = \frac{1}{n}\sum_{i=1}^{n}(y_i-\mu)^2$, with a scaled-inverse χ^2 prior with parameters as in Equation 14, we can form the posterior [3],

$$\begin{aligned}\mathsf{P}\left(\sigma^2 \mid y\right) &\propto \mathsf{P}\left(y \mid \sigma^2, \mu\right)\mathsf{P}\left(\sigma^2\right)\\ &\propto \left(\sigma^2\right)^{-\frac{n}{2}}\exp\left(-\frac{n}{2}\frac{\nu}{\sigma^2}\right)\left(\frac{\tau^2}{\sigma^2}\right)^{1+\frac{\nu_0}{2}}\exp\left(-\frac{\nu_0\tau^2}{2\sigma^2}\right)\end{aligned}$$

$$\propto \left(\sigma^2\right)^{-\left(\frac{n+\nu_0}{2}+1\right)} \exp\left(-\frac{\nu_0\tau^2 + n\nu}{2\sigma^2}\right). \tag{16}$$

We then have,

$$\sigma^2 \,|\, y \sim \mathrm{Inv}-\chi^2\left(\nu_0 + n, \frac{\nu_0\tau^2 + n\nu}{\nu_0 + n}\right) \tag{17}$$

1.2.4 Hierarchical Priors

Despite their mathematical convenience, the parametric form of conjugate priors may not be entirely suitable for representing the state of prior knowledge. We can improve this by using additional distributions to represent the hyperparameters, instead of fixing them to specific values [7]. Using the example in the previous section, we could fix the value of β, while allowing α to take an Exponential(λ) distribution. We call $\mathsf{P}(\alpha \,|\, \lambda)$ a hyperprior or a hierarchical prior [7]. We can now represent the prior for θ as the mixture prior,

$$\mathsf{P}(\theta \,|\, \lambda, \beta) = \int \mathsf{P}(\theta \,|\, \alpha, \beta)\mathsf{P}(\alpha \,|\, \lambda)\, d\alpha \tag{18}$$

2 Censored Data

We now wish to develop Bayesian methods for dealing with censored data whereby data is either missing or partially available due to the data collection mechanism. In this section, we will discuss 2 forms of censoring, namely missing data, and interval censored data.

2.1 Missing Data

We first deal with the case where data is missing. To do this, first suppose that we have data in the form of a matrix y, such that the ij^{th} entry represents observed covariate j on the i^{th} data point, where we have $i = 1 \dots n, j = 1 \dots m$. We then introduce the inclusion matrix I_{ij} such that $I_{ij} = 1$ if y_{ij} is observed, and $I_{ij} = 0$ if y_{ij} is missing. We also introduce y_{obs} and y_{mis} which refer to the group of elements of y which are observed and missing respectively.

As developed by Gelman et al., we can form the complete data likelihood as follows [3],

$$\mathsf{P}(y, I \mid \theta, \phi) = \mathsf{P}(y \mid \theta)\mathsf{P}(I \mid y, \phi). \tag{19}$$

The complete data likelihood is formed as a product of the individual likelihoods of y and I. The parameters for y are given by θ, while ϕ is an additional indexing parameter for I. Here we made the stable unit treatment assumption, such that the observed value of y is not affected by the data collection process [11]. However, the likelihood function in Equation 19 is not representative of the information that is present. Instead, we form the observed data likelihood by integrating out the missing data to obtain [3],

$$\mathsf{P}(y_{obs}, I \mid \theta, \phi) = \int \mathsf{P}(y, I \mid \theta, \phi)\, dy_{mis}. \tag{20}$$

We can now form the posterior distribution as follows,

$$\begin{aligned}
\mathsf{P}(\theta, \phi \mid y_{obs}, I) &\propto \mathsf{P}(\theta, \phi)\mathsf{P}(y_{obs}, I \mid \theta, \phi) && (21)\\
&= \mathsf{P}(\theta, \phi) \int \mathsf{P}(y, I \mid \theta, \phi)\, dy_{mis} && (22)\\
&= \mathsf{P}(\theta, \phi) \int \mathsf{P}(y \mid \theta)\mathsf{P}(I \mid y, \phi)\, dy_{mis}, && (23)
\end{aligned}$$

where we have used Equation 20 to arrive at Equation 22, and Equation 20 to arrive at Equation 23. Now, by integrating over ϕ, we arrive at

$$\mathsf{P}(\theta \mid y_{obs}, I) = \mathsf{P}(\theta) \int \int \mathsf{P}(\phi \mid \theta)\mathsf{P}(y \mid \theta)\mathsf{P}(I \mid y, \phi)\, dy_{mis} d\phi. \tag{24}$$

2.2 Interval Censored Data

We may also wish to employ Bayesian methods for interval censored data, where we only have information about a range of values which the censored quantity may take. Suppose we wish to form a posterior distribution of parameter θ, and we have i.i.d random samples, $y_1, \ldots, y_n$, some of which are right-censored beyond R, such that we do not observe the exact value of y_i, and instead only know that $y_i > R$. We can then form our inclusion model as follows [3],

$$\mathsf{P}(I_i = 1 \mid y_i) = \begin{cases} 1, & \text{if } y_i \leq R \\ 0, & \text{otherwise} \end{cases}, \tag{25}$$

and we can now form the posterior distribution,

$$\begin{aligned} \mathsf{P}(\theta \mid y_{obs}, I, R) &= \mathsf{P}(\theta)\mathsf{P}(y_{obs}, I \mid \theta, R) \\ &\propto \mathsf{P}(\theta) \prod_{i:I_i=1} f(y_{obs} \mid \theta) \prod_{i:I_i=0} \int_R^\infty f(y_i \mid \theta), \end{aligned} \tag{26}$$

where $f(y_i \mid \theta)$ is the probability distribution function of y_i.

3 Sampling Methods

Suppose we wish to find the posterior mean, $\mathbb{E}(\theta \mid y)$. We first notice that for a function $g(\theta)$, we have that

$$\mathbb{E}(g(\theta)) = \int \mathsf{P}(\theta \mid y) g(\theta) d\theta. \tag{27}$$

Setting $g(\theta) = \theta$ in Equation 27 will give us the mean of $\mathsf{P}(\theta \mid y)$. Now, given independent samples $\theta_1 \ldots \theta_n$, we can use

$$\mathbb{E}(g(\theta)) \approx \frac{1}{n} \sum_{i=1}^{n} g(\theta_i) \tag{28}$$

as an approximation to the integral in Equation 29, thus allowing us to estimate $\mathbb{E}(\theta \mid y)$ [12]. Obtaining samples of θ from the posterior is thus useful, and necessary if we do not have an analytical form of the posterior distribution available.

In such cases where it is impossible to sample from the posterior distribution analytically, we have to construct an approximate posterior distribution instead. This is often done through Markov Chain Monte Carlo (MCMC), a class of algorithms which work by drawing values from an approximate distribution, then correcting them to move closer towards the desired distribution [3].

In this section, we will look at two such algorithms, namely the Metropolis-Hastings algorithm and the Gibbs sampler, the latter of which we examine in greater detail because it can be easily extended to deal with censored data [13].

3.1 Metropolis-Hastings Algorithm

Suppose we wish to sample from a distribution that we are able to evaluate up to a normalising constant, $p(x)$. We can employ the Metropolis-Hastings algorithm which aims to construct a Markov Chain, denoted by $\mathsf{P}(x,y)$, which is the one-step transition probability from state y to state x, that converges to our desired distribution, $\mathsf{P}(x)$. We can do this by finding $p(x,y)$ that is time-reversible, and thus satisfies $\mathsf{P}(x)p(x,y) = \mathsf{P}(y)p(y,x)$. With a selected proposal distribution, $J(x \mid y)$, we can show that $p(x,y) = J(x \mid y)\alpha(x,y)$, where $\alpha(x,y) = \min\left(\frac{\mathsf{P}(y)p(y,x)}{\mathsf{P}(x)p(x,y)}, 1\right)$ is the acceptance probability [14].

We outline the Metropolis-Hastings algorithm as follows [3]:

1. Form an initial estimate of the parameter $\theta = \theta^0$ such that $\mathsf{P}\left(\theta^0 \mid y\right) > 0$.
2. Sample θ^* from a proposal distribution, $J_t\left(\theta^* \mid \theta^{t-1}\right)$, at time t.

3. Calculate the ratio,

$$r = \frac{\mathsf{P}(\theta^* \mid y)/J_t\left(\theta^* \mid \theta^{t-1}\right)}{\mathsf{P}(\theta^{t-1} \mid y)/J_t(\theta^{t-1} \mid \theta^*)} \tag{29}$$

4. Set

$$\theta^t = \begin{cases} \theta^* \text{ with probability } \min(r,1) \\ \theta^{t-1} \text{ otherwise} \end{cases} \tag{30}$$

5. Repeat steps 2-4 until convergence.

3.2 Gibbs Sampling

We can also employ the Gibbs sampler if we have a multivariate probability distribution which we cannot sample directly from, but we are able to sample from the conditional distributions of each variable. We make use of this by repeatedly sampling from the conditional distribution of each parameter, but updating the values of the other parameters at every iteration. The Gibbs sampler can also be viewed as a special case of the Metropolis-Hastings algorithm [3].

We outline how the Gibbs sampler is carried out in general. Suppose our posterior distribution is of the form $\mathsf{P}(\theta_1,\ldots,\theta_n \mid y)$. There are 3 main steps to sampling from the target posterior distribution [15]:

1. Form an initial estimate of the parameters, $\theta^0 = \left(\theta_1^0,\ldots,\theta_n^0\right)$. This could be a rough estimate, a classical estimator such as the maximum likelihood estimate, or even an arbitrary estimate. We denote θ_i^t to be the draw for the i^{th} parameter of θ at iteration t.

2. Sample from the conditional distribution of each parameter, and update the parameter as follows,

$$\begin{aligned} &\theta_1^1 \sim \mathsf{P}\left(\theta_1 \mid \theta_2^0,\ldots,\theta_n^0\right) \\ &\quad\vdots \\ &\theta_i^1 \sim \mathsf{P}\left(\theta_i \mid \theta_1^1,\ldots,\theta_{i-1}^1,\theta_{i+1}^0\cdots,\theta_n^0\right) \\ &\quad\vdots \\ &\theta_n^1 \sim \mathsf{P}\left(\theta_n \mid \theta_1^1,\ldots,\theta_{n-1}^1\right). \end{aligned}$$

This forms one iteration of the Gibbs sampler.

3. Repeat step 2 to obtain a sample closer to the target posterior distribution.

We note that the first few samples of any MCMC method, including both of those discussed in this section, are highly reliant on the initial values and these samples may not be from the desired probability density. We deal with this by discarding the first few values as part of a 'burn-in' phase to ensure that the final samples are drawn only after we have converged on our target posterior distribution [16].

In addition to this, samples which are drawn nearby to each other, for example at $t = 100$ and $t = 101$ may be correlated with each other and do not reflect the target distribution. We cope with this by picking samples at regular intervals, the length of which we can decide by measuring the autocorrelation between samples [17]. For example, we can run the Gibbs sampler 2000 times, discarding the first 1000 values, and picking every 10th value after that to obtain a sample of size 100.

To incorporate censored data into the Gibbs sampler, we employ a technique known as data augmentation [13]. We do this by incorporating the censored values into the Gibbs sampler as further unknowns, say y_s. In step 1, we form an estimate of y_s, and in step 2, we sample from its conditional distribution, $\mathsf{P}(y_s \,|\, \theta_1, \ldots, \theta_n)$.

3.3 Assessing Convergence

In the previous two algorithms, we want to iterate over the steps until convergence is reached so that the samples which we draw are indeed from the target posterior distribution. In order to ensure that convergence is indeed reached, some authors recommend to discard approximately half of the samples as part of the burn-in phase. However, it is also acknowledged to be a conservative choice, and ultimately arbitrary and may not be suitable for all situations [18]. Thus, we may need to monitor the Markov chain in order to ensure that it has indeed reached convergence. Here, we outline Gelman and Rubin's diagnostic which we used to assess convergence for the Gibbs Samplers implemented in the later sections.

The given approach uses a scalar quantity to estimate the convergence of each parameter of interest, say θ. The steps given by Gelman and Rubin are as follows [18]:

1. Simulate $m \geq 2$ sequences in parallel with starting values drawn from an overdispersed distribution, and discard the values obtained in the first half to obtain m sequences each of length n.

2. Calculate the average variance between sequences, B, and the average variance within sequences, W, as follows,

$$B = \frac{n}{m-1} \sum_{j=1}^{m} \left(\bar{\theta}_{\cdot j} - \bar{\theta} \right)^2 \text{ where } \bar{\theta}_{\cdot j} = \frac{1}{n} \sum_{i=1}^{n} \theta_{ij} \text{ and } \bar{\theta} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \theta_{ij} \tag{31}$$

$$W = \frac{1}{m}\sum_{j=1}^{m} s_j^2 \text{ where } s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(\theta_{ij} - \bar{\theta}_{\cdot j}\right)^2. \tag{32}$$

3. Estimate σ_θ^2 using

$$\sigma_\theta^2 = \frac{n-1}{n}W + \frac{1}{n}B. \tag{33}$$

4. Estimate the scale reduction factor $\hat{R}$ using

$$\hat{R} = \sqrt{\frac{\sigma_\theta^2}{W}}. \tag{34}$$

Since $\hat{R} \to 1$ as $n \to \infty$ [18], we can thus use this quantity to assess whether convergence has been reached $\left(\hat{R} \approx 1\right)$ and we have samples from the target distribution, or that we should continue running more simulations $\left(\hat{R} > 1\right)$.

3.4 Approximate Bayesian Computation

After obtaining the data, say y, to form a better approximate of the posterior distribution from the Gibbs sampler, we can also employ the Approximate Bayesian Computation rejection algorithm outlined below [19]:

1. Set a distance function $d(\cdot,\cdot)$ and tolerance threshold ϵ.
2. Sample a value of θ_i and generate simulated data $y_i \mid \theta_i$.
3. If $d(y, y_i) < \epsilon$, we add θ_i into our sample. If not, reject this value of θ_i and repeat step 2.
4. Repeat steps 2-3 until a sample of desired size is obtained

We observe that if we set the tolerance, ϵ, too high, the obtained distribution may not be accurate enough, but setting it too low will be computationally expensive [19]. In addition, we also need to select an appropriate distance function depending on the distributions present [19]. For example, the Euclidean distance function may not provide information on skewness of asymmetric distributions.

The case outlined above is applicable for continuous data. For discrete data, we can form an exact sample of the posterior distribution without the need for a distance function and tolerance. We have the steps of the algorithm as follows:

1. Sample a value of θ_i and generate simulated data $y_i \mid \theta_i$.
2. If $y_i = y$, we add θ_i into our sample. If not, reject this value of θ_i and repeat step 2.
3. Repeat steps 1-2 until a sample of desired size is obtained

4 Illustrative Example

We first illustrate the use of Bayesian statistics in a simple mark and recapture experiment. A fisherman wishes to find the population of fish in a pond, N. He catches 100 fish and tags them. A few days later, he fishes until he catches 20 tagged fish, and which point he has also caught 70 untagged fish, giving a total of 90 fish on the second day. Assuming that all fish are sampled independently and with equal probability, we wish to find a posterior distribution for N, using a noninformative prior distribution [3].

From the problem alone, we have the following variables:

N :	population of fish in the pond
n_1 :	number of fish caught on the first day
n_2 :	number of fish caught on the second day
n_t :	number of tagged fish caught on the second day
p_c :	probability of catching any single fish (only in Method 1)

We illustrate how to form the posterior distribution via two different interpretations.

4.1 Method 1

Using Bayes' Theorem,

$$\mathsf{P}(N,p_c\,|\,n_1,n_2,n_t) = \frac{\mathsf{P}(n_1,n_2,n_t\,|\,N,p_c)\mathsf{P}(N,p_c)}{\mathsf{P}(n_1,n_2,n_t)} \tag{35}$$

$$\propto \mathsf{P}(n_1,n_2,n_t\,|\,N,p_c)\mathsf{P}(N,p_c). \tag{36}$$

We assume the prior distribution, $\mathsf{P}(N,p_c)$, to be noninformative, and thus we have

$$\mathsf{P}(N,p_c) \propto \Theta(N)\Theta(p_c)\Theta(1-p_c). \tag{37}$$

Here, we use the Heaviside step function, $\Theta(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq n \end{cases}$. Since n_2 and n_t are dependent on N, p_c and n_1, we then have

$$\mathsf{P}(N,p_c\,|\,n_1,n_2,n_t) \propto \mathsf{P}(N,p_c)\mathsf{P}(n_1\,|\,N,p_c)\mathsf{P}(n_2,n_t\,|\,N,p_c,n_1), \tag{38}$$

and since n_2 is dependent on only N and p_c, while n_t is dependent on N, n_1, n_2, we arrive at the posterior distribution,

$$\mathsf{P}(N,p_c\,|\,n_1,n_2,n_t) \propto \mathsf{P}(N,p_c)\mathsf{P}(n_1\,|\,N,p_c)\mathsf{P}(n_2\,|\,N,p_c)\mathsf{P}(n_t\,|\,N,n_1,n_2) \tag{39}$$

4.2 Method 2

However, from the way the experiment is conducted, we may construct the posterior distribution in a different way. We use the same variables as above, except that this time n_1 and n_t are known parameters, since the fisherman could catch fish until a specified number of fish are caught on the first day, and tagged fish on the second. In addition, we no longer need to consider the probability p_c.

If we adopt this interpretation of the experiment, we have a case of sampling without replacement, and the distribution which reflects this most accurately is the negative hypergeometric distribution [9], which in this case describes the likelihood of catching $n_2 - n_t$ untagged fish with exactly n_t tagged fish, in a pond with $N - n_1$ untagged fish,

$$\begin{aligned} \mathsf{P}(N \mid n_1, n_2, n_t) &\propto \mathsf{P}(N)\,\mathsf{P}(n_2 \mid N, n_1, n_t) && (40) \\ &\propto \Theta(N)\,\frac{\binom{n_2-1}{n_2-n_t}\binom{N-n_2}{N-n_1-n_2+n_t}}{\binom{N}{N-n_1}}. && (41) \end{aligned}$$

Suppose we also have $n_1 \ll N$. We introduce $p_n = \frac{n_1}{N}$, the probability of catching a tagged fish on the second day. We can then approximate the negative hypergeometric distribution in Equation 33 to a negative binomial distribution as follows with the binomial coefficient $\binom{n}{x} = \frac{n!}{x!(n-x)!}$,

$$\mathsf{P}(N \mid n_1, n_2, n_t) \propto \Theta(N)\binom{n_2-1}{n_t-1}\left(\frac{n_1}{N}\right)^{n_t}\left(1-\frac{n_1}{N}\right)^{n_2-n_t}. \qquad (42)$$

Indeed, we can show that for a negative hypergeometric distribution with N elements of which K are successes, and elements are drawn successively until r failures are obtained, the distribution of X, that is the number of successes obtained, is as follows

$$\mathsf{P}(X=k) = \frac{\binom{k+r-1}{k}\binom{N-r-k}{K-k}}{\binom{N}{K}}. \qquad (43)$$

This tends to a negative binomial distribution with number of failures r and success probability p as follows,

$$\begin{aligned} \mathsf{P}(X=k) &= \binom{k+r-1}{k}\cdot\frac{(N-r-k)!}{(K-k)!\,(N-r-K)!}\cdot\frac{K!\,(N-k)!}{N!} && (44) \\ &= \binom{k+r-1}{k}\cdot\frac{K!\,(N-k)!}{(K-k)!\,N!}\cdot\frac{(N-r-k)!}{(N-r-K)!}, && (45) \end{aligned}$$

and if we have $N, K \to \infty$ and $\frac{K}{N} \to p$,

$$\mathsf{P}(X=k) = \binom{k+r-1}{k}\cdot\left(\frac{K}{N}\right)^{k}\left(\frac{N-K}{N}\right)^{r}. \qquad (46)$$

An example of this can be seen in Figure 1.

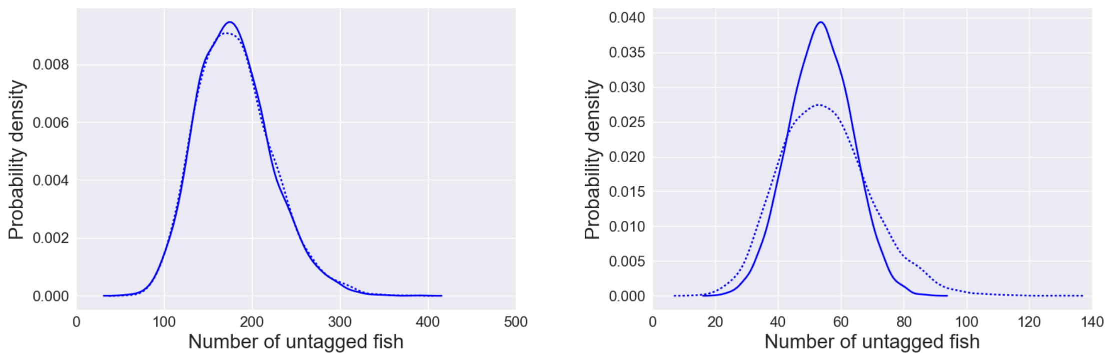

Figure 1: Sampling distributions of n_2. On the left, for both the negative hypergeometric (solid) and negative binomial (dotted) cases, where $N = 10000, n_1 = 1000, n_t = 20$. We see that the distributions are similar for large N and n_1, and $N \ll n_1$. On the right, we have $N = 150, n_1 = 40, n_t = 20$. We see that the approximation fares badly as the assumption that $\frac{n_1}{N}$ stays relatively constant in the negative binomial case (dotted) with each catch does not hold in the negative hypergeometric case (solid).

4.3 Prediction

Suppose we wish to now evaluate the probability that the next fish caught is tagged, $\tilde{n}$. Regardless of what we choose to be the posterior distribution, as we have seen in Equation 5, we can express the posterior predictive distribution as follows,

$$\mathsf{P}(\tilde{n} \mid n_1, n_2, n_t) = \sum_{N=0}^{\infty} \mathsf{P}(\tilde{n}, N \mid n_1, n_2, n_t) \tag{47}$$

$$= \sum_{N=0}^{\infty} \mathsf{P}(\tilde{n} \mid N)\mathsf{P}(N \mid n_1, n_2, n_t). \tag{48}$$

4.4 Missing Data

In this example, we will be looking at missing data, where the values of some variables are completely unknown. This is related to censoring, where the variables are only partially unknown.

Suppose now that, of the fish caught on the second day, 10 have missing fins, and thus we do not know if they were tagged or not. The actual number of tagged fish caught is therefore between 20 and 30. For simplicity, we assume that p_m, the probability of a fin being missing, is independent of N and n_2, that is the presence of a fin is not affected by whether or not it was tagged.

We form an inclusion model for the i^{th} fish, $I_i \sim \text{Bernoulli}(p_m)$. Suppose we now have the number of fish with observable fins caught on the second day, n_{obs}, and assuming a uniform prior for p_m, we can now form a new joint posterior distribution which includes the inclusion vector I [3],

$$\mathrm{P}(N,p_m,I\,|\,n_1,n_{obs},n_t) \propto \mathrm{P}(N,p_m)\mathrm{P}(n_{obs},I\,|\,n_1,n_t,N,p_m) \tag{49}$$

$$\propto \mathrm{P}(N)\mathrm{P}(p_m)\mathrm{P}(n_{obs}\,|\,n_1,n_t,N)\mathrm{P}(I\,|\,p_m,n_t). \tag{50}$$

We can then integrate over p_m to obtain the posterior for N,

$$\mathrm{P}(N\,|\,n_1,n_{obs},n_t,p_m,I) \tag{51}$$

$$\propto \mathrm{P}(N)\mathrm{P}(n_{obs}\,|\,n_1,n_t,N)\int \mathrm{P}(p_m)\mathrm{P}(I\,|\,p_m,n_t)\,dp_m \tag{52}$$

$$\propto \Theta(N)\frac{\binom{n_2-1}{n_{obs}-n_t}\binom{N-n_{obs}}{N-n_1-n_{obs}+n_t}}{\binom{N}{N-n_1}}\int \Theta(p_m)\Theta(1-p_m)\binom{n_2}{n_{obs}}p_m^{n_{obs}}(1-p_m)^{n_2-n_{obs}}\,dp_m. \tag{53}$$

4.5 Application of Gibbs Sampling

We illustrate the use of the Gibbs sampler with data augmentation using the fish example described earlier.

1. We use an initial estimate of $N = 500, n_2 = 100, t_1 = \cdots = t_5 = 0$ where $t_i = 1$ indicates that fish i with missing fin is actually tagged, and $t_i = 0$ if it is not. We also have data $n_1 = 100$ and $n_t = 20$.

2. We then iterate over the following 3 distributions.

$$t_i\,|\,N,n_1,n_2,n_t,t_1,\ldots,t_5 \sim \text{Bernoulli}\left(\frac{n_1-n_t-\sum_{j\neq i}t_j}{N-n_2-1}\right) \tag{54}$$

   For each fish with missing data, we incorporate the data augmentation step and treat it as a further unknown. Here, we draw from a Bernoulli distribution with the probability of the missing fin being tagged being the proportion of tagged fish caught out of the total number of fish caught on the second day.

$$n_2\,|\,N,n_1,n_t,t_1,\ldots,t_5 \sim \text{NHG}\frac{\binom{n_2-1}{n_2-n_t}\binom{N-n_2}{N-n_1-n_2+n_t+\sum_i t_i}}{\binom{N}{N-n_1}} \tag{55}$$

   We then draw n_2 from the negative hypergeometric distribution which we established in Method 2 previously. We note that values have been assigned to the missing data in the previous step, and we make use of them here.

$$\mathrm{P}(N\,|\,n_1,n_2,n_t,t_1,\ldots,t_5) \sim \frac{\binom{N-n_2}{N-n_1-n_2+n_t+\sum_i t_i}}{\binom{N}{N-n_1}}. \tag{56}$$

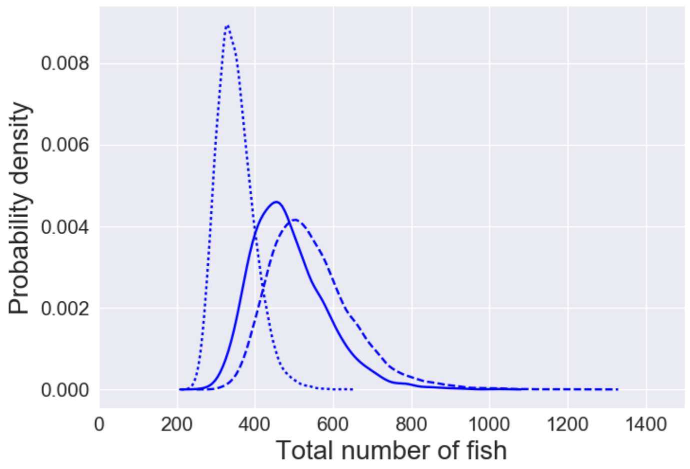

Figure 2: Gibbs posterior for the total number of fish in the pond,N. For the analytical posterior distributions, we have $n_1 = 100, n_2 = 100$, $n_t = 30$(dotted) and $n_t = 20$ (dashed). We see that the Gibbs posterior (solid) takes on an 'intermediate' distribution between the two cases.

For the posterior distribution, we used a uniform prior for N, with an upper limit of 2500 and a lower limit of $n_1 + n_2 - n_t$. We obtained the upper limit empirically by first obtaining an estimate, $\hat{N} = \frac{n_2}{n_t} n_1$, and then imposing the upper limit, $N_{max} = 5\hat{N}$

3. We then repeat step 2 20,000 times, discarding the first 10,000 values and taking every 10^{th} value thereafter to obtain a sample of size 1000.

We compare the distribution of $n_2 - n_t$ from the Gibbs sampler with the analytic distributions where $n_t = 20$ and $n_t = 25$ in Figure 2. We see that the peak of the Gibbs posterior is approximately in between that of the other 2 distributions to reflect the presence of missing data.

5 Linear Regression Model

In this section, we adapt the work of Kelly [2] and show how his model can be employed generally. Linear regression is an extremely common statistical approach used to describe relationships between a scalar dependent variable, and one or more independent variables, and is frequently used in scientific studies for both estimation and prediction. Censored data is common in such studies, and we outline a Bayesian approach to adapting the linear model to cope with censored data.

5.1 The Statistical Model

A linear regression model is formed as follows,

$$\eta_i = \alpha + \beta\xi + \epsilon_i, \tag{57}$$

where η_i are the dependent variables, ξ_i are the independent variables, α and β are the regression coefficients and the intrinsic scatter ϵ_i follows a Normal distribution with mean 0 and variance σ^2. For simplicity, we only use one covariate term, so β is a scalar quantity. The posterior distribution can be formed as,

$$\mathsf{P}\left(\eta_i \,|\, \alpha, \beta, \sigma^2, \xi_i\right) \sim \text{Normal}\left(\alpha + \beta\xi_i, \sigma^2\right). \tag{58}$$

However, as with most actual studies, measured values of the variables are observed instead of the true values, which could be attributed to limitations of measurement instruments. An error term is thus introduced as follows,

$$x_i = \xi_i + \epsilon_{x,i} \qquad y_i = \eta_i + \epsilon_{y,i} \tag{59}$$

where x_i and y_i are the observed independent and dependent variables respectively, and the error terms, $\epsilon_{x,i}$ and $\epsilon_{y,i}$, follow a bivariate Normal distribution with mean $(0,0)$, variances $\sigma^2_{x,i}$and $\sigma^2_{y,i}$, and covariance $\sigma_{xy,i}$. Here, we have set $\sigma_{xy,i} = 0$ for all i.

The distribution of the true dependent variables, ξ_i, are further modeled using a Gaussian mixture model,

$$\mathsf{P}\left(\xi_i \,|\, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\tau}^2\right) = \sum_{i=1}^{k} \frac{\pi_k}{\sqrt{2\pi\tau_k^2}} \exp\left[-\frac{(\xi_i - \mu_k)^2}{\tau_k^2}\right], \tag{60}$$

where $\sum_{i=1}^{k} \pi_k = 1$. We denote $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k), \boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)$ and $\boldsymbol{\tau}^2 = \left(\tau_1^2, \ldots, \tau_k^2\right)$. π_k represents the probability of using the kth Gaussian function with mean μ_k and variance τ_k^2 to simulate a specific ξ_i. The Gaussian mixture model is used because it can be flexibly adopted for many other distributions, and it is also conjugate to the Normal likelihood function which we have seen as part of linear regression in

Equation 58.

By integrating over ξ and η, the complete data likelihood function can thus be found as,

$$\mathrm{P}(x,y\,|\,\boldsymbol{\theta},\boldsymbol{\psi}) = \int\int \mathrm{P}(x,y\,|\,\xi,\eta)\mathrm{P}(\eta\,|\,\xi,\boldsymbol{\theta})\mathrm{P}(\xi\,|\,\boldsymbol{\psi})\,d\xi\,d\eta, \tag{61}$$

where $\boldsymbol{\theta} = \left(\alpha,\beta,\sigma^2\right)$ and $\boldsymbol{\psi} = \left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\tau}^2\right)$.

In addition to this, non-detections, or censoring of data can also be included in the model. The indicator variable, I_i is introduced such that $I_i = 1$ if y_i is detected, and $I_i = 0$ if y_i is censored. In the context of astronomical surveys, an object is considered to be detected if its measured flux falls is greater than a multiple of the background noise, say 3σ. Then, we have that $\mathrm{P}(I_i = 1\,|\,y_i) = 1$ if $y_i > 3\sigma$ and $\mathrm{P}(I_i = 0\,|\,y_i) = 1$ if $y_i < 3\sigma$. Due to the error from measurement, it is possible that objects with an intrinsic flux greater than the lower limit are censored, while objects with an intrinsic flux smaller than the lower limit are measured. Here, it is assumed that samples are selected based only on the independent variable.

The observed data likelihood function is then found as,

$$\mathrm{P}(x_{obs},y_{obs},\boldsymbol{I}\,|\,\boldsymbol{\theta},\boldsymbol{\psi_{obs}}) \tag{62}$$

$$\propto \prod_{i\in A_{obs}} \mathrm{P}(x_i,y_i\,|\,\boldsymbol{\theta},\boldsymbol{\psi_{obs}}) \prod_{j\in A_{mis}} \mathrm{P}\left(x_j\,|\,\boldsymbol{\psi_{obs}}\right)\int \mathrm{P}\left(I_j = 0\,|\,y_j,x_j\right)\mathrm{P}\left(y_j\,|\,x_j,\boldsymbol{\theta},\boldsymbol{\psi_{obs}}\right)dy_j \tag{63}$$

Here the likelihood function was split into two products for the observed and missing data, and then integrated over the missing data y_j, a method which we saw in Equation 20.

5.2 Implementation of Gibbs Sampler

A Gibbs sampler for the above model was described in detail by Kelly, and we implemented it for the case with only one covariate. The general approach to the Gibbs sampler follows what was described in Section 4.2, including the presence of data censoring.

5.2.1 Selection of Priors

Uniform priors are selected for α, β and σ^2. A Dirichlet$(1,\ldots,1)$ distribution is selected for $\pi_1,\ldots,\pi_k$ because it is conjugate to the Multinomial distribution [20], a property that we will make use of later. The Dirichlet distribution is the multivariate extension of the Beta distribution, and here, Dirichlet$(1,\ldots,1)$ is equivalent to a

Uniform distribution except that $\sum_{k=1}^{K}\pi_k = 1$.

$\mu_1,\ldots,\mu_K$ takes on a Normal distribution with mean μ_0 and variance u^2, while $\tau_1^2,\ldots,\tau_K^2$ are taken to have a Scaled-Inverse χ^2 distribution with 1 degree of freedom and scale parameter w^2. As highlighted in Section 2.2.3, the Scaled-Inverse χ^2 distribution has the useful property of being conjugate to the Normal distribution for a known mean and unknown variance.

The hyperparemeters introduced, μ_0, u^2 and w^2, do not have fixed values, but instead have their own distributions so that they are able to adapt to the data. μ_0 and w^2 are taken to have Uniform priors, while u^2 follows the same distribution as τ_k^2. The distributions are summarised as follows,

$$\alpha,\beta \sim \text{Uniform}(-\infty,\infty), \tag{64}$$
$$\sigma^2 \sim \text{Uniform}(0,\infty), \tag{65}$$
$$\boldsymbol{\pi} \sim \text{Dirichlet}(1,\ldots,1), \tag{66}$$
$$\mu_1,\ldots,\mu_K \sim \text{Normal}\left(\mu_0,u^2\right), \tag{67}$$
$$\tau_1^2,\ldots,\tau_K^2,u^2 \sim \text{Inv-}\chi^2\left(1,w^2\right), \tag{68}$$
$$\mu_0 \sim \text{Uniform}(-\infty,\infty), \tag{69}$$
$$w^2 \sim \text{Uniform}(0,\infty). \tag{70}$$

We note that the Uniform distributions specified above are improper. However, as mentioned in Section 1.2.2, in the presence of sufficient data, the posterior distributions will be proper distributions.

5.2.2 Outline of Gibbs Sampler

Now that the priors for the required parameters have been established, we now outline the steps for the Gibbs sampler provided by Kelly [2]:

1. Begin with initial guesses for $\boldsymbol{\eta},\boldsymbol{G},\boldsymbol{\theta},\boldsymbol{\psi}$ and the hyperparameters of $\boldsymbol{\psi}$, μ_0,u^2 and w^2. Here, $\boldsymbol{G}$ denotes the class membership of $\boldsymbol{\xi}$, that is which Gaussian function each ξ_i was drawn from, such that $G_{ik}=1$ if ξ_i was drawn from the kth Gaussian function, and $G_{ik}=0$ if it was not.

2. Draw values of y_i, if any are censored, from

$$\mathrm{P}(y_i\,|\,\eta_i,I_i=0) \propto \mathrm{P}(I_i=0\,|\,y_i)\mathrm{P}(y_i\,|\,\eta_i) \tag{71}$$

   In this case, we keep drawing each y_i from a Normal$\left(\eta_i,\sigma_{y,i}^2\right)$ distribution until a value for y_i is obtained that is below the censoring threshold.

3. Draw values of the true dependent variable, $\boldsymbol{\xi}$ from a Normal$\left(\hat{\xi}_i, \hat{\sigma}^2_{\hat{\xi},i}\right)$ distribution, where

$$\hat{\xi}_i = \sigma^2_{\hat{\xi},i}\left[x_i + \frac{\eta_i - \alpha}{\beta\sigma^2} + \frac{\mu_k}{\tau_k^2}\right], \tag{72}$$

$$\sigma^2_{\hat{\xi},i} = \left[\frac{1}{\sigma^2_{x,i}} + \frac{\beta^2}{\sigma^2} + \frac{1}{\tau_k}^2\right]^{-1}. \tag{73}$$

The first term for both quantities is the contribution from the measured data x_i, the second term is the contribution obtained from linear regression using the current estimates of the parameters α and β, and the third term is the contribution from the specific Gaussian function from which ξ_i was drawn.

We note here that there is an error in the calculation of $\hat{\xi}_i$ in the original paper, where the second term in Equation 72 was printed as $\frac{\beta(\eta_i-\alpha)}{\sigma^2}$ and we corrected it above. These values can be derived using the distribution in Equation 58

4. Draw new values of the true independent variable, $\boldsymbol{\eta}$ from a Normal$\left(\hat{\eta}_i, \sigma^2_{\hat{\eta},i}\right)$ distribution, where

$$\hat{\eta}_i = \sigma^2_{\hat{\eta},i}\left[\frac{y_i + (\xi_i - x_i)/\sigma^2_{x,i}}{\sigma^2_{y,i}} + \frac{\alpha + \beta\xi_i}{\sigma^2}\right], \tag{74}$$

$$\sigma^2_{\hat{\eta},i} = \left[\frac{1}{\sigma^2_{y,i}} + \frac{1}{\sigma^2}\right]^{-1}. \tag{75}$$

Similar to Equation 72, the first term for both quantities is the contribution from the measured data, and the second term is the contribution from regression.

5. Draw new values of the class membership, $\boldsymbol{G}$, where $\boldsymbol{G}_i$ is drawn from a Multinomial$(1, \mathbf{q})$ distribution. The probabilities $\boldsymbol{q} = (q_1, \ldots, q_K)$ are formed by,

$$q_k = \frac{\pi_k N\left(\xi_i \mid \mu_k, \tau_k^2\right)}{\sum_{j=1}^{K} \pi_j N\left(\xi_i \mid \mu_j, \tau_j^2\right)}. \tag{76}$$

The numerator for q_k is formed by multiplying the current estimates of π_k, that is the probability of drawing from the kth Gaussian function, and the probability density function of a Normal$\left(\mu_k, \tau_k^2\right)$ evaluated at ξ_i. This is normalised by summing over the K such products.

6. Draw new values of the regression parameters, α and β from a Normal$(\hat{\boldsymbol{c}}, \Sigma)$ distribution. This follows from simple linear regression, where

$$\hat{\boldsymbol{c}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{\eta}, \tag{77}$$

$$\Sigma = \sigma^2\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}, \tag{78}$$

and $\boldsymbol{X}$ is the $n \times 2$ matrix formed with the first column being a column of 1s, and the second column being $\boldsymbol{\xi}$.

7. Draw a new value for the intrinsic variance, σ^2 from an Inv-$\chi^2\left(n-2, s^2\right)$ distribution, where the unbiased estimator of variance, $s^2 = \frac{RSS}{n-p}$, is given by,

$$s^2 = \frac{1}{n-2}\sum_{i=1}^{n}\left[\eta_i - (\alpha + \beta\xi_i)\right]^2. \tag{79}$$

A scaled-inverse χ^2 distribution here because it is conjugate to the normally distributed likelihood function, as seen in Section 2.2.3.

8. Draw new values of $\boldsymbol{\pi}$ from a Dirichlet$(n_1 + 1, \ldots, n_K + 1)$ distribution, where $n_k = \sum_{i=1}^{n} G_{ik}$ is the number of ξ_i that have been drawn from the kth Gaussian function. As mentioned previously, the Dirichlet distribution is used here because it is conjugate to the Multinomial distribution from which $\boldsymbol{G}$ is drawn.

9. Draw new values of μ_k from a Normal$\left(\hat{\mu}_k, \sigma^2_{\hat{\mu}_k}\right)$ distribution, where

$$\hat{\mu}_k = \frac{\frac{\mu_0}{u^2} + \frac{n_k\bar{\xi}_k}{\tau_k^2}}{\sigma^2_{\hat{\mu}_k}}, \qquad \sigma^2_{\hat{\mu}_k} = \left(\frac{1}{u^2} + \frac{n_k}{\tau_k^2}\right)^{-1}, \qquad \bar{\xi}_k = \frac{1}{n_k}\sum_{i=1}^{n} G_{ik}\xi_i. \tag{80}$$

$\bar{\xi}_k$ represents the average of the ξ terms that have been drawn from the kth Gaussian function. $\hat{\mu}_k$ and $\sigma^2_{\hat{\mu}_k}$ thus take into account contributions from the hyperparameters u^2 and μ_0, and from the current values of $\boldsymbol{\xi}$.

10. Draw new values of τ_k^2 from an Inv-$\chi^2\left(n_k + 1, t_k^2\right)$ distribution, where,

$$t_k^2 = \frac{1}{n_k + 1}\left[w^2 + \sum_{i=1}^{n} G_{ik}\left(\xi_i - \mu_k\right)^2\right]. \tag{81}$$

This is similar to how s^2 was formed for σ^2 in Equation 79, except that now it takes into account the specific Gaussian functions from which each ξ_i was drawn, and the hyperparameter w^2.

11. Draw a new value of μ_0 from a Normal$\left(\bar{\mu}, \frac{u^2}{K}\right)$ distribution, where $\bar{\mu} = \frac{1}{K}\sum_{k=1}^{K}\mu_k$. This is the sampling distribution for the mean of K independent Normal$\left(\mu_0, u^2\right)$ distributions.

12. Draw a new value of u^2 from an Inv-$\chi^2\left(K+1, \hat{u}^2\right)$ distribution, where

$$\hat{u}^2 = \frac{1}{K+1}\left[w^2 + \sum_{k=1}^{K}(\mu_k - \mu_0)^2\right]. \tag{82}$$

This is similar to the case of τ_k^2 in Equation 81.

13. Draw a new value of w^2 from a Gamma(a, b) distribution, where

$$a = \frac{1}{2}(K+3), \qquad b = \frac{1}{2}\left(\frac{1}{u^2} + \sum_{k=1}^{K}\frac{1}{\tau_k^2}\right). \tag{83}$$

This arises because the posterior of w^2 takes the form of a Gamma distribution which we have derived as follows,

$$\begin{aligned} \mathsf{P}\left(w^2 \mid u^2, \tau^2\right) &\propto \mathsf{P}\left(u^2 \mid w^2\right)\mathsf{P}\left(\tau^2 \mid w^2\right) & (84) \\ &\propto \frac{\left(w^2/2\right)^{\frac{1}{2}}}{\Gamma\left(\frac{1}{2}\right)}\frac{\exp\left(-\frac{w^2}{2u^2}\right)}{(u^2)^{3/2}}\prod_{k=1}^{K}\frac{\left(w^2/2\right)^{\frac{1}{2}}}{\Gamma\left(\frac{1}{2}\right)}\frac{\exp\left(-\frac{w^2}{2\tau^2}\right)}{(\tau^2)^{3/2}} & (85) \\ &\propto \left(w^2\right)^{\frac{1}{2}(K+1)}\exp\left[-\frac{1}{2}\left(\frac{1}{u^2} + \sum_{k=1}^{K}\frac{1}{\tau_k^2}\right)\right]. & (86) \end{aligned}$$

This is indeed the form of a Gamma distribution as parameterised in Equation 83, and where we have taken $\tau_k^2, u^2 \mid w^2 \sim$ Inv-$\chi^2\left(1, w^2\right)$ as stated in Equation 68.

14. Repeat steps 2-13 until convergence is reached. We measured this by running in parallel the Gibbs sampler for 3 different starting points until $|\hat{R} - 1| < 10^{-2}$, where $\hat{R}$ is the Gelman and Rubin diagnostic highlighted in Section 4.3.

5.3 Simulation and Results

Before looking at the results from the Gibbs sampler, we first look at the mixture model used to simulate the distribution from which $\boldsymbol{\xi}$ is drawn. Kelly provided the following distribution,

$$\mathsf{P}(\xi) \propto e^{\xi}\left(1 + e^{2.75\xi}\right)^{-1}, \tag{87}$$

and we tested the mixture model for $K = 2$ to see how well it could replicate the above distribution, as well as a Normal$(0,1)$ and Uniform$(-1,1)$ distribution. With reference to the top left plot of Figure 3, for the distribution in Equation 87, we see that the mixture model simulates it well, especially considering the asymmetry

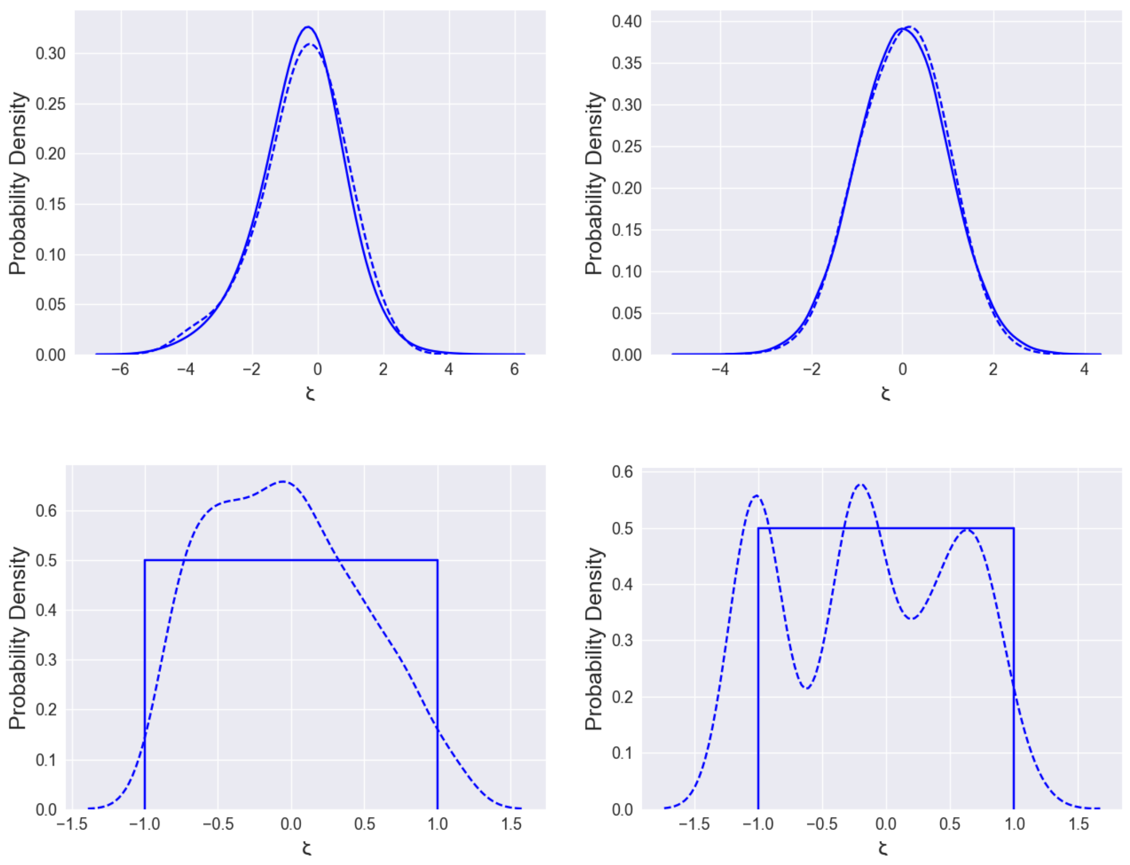

Figure 3: Simulated and analytical distributions for ξ. For all 4 plots, the solid line represents the relevant analytical distribution and the dashed line represents the simulation from the mixture model. The top left plot displays the distribution provided by Kelly, while the top right plot displays a Normal$(0, 1)$ distribution, with both mixture models using $K = 2$ Gaussian functions. The bottom left and right plots both display Uniform$(-1, 1)$ functions, with $K = 2$ and $K = 4$ respectively.

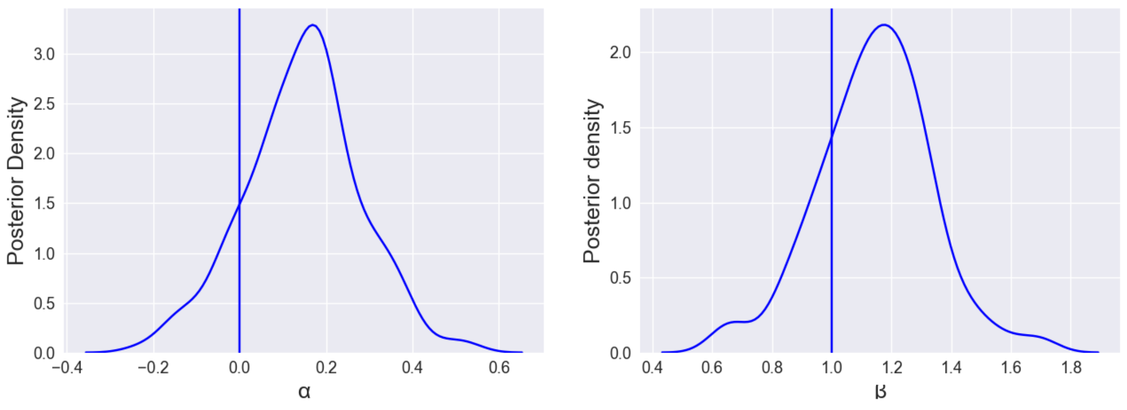

Figure 4: Posterior samples of α (left) and β (right). The true values for both regression coefficients are marked by solid vertical lines.

present in the distribution. We can see that the simulated distribution is almost exactly the same as the analytical one. This is also the case for the Normal$(0, 1)$ distribution in the top right plot of Figure 3, which we would expect since the mixture model is a sum of Normal distributions anyway.

However, from the bottom plots of Figure 3, we see that the mixture model is unsuitable for simulating the Uniform$(-1, 1)$ distribution, since the support of a Normal distribution is the real line and thus samples outside the support of the Uniform distribution will often be drawn. Should we wish to continue with the mixture model, then we would need to use $K > 4$ Gaussian functions, and even then it might still be unsuitable.

We simulated 100 data points, where $\boldsymbol{\xi}$ was drawn from a Normal$(0, 1)$ distribution, and we set $\alpha = 0$,$\beta = 1$ and $\sigma^2 = 1$ to form $\boldsymbol{\eta}$. We set $\sigma_x^2 = \sigma_y^2 = 0.1$ and $\sigma_{xy} = 0$ to form x and y, following which all values below $y = 0$ were censored. We then ran the Gibbs sampler until convergence was reached, and obtained a sample posterior distribution of size 200.

With reference to Figure 4, we observe that the true values of α and β lie quite close to the median of the respective sampled posterior densities. We can thus say that the bounds on the regression parameters from the posterior distribution are trustworthy.

Estimating the regression coefficients using the posterior medians from the Gibbs sampler provides a much more accurate estimate than if we had not factored in the censoring of data and estimated the coefficients normally. To show this, we found the ordinary least squares estimate of α and β which only took into account the observed data, and we see in Figure 5 that the estimates are markedly different from the true values.

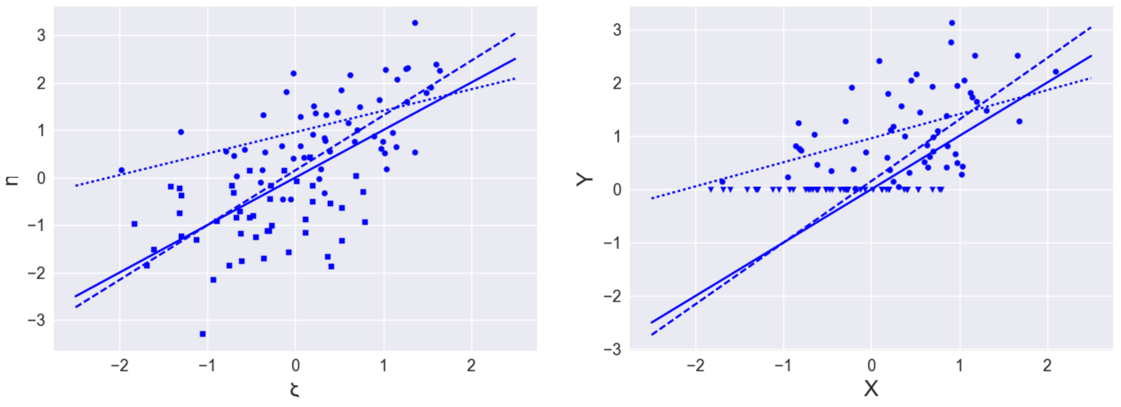

Figure 5: Regression estimates of α and β. The 3 lines drawn in each plot have coefficients which are the true values (solid), estimated using the posterior medians from the Gibbs sampler (dashed), and estimated using ordinary least squares after removing the censored data (dotted). The points on the left plot are the true values of $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$, with the points censored during measurement denoted by squares. The points on the right plot are the measured values, $\boldsymbol{x}$ and $\boldsymbol{y}$ with the censored points denoted by triangles.

The Gibbs sampler can also provide us with the joint posterior density of α and β, which we see in Figure 6. These 2 quantities exhibit a negative correlation because a lower intercept would require a higher gradient for the line to pass through the main body of the points. From the plot on the right, we see that the true values of the regression coefficients lie well within the sampled values, as the true regression line lies within the 'spread' of sampled regression lines.

We observe the two plots in the first row of Figure 7, where the plot on the left shows the 200 draws of ξ_i and η_i for a single censored data point, compared to the plot on the right which shows the same for an uncensored data point. For a single censored point, we only know the value of x_i. The Gibbs sampler thus draws values of η_i that are spread over a larger range below the censoring threshold, in this case $y = 0$, than if y_i had been previously known. Here we have calculated the variance over η for the censored point to be ≈ 0.532 while that for the uncensored point is ≈ 0.102.

We also observe that the presence of the censoring threshold does not necessarily imply that all sampled values of η_i will fall below it due to contribution of the regression parameters and the presence of measurement error. For the uncensored data point, the sample is concentrated between the predicted values based on the regression parameters, and the measured variables, which is what we expect. The bottom plot shows a single sample of the full $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$. We see that as the values of η move toward 0 and below, the points deviate more from the line. This corroborates our observations from the above two plots.

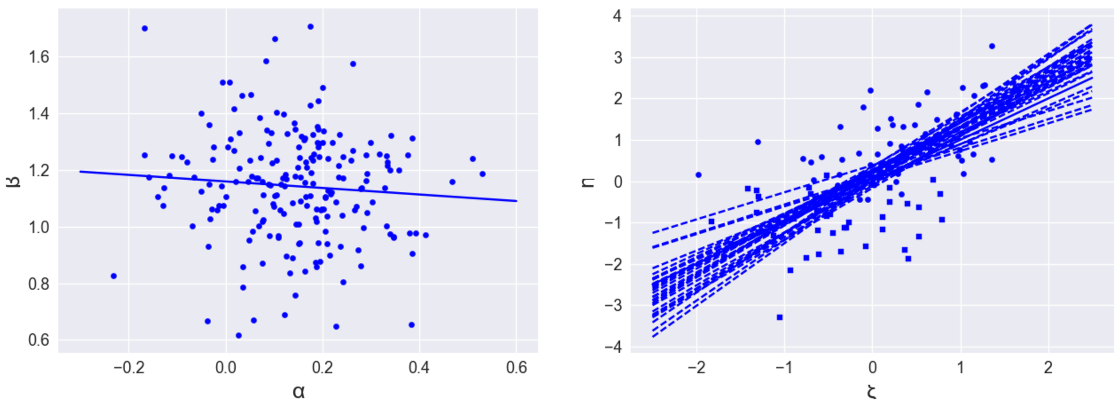

Figure 6: Joint posterior density of α and β. The figure on the left shows β plotted against α, along with the ordinary least squares regression line for these two parameters drawn, showing a slightly negative correlation. The true regression line (solid) on the figure on the right is not visible since it is lying within 30 lines (dashed) with α and β randomly drawn from the points in the plot on the left.

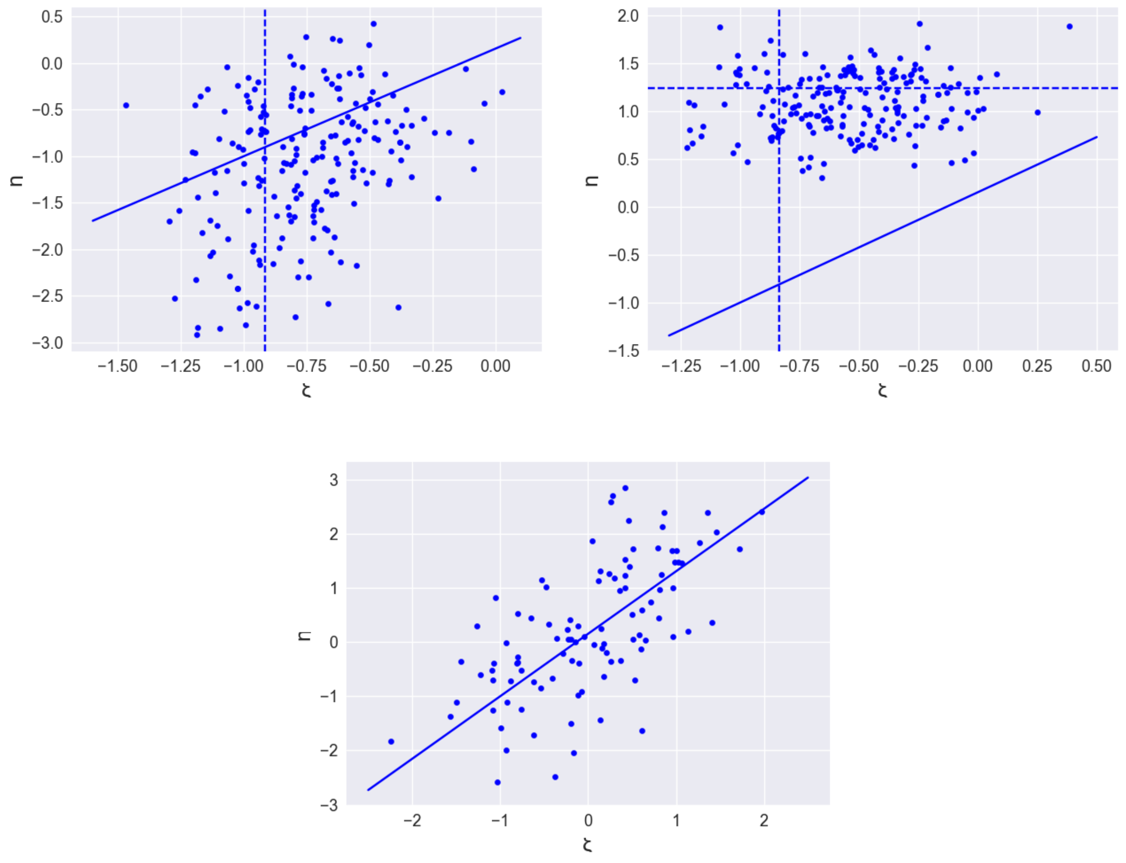

Figure 7: Samples of specific values of η and ξ. In all 3 plots, the solid line is the line given by the posterior medians of α and β. The figure on the top left shows the 200 samples of a single censored data point, with the measured value of x_i given by the dashed vertical line. The figure on the top right shows the 200 samples of a single uncensored data point, with the measured values of x_i and y_i given by the dashed vertical and horizontal lines respectively. The figure at the bottom plots all the values of η and ξ for a single sample.

6 Conclusion

In this report, we have seen the theoretical approach in factoring in data censoring during Bayesian inference, and this approach is generalisable to other forms of censorship. This includes the addition of an inclusion vector, and integrating over the missing data to obtain an analytical form for the observed data likelihood function and the posterior distribution where possible. We have also seen how computational methods such as Markov Chain Monte Carlo algorithms can incorporate a data augmentation step that allows it to sample from the posterior even with the presence of censored data.

In addition to this, we implemented such algorithms in two scenarios which looked at both missing data and interval-censored data, and showed that censored data does carry information, and cannot be ignored in a statistical model. Trends involving the posterior distributions of the dependent and independent variables, and the joint posterior of of the regression parameters were examined, and in the process corroborated Kelly's conclusion that the posterior distributions of the regression parameters provide reasonable bounds for the parameters [2], and in the presence of censored data may outperform other estimators.

Although we have only dealt with two specific scenarios in detail, the methods involved in both are generalisable. This is especially so for the linear regression model, because the Gaussian mixture model on the independent variables is flexible, and can admit a variety of data generating models. Possible areas for exploration in the future could include developing the theory further to cope with selection effects arising from experimental design, and applying the linear model to other fields of survival analysis, such as medical studies.

7 Acknowledgements

I would like to thank Dr. Daniel Mortlock for his invaluable support and guidance, as well as time and effort spent mentoring me throughout the course of this project, without which this project would not have been possible.

8 Bibliography

[1] E.D. Feigelson. Censoring in Astronomical Data Due to Nondetections, pages 221–237. Springer New York, New York, NY, 1992. ISBN 978-1-4613-9290-3. doi: 10.1007/978-1-4613-9290-3_24. URL https://doi.org/10.1007/978-1-4613-9290-3_24. pages 3

[2] B.C. Kelly. Some aspects of measurement error in linear regression of astronomical data. The Astrophysical Journal, 665(2):1489, 2007. URL http://stacks.iop.org/0004-637X/665/i=2/a=1489. pages 3, 20, 22, 31

[3] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. Bayesian Data Analysis, Third Edition. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013. ISBN 9781439840955. pages 3, 4, 5, 7, 9, 10, 11, 12, 15, 18

[4] R. Christensen, W. Johnson, A. Branscum, and T.E. Hanson. Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2011. ISBN 9781439803554. URL https://books.google.co.uk/books?id=qPERhCbePNcC. pages 4

[5] J.O. Berger and R.L. Wolpert. Chapter 3: The Likelihood Principle and Generalizations, volume Volume 6 of Lecture Notes–Monograph Series, pages 19–64. Institute of Mathematical Statistics, Hayward, CA, 1988. doi: 10.1214/lnms/1215466214. URL https://doi.org/10.1214/lnms/1215466214. pages 4

[6] A. Gelman and C. Hennig. Beyond subjective and objective in statistics. Journal of the Royal Statistical Society: Series A (Statistics in Society), 180(4):967–1033. doi: 10.1111/rssa.12276. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/rssa.12276. pages 5

[7] E. Lesaffre and A.B. Lawson. Bayesian Biostatistics. Statistics in Practice. Wiley, 2012. ISBN 9781118314579. URL https://books.google.co.uk/books?id=WV7KVjEQnJMC. pages 5, 6, 8

[8] R. Yang and J.O. Berger. A catalog of noninformative priors. Purdue University, 1998. pages 6

[9] E.T. Jaynes. Probability Theory: The Logic of Science. Cambridge University Press, 2003. doi: 10.1017/CBO9780511790423.005. pages 6, 16

[10] H. Akaike. The interpretation of improper prior distributions as limits of data dependent proper prior distributions. Journal of the Royal Statistical Society. Series B (Methodological), 42(1):46–52, 1980. ISSN 00359246. URL http://www.jstor.org/stable/2984737. pages 6

[11] D.B. Rubin. Randomization analysis of experimental data: The fisher randomization test comment. **Journal of the American Statistical Association**, 75 (371):591–593, 1980. ISSN 01621459. URL `http://www.jstor.org/stable/2287653`. pages 9

[12] A. Kak. Monte carlo integration in bayesian estimation. Purdue University, June 2014. URL `https://engineering.purdue.edu/kak/Tutorials/MonteCarloInBayesian.pdf`. pages 11

[13] A.F.M. Smith and G.O. Roberts. Bayesian computation via the gibbs sampler and related markov chain monte carlo methods. **Journal of the Royal Statistical Society. Series B (Methodological)**, 55(1):3–23, 1993. ISSN 00359246. URL `http://www.jstor.org/stable/2346063`. pages 11, 13

[14] A. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. **The American Statistician**, 49(4):327–335, 1995. doi: 10.1080/00031305.1995.10476177. URL `https://www.tandfonline.com/doi/abs/10.1080/00031305.1995.10476177`. pages 11

[15] A. Minin, K. Auranen, and M.E. Halloran. MCMC I. 8th Summer Institute in Statistics and Modeling in Infectious Diseases, July 2016. pages 12

[16] K. Sahlin. Estimating convergence of markov chain monte carlo simulations. Master's thesis, Stockholms Universitet, Sweden, 2011. pages 13

[17] A.B. Owen. Statistically efficient thinning of a markov chain sampler. Technical report, Stanford University, Department of Statistics, 11 2015. pages 13

[18] A. Gelman and D.B. Rubin. Inference from iterative simulation using multiple sequences. **Statistical Science**, 7(4):457–472, 1992. ISSN 08834237. URL `http://www.jstor.org/stable/2246093`. pages 13, 14

[19] B.M. Turner and T.V. Zandt. A tutorial on approximate bayesian computation. **Journal of Mathematical Psychology**, 56(2):69 – 85, 2012. ISSN 0022-2496. doi: https://doi.org/10.1016/j.jmp.2012.02.005. URL `http://www.sciencedirect.com/science/article/pii/S0022249612000272`. pages 14

[20] S. Tu. The dirichlet-multinomial and dirichlet-categorical models for bayesian inference. 2014. pages 21

9 Appendix

Contained in this section are the functions used in Sections 5 and 6. Exact code for plotting graphs are not included.

9.1 Code used in Section 4

The functions included in this section are ptrials, ncr, nhypergeom, bintrials, htrials, Ntrials and gibbs, with descriptions for each within the code itself.

```
import numpy as np
import matplotlib.pyplot as plt
import operator as op
from functools import reduce
import seaborn as sns
sns.set(color_codes=True)
import scipy


def ptrials(N,pcatch,ncatch1,ncatch2,ntrials):
    #Method 1 as described in Section 5.1

    #Inputs
    #N: Total number of fish in the pond
    #pcatch: Probability of catching a fish with each ncatch
    #ncatch1: Number of times to try catching a fish the first day
    #ncatch2: Number of times to try catching a fish the
    second day
    #ntrials: Number of times to run the experiment

    #Outputs
    #n1: Fish tagged on the first day
    #n2: Total fish caught on the second day
    #nt: Tagged fish caught on the second day

    n1 = np.zeros(ntrials)
    n2 = np.zeros(ntrials)
    nt = np.zeros(ntrials)

    for j in range(ntrials):
        n1[j] = np.random.binomial(ncatch1,pcatch)
        remaining = [N,n1[j]]
        for i in range(ncatch2):
            if np.random.binomial(1,pcatch) == 1:
                if np.random.binomial(1,remaining[1]
                /remaining[0]) == 1:
                    n2[j] = n2[j] + 1
                    nt[j] = nt[j] + 1
                    remaining[0] -= 1
                    remaining[1] -= 1
                else:
                    n2[j] = n2[j] + 1
                    remaining[0] -= 1

    return n1,n2,nt

def ncr(n, r):
    #'Choose' function ie computes n!/(n-r)!r!
    r = min(r, n-r)
    numer = reduce(op.mul, range(n, n-r, -1), 1)
    denom = reduce(op.mul, range(1, r+1), 1)

    return numer//denom

def nhypergeom(N, K, r):
    #Creates a list of probabilities following of an r.v.
    following a negativehypergeometric distribution

    #Inputs
    #N: Total number of objects
    #K: Total number of 'success' elements (i.e. untagged fish)
    #r: Total number of 'failure' elements to stop the experiment
    at (i.e. number of tagged fish to catch)

    #Output
    #k: Vector of probabilities over the support of K

    k = np.zeros(K+1) #Support of K

    for i in range(K+1):
        k[i] =  ncr(i+r-1,i)*ncr(N-r-i,K-i)/ncr(N,K)

    return k

def bintrials(nexp, N, n1, nt):
    #Negative binomial method as described in Section 5.2

    #Inputs
    #nexp: Number of times to run the experiment
    #N: Total number of fish in the pond
    #n1: Number of fish caught on the first day
    #nt: Number of tagged fish caught on the second day

    #Output
    #ntrials: nexp negative binomial trials , each with number of
    untagged fish

    pn = n1/N #probability of catching a tagged fish
    ntrials = np.zeros(nexp)

    for i in range(nexp):
        ntrials[i] = np.random.negative_binomial(nt,pn)

    return ntrials

def htrials(nexp, N, n1, nt):
    #Negative hypergeometric method as in Section 5.2

    #Inputs
    #nexp: Number of times to run the experiment
    #N: Total number of fish in the pond
    #n1: Total number of tagged fish in the pond
    #nt: Number of tagged fish caught to stop experiment

    #Output
    #ntrials: Nexp trials of untagged fish caught

    ntrials = np.zeros(nexp)
    probs = nhypergeom(N,N−n1,nt) #probabilities of each k
    support = np.arange(len(probs)) #support of K

    for i in range(nexp):
        #Sample from the negative hypergeometric distribution
        ntrials[i] = np.random.choice(support,p=probs)

    #plt.hist(ntrials)
    #plt.show()

    return ntrials

def Ntrials(nexp, n1, n2, nt):
    #Sampling from the posterior distribution of N
    #Inputs
    #nexp: Number of times to run the experiment
    #n1: Total number of tagged fish in the pond
    #n2: Total number of fish caught on second day
    #nt: Number of tagged fish caught to stop experiment

    #Output
    #ntrials: nexp trials

    ntrials = np.zeros(nexp)
    support = np.arange(n1+n2-nt,2500) #support of N
    probs = np.zeros(len(support)) #probability vector

    for i in range(len(support)):
        num = ncr(support[i]-n2,support[i]-n1-n2+nt)
        den = ncr(support[i],support[i]-n1)
        probs[i] = num/den

    probs = probs/sum(probs)
    for i in range(nexp):
        #Sample from the negative hypergeometric distribution
        ntrials[i] = np.random.choice(support,p=probs)

    return ntrials

def gibbs(N, n1, n2, nt, t, iters):
    #Gibbs sampler with t number of fish with unknown tags

    #Input:
    #N: Total number of fish in the pond
    #n1: Total number of tagged fish in the pond
    #n2: Total number of fish caught on second day
    #nt: Number of tagged fish caught to stop the experiment
    #t: Vector of fish with missing fins
    #iters: Number of iterations

    #Output:
    #nsample: Sample of total number of fish in the pond
    #(Can also return number of untagged fish caught)

    nsample = np.zeros(int(iters/20))

    count = 0
    fill = 0
    while nsample[-1] == 0:
        count += 1
        print(count)
        t = t.astype(int)
        #Sample the fish with missing fins
        for i in range(len(t)):
            t[i] = np.random.binomial(1, ((n1-nt-sum(t))/
            (N-n2-1)))

        #Sample the total number of fish on the second day, n2
        nhgprobs = nhypergeom(N,N-n1,nt+sum(t))
        nhgsupport = np.arange(len(nhgprobs))
        untagged = np.random.choice(nhgsupport,p=nhgprobs)+ 5
        - sum(t)
        n2 = untagged + nt + sum(t)

        #Sample the total number of fish in the pond
        Nsupport = np.arange(n1+n2-nt,2501,dtype=int)
        Nprobs = np.zeros(len(Nsupport))
        for i in range(len(Nsupport)):
            Nprobs[i] = ncr(Nsupport[i]-n2,Nsupport[i]-n1-
            n2+nt+sum(t))/ncr(Nsupport[i],Nsupport[i]-n1)
        N = np.random.choice(Nsupport, p=Nprobs/sum(Nprobs))
        if count % 5 == 0 :
            if n2 == 200:
                nsample[fill] = N
                fill += 1
                print('fill', fill)

    return nsample
```

9.2 Code used in Section 5

The functions included in this section are inv_chisq, convg, kellysim, obsdata, and kelly, with descriptions for each within the code itself.

```
def inv_chisq(n, df, wsq):
    #Sample of size n from an inverse chi squared
    #distribution with
    #df degrees of freedom and scale parameter wsq

    #Input:
    #n: Number of samples
    #df: Degrees of freedom
    #wsq: Scale parameter

    #Output:
    #nsample: n sized vector

    #First sample from a chi squared distribution with
    #df degrees of freedom
    temp = np.random.chisquare(df, n)

    #Calculate the scaled inverse
    sample = float(df)*float(wsq)/temp

    return sample

def convg(array,n,m):
    #Asseses parallel chains of the Gibbs sampler for convergence
    #using the Gelman−Rubin Diagnostic

    #Input:
    #array: n*m array containing m chains each of length n
    #n: Length of each chain
    #m: Number of chains

    groupmean = np.mean(array, axis=0)
    W = np.mean(np.var(array,axis=0))
    B = n*np.mean(np.var(groupmean,axis=0))

    varplus = (n−1)*W/n + B/n
    Rhat = np.sqrt(varplus/W)

    return Rhat

def kellysim(n, alpha, beta, var):
    #Draws n samples from the distribution given in the Kelly
    #paper for simulated data. Drawing samples from a Uniform and
    #Normal random variable also present but commented out

    #Inputs:
    #n: Number of data points
    #alpha: Intercept term
    #beta: Regression coefficient
    #var: Variance of error terms
    #Output:
    #xi: Simulated covariate terms
    #eta: Simulated response terms

    #Covariates
    support = np.linspace(-4.7,5.5,500)
    vals = np.exp(support)/(1+np.exp(2.75*support))
    vals /= sum(vals)
    xi = np.random.choice(support,size=n,p=vals)
    #xi = np.random.normal(0,1,n)
    #xi = np.random.uniform(-1,1,n)

    #Error terms
    epsilon = np.random.normal(0,np.sqrt(var),n)

    eta = alpha + beta*xi + epsilon

    return xi, eta

def obsdata(xi,eta,sigma, censor):
    #Simulates measured data given 'actual' data and measurement
    #error variance

    #Input:
    #xi: Vector of covariates
    #eta: Vector of responses
    #sigma: 2x2 covariance matrix for measurement errors
    #censor: 1 to censor all values below 0, and 0 to keep all
    #values

    #Output:
    #X: Vector of measured covariates
    #Y: Vector of measured responses

    err = np.random.multivariate_normal([0,0],sigma,len(xi))
    X = xi + err[:,0]
    Y = eta + err[:,1]

    if censor == 1:
        Y[Y < 0] = 0

    return X,Y

def kelly(k, sigmax, sigmay, sigmaxy, x, y, etai, censor):
```

```
#Kelly Gibbs sampler in Section 6.2

#Input:
#k: Number of Gaussian functions for the mixture model
#sigmax: n sized vector containing variance of each x
#sigmay: n sized vector containing variance of each y
#sigmaxy: n sized vector containing covariance between each
#x,y pair
#x: n sized vector containing x
#y: n sized vector containing y
#etai: n sized vector containing estimate of etai
#censor: Values below which are censored

#Output:
#Note that the following outputs were picked specifically
#for plotting purposes for the report. Samples of any
#other parameter can also be picked
#postalpha: Sample for alpha
#postbeta: Sample for beta
#postsigma: Sample for intrinsic variance sigma
#postcensoreta: Sample for eta of a single censored point
#postcensorxi: Sample for xi of a single censored point
#posteta: Sample for eta of a single uncensored point
#postxi: Sample for xi of a single uncensored point
#uncensoredlist: Indicies of uncensored points
#censorlist: Indicies of censored points
#measurexc: Observed x for selected censored point
#measurexu: Observed x for selected uncensored point
#measureyu: Observed y for selected uncensored point
#etai: Single sample for full eta
#xi: Single sample for full xi

n = len(sigmax)
#Autocorrelation for each x,y pair
rhoxy = np.divide(sigmaxy, np.sqrt(np.multiply(sigmax,
sigmay)))

#Initialise output
postalpha = np.zeros(200)
postbeta = np.zeros(200)
postsigma = np.zeros(200)
postcensoreta = np.zeros(200)
postcensorxi = np.zeros(200)
posteta = np.zeros(200)
```

```
postxi = np.zeros(200)

#Parameter values drawn from prior densities:
    #wsq: Scale hyperparameter of prior for tausq
    #mu0: Mean hyperparameter of means of Gaussian
    #functions
    #tausq: k sized vector containing variance of k
    #Gaussian functions
    #usq: Variance hyperparameter of means of
    #Gaussian functions
    #muk: k sized vector containing mean of k
    #Gaussian functions
    #pi: k sized vector containing the probability of
    #drawing data from the kth Gaussian function
    #sigma: Variance of regression
    #alpha: Regression intercept term
    #beta: Regression covariate
    #G: n*k sized matrix containing class membership of
    #the mixture model

#Draw initial values based on prior distributions
wsq = np.random.uniform(0,1)
mu0 = np.random.uniform(-1,1)
tausq = inv_chisq(k,n/2,wsq)
usq = inv_chisq(1,1,np.sqrt(wsq))
muk = np.random.normal(mu0,np.sqrt(usq),k)
pi = np.random.dirichlet(np.ones(k))
sigma = np.random.uniform(0,2)
alpha,beta = np.random.uniform(-1,1,2)
G = np.zeros(np.array([n,k]))
for i in range(n):
    G[i,:] = np.random.multinomial(1,pi)

#List of censored and uncensored data points
censorlist = []
uncensoredlist = []
for i in range(n):
   if y[i] == censor :
       censorlist.append(i)
   else:
       uncensoredlist.append(i)
count = 0
fill = 0
```

```
#Picks a single point from the censored and uncensored list
measurexc = np.copy(x[censorlist[0]])
measurexu = np.copy(x[uncensoredlist[0]])
measureyu = np.copy(y[uncensoredlist[0]])

while postalpha[-1] == 0:
    #Censoring
    for vals in censorlist:
        #Simulate new value for censored y
        y[vals] =
        np.random.normal(etai[vals],np.sqrt(sigmay[vals]))
        #Only accept if value is less than threshold
        while y[vals] >= censor:
            y[vals] =
            np.random.normal(etai[vals],np.sqrt(sigmay[vals]))
        if count == 0:
            print(vals)

    #Equation 55
    col = np.reciprocal(np.multiply(sigmax,(1-
    np.square(rhoxy))))+(beta**2/sigma)
    sigmaxik = np.tile(col,(k,1))
    sigmaxik = sigmaxik.transpose()
    sigmaxik = np.reciprocal(np.tile(1/tausq,(n,1)) +
    sigmaxik)
    print('Equation 58')
    print('sigmaxik',sigmaxik)

    #Equation 57
    sigmaxi = np.multiply(G,sigmaxik)
    sigmaxi = sigmaxi.sum(axis=1)
    print('Equation 57')
    print('sigmaxi',sigmaxi)

    #Equation 56
    xixyhat = x + (sigmaxy/sigmay)*(etai-y)
    print('Equation 56')
    print('xixyhat',xixyhat)

    #Equation 55
    xiikhat = np.divide(xixyhat,np.multiply(sigmax,
    (1-np.square(rhoxy))))
    xiikhat = xiikhat + ((1/beta)*(etai-alpha)/sigma)
    xiikhat = np.tile(xiikhat, (k,1))
    xiikhat = xiikhat.transpose()
    temp = np.tile(muk/tausq, (n,1))
    temp2 = np.tile(sigmaxi, (k,1)).transpose()
    xiikhat = np.multiply((xiikhat + temp),temp2)
    print('Equation 55')
    print('xiikhat',xiikhat)

    #Equation 54
    xihat = np.multiply(G,xiikhat)
    xihat = xihat.sum(axis=1)
    print('Equation 54')
    print('xihat',xihat)

    #Equation 53
    xi = np.random.normal(xihat, np.sqrt(sigmaxi))
    print('Equation 53')
    print('xi',xi)

    #Equation 68
    first = 1/(sigmay*(1-np.square(rhoxy)))
    second = 1/sigma
    sigmaetai = 1/(first+second)
    print('sigmaetai',sigmaetai)

    #Equation 67
    tempnum = y + sigmaxy*(xi-x)/sigmax
    tempden = sigmay*(1-np.square(rhoxy))
    etaihat = sigmaetai*(tempnum/tempden + (alpha +
    beta*xi)/sigma)
    print('Equation 67')
    print('etaihat',etaihat)

    #Equation 66
    etai = np.random.normal(etaihat, np.sqrt(sigmaetai))
    print('Equation 66')
    print('etai',etai)

    #Equation 73 and 74
    for i in range(n):
        qk = scipy.stats.norm.pdf(xi[i],muk,np.sqrt(tausq))
        qk = qk*pi
        qk /= sum(qk)
        G[i,:] = np.random.multinomial(1,qk)
    print('Equation 74')
```

```
#Equation 76
X = np.transpose(np.vstack((np.ones(len(xi)),xi)))
XtXinv = np.linalg.inv(np.matmul(np.transpose(X),X))
Xteta = np.matmul(np.transpose(X),etai)
chat = np.matmul(XtXinv, Xteta)
print('Equation 76')
print('chat',chat)

#Equation 77
sigmachat = XtXinv*sigma
print('Equation 77')
print('sigmachat',sigmachat)

#Equation 75
alpha, beta =
np.random.multivariate_normal(chat,sigmachat)
print('Equation 75')
print('alpha',alpha)
print('beta',beta)

#Equation 80
ssq = sum(np.square(etai-alpha-beta*xi))/(n-2)
print('Equation 80')
print('ssq',ssq)
print('ssum',sum(np.square(etai-alpha-beta*xi)))

#Equation 78
sigma = (inv_chisq(1,n-2,ssq))
print('Equation 78')
print('sigma',sigma)

#Equation 82
nk = G.sum(axis=0)
print('Equation 82')
print('nk',nk)

#Equation 81
pi = np.random.dirichlet(nk+1)
print('Equation 81')
print('pi',pi)

#Equation 86
sigmamuhatk = 1/(1/usq + nk/tausq)
```

```
    print('Equation 86')
    print('sigmamuhatk',sigmamuhatk)

    #Equation 85
    xikbar = np.zeros(k)
    temp = np.matmul(np.transpose(G),xi)
    for i in range(k):
        #Prevent dividing by 0
        if nk[i] == 0:
            xikbar[i] = 0
        else:
            xikbar[i] = temp[i]/nk[i]
    print('Equation 85')
    print('xikbar',xikbar)

    #Equation 84
    right = mu0/usq + nk*xikbar/tausq
    mukhat = sigmamuhatk*right
    print('Equation 84')
    print('mukhat',mukhat)

    #Equation 83
    muk = np.random.normal(mukhat,np.sqrt(sigmamuhatk))
    print('Equation 83')
    print('muk',muk)

    #Equation 89
    tksq = np.zeros(k)
    for j in range(k):
        temp1 = np.square(xi-muk[j])
        temp2 = np.matmul(np.transpose(G[:,j]),temp1)
        tksq[j] = (temp2+wsq)/(nk[j]+1)
    print('Equation 89')
    print('tksq',tksq)

    #Equation 87
    for i in range(k):
        tausq[i] = inv_chisq(1,nk[i]+1.0,tksq[i])
    print('Equation 87')
    print('tausq',tausq)

    #Equation 94
    mubar = np.mean(muk)
    print('Equation 94')
```

```
        print('mubar',mubar)

        #Equation 93
        mu0 = np.random.normal(mubar, np.sqrt(usq/k))
        print('Equation 93')
        print('mu0',mu0)

        #Equation 97
        usqhat = (sum(np.square(muk-mu0))+wsq)/(k+1)
        print('Equation 97')
        print('usqhat',usqhat)

        #Equation 95
        usq = inv_chisq(1,k+1,usqhat)
        print('Equation 95')
        print('usq',usq)

        #Equation 103,101
        b = (1/usq+sum(1/tausq))/2
        wsq = np.random.gamma((k+3)/2,1/b)
        print('Equation 103')
        print('wsq',wsq)

        #Burn first 1000 samples, and thin by recording every
        #5th sample after
        if count >= 1000 and count % 5 == 0:
            postalpha[fill] = alpha
            postbeta[fill] = beta
            postsigma[fill] = sigma
            postcensoreta[fill] = etai[censorlist[0]]
            postcensorxi[fill] = xi[censorlist[0]]
            posteta[fill] = etai[uncensoredlist[0]]
            postxi[fill] = xi[uncensoredlist[0]]
            fill += 1
        print('count',count)
        count += 1

    return (postalpha, postbeta, postsigma, postcensoreta,
    postcensorxi, posteta, postxi,uncensoredlist, censorlist,
    measurexc, measurexu, measureyu,etai,xi)
```