BAYESIAN MODELLING AND SOFTWARE DEVELOPMENT FOR THE POST PROJECT

by

Saman Muthukumarana

B.Sc. Special Degree in Statistics, University of Sri Jayewardenepura, Sri Lanka, 2002

a Project submitted in partial fulfillment

of the requirements for the degree of

Master of Science

in the Department

of

Statistics and Actuarial Science

SIMON FRASER UNIVERSITY

Spring 2007

APPROVAL

Name: Saman Muthukumarana

Degree: Master of Science

Title of Project: Bayesian Modelling and Software Development for the POST Project

Examining Committee: Dr. Richard Lockhart
Chair

Dr. Tim Swartz
Senior Supervisor
Simon Fraser University

Dr. Carl Schwarz
Supervisor
Simon Fraser University

Dr. Steven Thompson
External Examiner
Simon Fraser University

Date Approved: _____

Abstract

The Pacific Ocean Shelf Tracking (POST) project is part of the Census of Marine Life Study. In this project, acoustic transmitters are surgically implanted into salmon and the salmon are tracked during their migration over a series of listing lines placed along the ocean floor. At the moment, researchers observe the simple descriptive statistics at different locations based on the actual number of radio detections. However, these methods are not sufficient to study their movement patterns and we need to employ advanced mark-recapture models for better understanding of the movement patterns. Estimating between locations survival probabilities of animals is a key component in mark-recapture studies. Detection probabilities at listening lines are nuisance parameters. They are high, but not 100% and also need to be estimated.

In our project, we develop a Bayesian model for estimating detection probabilities and survival probabilities that is well suited for the POST project. Previous mark-recapture models do not make any adjustments in survival probabilities between listening lines for travel times of fish whereas our model treats survival probabilities as a function of travel times. This plays a key role when distances between listening lines vary greatly. The model is implemented via Markov chain Monto Carlo using WinBUGS. Simulation results indicate that the model is well behaved in estimating parameters. We also submit our model to the POST project for their consideration in future studies.

Keywords: mark-recapture, Bayesian analysis, Markov chain Monto Carlo, salmon, latent variables, simulation, WinBUGS

Acknowledgements

I would like to begin by thanking the faculty in the Department of Statistics and Actuarial Science for their great commitment to their students. Their kind and deep attention in teaching always motivated me to finish my studies in a high standard.

I'm deeply indebted to my senior supervisor Dr. Tim Swartz for mentoring me in limitless ways. His support on academic, professional and personal issues has been priceless. He was always ready to listen to my ideas, to correct my chapters and asked me challenging theoretical and computational questions to get me on to the right track. He taught me how to write scientific papers and how to be persistent in accomplishing any goal throughout the project. It is not possible to thank him enough for understanding my research interest and nurturing it. I was lucky to continue my higher studies under his supervision.

My big thanks go to Dr. Carl Schwarz who is responsible for this project. He always directed me to the right path from the very beginning of the project. Thank you for taking me into the mark-recapture world and introducing me to Dr. Ken Newman who gave valuable comments on the WinBUGS code. I also thank my examining committee member Dr. Steve Thomson for commenting on the project.

My special thanks go to Dr. Richard Lockhart for understanding my academic background and giving me an opportunity to pursue my studies at SFU. I can't forget his prompt actions on my visa issues.

I'm grateful for the financial support provided by the Department of Statistics and Actuarial Science. Sincere gratitude to Sadika, Kelly and Charlene for your kindness and help.

I also thank Dr. Laura Cowen for her willingness to share the Columbia River data

with me. I would also like to thank all of the graduate students who I have come to know throughout my time at SFU, especially Crystal, Pritam and Matt for helping me to settle on my arrival.

To finish, but not least, I appreciate Aruni for her patient support and sacrifices. This would not have been finished without her understanding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background of the Project

The Pacific Ocean Shelf Tracking (POST) project (www.postcoml.org) is one of thirteen field programs contributing to the Census of Marine Life. The Census of Marine Life is an international collaboration of scientists that seek to assess and explain the diversity, distribution and abundance of marine life in the oceans. The POST project plans to build acoustic tracking arrays along the west coast of North America to study the migration patterns, life spans, movements and behaviours of Pacific Ocean aquatic animals such as salmon or other fish species. The array will have 2000 receivers and 30 listening lines, each up to 50 km long. They are capable of recording up to 250,000 animals at once. The POST project expects to complete this task by 2010.

Firstly, acoustic transmiters are surgically implanted into animals and then the animals are released at a release point. Fixed listening lines placed on the ocean floor pick up the signals when tagged animals pass over it. Receivers store the unique ID number of the tag, detection date and time in a database that can be queried by researchers. Tag implantation protocols and listening lines technology have been specially developed for the POST project. A pilot program in 2004 has demonstrated the feasibility of the technology. It has also revealed that the listening lines have a detection efficiency of 91%. They further investigate the methods for extending battery life, identifying suitable areas for listening lines and tag implantation technology. In 2006, acoustic tags with 10-20 years lifespans

were developed to study almost the entire ocean life of animals.

The idea for this project came during my Mitacs internship involving the analysis of mark-recapture data. The POST project is a combination of two types of mark-recapture experiments. In the first type of mark-recapture experiment (Lebreton et al. 1992), animals are initially marked, and then recaptured at yearly intervals (for example). Not all marked and living animals are captured at future time points. This corresponds to salmon in the POST project that do not pass sufficiently close to the receiver, and hence are not detected. In this type of experiment, there is interest in the temporal dimension of survival. For example, one may be interested in the survival rates of species from year to year.

In the second type of mark-recapture experiment, marked fish are released, and are detected as they swim past landmarks (Burnham et al. 1987). In this type of experiment, there is interest in the spatial dimension of survival. For example, one may be interested in the survival rates of species between particular dams.

This project considers methods to combine both the temporal and the spatial dimensions of the problem into a single mark-recapture model.

1.2 History of Mark-Recapture Models

Mark-recapture models are popular in estimating animal population sizes, birth rates, survival rates and migration rates. Basically, mark-recapture models can be broken down into two categories as open and closed populations models.

The Peterson estimator (Peterson, 1896) is the simplest estimator which is based on two sample periods, one involving the initial marking of n_1 individuals and then m_2 are recaptured amongst the n_2 individuals caught on the second occasion. The Peterson estimator is used to estimate the population total N. The marked fraction in the population is estimated by the marked fraction in the second sample. That is,

$$m_2/n_2 = n_1/\hat{N} \Rightarrow \hat{N} = n_1 n_2/m_2$$

Here, the second sample must be a random sample for the method to be valid. That is, marked and unmarked individuals must have the same chance of being captured in the

second sample.

Schnabel (1938) extended the Peterson estimator to a series of samples. Individuals caught at each sample are examined for markers and are then released. Here, only the same type of mark is used for all animals since we need to distinguish only marked and unmarked animals. The basic assumptions for these two models are

- marks/tags are not lost
- the population is closed (i.e. the population size N is constant)
- capture-recapture probabilities are constant at each sampling location

It is often the case that these assumptions are unrealistic, so further developments are needed.

Cormack (1964), Jolly (1965) and Seber (1965) introduced a multiple sample capture-recapture models for open populations. Open population models do not assume that the population is constant over the study period. The CJS (Cormack/Jolly/Seber) model allows to estimate survival and recapture probabilities for single group of individuals conditioning on first capture. The JS (Jolly/Seber) model extends the CJS to estimate the population size and new birth and immigrations at each sampling locations. The JS model is fairly general by not conditioning on first capture. The following assumptions must be satisfied for the CJS (Cormack/Jolly/Seber) model to be valid.

- every marked or unmarked individual present in the population in each sample period has the same probability being captured.
- every marked individual in the population immediately after each sample period has the same probability of survival until the next sampling period
- marks are not lost
- all samples are instantaneous and each release is made immediately after the sample period
- all emigrations from the population are permanent

- the survival and capture of every individual is independent of the survival and capture of all other individuals

Recently, various models have been developed by researchers by considering violations of some of the above assumptions. Shirley, Pollock and Norris (2003) proposed a flexible framework to relax assumption 1. They relaxed the homogeneity in survival and capture probabilities using the finite mixtures to model the heterogeneity. Cowen and Schwarz (2005) considered the violation of assumption 3 due to tag loss. Bonner and Schwarz (2006) extended the CJS model for continuous covariate which is assumed to have a Weiner process. They treated survival and capture probabilities as a function of covariate using the logistic link function. Complete details on the use of covariates via link functions was discussed by Lebreton et al. (1992). They extended the CJS model to allow for multiple groups and various covariates using appropriate link functions.

1.3 Organization of the Project

In chapter 2, we provide the details of the Bayesian model development and implementation. By treating the latent variables as though they were known, the complete data likelihood is derived where survival probabilities depend on travel times. Appropriate prior distributions are then selected for the model parameters. As the posterior distribution is complex and high-dimensional, we obtain posterior summary statistics which describe key features in the study. In particular, posterior expectations are approximated through Markov chain Monte Carlo (MCMC) methods using WinBUGS software (Spiegelhalter, Thomas and Best, 2003). We then provide details of implementation of the model via WinBUGS. In chapter 3, we apply our model to real data obtained from the Columbia River system and the POST project. The reliability of the model is demonstrated using simulated data. We also provide sensitivity analyses with respect to some of the model assumptions. We conclude with a discussion in chapter 4.

Chapter 2

Bayesian Model Development and Implementation

2.1 Why Bayesian Modelling?

The Bayesian framework was developed by the Reverand Thomas Bayes (1702-1761) and the Bayesian approach to obtain population estimates was first used by LaPlace in 1786. At the moment, Bayesian statistics is widely used by researchers in widespread fields due to significant computational advancements including MCMC, BUGS and WinBUGS software. Recently, researchers in the capture-recapture area have also taken a Bayesian approach instead of classical likelihood approach. Edward and Christian (1992) showed how Gibbs sampling can be applied in mark-recapture experiments. Dupuis (1995) discussed multiple recapture analysis with missing data via Gibbs sampling. Schwarz and Seber (1999) discussed the importance of Bayesian methods in mark-recapture models. Brooks, Catchpole and Morgan (2000) provided a Bayesian treatment for the CJS model. Bonner and Schwarz (2006) also used a Bayesian approach based on Metropolis-Hasting algorithm to estimate model parameters.

The Bayesian approach has many attractive features over the standard likelihood approach. In the Bayesian approach, models can often be as complex as reality demands and missing data and latent variables can handle in a flexible way. It also provides a way to include expert prior knowledge concerning the parameters of interest. Another method

to handle the missing data is the Expectation-Maximization (EM) algorithm. Van Deusen (2002) used the EM algorithm to maximize a complete data likelihood but assumed survival probabilities independent of travel times. Cowen and Schwarz (2005) also assumed survival probabilities independent of travel times but take a different approach by working with the observed likelihood. The observed likelihood is somewhat more complex than the complete data likelihood as it involves integrals with respect to the unobserved (i.e. latent) variables. All recent models do not make adjustments in survival probabilities between listening lines for the time of travel.

It is reasonable to assume that a fish that takes a longer time to swim between the lines may have a lower overall survival rate than a fish that takes only a short time to swim between listening lines. We treat survival probabilities as a function of travel time. When the distances between listening lines vary greatly, this dependance structure is clearly important. We are also interested in acoustic detection probabilities and allow them to vary over listening lines. Our model is a combination of both the temporal and the spatial dimensions of the problem into a single mark-recapture model. A complete data likelihood is constructed by treating latent variables as though they are observable. We then make inferences about model parameters based on the posterior distribution which is derived from the prior distributions of parameters and the complete data likelihood.

2.2 Notation

A summary of our mark-recapture experiment is as follows. Acoustic tags are surgically implanted into animals and then the animals are released at an initial release point. It is also possible to release them at a listening line following the initial release point. Listening lines placed on the ocean floor pick up the signals when tagged animals pass over it. Receivers store the unique ID number of the tag, detection date and time in a database. For convenience, we summarize the following symbols which are used in our model development.

- m the number of listening lines following the release point

- $\omega = \{\omega_{ij}\}$ the detection history vector of all fish such that

$$\omega_{ij} = \begin{cases} 1 & \text{if the } i\text{-th fish is detected at } j\text{-th listening line} \\ 0 & \text{if the } i\text{-th fish is not detected at } j\text{-th listening line} \end{cases}$$

  Note that $\omega_{i0} = 1$ for all fish.

- T_{ij} the time required for the i-th fish to travel from the point of release to j-th listening line

- T^{obs} the observed cumulative travel times vector

- T^{mis} the missing or latent cumulative travel times vector

- T the complete cumulative travel time vector. Note that $T = (T^{obs}, T^{mis})$.

- S_{ij} the survival status of the i-th fish at j-th listening line such that

$$S_{ij} = \begin{cases} 1 & \text{if the } i\text{-th fish is alive at } j\text{-th listening line} \\ 0 & \text{if the } i\text{-th fish is not alive at } j\text{-th listening line} \end{cases}$$

  Note that $S_{i0} = 1$ for all fish.

- S^{obs} the observed survival states vector

- S^{mis} the missing or latent survival states vector

- S the complete survival states vector. Note that $S = (S^{obs}, S^{mis})$.

- t_{ij} the interval travel time for the i-th fish from listening line $j-1$ to j

- $t = \{t_{ij}\}$ the complete interval travel time vector

- p_j the probability of detection at the j-th listening line

- q_j the daily survival probability when travelling between listening lines $j-1$ and j

- ϕ_{ij} the survival probability of the i-th fish when travelling from listening line $j-1$ to listening line j given that the fish was alive at listening line $j-1$

  Note that $\phi_{ij} = q_j^{t_{ij}}$.

For example, consider a situation with 5 listening lines. The i-th animal may have $(1, 0, 0, 1, 1, 1)$ as the capture history. Note that this animal was not detected at listening lines 1 and 2. The probability of observing this history is $\phi_{i1}(1 - p_1)\phi_{i2}(1 - p_2)\phi_{i3}p_3\phi_{i4}p_4\phi_{i5}p_5$. A capture history of $(0, 0, 1, 1, 1, 0)$ implies that the i-th animal is first released at the second listening line. The probability of observing this capture history is $\phi_{i3}p_3\phi_{i4}p_4\left[(1 - \phi_{i5}) + \phi_{i5}\,(1 - p_5)\right]$. The last term in brackets is the probability that the animal died before listening line 5 plus the probability that the animal is alive but not captured at listening line 5. It is clear that these probabilities become complicated when the animal is unobserved and when the number of listening lines increases.

2.3 Development of the Complete Data Likelihood

Consider a population of n fish where each fish is implanted with an acoustic transmitter. Without loss of generality, assume that all fish are released at location $j = 0$, and that listening lines are set up at fixed locations $j = 1, \ldots, m$. It is also reasonable to assume that cumulative travel times from the point of release to the listening lines are available since receivers store the unique ID number of the tag, the detection date and time. If the listening line does not detect when a tagged fish passes over it, then the cumulative travel time from the point of release to that particular listening line is unknown (latent). Note that when a fish has died (and is therefore not detected), we still imagine that there is a cumulative travel time associated with the fish. The value is missing but it represents the cumulative travel time that the fish would have taken had it been alive. When a fish is not detected, then there is no observed cumulative travel time. We refer to the vector of missing or latent cumulative travel times as T^{mis} and the complete cumulative travel times vector $T = (T^{obs}, T^{mis})$.

The quantities $S^{obs} = \{S_{ij}\}$ and $t = \{t_{ij}\}$ are associated with ω and T as follows.

$$S_{ik} = 1 \quad \text{for } k = 1, \ldots, j \quad \text{if } \omega_{ij} = 1$$
$$t_{ij} = T_{ij} - T_{i,j-1}$$

Note that whereas the entire vector ω is observed, some of the entries S_{ij} are latent. This is due to the fact that an undetected fish may be either alive or dead. As an example, consider

the observed data $(\omega_{i0}, \ldots, \omega_{i5}) = (1, 0, 0, 1, 0, 0)$. In this case, $(S_{i0}, \ldots, S_{i3}) = (1, 1, 1, 1)$ but S_{i4} and S_{i5} are latent. We supplement the observed S^{obs} with the missing or latent S^{mis} to give the complete survival history $S = (S^{obs}, S^{mis})$. The variable t_{ij} may be missing because some of the T_{ij} may be missing. In fact, there are at least as many missing t_{ij}'s as there are missing T_{ij}'s. As an example, consider $(T_{i0}, T_{i1}, T_{i2}, T_{i3}, T_{i4}, T_{i5}) = (0, x, \text{NA}, \text{NA}, y, z)$ where NA denotes "Not Available". Then $(t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}, t_{i5}) = (0, x, \text{NA}, \text{NA}, \text{NA}, z-y)$. Therefore, the vector t consists of both observed and latent data.

We now describe the two primary parameters in the model. As the acoustic transmitters are identical and the fish comprise a sample from an underlying population, one typically assumes that the probability p_j, the probability of detection at the j-th listening line does not depend on fish i. In some instances, it may be reasonable to assume a common probability of detection (i.e. $p_j = p$ for all locations) although the general case causes no additional difficulty. The second parameter of interest, ϕ_{ij} concerns survival of the i-th fish. In Cowen and Schwarz (2005), the modelling assumption $\phi_{ij} = \phi_j$ implies that survival probabilities are independent of travel times. In our model, we consider $\phi_{ij} = f(t_{ij})$ where f is a specified decreasing function. Using this parametrization, the longer that it takes a fish to travel between listening lines $j-1$ and j, the greater the chance that the fish does not survive. In our model, travel times are measured in days, and we define $\phi_{ij} = q_j^{t_{ij}}$ such that q_j denotes the daily survival probability when travelling between listening lines $j-1$ and j. Our modelling assumption implies that survival is independent across days. Therefore, the proposed framework reduces the primary parameters of interest to (p, q) where $p = \{p_j\}$ and $q = \{q_j\}$.

In Cowen and Schwarz (2005), an observed likelihood is obtained based on the observed data (ω, T^{obs}). The observed likelihood is complex as it involves integrals with respect to the latent cumulative travel times T^{mis}. We take an approach based on the complete data likelihood as in van Deusen (2002). The complete data likelihood treats latent variables as though they are available, and is especially well suited to a Bayesian analysis. An advantage of the complete data likelihood over the observed likelihood is that it has a much simpler form. In our approach, we develop the complete likelihood based on (ω, S, t).

In obtaining the complete data likelihood, we follow the development in the companion

paper by Muthukumarana, Schwarz and Swartz (2007). Let $[A \mid B]$ generically denote the density function or probability mass function corresponding to A given B. In addition, let $\omega_i = (\omega_{i0}, \ldots, \omega_{im})$, $S_i = (S_{i0}, \ldots, S_{im})$, and $t_i = (t_{i0}, \ldots, t_{im})$. Then the complete data likelihood is given by

$$\begin{aligned} [\omega, S, t] &= \prod_{i=1}^{n} [\omega_i, S_i, t_i] \\ &\quad \prod_{i=1}^{n} [\omega_i \mid S_i, t_i]\, [S_i, t_i] \\ &\quad \prod_{i=1}^{n} [\omega_i \mid S_i, t_i]\, [S_i \mid t_i]\, [t_i] \end{aligned} \tag{2.1}$$

where the independence of fish is assumed and the expressions in (2.1) are based on conditional probability. Then,

$$\begin{aligned} [\omega_i \mid S_i, t_i] &= [\omega_i \mid S_i] \\ &= \prod_{j=1}^{m} [\omega_{ij} \mid S_{ij}] \\ &= \prod_{j=1}^{m} (p_j^{\omega_{ij}}(1-p_j)^{1-\omega_{ij}})^{S_{ij}} \end{aligned} \tag{2.2}$$

where the key assumption in (2.2) is that detection at location j does not depend on other locations, and we note that when a fish dies (i.e. $S_{ij} = 0$), then detection is impossible and there is no contribution to the complete data likelihood. Now,

$$\begin{aligned} [S_i \mid t_i] &= [S_{im} \mid S_{i0}, \ldots, S_{i,m-1}, t_i]\, [S_{i,m-1} \mid S_{i0}, \ldots, S_{i,m-2}, t_i]\, \cdots [S_{i1} \mid S_{i0}, t_i] \\ &= [S_{im} \mid S_{i,m-1}, t_i]\, [S_{i,m-1} \mid S_{i,m-2}, t_i]\, \cdots [S_{i1} \mid S_{i0}, t_i] \\ &= \prod_{j=1}^{m} [S_{ij} \mid S_{i,j-1}, t_{ij}] \\ &= \prod_{j=1}^{m} (\phi_{ij}^{S_{ij}}(1-\phi_{ij})^{1-S_{ij}})^{S_{i,j-1}} \\ &= \prod_{j=1}^{m} (q_j^{t_{ij}S_{ij}}(1-q_j^{t_{ij}})^{1-S_{ij}})^{S_{i,j-1}} \end{aligned} \tag{2.3}$$

where there is no survival contribution to the likelihood when a fish has already died (i.e. $S_{i,j-1} = 0$). Putting (2.1), (2.2) and (2.3) together, we have the complete data likelihood

$$[\omega, S, t] = \prod_{i=1}^{n} [t_i] \prod_{j=1}^{m} (p_j^{\omega_{ij}}(1-p_j)^{1-\omega_{ij}})^{S_{ij}} (q_j^{t_{ij}S_{ij}}(1-q_j^{t_{ij}})^{1-S_{ij}})^{S_{i,j-1}}. \tag{2.4}$$

The last step in the determination of the complete data likelihood (2.4) is the specification of $[t_i]$. As the fish arise from the same population and travel times are non-negative, it may be reasonable to consider a multivariate lognormal distribution. The convenient covariance structure in the multivariate normal distribution is appealing as one might imagine

that a fish that is fast (slow) in travelling between two locations may be fast (slow) in travelling between other locations. Specifically, we assume

$$(\log(t_{i1}), \ldots, \log(t_{im}))' \sim \text{Normal}_m(\mu, \Sigma). \tag{2.5}$$

A simpler (but perhaps less realistic) alternative to (2.5) is $\log(t_{ij}) \sim \text{Normal}(\mu, \sigma^2)$ with independence over i and j.

2.4 Bayesian Model Ingredients

In a classical approach, the sample data are taken as random while parameters are taken as fixed. In a Bayesian approach, parameters themselves follow a probability distribution. Furthermore, parameters may be model parameters, missing data or events that are not observed (latent). The following components are required in order to carry out a Bayesian analysis.

- the prior distribution
- the likelihood of the data

A prior distribution must be specified for the parameter vector in the model. It quantifies the uncertainty about the parameters before the data are observed. It is important that priors should be selected such that they represent the best knowledge that we have about parameters before looking at the data. If it is not possible, we can still use non-informative priors which often produce useful results provided that there is sufficient information in the likelihood. Referring to (2.4) and (2.5), we consider the prior density

$$[p, q, \mu, \Sigma] \quad = \quad [p]\ [q]\ [\mu]\ [\Sigma] \tag{2.6}$$

where prior independence is assumed. As the p's and q's are probabilities defined on the simplex, we assign Beta priors for them. Specifically, we assume independent p_j where

$$[p_j] \propto p_j^{a_p - 1}(1 - p_j)^{b_p - 1}\ ;\ 0 \leq p_j \leq 1$$

and independent q_j where

$$[q_j] \propto q_j^{a_q - 1}(1 - q_j)^{b_q - 1}\ ;\ 0 \leq q_j \leq 1.$$

The a's and the b's are pre-specified based on one's subjective understanding of the listening devices and the daily survival rates. As is customary, we impose a diffuse improper prior,

$$[\mu] \propto 1$$

for the mean travel time and

$$\Sigma^{-1} \sim \text{Wishart}((1/m)I, m) \text{ (i.e. } [\Sigma] \propto \exp\{-(m/2)\text{tr}\Sigma\})$$

for the inverse variance covariance matrix.

The second ingredient, the likelihood of the data relates the parameter vector to a probability model which is the complete data likelihood in our development. The complete data likelihood ((2.4) and (2.5)) and the prior (2.6) provide the ingredients for the Bayesian analysis. We next discuss the computation of the posterior distribution of parameters which is obtained from the prior and the complete data likelihood through Bayes theorem.

2.5 Computations

Recall that Bayes formula gives us the posterior distribution

$$\pi(\theta \mid y) = \tfrac{f(y|\theta)\pi(\theta)}{f(y)} \ .$$

In our scenario, θ is the vector of parameters of interest and y is the vector of observed data. Both θ and y are considered to be random. The function $f(y \mid \theta)$ is the likelihood of the data given θ, $\pi(\theta)$ is the prior density and $f(y)$ is the marginal density of the observed data. This implies that

$$\pi(\theta \mid y) \quad \propto \quad f(y \mid \theta)\ \pi(\theta).$$

So, if the prior and likelihood are known, we can obtain the posterior, and the inverse normalizing constant can be calculated as

$$f(y) = \int f(y \mid \theta)\pi(\theta)d\theta.$$

In order to perform inferences about components of θ, we need to calculate the marginal posterior density of individual elements of θ. This requires integration of the posterior with respect to other parameters. As an example, the posterior mean of θ_1 is given by

$$E(\theta_1 \mid y) = \int \theta_1 \pi(\theta_1 \mid y) d\theta_1 = \int \theta_1 \left[\int \pi(\theta \mid y) d\theta_2 \, d\theta_3 \, d\theta_4 \, d\theta_5 \right] d\theta_1.$$

In simple models, the integration problems can be avoided by choosing particular types of priors. If the prior and likelihood are natural conjugate distributions, then the posterior is in the same family and the above integrations may have an easy analytical solution. For more complex models, the integrations are often difficult and even impossible. Sometimes numerical approaches such as quadrature and Laplace methods can be used to approximate the expectations. Evans and Swartz (1995) give a complete discussion of the major techniques available for the approximations of integrals in statistics. They discuss the applicability, merits and demerits of these methods.

For our problem, we re-express the complete data likelihood $[\omega, S, t]$ appearing in (2.4) as $[X^{obs}, X^{mis} \mid p, q, \mu, \Sigma]$ to emphasize the dependency on the unknown parameters and to emphasize that (ω, S, t) consists of both observed and missing values. We then obtain the following expression for the posterior

$$\begin{aligned} [p, q, \mu, \Sigma \mid X^{obs}] &\propto [X^{obs} \mid p, q, \mu, \Sigma] \; [p, q, \mu, \Sigma] \\ &= \int [X^{obs}, X^{mis} \mid p, q, \mu, \Sigma] \; [p, q, \mu, \Sigma] \; dX^{mis}. \end{aligned} \tag{2.7}$$

In theory, the functional form of the posterior density (2.7) provides a complete description of the uncertainty in the parameters defined in the mark-recapture experiment. However, the dimensionality and the complexity of (2.7) is such that it is impossible to gain any meaningful insight. Alternatively, we consider the following expression

$$\begin{aligned} [p, q, \mu, \Sigma, X^{mis} \mid X^{obs}] &\propto [p, q, \mu, \Sigma, X^{obs}, X^{mis}] \\ &\propto [X^{obs}, X^{mis} \mid p, q, \mu, \Sigma] \; [p, q, \mu, \Sigma] \end{aligned} \tag{2.8}$$

where the last expression in (2.8) is the product of the complete data likelihood and the prior density which are defined in familiar forms in (2.4) and (2.6).

In this model, there are $\left(3m + \frac{m}{2}[m+1]\right)$ primary parameters in addition to the missing data. Recall that m is the number of listening lines. As an example, if there are 10 listening lines, then there are 85 primary parameters which gives rise to a high-dimensional posterior. It is now clear that our posterior is complex and can not be integrated analytically. We instead consider a simulation approach, whereby if we are able to sample variates

$(p, q, \mu, \Sigma, X^{mis})$ from (2.8), then we can use the sampled components (p, q, μ, Σ) as realizations from the posterior distribution. However, sampling directly from (2.8) is also difficult and there are some alternative sampling strategies which may be useful in sampling from complex models. The most widely used sampling methods are

- importance sampling
- Markov chain Monte Carlo (MCMC)

In Evans and Swartz (1995), these two methods are discussed where Markov chain methods are recommended for high-dimensional problems such as our situation. In importance sampling, samples are drawn from a tractable density that is similar to the posterior distribution. In MCMC, variates are drawn from a distribution which has the posterior distribution as its equilibrium distribution. In both strategies, the output may be averaged to obtain approximations to posterior expectations. A Markov chain is a random process where the variate at iteration i depends only on the variate at iteration $i-1$. Various algorithms have been developed to implement MCMC. The most popular algorithms are

- Metropolis-Hastings
- Gibbs sampling

The Metropolis-Hasting algorithm proceeds by using a proposal density, $p(\theta, \theta^k)$ to generate the next value θ^* where θ^k is the value generated at k-th iteration. This generated value θ^* is accepted with probability

$$min\left(1, \frac{p(\theta^k, \theta^*)\pi(\theta^* \mid y)}{p(\theta^*, \theta^k)\pi(\theta^k \mid y)}\right). \tag{2.9}$$

If θ^* is not accepted as the next value, then it is set to θ^k. The rate at which the new values are accepted is called the acceptance rate. The process is repeated to obtain the sequence $\theta^1, \theta^2, \ldots$ where θ^k is approximately a realization from the posterior for sufficiently large k. The Metropolis-Hastings algorithm requires an initial value θ^0 in order to start the simulation. The choice of initial value may effect the rate of convergence of the algorithm. Initial values which are far away from the range covered by the posterior distribution often lead to chains that take more iterations to attain convergence.

The Gibbs sampling algorithm is a special case of the Metropolis-Hastings algorithm in which samples are drawn by turning the multivariate problem into a sequence of lower-dimensional problems. In Gibbs sampling, the value θ^* is obtained by generating from distributions with a 100% acceptance rate.

Fortunately, the software package WinBUGS implements MCMC without programming any of the Metropolis-Hasting or Gibbs algorithm. The default option in WinBUGS for well behaved models with log concave densities is the Gibbs sampling algorithm. However, Metropolis-Hasting is invoked for nonstandard models. In WinBUGS, we need only to specify the complete data likelihood, the priors, the observed data and the initial values. WinBUGS then produces an appropriate Markov chain.

However, we need to make sure that the sequence has converged before inferences are obtained on the simulated sequence. The number of iterations taken for the practical convergence to the stationary distribution depends on various factors including

- the complexity of the model (models with few parameters generally converge faster)
- whether the prior and likelihood are conjugate
- the closeness of the initial value to the posterior mean
- the parameterization of the problem
- the sampling scheme adopted

The number of iterations prior to convergence is called the burn-in, and we discard these variates for the purpose of inference. WinBUGS provides several statistics and graphical tools to check the convergence of Markov chains. Brooks and Gelman (1997) discussed these alternative methods monitoring convergence.

We are now in a position to fit the model. In the next section, we discuss the model implementation and Bayesian inference via WinBUGS.

2.6 Markov Chain Monte Carlo (MCMC) Simulation via WinBUGS

2.6.1 Introduction to WinBUGS

WinBUGS is a product of the BUGS (Bayesian inference Using Gibbs Sampling) project which is a joint program of the Medical Research Council of Biostatistics Unit at Cambridge University and the Department of Epidemiology and Public Health of Imperial College at St.Mary's Hospital in London. The software is freely distributed from their web page at (www.mrc-bsu.cam.ac.uk/bugs). Models can be implemented in two ways.

- using the BUGS language
- using the graphical feature, DoodleBUGS which allows the specification of models in terms of a directed graph

We believe that WinBUGS is a very handy tool in fitting complex models although it is a difficult and frustrating package to master. Bayesian analysis using WinBUGS requires three major tasks as follows.

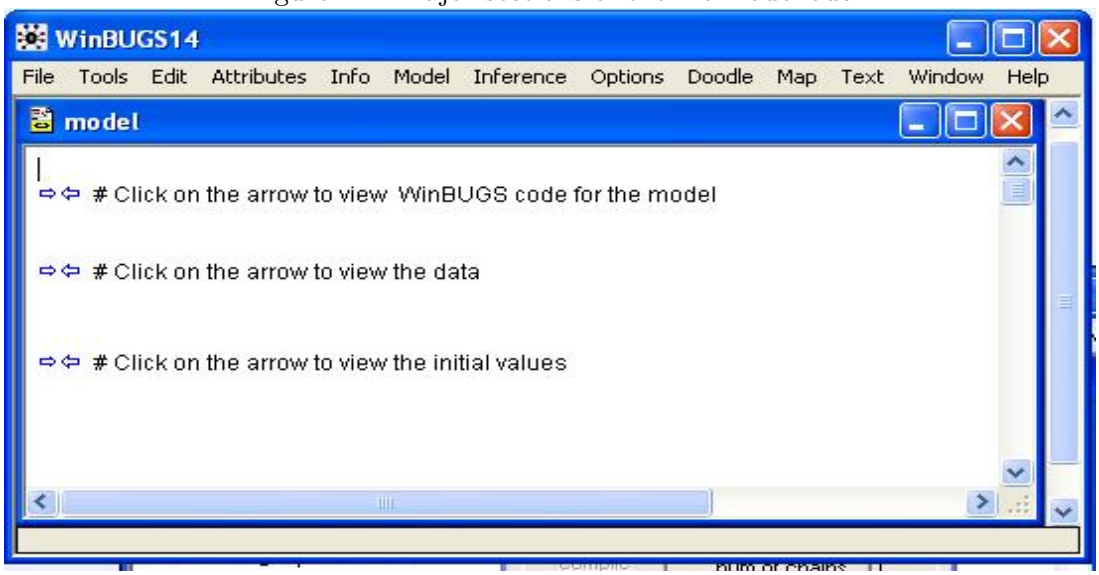
- model specification
- running the model
- Bayesian inference

2.6.2 Model Specification in the WinBUGS Language

In model specification, we need to create a WinBUGS file for implementation. We have provided the POST project with the file *model.odc*. This file has three major sections. When you open *model.odc*, three choices appear as given in figure 2.1.

By clicking an arrow, the expanded version of the relevant section of *model.odc* appears. A complete listing of *model.odc* is given in Appendix A. We now provide some preliminaries that are necessary to understand the code.

Figure 2.1: Major sections of the file model.odc



In WinBUGS, there are three types of nodes referred to as constant, stochastic and deterministic. Constant nodes are used to declare constant terms. Stochastic nodes represent data or parameters that are assigned a distribution. Currently WinBUGS provides 23 familiar distributions. Deterministic nodes are logical expressions of other nodes. Logical expressions can be built using the operators +, -, *, / and various WinBUGS functions. Note that WinBUGS has some special syntax which differs from other languages such as Splus and C++. As an example, WinBUGS requires that each node appear exactly once on the left hand side of an equation.

We now describe the flow of the code given in Appendix A. Our complete data likelihood in (2.4) has three separate probabilities in evaluating the likelihood. Log travel times are multivariate normal as given in (2.5), S_{ij} is Bernoulli($q_j^{t_{ij}}$) and ω_{ij} is Bernoulli(p_j) given S_{ij}. Since this is not one of the 23 WinBUGS distributions, we utilize the 'Zeros trick' to specify the likelihood. We create a variable 'zeros' which is assigned the value of zero as given below.

```
term1[i,j] <- s[i,j]*log(pow(p[j],c[i,j])*pow(1-p[j],1-c[i,j]))
zeros[i,j] <- 0
```

```
lambda[i,j] <- -term1[i,j]+k
zeros[i,j] ~ dpois(lambda[i,j])
```

The idea behind the Zeros trick is simple. Suppose that we have a Poisson observation of zero with parameter λ. Then the likelihood of this observation is $e^{-\lambda}$. If we set λ as the negative log-likelihood of a non-standard distribution, this gives the correct likelihood contribution. We add a large constant k to make sure that the mean of the Poisson variable is positive.

We then specify the prior density (2.6). We assign independent Beta(1,1) distributions for p_j and q_j such that any value between 0 and 1 is equally likely. We assign a Normal$(0, 10^5)$ distribution for μ which suggests that any value in the proximity of zero is equally likely for μ. With the normal distribution, WinBUGS parametrizes precision rather than variance. This explains the term 10^{-5} appearing in the code. We then assign a Wishart prior for the inverse variance-covariance matrix $G{=}\Sigma^{-1}$ as described in section 2.4. Finally, we calculate the inverse of G to obtain the variance-covariance matrix.

The next section in the WinBUGS code corresponds to the data. The data appear as three matrices, c for capture history, s for complete survival history and t for interval travel time. Note that missing data are denoted by NA in WinBUGS. Data can be viewed by clicking on the second arrow in figure 2.1. It is required to extract the capture history matrix, the survival history matrix and the interval travel time matrix from the observed cumulative travel time matrix. An R program for extracting $[\omega, S, t]$ from T^{obs} is given in Appendix B. Note that it is also possible to upgrade the POST database to produce these matrices.

The third section contains the initial values of the parameters in order to carry out MCMC. We can also allow WinBUGS to assign initial values for the parameters and this is described in the WinBUGS manual. The WinBUGS manual advises users to provide sensible initial values. We provide three sets of initial values for p, q and μ and ask WinBUGS to create initial values for Σ as described in section 2.6.3. We are now ready to run the model and we run 3 Markov chains. Up to now, we have described the code in Appendix A. The POST user does not yet need to do anything.

2.6.3 Running the Model in WinBUGS

The POST user can run the model and obtain posterior estimates of daily survival probabilities between listening lines, detection probabilities at listening lines, log travel times between listening lines and the correlation structure by following the steps given below.

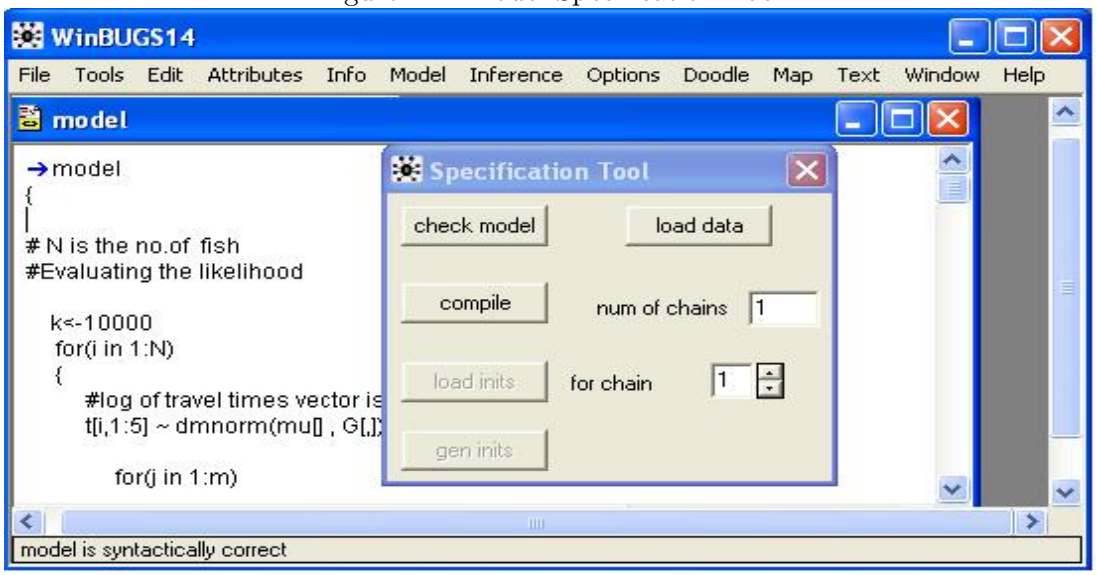
1. Open the file *model.odc* from WinBUGS.

   The file will appear as Figure 2.1. The user must make sure that they have unrestricted access to the full version of WinBUGS. This can be freely obtained by filling in the restriction form and stating your purpose.

2. Choose the first arrow in figure 2.1 and place the cursor somewhere inside of the expanded code.

3. Click on 'Specification' by pulling down the 'Model' menu. The screen will then appear as given in Figure 2.2.

Figure 2.2: Model Specification Tool



4. Choose the option 'check model' and watch the status bar in the lower left side of WinBUGS for the message 'model is syntactically correct'. This is the area where WinBUGS prints messages. This step verifies that the syntax of the code is correct.

5. The user should now click on the bold arrow located at the top of the WinBUGS code. This returns the user to the display in Figure 2.1. Next, choose the second arrow of figure 2.1 and highlight the keyword 'list'. Then select the option 'load data'. If the data are successfully loaded, the message 'data loaded' will appear in the message area.

6. Enter the number 3 in the field 'num of chains'.

7. Choose the option 'compile'. This step checks that the model structure for WinBUGS is correct. The message 'model compiled' then appears in the message area.

8. Click again on the bold arrow in the WinBUGS code. Choose the third arrow in Figure 2.1 and highlight the first occurrence of the keyword 'list'. Click on 'load inits'. The second chain is now invoked.

9. Highlight the second occurrence of the keyword 'list'. Click on 'load inits'. The third chain is now invoked.

10. Highlight the third occurrence of the keyword 'list'. Click on 'load inits'. In the later stages, POST users have the option to provide their previous estimates as the initial values in subsequent stages of estimation. This may reduce the computation time.

11. Click on 'gen inits' to ask WinBUGS to create initial values for Σ.

12. Choose 'samples' from the 'Inference' menu. It gives the options shown in Figure 2.3. Note that only some of the options (in bold) are accessible.

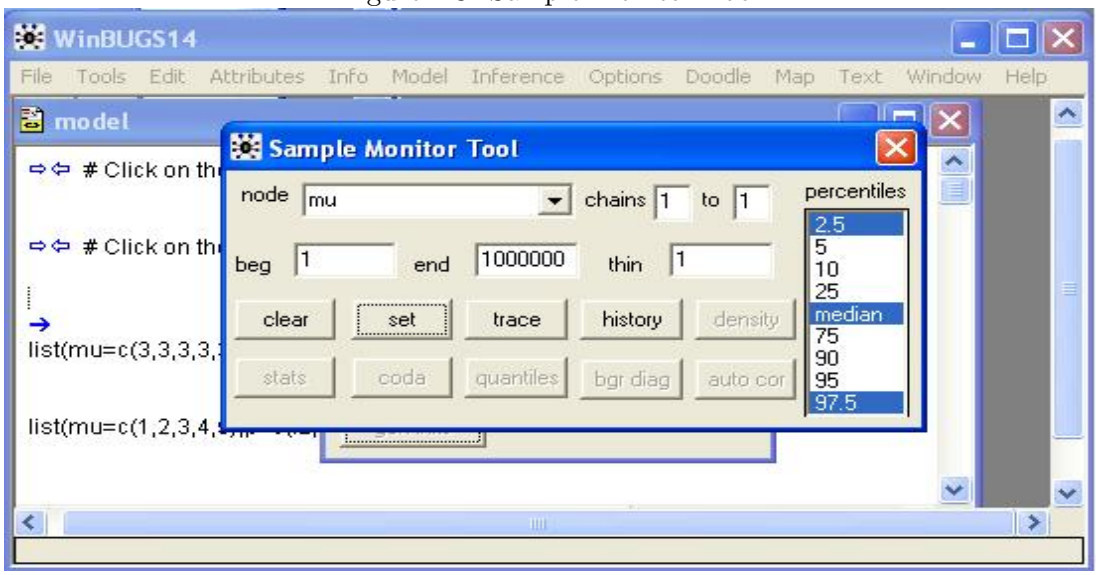
13. Type 'mu' in the field 'node' and click on 'set'. Do the same thing for p, q and varcov. By doing this, we request that Markov chain output is stored for these parameters. Markov chain output is not stored for parameters which are not specified.

14. Choose 'Update' from the 'Model' menu. The screen will appear as given in Figure 2.4.

15. Enter 10000 in the field 'updates'. This requests 10000 iterations of MCMC.

16. Click on 'update" to start the MCMC simulation.

Figure 2.3: Sample Monitor Tool



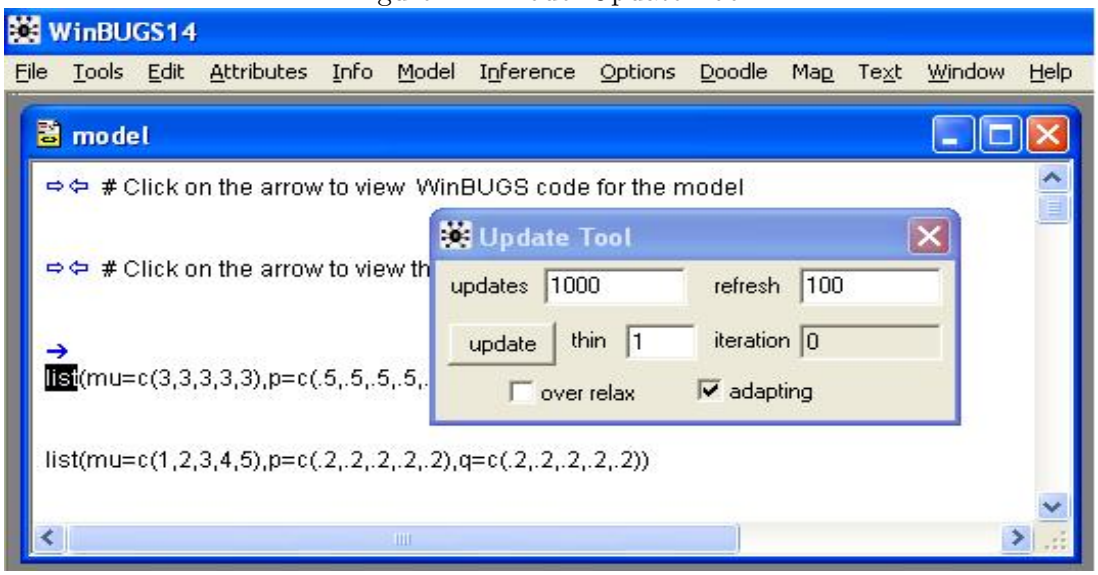
17. At the end of the simulation, Figure 2.3 again appears. POST users obtain the posterior estimates of daily survival probabilities between listening lines, detection probabilities at listening lines, log travel times between listening lines and the correlation structure by entering '*' in the field node of the 'Sample Monitor Tool' and clicking on 'stats'. On a typical 3.00GHz computer, the simulation phase requires 3-7 minutes of computation for 100 fish depending on the number of missing data points in the data set.

2.6.4 Bayesian Inference using WinBUGS

During and after MCMC simulation, WinBUGS provides several numerical and graphical summaries for the parameters. We briefly discuss some of these which we use in the next chapter.

The Option 'trace' in the Sample Monitor Tool provides a dynamic trace for each parameter which gets updated each time a variate is generated. This is a handy tool for investigating convergence. The 'history' button provides a trace plot of the entire Markov chain at the end of the simulation. The 'density' button provides empirical posterior density

Figure 2.4: Model Update Tool



plots of parameters using a kernel smoother. The 'stats' and 'quantiles' buttons provide basic posterior summaries of parameters. The full sequence of simulated values of each parameter is available from the 'coda' button. Coda output is easily accessible from other software platforms for further analysis. The 'bgr diag' button provides Brooks-Gelman-Rubin convergence statistics (Brooks and Gelman 1997) which converge to one when the Markov chain converges to the equilibrium distribution. Brooks and Roberts (1997) further discussed assessing convergence of MCMC with an emphasis on implementational issues and possible extensions. The 'auto cor' button provides the autocorrelation plots of the sequences. The autocorrelation plots illustrate the dependance between successive observations of the Markov chain.

In the next chapter, we test our model via simulated data and also consider the analysis of real data.

Chapter 3

Data Analysis

3.1 Model Adequacy via Simulated Data

In order to test the model, several simulation case studies were carried out. We wrote an R program to simulate the datasets.

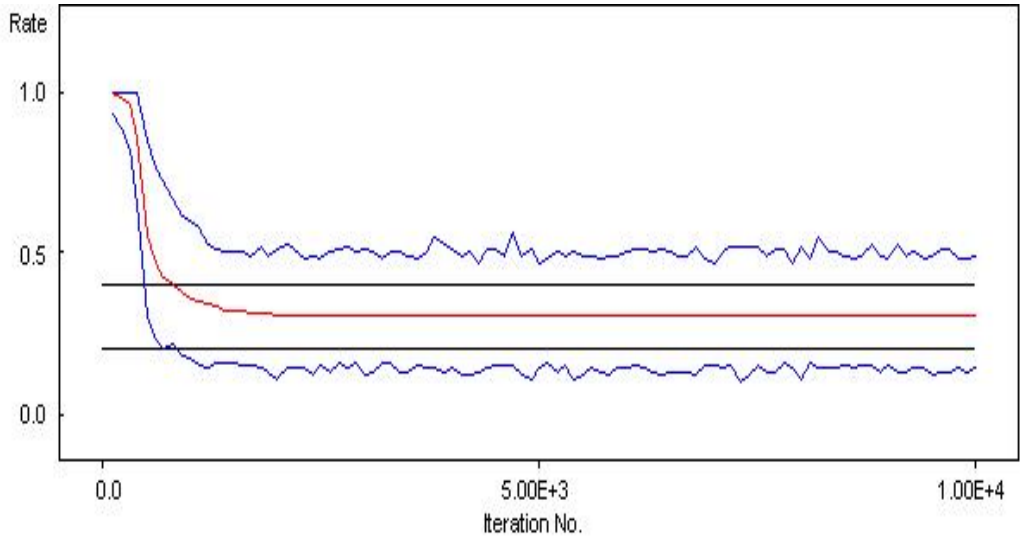
3.1.1 Case Study I

A dataset corresponding to $n = 500$ fish with $m = 5$ listening lines was simulated. Detection probabilities at each listening line were set to $p_j = p = 0.8$ while daily survival probabilities between listening lines were set to $q_j = q = 0.98$, $j = 1, ..., 5$. The log travel times of fish between listening lines were generated from a multivariate normal distribution with mean $\mu = [1, 2, 3, 4, 5]$ and variance-covariance matrix Σ where

$$\Sigma = \begin{bmatrix} 1 & .8 & .8 & .8 & .8 \\ .8 & 1 & .8 & .8 & .8 \\ .8 & .8 & 1 & .8 & .8 \\ .8 & .8 & .8 & 1 & .8 \\ .8 & .8 & .8 & .8 & 1 \end{bmatrix}.$$

The model was fit using 3 parallel chains as described in section 2.6.3. Figure 3.1 shows the minimum, maximum and average acceptance ratios from (2.9) using the Metropolis-Hastings algorithm averaged over all variates and 100 iterations. The average ratio lies

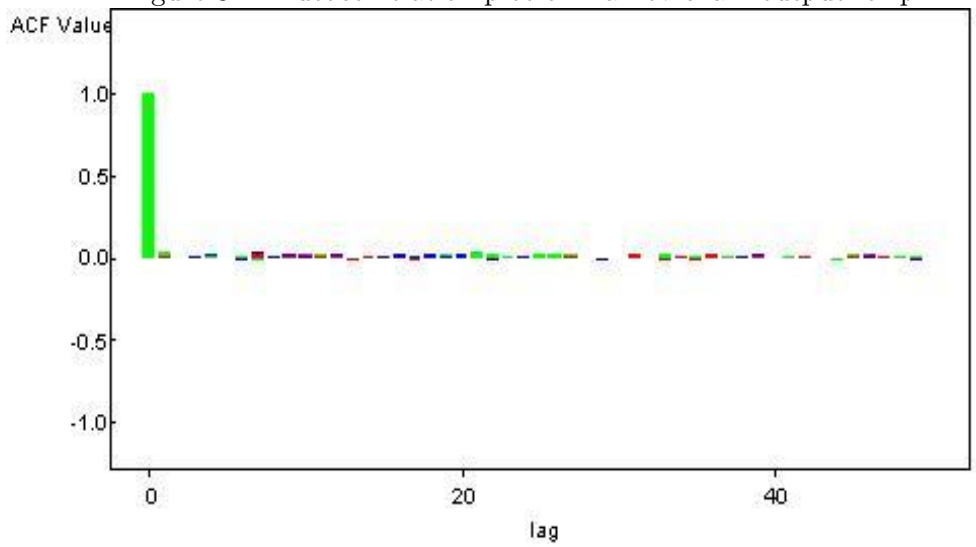
Figure 3.1: Minimum, maximum and average acceptance ratios for the Metropolis-Hastings algorithm



between 0.2 and 0.4 as desired by the WinBUGS program. The autocorrelation plot of the Markov chain for p is given in Figure 3.2. The autocorrelations appear to dampen quickly which indicates that successive variates are not strongly correlated. We observe that the autocorrelation plots of the remaining 21 parameters also die out in the same style. This suggests that it may be appropriate to average Markov chain output as though the variates were independent. Additionally, the trace plots also appear to converge quickly. The trace plots for μ are given in Figure 3.3. As can be seen, they appear to stabilize immediately and hence provide no indication of lack of convergence in the Markov chains. Figure 3.3 also indicates that 4000 iterations for the burn-in period is adequate as there is very little change between 4000 and 10000 iterations.

We also monitor the Brooks-Gelman-Rubin convergence statistic to assess convergence. The Brooks-Gelman-Rubin convergence statistic for q is given in Figure 3.4. It appears with the between chain variation plot and the within chain variation plot. As we simulated the variates from three independent chains, convergence of the within chain variability, the pooled chain variability and their ratio (the Brooks-Gelman-Rubin statistic) to one provides additional evidence of convergence. The Brooks-Gelman-Rubin convergence statistics

Figure 3.2: Autocorrelation plot of Markov chain output for p



for the remaining parameters also have similar appearances providing strong evidence of convergence.

Table 3.1 provides estimates of the posterior means and posterior standard deviations of the parameters. These are based on 18000 iterations after the 4000 burn-in period. As can be seen, the posterior means of the primary parameters p and q are close to the pre-set values. The posterior means of the secondary parameters μ and Σ also appear in agreement with the pre-set values.

Finally, Figure 3.5 provides estimates of the posterior density of μ using a kernel smoother. The plots suggest nearly symmetric unimodal distributions as might be expected.

3.1.2 Case Study II

We now investigate the sensitivity of the analysis with respect to the assumption of the normality of the log travel times. We simulated a dataset exactly as in Case Study I except that we generated $(t_{i1}, \ldots, t_{i5})' \sim \mathrm{N}_5(\mu^*, \Sigma)$ where $\mu^* = [4, 6, 8, 10, 12]$. We continued to use (2.5) as a modelling assumption. The posterior estimates of p and q were 0.78 and 0.99 respectively which suggest that the precise shape of the distributions of travel times is not

Figure 3.3: The trace plots for μ

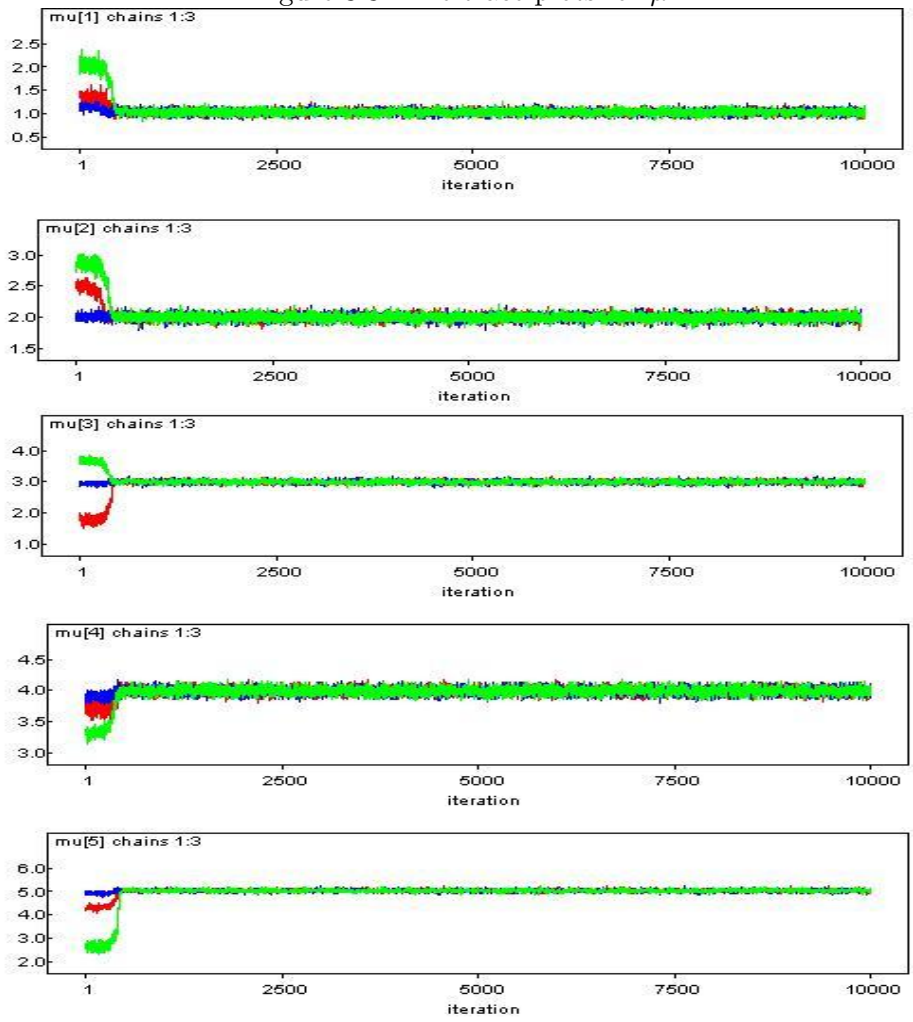


Figure 3.4: The Brooks-Gelman-Rubin convergence statistic for q along with the within chain variation and the between chain variation

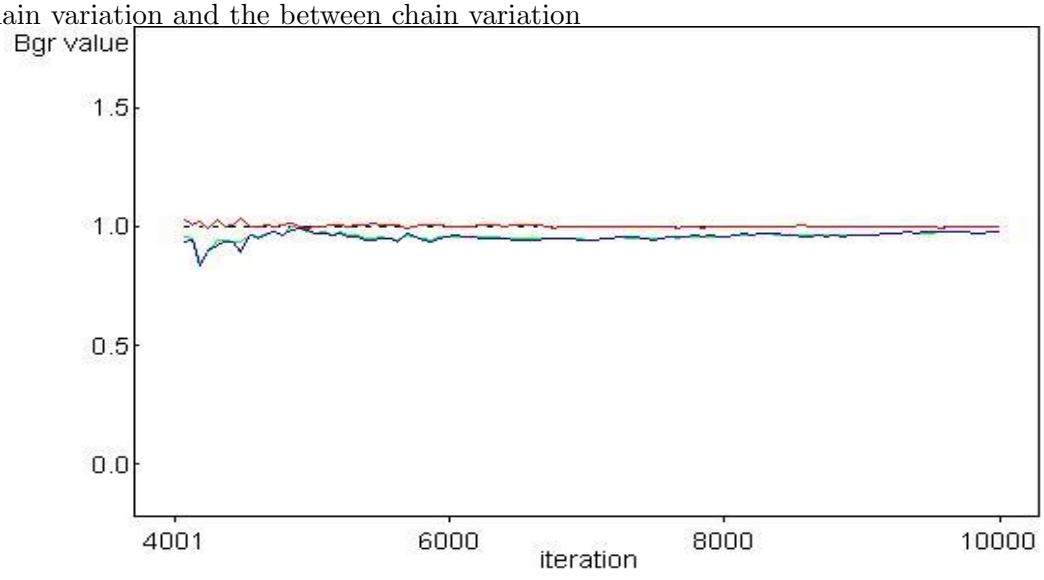


Figure 3.5: Estimates of posterior densities of μ in Case Study I

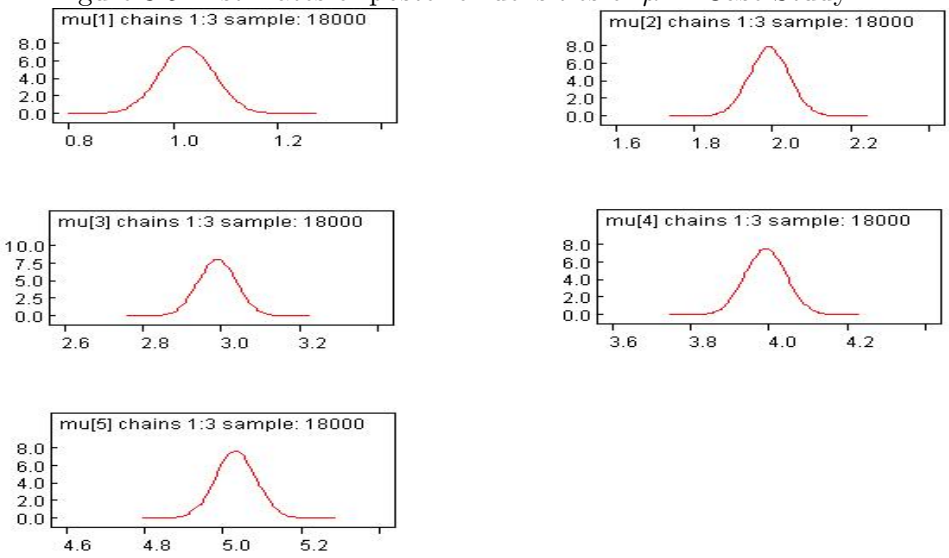


Table 3.1: Posterior estimates in Case Study I

Parameter	Mean	SD	Parameter	Mean	SD
p	0.77	0.08	Σ_{15}	0.88	0.06
q	0.99	0.00	Σ_{22}	0.99	0.07
μ_1	1.03	0.05	Σ_{23}	0.80	0.06
μ_2	1.99	0.05	Σ_{24}	0.84	0.06
μ_3	2.99	0.05	Σ_{25}	0.83	0.06
μ_4	3.99	0.05	Σ_{33}	0.92	0.07
μ_5	5.03	0.09	Σ_{34}	0.81	0.06
Σ_{11}	1.09	0.07	Σ_{35}	0.81	0.06
Σ_{12}	0.86	0.06	Σ_{44}	1.04	0.07
Σ_{13}	0.82	0.06	Σ_{45}	0.87	0.06
Σ_{14}	0.88	0.06	Σ_{55}	1.39	0.15

a critical assumption.

3.1.3 Case Study III

We now consider sensitivity issues related to the travel time assumption (2.5). We use the same data set that we used in Case Study I but consider the simpler travel time assumption $\log(t_{ij}) \sim \text{Normal}(\mu, \sigma^2)$ where a diffuse inverse gamma prior is assigned to σ^2. Under this simpler assumption, we ignore the correlation structure in the travel time data. Posterior estimates of parameters under this simpler model are given in Table 3.2. It seems that the prior dependence structure in the log travel times is not needed in this example. When datasets are smaller and there are more missing data, it may be important to use subjective knowledge in specifying a more informative Wishart prior.

3.1.4 Case Study IV

We now investigate the behaviour of the model with respect to missing data. We consider an extreme situation where 75% of the survival histories and interval travel times are missing. We generated a data set exactly as in Case Study I except with p=[0.9, 0.85, 0.8, 0.75, 0.7] and q=[0.95, 0.93, 0.91, 0.89, 0.87]. Note that 3711 simulated survival histories and interval travel times are missing in the dataset. We assume that p_j and q_j lie between 0.5 and 0.99. This is a realistic assumption in the POST project. We incorporate this little bit of information with priors by providing truncated beta(1,1) priors for p_j and q_j such that any value between 0.5 and 0.99 is equally likely. Note that the priors are still diffuse.

Table 3.2: Posterior estimates in Case Study III

Parameter	Mean	SD
p	0.76	0.08
q	0.99	0.00
μ_1	1.00	0.04
μ_2	1.94	0.06
μ_3	2.93	0.05
μ_4	3.90	0.05
μ_5	5.01	0.06
σ^2	1.01	0.03

We first apply the simpler model (as in Case Study III) with the travel time assumption $\log(t_{ij}) \sim \text{Normal}(\mu, \sigma^2)$. Posterior estimates of the parameters are given in Table 3.3. It appears that some of the estimates are less accurate and have higher standard errors with respect to the true values. This may be due to two reasons. Here, the likelihood may not provide sufficient information as in previous situations and also the model does not consider the prior dependence structure in the log travel times. We now apply the full model which takes into account the prior dependence structure in the log travel times. The posterior estimates of the parameters are given in Table 3.4. We observe that estimates are improved relative to the estimates which we obtain under the simpler model. This clearly suggests that the prior dependence structure in the log travel times is important when there are more missing data. Note that our priors are still diffuse and estimators may be improved by providing more informative priors if it is possible. In the POST project, there may be more informative prior knowledge about parameters.

We now investigate the posterior correlation matrices of p and q in order to check the independence assumption of p_j and q_j.

$$\hat{\Sigma}_q = \begin{bmatrix} 1 & 0.35 & 0.30 & 0.11 & 0.13 \\ & 1 & -0.13 & 0.15 & 0.07 \\ & & 1 & 0.21 & 0.14 \\ & & & 1 & 0.12 \\ & & & & 1 \end{bmatrix}$$

$$\hat{\Sigma}_p = \begin{bmatrix} 1 & -0.05 & -0.16 & -0.21 & -0.04 \\ & 1 & 0.43 & 0.21 & 0.03 \\ & & 1 & 0.26 & 0.07 \\ & & & 1 & 0.16 \\ & & & & 1 \end{bmatrix}$$

The correlation structures in $\hat{\Sigma}_p$ and $\hat{\Sigma}_q$ indicate that it is reasonable to assume the independence of the p_j and the independence of the q_j as is done in our prior specification.

Table 3.3: Posterior estimates in Case Study IV using simpler model

Parameter	Mean	SD	Parameter	Mean	SD
p_1	0.76	0.01	q_4	0.80	0.04
p_2	0.55	0.04	q_5	0.90	0.02
p_3	0.53	0.03	μ_1	0.80	0.04
p_4	0.64	0.11	μ_2	1.36	0.08
p_5	0.73	0.14	μ_3	2.41	0.11
q_1	0.99	0.00	μ_4	3.46	0.23
q_2	0.92	0.01	μ_5	5.02	0.34
q_3	0.86	0.01	σ^2	0.83	0.03

Table 3.4: Posterior estimates in Case Study IV using full model

Parameter	Mean	SD	Parameter	Mean	SD
p_1	0.85	0.03	Σ_{11}	1.03	0.07
p_2	0.88	0.03	Σ_{12}	0.96	0.09
p_3	0.74	0.06	Σ_{13}	1.43	0.23
p_4	0.74	0.12	Σ_{14}	1.31	0.27
p_5	0.76	0.14	Σ_{15}	0.37	0.18
q_1	0.97	0.01	Σ_{22}	1.29	0.16
q_2	0.90	0.01	Σ_{23}	1.69	0.28
q_3	0.91	0.01	Σ_{24}	1.39	0.32
q_4	0.83	0.03	Σ_{25}	0.48	0.19
q_5	0.90	0.03	Σ_{33}	2.95	0.71
μ_1	0.90	0.05	Σ_{34}	2.29	0.54
μ_2	2.07	0.08	Σ_{35}	0.74	0.29
μ_3	3.45	0.29	Σ_{44}	2.02	0.76
μ_4	4.43	0.47	Σ_{45}	0.63	0.35
μ_5	4.49	0.48	Σ_{55}	0.49	0.21

3.2 Columbia River Data

In this section, we apply the model to data obtained from the Columbia river system. From April 25/2001 to May 30/2001, $n = 324$ radio tagged chinook salmon were released from the Rock Island Dam. Data were recorded at listening lines established at the $m = 3$ dams downstream at Wanapum, Priest Rapids and Hanford Reach. A map of the dams is given in Figure 3.6. The interdam distances are approximately 37.6 miles, 18.7 miles and 15.0 miles respectively. Cowen and Schwarz (2005) also studied this data set in the context of radio failure. As before, the trace plots, the autocorrelation plots and the

Figure 3.6: Map of the dams Rock Island, Wanapum, Priest Rapids and Hanford Reach from 'Save Our Wild Salmon - www.wildsalmon.org'

Brooks-Gelman-Rubin convergence statistics of each parameter were examined to assess convergence. The trace plots of Markov chain output for capture probabilities, Brooks-Gelman-Rubin convergence statistics of daily survival probabilities and the autocorrelation plots of Markov chain output for μ are given in Figures 3.7, 3.8 and 3.9 respectively. They all provide evidence of convergence which is necessary in MCMC simulation. Estimates of the posterior means of the parameters are given in Table 3.5. We observe that detection probabilities decrease markedly as the fish travel to subsequent dams. An explanation for

this is that the radio tags may suffer degradation over time and are less likely to be detected as the fish travel downstream. The daily survival probabilities at the three dams are almost the same. The density plots of the log travel times (actual data) between dams are given in Figure 3.10. These plots provide some corroboration of the estimates of μ in Table 3.5. Note that the mean log travel times are not roughly proportional to the interdam distances. This may be an artefact of radio failure. Interestingly, the log travel time variances are increasing and this may be partly due to the fact that the region between Priest Rapids and Hanford Reach is a free-flowing river system. We also note that there is little correlation in the Σ matrix. This indicates that there is no evidence that fish which are faster between a set of dams also move faster between other sets of dams.

Figure 3.7: The trace plots for p

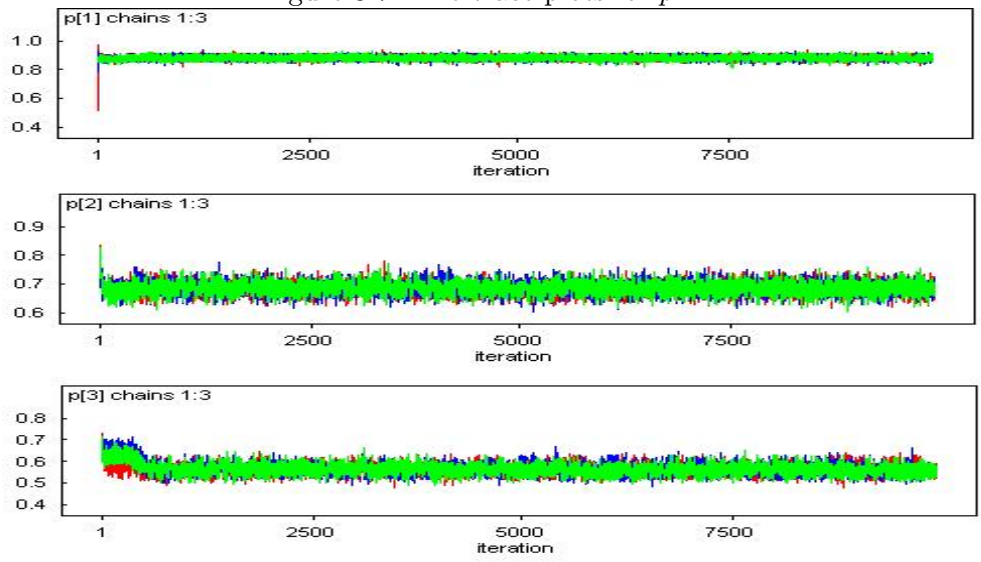


Figure 3.8: The Brooks-Gelman-Rubin convergence statistic for q along with the within chain variation and the between chain variation

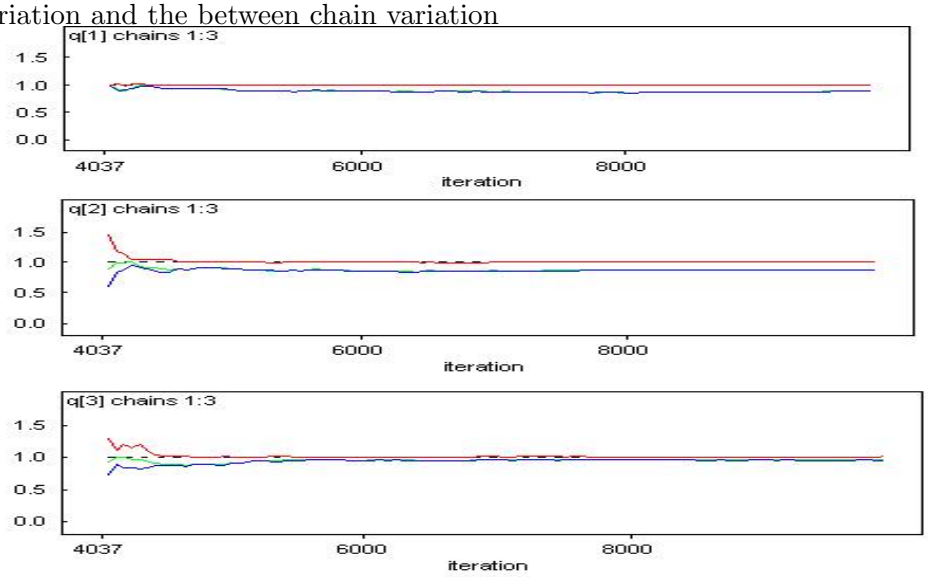


Figure 3.9: Autocorrelation plot of Markov chain output for μ

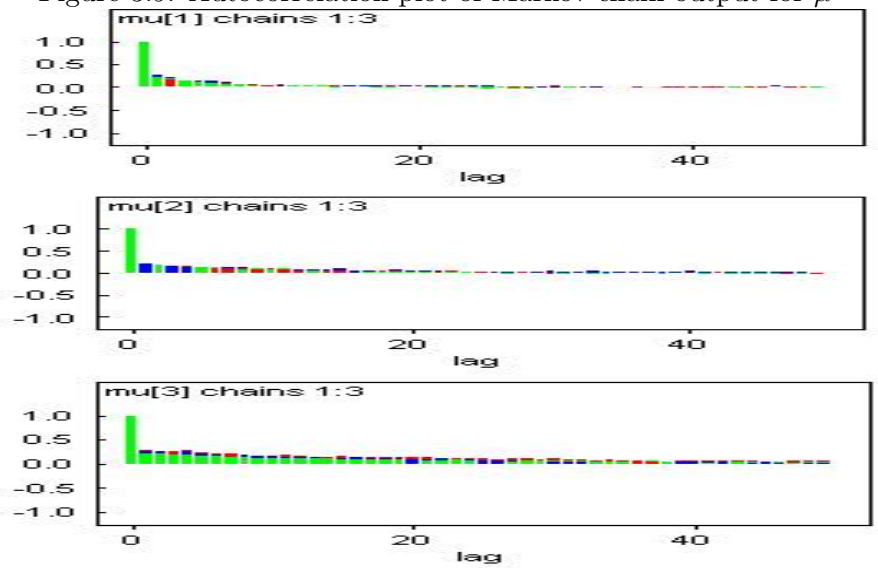
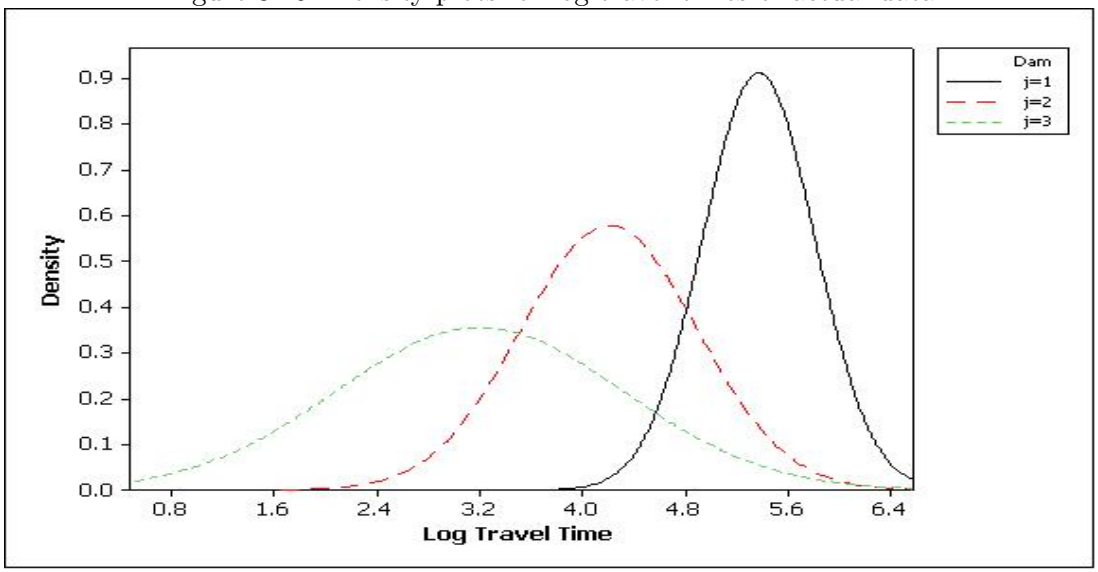


Table 3.5: Posterior estimates of parameters in the Columbia river data

Parameter	Mean	SD	Parameter	Mean	SD
p_1	0.881	0.01	μ_3	3.274	0.10
p_2	0.685	0.02	Σ_{11}	0.191	0.01
p_3	0.562	0.02	Σ_{12}	-0.026	0.02
q_1	0.999	0.00	Σ_{13}	-0.029	0.04
q_2	0.999	0.00	Σ_{22}	0.483	0.04
q_3	0.998	0.00	Σ_{23}	0.079	0.06
μ_1	5.380	0.02	Σ_{33}	1.341	0.16
μ_2	4.225	0.04			

Figure 3.10: Density plots for log travel times of actual data



Chapter 4

Conclusions

The goal of this project was to develop a model which estimates parameters of interest in the POST project. First, we investigated the theory behind mark-recapture models and then developed a model via the Bayesian framework due to the complexity of the problem. The Bayesian approach not only provided a flexible way to handle the complexity, it also allowed us to include prior knowledge of the parameters.

Recognizing the existence of missing data in interval travel times and survival status encouraged us to utilize the complete data likelihood as it was less complex than the observed likelihood. This required the introduction of latent variables. We specified diffuse priors but maintain that subjective priors are preferable when good prior information is available. Bayesian inference was carried out via Markov chain Monte Carlo simulation. MCMC was implemented via the Metropolis-Hastings algorithm using WinBUGS software which is a very useful and powerful tool for Bayesian computation.

Standard problems in MCMC simulation are the assessment of convergence and the determination of the length of the burn-in period. We used several graphical tools and the Brooks-Gelman-Rubin convergence statistic from WinBUGS to assess convergence and determine the length of the burn-in period. Several simulation case studies were conducted in order to test model adequacy. Results in section 3.1.2 indicate that the precise shape of the travel time distribution is not critical in parameter estimation. Section 3.1.4 indicates that the prior dependence structure in the log travel time should be used in parameter estimation, especially when there is considerable missing data. This also allows us to check

whether fish which are faster between a set of markers also move faster between other sets of markers. The simulation results indicate that our model is well behaved in estimating primary parameters (i.e. the detection probabilities at listening lines and the daily survival probabilities between listening lines), as well as the secondary parameters (i.e. log travel times between listening lines and the correlation structure).

Our model can also be applied in some health science problems. As an example, suppose that we are interested in the detection of a certain disease and surviving from it. In this example, survival probabilities correspond to the probability of surviving from the disease and it is reasonable to assume that they vary from patient to patient. Capture probabilities correspond to the probability of detecting the disease. Interval travel times can be replaced by time between checkups. It is also a known fact that survival probability may be a function of some other covariates. We can easily extend our model for such situations using a suitable link function to connect survival probability with covariates and assume a suitable distribution for covariates.

The model can further improve by using cumulative travel times instead of interval travel times or by setting some constraints based on cumulative travel times. This work is underway and we refer readers to paper by Muthukumarana, Schwarz and Swartz (2007) for details of improvement.

Appendix A

WinBUGS Code for the Model

```
model
{

# N is the no.of  fish
# Evaluating the likelihood

    k<-10000
for(i in 1:N)
{
#log of travel times vector is MVN, i.e., (2.5) in model development
t[i,1:5] ~ dmnorm(mu[] , G[,])

for(j in 1:m)
{
phi[i,j]<-pow(q[j],exp(t[i,j]))
     s[i,j]~dbern(phi[i,j])


     #Use of  0's trick
     # k is to make sure that the mean of the Poisson variable is positive
      term1[i,j] <- s[i,j]*log(pow(p[j],c[i,j])*pow(1-p[j],1-c[i,j]))
      zeros[i,j] <- 0
      lambda[i,j] <- -term1[i,j]+k
      zeros[i,j] ~ dpois(lambda[i,j])
   }
}
    #Prior distributions
      q[1]~dbeta(1,1)          #daily survival probability
      q[2]~dbeta(1,1)
      q[3]~dbeta(1,1)
      q[4]~dbeta(1,1)
```

```
  q[5]~dbeta(1,1)
  p[1]~dbeta(1,1)          #capture probability
  p[2]~dbeta(1,1)
  p[3]~dbeta(1,1)
  p[4]~dbeta(1,1)
  p[5]~dbeta(1,1)

  #log travel time mean vector
  mu[1:5] ~ dmnorm(g0[] , gv[,])

   g0[1] <- 0;  g0[2] <-0; g0[3] <- 0; g0[4] <- 0;  g0[5] <- 0

   gv[1,1] <- .00001;  gv[1,2] <- 0; gv[1,3] <- 0; gv[1,4] <- 0; gv[1,5] <- 0
   gv[2,1] <- 0;  gv[2,2] <- .00001; gv[2,3] <- 0; gv[2,4] <- 0; gv[2,5] <- 0
   gv[3,1] <- 0;  gv[3,2] <- 0; gv[3,3] <- .00001; gv[3,4] <- 0; gv[3,5] <- 0
   gv[4,1] <- 0;  gv[4,2] <- 0; gv[4,3] <- 0; gv[4,4] <- .00001; gv[4,5] <- 0
   gv[5,1] <- 0;  gv[5,2] <- 0; gv[5,3] <- 0; gv[5,4] <- 0; gv[5,5] <- .00001

#Variance  covariance matrix

G[1:5,1:5] ~ dwish(R[,],m)

  R[1,1]<-1/m; R[1,2]<- 0; R[1,3]<- 0;R[1,4]<- 0; R[1,5]<- 0
  R[2,1]<- 0; R[2,2]<-1/m; R[2,3]<- 0;R[2,4]<- 0; R[2,5]<- 0
  R[3,1]<- 0; R[3,2]<- 0; R[3,3]<-1/m;R[3,4]<- 0; R[3,5]<- 0
  R[4,1]<- 0; R[4,2]<- 0; R[4,3]<- 0;R[4,4]<-1/m; R[4,5]<- 0
  R[5,1]<- 0; R[5,2]<- 0; R[5,3]<- 0;R[5,4]<- 0; R[5,5]<-1/m

      varcov[1:5,1:5] <- inverse(G[,])

}

    # Click on the arrow to view  WinBUGS code for the model


list(N=10,m=5,

#capture data
     c=structure(.Data=c(
0 ,0 ,1 ,1 ,1 ,
0 ,0 ,1 ,0 ,1 ,
1 ,0 ,1 ,0 ,1 ,
1 ,1 ,1 ,1 ,1 ,
1 ,1 ,1 ,1 ,0 ,
0 ,0 ,1 ,1 ,1 ,
```

```
1 ,1 ,1 ,1 ,1 ,
1 ,1 ,1 ,1 ,0 ,
1 ,1 ,1 ,1 ,1 ,
1 ,1 ,1 ,1 ,1 ),.Dim=c(10,5)),

#Survival data
     s=structure(.Data=c(
1 ,1 ,1 ,1 ,1 ,
1 ,1 ,1 ,1 ,1 ,
1 ,1 ,1 ,1 ,1 ,
1 ,1 ,1 ,1 ,1 ,
1 ,1 ,1 ,1 ,NA ,
1 ,1 ,1 ,1 ,1 ,
1 ,1 ,1 ,1 ,1 ,
1 ,1 ,1 ,1 ,NA ,
1 ,1 ,1 ,1 ,1 ,
1 ,1 ,1 ,1 ,1),.Dim=c(10,5)),

#log of travel times
     t=structure(.Data=c(
NA ,NA ,NA ,4.570213847 ,5.147049414 ,
NA ,NA ,NA ,NA ,NA ,
1.569668475 ,NA ,NA ,NA ,NA ,
0.332605771 ,0.811377733 ,2.155025381 ,3.338856666 ,4.454989111 ,
2.676427836 ,3.886440009 ,4.405275821 ,5.294069073 ,NA          ,
NA ,NA ,NA ,2.993772157 ,3.737354068 ,
0.774916249 ,2.315602835 ,2.153099099 ,3.864518223 ,5.614917829 ,
1.81593285 ,2.637515863 ,3.672125569 ,3.72029615 ,NA ,
2.06601186 ,3.874599418 ,4.407558935 ,4.241067378 ,5.762918364 ,
2.834203738 ,2.698174478 ,4.163108617 ,5.899092117 ,6.396100374),.Dim=c(10,5)))


   # Click on the arrow to view the data

list(mu=c(3,3,3,3,3),p=c(.5,.5,.5,.5,.5),q=c(.5,.5,.5,.5,.5))


list(mu=c(1,2,3,4,5),p=c(.2,.2,.2,.2,.2),q=c(.2,.2,.2,.2,.2))


list(mu=c(5,4,3,2,1),p=c(.4,.4,.4,.4,.4),q=c(.7,.7,.7,.7,.7))

  # Click on the arrow to view the initial values
```

Appendix B

An R Program for Extracting $[\omega, S, t]$ from the T^{obs} matrix

```
his=function(T)
{

# T is the cumulative travel time matrix
# n is the number of fish
# m is the number of listening lines
# c is the capture history matrix
# S is the survival history matrix
# t is the interval travel time matrix

nm=dim(T)
n=nm[1]
m=nm[2]

i=1
j=1

c=matrix(ncol=m,nrow=n)
s=matrix(ncol=m,nrow=n)
t=matrix(ncol=m,nrow=n)


 for(i in 1:n)
 {

   c[i,1]=ifelse(T[i,1]==-999,0,1)

   t[i,1]=ifelse(T[i,1]!=-999,T[i,1],NA)
```

```
   for(j in 2:m)
   {

   c[i,j]=ifelse(T[i,j]==-999,0,1)

   t[i,j]=ifelse(T[i,j]!=-999 & T[i,j-1]!=-999,T[i,j]-T[i,j-1],NA)


   }
 }

     for(i in 1:n)
     {

      s[i,m]=ifelse(c[i,m]==1,1,NA)

         for(j in 2:m)
         {

 s[i,m+1-j]=ifelse( c[i,m+1-j]==0 & s[i,m+2-j]=='NA',NA,1)
         }
    }

list(c,s,t,T)

}
```

Bibliography

[1] Bonner, S.J. and Schwarz, C.J. An extension of the Cormack-Jolly-Seber model for continuous covariates with application to Microtus pennsylvanicus. *Biometrics*, 62:142–149, 2006.

[2] Brooks, S.P. and Gelman, A. Alternative methods for monitoring convergence of iterative simulations. *Computational and Graphical Statistics*, 7:434–455, 1997.

[3] Brooks, S.P. and Roberts, G.O. Assessing convergence of markov chain monte carlo algorithms. *Statistics and Computing*, 8:319–335, 1998.

[4] Brooks, S.P., Catchpole, E.A. and Morgan, J.T. Bayesian animal survival estimation. *Statistical Science*, 15:357–376, 2000.

[5] Burnham, K.P., Anderson, D.R., White, G.C., Brownie, C. and Pollock, K.H. Design and Analysis Methods for Fish Survival Experiments based on Release-Recapture. *Bethesda, Maryland: American Fisheries Society*, 1987.

[6] Cormack, R.M. Estimates of survival from the sighting of marked animals. *Biometrika*, 51:429–438, 1964.

[7] Cowen, L. and Schwarz, C.J. Capture-recapture studies using radio telemetry with premature radio-tag failure. *Biometrics*, 61:657–664, 2005.

[8] Dupuis, J.A. Bayesian estimation of movement and survival probabilities from capture-recapture data. *Biometrika*, 82:761–772, 1995.

[9] Edward, I.G. and Christian, P.R. Capture-recapture estimation via Gibbs sampling. *Biometrika*, 79:677–683, 1992.

[10] Evans, M. and Swartz, T.B. Methods for approximating integrals in Statistics with special emphasis on Bayesian integration problems. *Statistical Science*, 10:254–272, 1995.

[11] Evans, M. and Swartz, T.B. *Approximating Integrals via Monte Carlo and Deterministic Methods*. Oxford University Press, 2000.

[12] Jolly, G.M. Explicit estimates from capture-recapture data with both death and immigration-stochastic model. *Biometrika*, 52:225–247, 1965.

[13] Lebreton, J., Burnham, K.P., Clobert, J. and Anderson, D.R. Modelling survival and testing biological hypotheses using marked animals: a unified approach with case studies. *Ecological Monographs*, 62:67–118, 1992.

[14] Muthukumarana, S., Schwarz, C.J. and Swartz, T.B. Bayesian analysis of mark-recapture data with travel-time-dependent survival probabilities. *Manuscript*, 2007.

[15] Peterson, C.G.J. The yearly immigration of young plaice into the limfjord from the German sea. *Report of Danish Biological Station*, 6:1–48, 1896.

[16] Pollock, K.H., Bunck, C.M., Winterstein, S.R. and Chen, C. A capture-recapture survival analysis model for radio-tagged animals. *Journal of Applied Statistics*, 22:661–672, 1995.

[17] Schnabel, Z.E. The estimation of the total fish population of a lake. *American Mathematical Monthly*, 45:348–352, 1938.

[18] Schwarz, C.J. and Seber, G.A.F. Estimating animal abundance: Review III. *Statistical Science*, 14:427–456, 1999.

[19] Seber, G.A.F. A note on the multiple recapture census. *Biometrika*, 52:249–259, 1965.

[20] Shirley, P., Pollock, K.H. and Norris, J.L. Open capture-recapture models with heterogeneity: I. Cormack-Jolly-Seber model. *Biometrics*, 59:786–794, 2003.

[21] Spiegelhalter, D., Thomas, A. and Best, N. WinBUGS (Version 1.4) User Manual. *Cambridge: MRC Biostatistics Unit*, 2003.

[22] van Deusen, P.C. An EM algorithm for capture-recapture estimation. *Environmental and Ecological Statistics*, 9:151–165, 2002.