ABSTRACT

Bayesian Methods to Estimate the Accuracy of a Binary Measurement System

Mark Eschmann, Ph.D.

Chairperson: James D. Stamey, Ph.D.

Binary Measurement Systems (BMS) are frequently used in such applications as quality control. They are important enough that their operating characteristics, the repeatability and reproducibility are important because of the clues that provide engineers in improving the BMS. We have developed a Bayesian single inspector model that incorporates baseline information. Simulation studies show that this can cut credible set width of the misclassification parameters in half. We have also developed Bayesian fixed effects and random effects models for BMS's with multiple inspectors. Simulation studies that demonstrated acceptable performance characteristics. Several applications are also worked out included sample size determination for the fixed effects model, and superior subset selection with the random effects model.

Bayesian Methods to Estimate the Accuracy of a Binary Measurement System

by

Mark Eschmann, B.S., M.S.

A Dissertation

Approved by the Department of Statistical Science

_____

Jack D. Tubbs, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of
Doctor of Philosophy

Approved by the Dissertation Committee

_____

James D. Stamey, Ph.D., Chairperson

_____

Steve Green, Ph.D.

_____

David Kahle, Ph.D.

_____

Jack D. Tubbs, Ph.D.

_____

Dean Young, Ph.D.

Accepted by the Graduate School
May 2016

_____

J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, I would like to thank the members of my committee. I especially would like to thank my advisor Dr. Stamey. Because of your help and patience, I have accomplished something which, at several points along the way, I did not think I would.

I would also like to thank the Statistics department at Baylor. I am thankful that I studied in a department whose faculty were very open, helpful, and flexible. I am also thankful for the students in the department during my time at Baylor. I am very proud to call many of you my friends.

Lastly, I would like to thank Carl Bell in the Academic and Research Computing Services for all of his help in getting my simulations to run on Baylor's cluster computer. His help greatly assisted and shortened the time to conduct my research.

CHAPTER ONE

Introduction

Repeatability and reproducibility (R& R) is an important aspect in quality control. In this procedure an object is measured multiple times to assess the degree of similarity of the measurements. There is a long history of these sorts of studies when responses are continuous. See, for example, Burdick et al. (2003) and the references therein. Assessing variability and error in measurement, measurement system analysis (MSA), is of course vital to process improvement. Recently there has been considerable interest in this type of analysis for binary data using a variety of approaches. Boyles (2001) considers the case where a single system (rater) rates each part multiple times. The repeated testing for binary data paradigm has been extended by van Wieringen and de Mast (2008), Danila et al. (2008, 2010), and de Mast et al. (2011).

In Chapter Two, we develop a Bayesian single inspector model that incorporates baseline data. This is an extension of Beavers and Stamey (2012) in which baseline information is incorporated and Danila et al. (2010) by developing it from Bayesian perspective. An example analysis is shown in Section 2.3 and a simulation study is performed in Section 2.4. The simulation study examines the performance characteristics of the model and demonstrates the superiority of the model over one in which the baseline data is not incorporated.

In Chapter Three, we develop a Bayesian fixed effects model for multiple inspectors. This is an extension of Beavers and Stamey (2012) in which there are multiple inspectors and van Wieringen and de Mast (2008) by developing it from Bayesian perspective. A simulation study is performed in Section 3.3 which exam-

ines the performance characteristics of the model. In Section 3.4, an algorithm for sample size determination for the model is presented, and an example is worked out.

In Chapter Four, we develop a Bayesian random effects model for multiple inspectors. This is an extension of Beavers and Stamey (2012). An example analysis is presented in Section 4.3, and a simulation study is performed in Section 4.4. The simulation study examines the performance characteristics of the model. In Section 4.5, a second model for random effects is presented which incorporates baseline in a way similar to Chapter Two. It should be regarded as a proof of concept model as there are some approximations in the model. A simulation study demonstrating the performance characteristics of the model as well as the superiority it is also in the section. Lastly, a ranking and best subset selection model is presented in Section 4.6. This model assumes both multiple plants with different misclassification rates and a desire to rank and characterize them. A small simulation study is done with this model.

CHAPTER TWO

Estimating Performance of a BMS Incorporating Existing Baseline Information

2.1 Introduction

Repeated binary, pass-fail testing, often called a binary measurement system (BMS), is regularly used in quality control studies as a means of assessing the quality of the production units. A gold standard inspection system is often too expensive or not available. Thus, these inspection methods are highly dependent on the quality of the individual inspectors. This makes the study of the inspection attributes itself an integral part of the quality control process.

One important application of BMS is in the evaluation of online manufacturing processes. These evaluations differ from evaluations of new systems in both that samples of previously passed and failed parts are usually readily available, and in that the overall pass rate of the BMS is frequently known. Examples might be the inspection of credit cards or of electronic components. In the present chapter, the incorporation of this baseline data for the evaluation of single inspector BMS is introduced from a Bayesian perspective. This in effect makes two assumptions about the inspection process: one that it is sufficiently high volume for the incorporation of this overall pass rate to be useful and two that the inspection process is relatively homogeneous and can thus be represented as a single inspector. This is an extension of Beavers and Stamey (2012) in which baseline information is incorporated. It is also extends Danila et al. (2010) by developing it from Bayesian perspective.

There are multiple advantages to using a Bayesian approach. The first is that prior knowledge can be incorporated into the study in a natural way with the use of informative prior distributions. Another is that even in the case of no prior knowledge where the asymptotic dominance of the prior by the likelihood is

necessary (or some form of sensitivity analysis is required), the interval estimates generated from the Bayesian paradigm are based largely on the likelihood which has been shown superior to other interval estimation methods for this problem type (Boyles 2001).

After the mathematical presentation of the model, a Gibbs sampling approach to the problem is described. Markov Chain Monte Carlo methods are then used to sample from the model's posterior distribution. The programs JAGS and R were used to do the MCMC computations necessary for the model implementation.The R package RJAGS was used to interface between them.

The rest of this chapter is organized as follows. Section 2.2 introduces the single inspector model that incorporates baseline data. An example is worked out in Section 2.3. In Section 2.4 a simulation study is performed that both demonstrates the operating characteristics of the model and compares its performance to a model that does not incorporate baseline data.

2.1.1 Definitions and Basic Assumptions

Let the true quality state of a potentially defective item be denoted by T such that

$$T = \begin{cases} 0 & \text{for a non conforming item} \\ 1 & \text{for a conforming item.} \end{cases} \tag{2.1}$$

The probability τ_c will denote the overall conforming rate. We assume that no gold standard is used. Thus, T is a latent variable, and we further assume that T is distributed Bernoulli(τ). We will be most interested in the case that $\tau > .8$ as that is the case most often seen in real production processes.

A BMS classifies parts as either passing or failing. We denote the results of this classification by

$$y = \begin{cases} 0 & \text{for a part that fails inspection} \\ 1 & \text{for a part that passes inspection.} \end{cases} \tag{2.2}$$

We further note that the quantities of primary interest in the analysis of an inspection process are the false positive and false negative rates. We define

$$\theta_+ = P(y = 1|T = 0), \text{ and} \tag{2.3}$$

$$\theta_- = P(y = 0|T = 1), \tag{2.4}$$

to be these quantities of interest. This notation is consistent with Beavers et al. (2011). These quantities are also known as the consumer's risk, θ_+ and the producer's risk, θ_-. Generally, the consumer's risk, θ_+ is considered more important because it is the chance that a bad item reaches the consumer Danila et al. (2008). We further implicitly assume here, that these misclassification rates remain constant across all parts. That is, we assume that every conforming (nonconforming) part has the same misclassification chance θ_- (θ_+). One particular and common violation of this assumption is if the classification is based on an underlying continuous measurand (de Mast et al. 2011).

2.1.2 Sampling Plans

A number of different sampling plans have been proposed to use in evaluating a BMS. De Mast et al. (2011) note that there are two very important things to consider that effect the choice of a sampling plan.

The first, and well known, important consideration is the application of gold standard inspection systems. Sometimes a gold standard system is entirely unavailable, while in other applications it may just be prohibitively expensive to use, and thus rarely used. Sampling plans can and should be designed to take advantage of

whatever gold standard inspection system is available. It is assumed in all the literature that a gold standard system is unavailable for all items as that would render questions about the quality of the inspection system superfluous.

The second important consideration is whether the data is truly dichotomous or is artificially dichotomized. De Mast et al. (2011) discuss this situation at length and its important ramifications on both sampling methods and model building. They show, among other things, the importance of truly random sampling when the underlying measurand is continuous. Erdmann and de Mast (2012) extend this analysis to include hybrid measurands. That is, if the measurand is a hybrid between a continuum and a dichotomy.

These two considerations are not independent from the perspective of experimental design, and all four possible cases are analyzed in greater detail from this perspective in de Mast et al. (2011). Where there is a continuous measureand, their recommendation, in so far as possible, is to not artificially dichotomize the results, and to instead use the continuous measurand. They do discuss how random sampling can alleviate the issue some, but because their primary recommendation is to basically treat continuous variables as continuous the emphasis in this chapter will instead be on true dichotomies. The likelihoods for when you do gold standard inspections versus when you don't do them are similar enough that both likelihoods will be produced in the present chapter. It will, however, be assumed that a gold standard measurement system, if it exists, is too expensive to use as liberally as Farnum's plan calls for (Farnum 1994).

In the absence of a gold standard inspection, what must instead be used is a latent class model. Latent class models rely on measuring each part multiple times, critically assuming that the BMS classifications are independent.

Danila et al. (2010), analyzing the problem from a frequentist perspective, defines three different sampling plans relevant given a true dichotomy and limited availability of a gold standard inspection system:

- The standard plan (SP): randomly sample from the overall population of items
- The random sample plan (RS): randomly sample from the streams of passed and failed items
- The conditional sampling plan (CS): randomly sample n_0 items from the stream of failed items and n_1 items from the stream of passed items.

They show that the conditional sampling plan is uniformly better than the other plans. More precisely, they show that the standard error of the estimates for θ_-, θ_+, and τ are smaller in all cases for the conditional sampling plan.

To explain a little further why this is, it is necessary to examine in greater detail the increased information utilized by each successive plan. A key statistic that can be utilized by all three plans is knowledge of the overall pass rate, $p = P(Y = 1)$. An important mathematical relationship is that

$$p = \tau(1 - \theta_-) + (1 - \tau)\theta_+. \tag{2.5}$$

Equation 2.5 allows known information about an inspection process' overall pass rate to be incorporated into an experiment. It is also the means by which τ can be estimated in the RS and CS plans.

The issue that the standard plan faces is that in most production processes τ is quite high which means that there are few bad items in the sample from which to estimate θ_+. The other two plans remedy this by partially sampling from the stream of items which have failed an inspection, and thus are more likely to be non conforming. The key difference between the RS and CS plans, is that the CS plan

keeps track of whether an item originally passed or failed. The importance of this, is far greater than the weight of a single inspection might indicate. Rather, the first inspection, the baseline data, changes the population that the sample is coming from. That is,

$$P(T = 1|Y = 1) = \frac{\tau(1 - \theta_-)}{\tau(1 - \theta_-) + (1 - \tau)\theta_+} \tag{2.6}$$

$$P(T = 1|Y = 0) = \frac{\tau\theta_-}{\tau\theta_- + (1 - \tau)(1 - \theta_+)}. \tag{2.7}$$

These probabilities are also known as the positive predictive value (PPV) and the false omission rate (FOR) respectively in the biomedical literature (Altman and Bland 1993).

For the CS plan with l repeated inspections, and defining $X = \sum_{i=1}^{N} Y_i$ to be the number of times an item passes inspection, we have that

$$X \sim \begin{cases} \text{Binomial}(l, P(T = 1|Y = 0)) & \text{for a part that fails inspection} \\ \text{Binomial}(l, P(T = 1|Y = 1)) & \text{for a part that passes inspection.} \end{cases}$$

Because the conditional sampling plan has been shown to be uniformly better than the other plans, it is the one utilized through the rest of this chapter.

2.2 The Model

Assume that there are N_0 randomly selected items from the stream of failed parts, and N_1 randomly selected items from the stream of passed parts. Let $N = N_0 + N_1$, and $\xi_i = 1$ if an item passed its first inspection and 0 otherwise. Repeated independent and fallible observations are then made on the i^{th} unit to indirectly assess the true state of the i^{th} unit where $i \in \{1, \cdots, N\}$. Let $Y_{i,j}$ be the result of the j^{th} inspection on the i^{th} item where $j \in \{1, \cdots, l_i\}$. Also let $Y_{i,j} = 1$ be a passed inspection and $Y_{i,j} = 0$ be a failed inspection. For each $Y_{i,j}$, further define the conditional probabilities $\theta_+ = P(Y_{i,j} = 1|T_i = 0)$ (false positive rate) and $\theta_- = P(Y_{i,j} = 0|T_i = 1)$ (false negative rate) with respect to the true state of the

item, T_i. Further, assume that the conditional distribution of $Y_{i,j}$ is

$$Y_{i,j}|T_i = 0 \sim \text{Bernoulli}(\theta_+),$$
$$Y_{i,j}|T_i = 1 \sim \text{Bernoulli}(1-\theta_-).$$

Let $X_i = \sum_{k=1}^{l_i} Y_{i,j}$. Due to the independence of the inspections we have that the conditional distributions $X_i|T_i = 0 \sim \text{Binomial}(l_i, \theta_+)$ and similarly $X_i|T_i = 1 \sim \text{Binomial}(l_i, 1-\theta_-)$. Thus, part of the likelihood of the latent vector $\mathbf{t} = \{t_1, \cdots, t_N\}'$ and the observed data matrix $\mathbf{x} = \{x_1, \cdots, x_N\}$ is

$$\begin{aligned} L(\mathbf{x}|\theta_-, \theta_+, \mathbf{t}) \propto & (1-\theta_-)^{\sum_{i=1}^N t_i x_i} (\theta_-)^{\sum_{i=1}^N t_i (l_i - x_i)} \\ & \times (\theta_+)^{\sum_{i=1}^N (1-t_i) x_i} (1-\theta_+)^{\sum_{i=1}^N (1-t_i)(l_i - x_i)}. \end{aligned} \tag{2.8}$$

To incorporate information on whether the part comes from the previously passed parts or previously failed parts requires knowledge of the previous result and Equations 2.6 and 2.7. Assume without loss of generality that the first N_0 items failed their first inspection and the last N_1 passed it. If this information is available then the additional likelihood information is

$$\begin{aligned} L(\mathbf{t}|\tau, \theta_-, \theta_+, \xi) \propto & \left(\frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)} \right)^{\sum_{i=1}^{N_0} t_i} \\ & \times \left(1 - \frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)} \right)^{N_0 - \sum_{i=1}^{N_0} t_i} \\ & \times \left(\frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+} \right)^{\sum_{i=N_0+1}^{N} t_i} \\ & \times \left(1 - \frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+} \right)^{N_1 - \sum_{i=N_0+1}^{N} t_i}. \end{aligned} \tag{2.9}$$

Also incorporated in this model is information about the overall pass rate p in Equation 2.5. Let N_{pop} be the total number of items inspected by an inspection process, and N_{pass} be the total number of items passed by an inspection process.

Then the additional contribution this information makes to the likelihood is

$$L(N_{pass}|\theta_-,\theta_+,\tau) \propto (\tau(1-\theta_-)+(1-\tau)\theta_+)^{N_{pass}} \times (1-\tau(1-\theta_-)-(1-\tau)\theta_+)^{N_{pop}-N_{pass}}. \quad (2.10)$$

Combining the previous parts, the full likelihood is

$$\begin{aligned} L(\mathbf{x},\mathbf{t},N_{pass}|\xi,\theta_-,\theta_+,\tau) \propto & L(N_{pass}|\theta_-,\theta_+,\tau)L(\mathbf{t}|\tau,\theta_-,\theta_+,\xi)L(\mathbf{x}|\theta_-,\theta_+,\mathbf{t}) \\ \propto & (1-\theta_-)^{\sum_{i=1}^N t_i x_i}(\theta_-)^{\sum_{i=1}^N t_i(l_i-x_i)} \\ & \times (\theta_+)^{\sum_{i=1}^N (1-t_i)x_i}(1-\theta_+)^{\sum_{i=1}^N (1-t_i)(l_i-x_i)} \\ & \times \left(\frac{\tau\theta_-}{\tau\theta_-+(1-\tau)(1-\theta_+)}\right)^{\sum_{i=1}^{N_0} t_i} \\ & \times \left(1-\frac{\tau\theta_-}{\tau\theta_-+(1-\tau)(1-\theta_+)}\right)^{N_0-\sum_{i=1}^{N_0} t_i} \\ & \times \left(\frac{\tau(1-\theta_-)}{\tau(1-\theta_-)+(1-\tau)\theta_+}\right)^{\sum_{i=N_0+1}^{N} t_i} \\ & \times \left(1-\frac{\tau(1-\theta_-)}{\tau(1-\theta_-)+(1-\tau)\theta_+}\right)^{N_1-\sum_{i=N_0+1}^{N} t_i} \\ & \times (\tau(1-\theta_-)+(1-\tau)\theta_+)^{N_{pass}} \\ & \times (1-\tau(1-\theta_-)-(1-\tau)\theta_+)^{N_{pop}-N_{pass}}. \end{aligned} \quad (2.11)$$

2.2.1 Identifiability

There are several additional model assumptions required for inference most notably those pertaining to the identifiability of the model. The first is that

$$\theta_+ + \theta_- < 1. \quad (2.12)$$

This condition essentially says that the chance of correctly classifying an item is greater than the chance of misclassifying it. It is necessary for identifiability due to the bimodal nature of the likelihood (Boyles 2001). If this condition were to not hold it would mean that $\theta_{j,+}$ and $\theta_{j,-}$ were classification rates instead of misclassification rates.

The second identifiability assumption ensures that there are enough degrees of freedom to estimate all the parameters. Letting l be the number of inspections per item, we require that $l \geq 3$. This assumption can be relaxed by the use of informative priors as shown in Dendukuri and Joseph (2001).

The third identifiability assumption is that both true negatives ($T_i = 0$) and true positives ($T_i = 1$) exist in the sample. This is because the absence, for instance, of any true negatives means there is no data from which to draw inferences about the false negative rates. Thus, Danila et al. (2012) demonstrate that this absence of either true negatives or true positives essentially means that there is only data to estimate one of the parameters (θ_+ or θ_-).

2.2.2 The Bayesian Approach

We employ the Bayesian approach to parameter estimation, which is fundamentally different from the frequentist approach. Information concerning the model parameters or the lack thereof is incorporated into a prior distribution which then describes our uncertainty concerning the model parameters prior to data analysis. This prior is then updated via Bayes rule with the likelihood to produce a posterior distribution that is then used for inference. For more information on this, and on the elicitation of priors see Robert (2001). Also the last two identifiability requirements can be alleviated through the use of informative priors as discussed in the next subsection.

Due to the flexibility of the beta distribution independent beta priors will be placed on τ, θ_+, and θ_-. Thus, the prior distribution is

$$\begin{aligned} p(\theta_-, \theta_+, \tau) &= p(\theta_-)p(\theta_+)p(\tau) \\ &\propto \theta_-^{\alpha_- - 1}(1-\theta_-)^{\beta_- - 1}\theta_+^{\alpha_+ - 1}(1-\theta_+)^{\beta_+ - 1}\tau^{\alpha_\tau - 1}(1-\tau)^{\beta_\tau - 1}. \end{aligned} \tag{2.13}$$

This results in the joint posterior distribution below.

$$
\begin{aligned}
p(\theta_-, \theta_+, \tau, \mathbf{t}|\xi, \mathbf{x}, N_{pass}) \propto & \tau^{\alpha_\tau - 1}(1-\tau)^{\beta_\tau - 1} \\
& \times (1-\theta_-)^{\sum_{i=1}^{N} t_i x_i + \beta_- - 1} (\theta_-)^{\sum_{i=1}^{N} t_i (l_i - x_i) + \alpha_- - 1} \\
& \times (\theta_+)^{\sum_{i=1}^{N} (1-t_i) x_i + \alpha_+ - 1} (1-\theta_+)^{\sum_{i=1}^{N} (1-t_i)(l_i - x_i) + \beta_+ - 1} \\
& \times \left(\frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)} \right)^{\sum_{i=1}^{N_0} t_i} \\
& \times \left(1 - \frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)} \right)^{N_0 - \sum_{i=1}^{N_0} t_i} \\
& \times \left(\frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+} \right)^{\sum_{i=N_0+1}^{N} t_i} \\
& \times \left(1 - \frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+} \right)^{N_1 - \sum_{i=N_0+1}^{N} t_i} \\
& \times (\tau(1-\theta_-) + (1-\tau)\theta_+)^{N_{pass}} \\
& \times (1 - \tau(1-\theta_-) - (1-\tau)\theta_+)^{N_{pop} - N_{pass}} . \qquad (2.14)
\end{aligned}
$$

It is now standard practice in Bayesian analysis for many problems especially high dimensional ones to turn to Markov Chain Monte Carlo methods to calculate posterior distributions. This particular posterior distribution can be sampled from the Metropolis-Hasting algorithm. The full conditional distributions for the Metropolis-Hastings algorithm are in Appendix A. See Carlin and Louis (2008) for further details on MCMC methods as applied to Bayesian problems.

2.3 Example

We now consider an example analysis of this model complete with the diagnostic plots that indicate convergence of the MCMC simulation. Suppose that 200 electronic items are sampled from previously failed items from a manufacturing process. We are making use of the work in Danila et al. (2010) to justify sampling only from previously failed part. It is known that of the previous 100000 items that have been inspected from this manufacturing process 81887 have passed. However,

no previous assessment of the inspection process has been done so it is decided to use relatively uninformative uniform priors on τ, θ_-, and θ_+. Each part is inspected $l = 11$ times in the experiment. The simulated data for the experiment is in Table

Table 2.1: Data

Passes	Frequency
0	26
1	37
2	24
3	5
4	4
5	0
6	0
7	2
8	3
9	26
10	44
11	29

2.1. The true values are $\theta_- = \theta_+ = .1$ and $\tau = .9$. The observations were the number of instances that that particular number of passes was observed. The program JAGS was used in conjuction with R and the R package rjags to do the MCMC computations. A burn-in of 5000 samples was used followed by a further sampling of 25000 with a thinning of 25. One additional variable kept track of, called *stau*, is the proportion of good parts in the sample. Ideally this value is close to 0.5 (Danila et al. 2008).

Figure 2.1 shows the autocorrelation plots while Figure 2.2 shows the trace and density plots from the MCMC iterations. Both provide evidence that the MCMC process has converged.

A summary of the posterior results is contained in Table 2.2. We see that the true value of the parameter is covered by the 95% credible set in all three cases, and that the proportion of conforming parts in the sample, *stau*, is close to the ideal

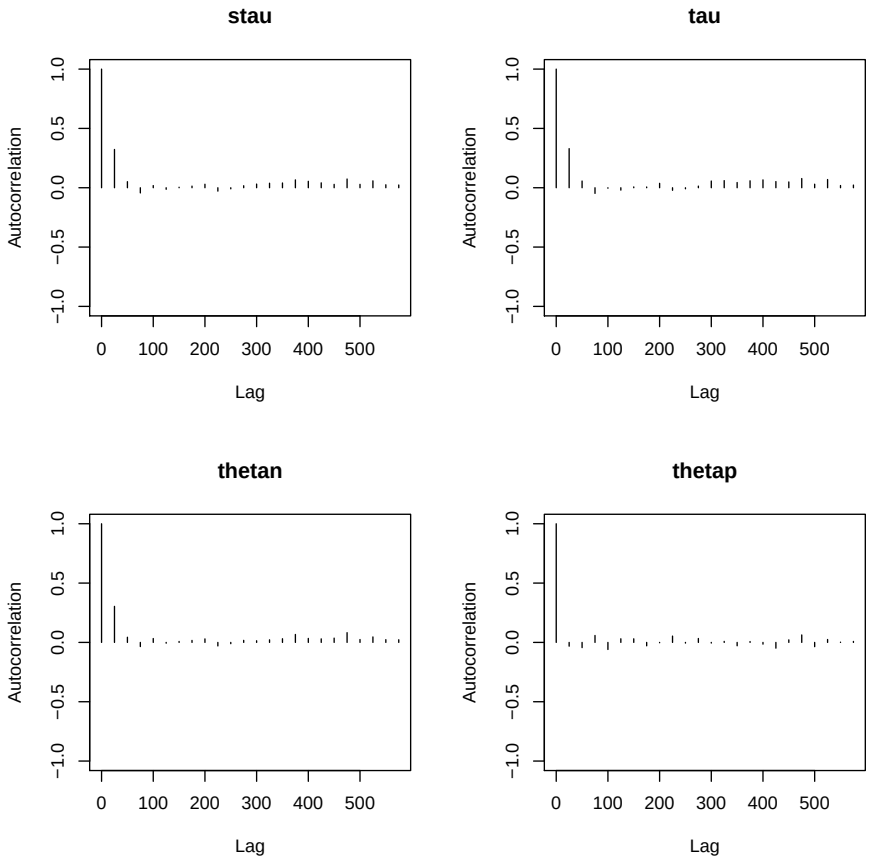

Figure 2.1: Autocorrelation plots of τ, θ_-, θ_+, as well as the proportion of good items in the sample, *stau*.

value of 50%. It can also be seen the estimate of the θ_- parameter is much more precise than that of the θ_+ parameter. This will be remarked on in greater detail in the Section 2.4.

Table 2.2: Posterior Summary

Parameters	Mean	SD	2.5%	50%	97.5%
stau	0.51235	0.02774	0.45847	0.51257	0.56395
tau	0.90067	0.00582	0.88970	0.90093	0.91165
thetan	0.10299	0.00496	0.09363	0.10298	0.11245
thetap	0.11076	0.00951	0.09216	0.11067	0.12943

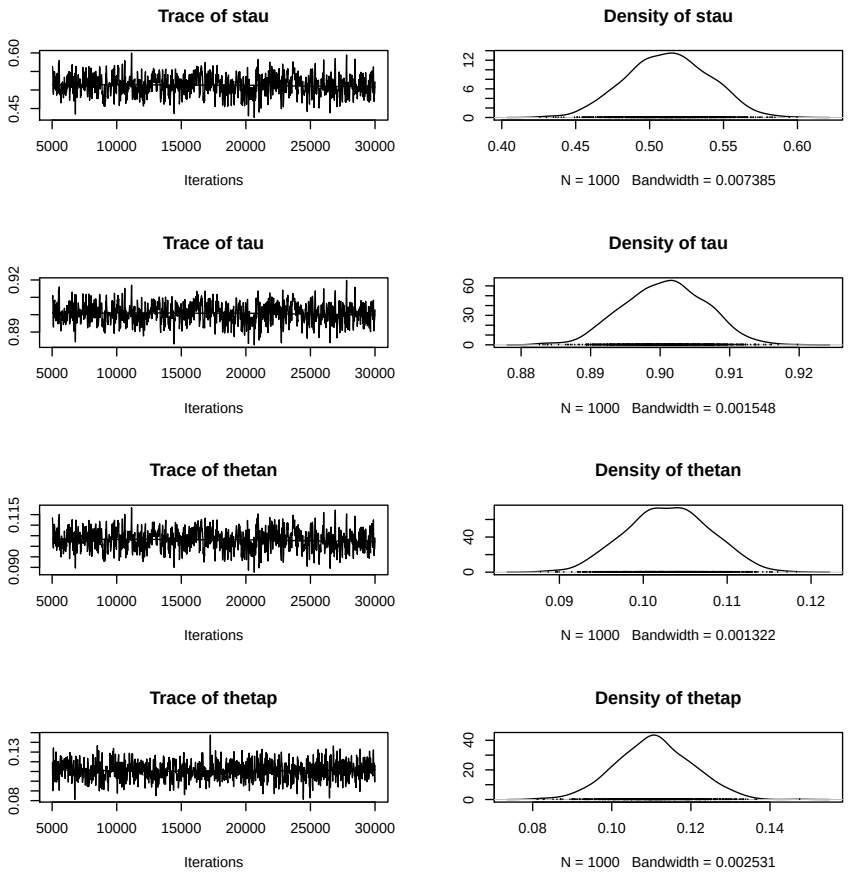

Figure 2.2: Trace and Density plots of τ, θ_-, θ_+, as well as the proportion of good items in the sample, *stau*.

2.4 Simulation Study

A simulation study was done both to demonstrate the operating characteristics of the model, and to demonstrate the importance of including the baseline data.

The parameters varied in the simulation study were τ, θ_+, θ_-, l, N_0, and N_{pop}. Slices (the change of one parameter of interest at a time) of the design space were investigated rather than doing a full factorial design. What might be thought of as the base design point was $\tau = .9$, $\theta_+ = .1$, $\theta_- = .1$, $l = 11$, $N_0 = 200$, $N_1 = 0$, and $N_{pop} = 100000$. All the slices varied one of the parameters while the rest of the parameters were set at their base design point value.

The slices were $\tau \in .7, .71, \cdots, .95$, $\theta_+ \in .01, .02, \cdots, .3$, $\theta_- \in .01, .02, \cdots, .3$, $l \in 7, 8, \cdots, 30$, $N_0 \in 100, 120, \cdots, 400$, and $N_{pop} \in 1000, 10000, \cdots, 10000000$. Additionally, a study involving just the variation of N_1 was done where $N_1 \in 200, 220, \cdots, 600$ and $N_0 = 0$. The higher simulation runs were necessary, in this case, because of the small number of true negatives when sampling from previously passed parts.

The prior distributions for all simulation runs were

$$\theta_+ \sim \mathrm{U}(0,1)$$

$$\theta_- \sim \mathrm{U}(0,1)$$

$$\tau \sim \mathrm{U}(0,1).$$

It was discovered that occasionally, even with good starting values the misclassification rate parameters would instead converge to the classification rates. This occurred about 2% of the time, and these simulations were discarded.

The simulations were done in R using RJAGS to interface with JAGS. The simulations including the baseline data had a sample size of 25000 with a thinning of 25 following a burn-in of 5000, while simulations without baseline data had a sample size of 50000 with a thinning of 25 following a burn-in of 5000. 200 simulations were run per design point.

2.4.1 Simulation Results

Figure 2.3 shows the simulation results for the τ slice. A higher quality production process results in a higher proportion of good items in the sample even when exclusively sampling from previously failed parts (see Equation 2.7). This explains the increasing precision of θ_- estimates and the decreasing precision of θ_+ estimates. Also there is large gain in the precision of the θ_- and τ parameters when using baseline data. This improvement from incorporating the baseline data is observed in all the simulation results. Coverage was good for all the simulation runs. The

95% credible sets covered the true parameter value for τ, θ_+, and θ_- over 90% of the time for every simulation run. Examples of typical convergence plots were included in the previous section (Section 2.3).

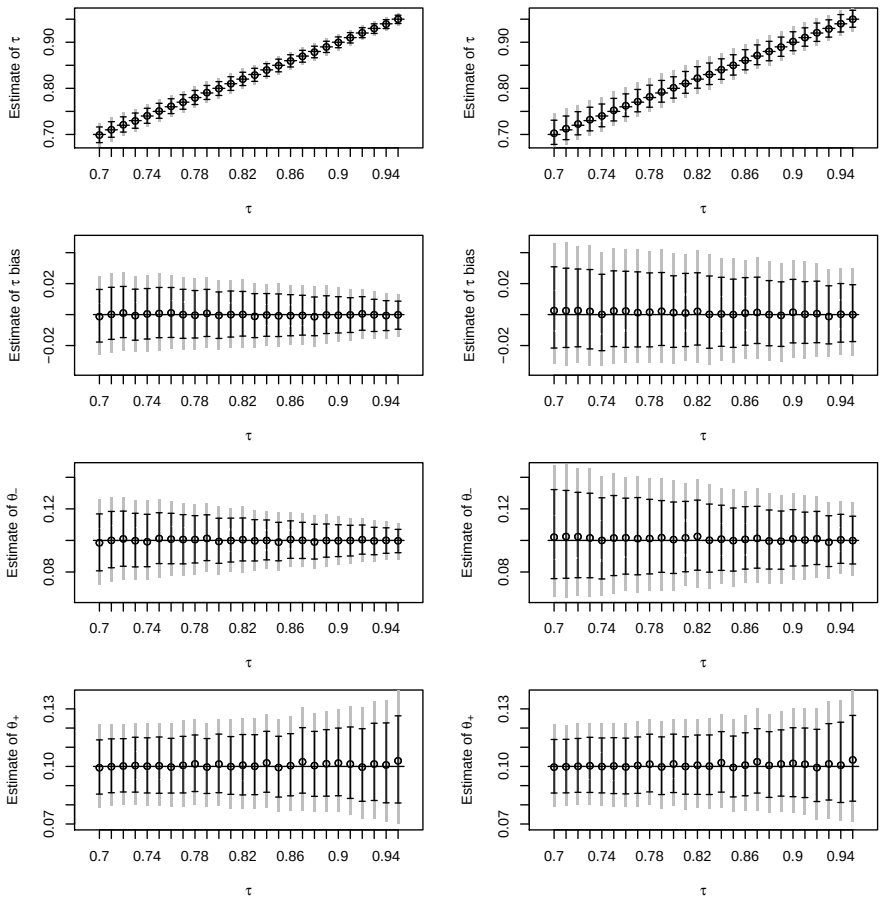

Figure 2.3: Plots summarizing the posterior results as τ changes from .7 to .95. On the left are the results incorporating the baseline data while results on the right do not incorporate baseline data. Top to bottom, the parameters are τ, bias of τ, θ_-, and θ_+.

Figure 2.4 shows the simulation results for the θ_- slice. Some values of θ_- are shown to result in skewed estimates for both θ_- and τ.

Figure 2.5 shows the simulation results for the θ_+ slice. Unlike with θ_-, small values of θ_+ are not shown to result in skewed posteriors.

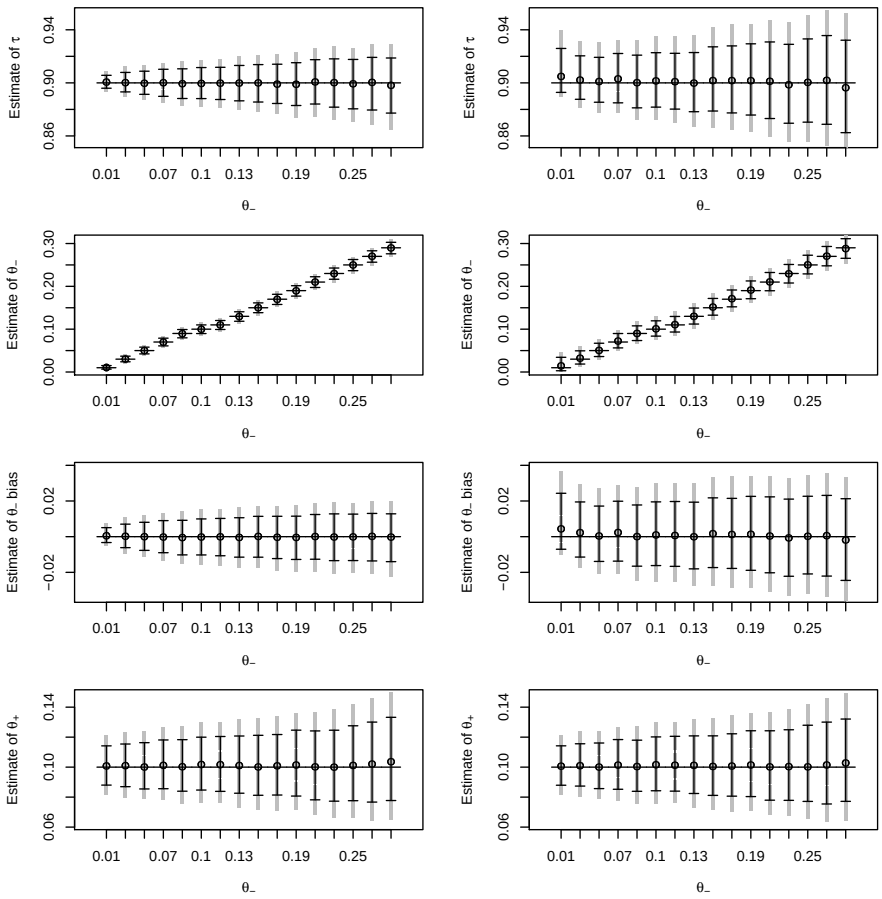

Figure 2.4: Plots summarizing the posterior results as θ_- changes from .01 to .3. On the left are the results incorporating the baseline data while results on the right do not incorporate baseline data. Top to bottom, the parameters are τ, θ_-, bias of θ_-, and θ_+.

Figure 2.6 shows the simulation results for the N_0 slice. As N_0 increases, all estimates get improve. The proportion by which the incorporation of baseline data improves the τ and θ_- estimates appears to remain constant as well.

Figure 2.7 shows the simulation results for the N_{pop} slice. This graphs shows most clearly that when baseline data involving the incorporation of all previously failed parts is included, there is no difference in the estimate of θ_+.

Figure 2.8 shows the simulation results for the l slice. When baseline information is incorporated (and all the parts have previously failed inspection) there is very little increase in the precision of the θ_- and τ estimates as the number of

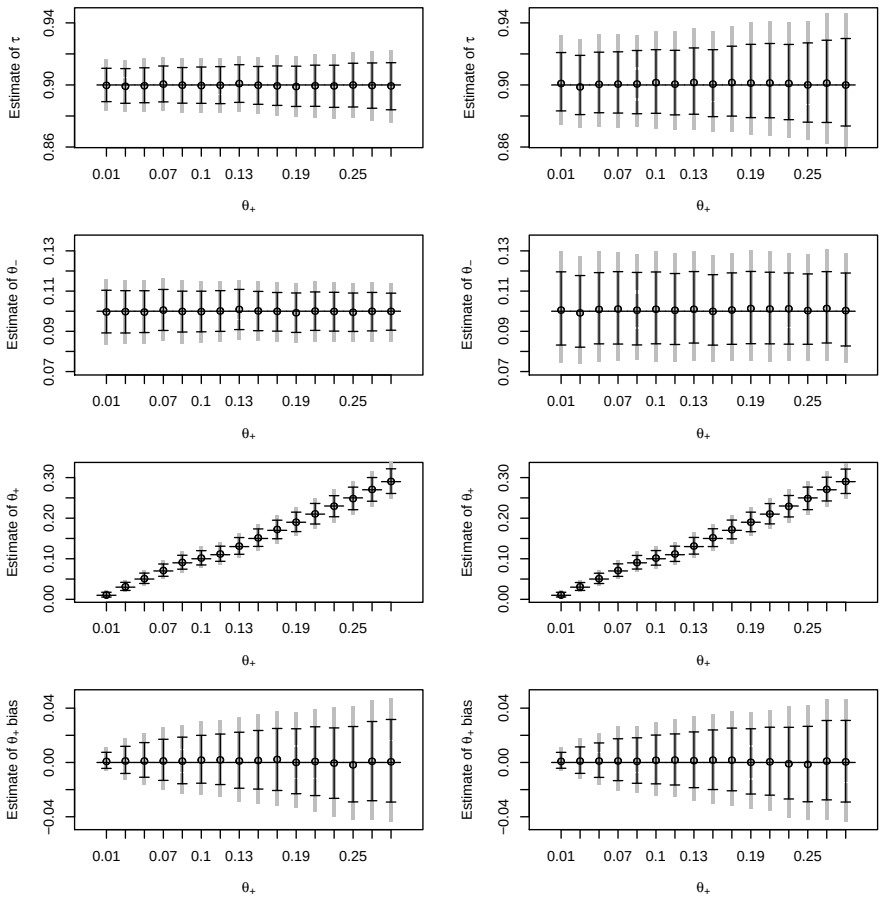

Figure 2.5: Plots summarizing the posterior results as θ_+ changes from .01 to .3. On the left are the results incorporating the baseline data while results on the right do not incorporate baseline data. Top to bottom, the parameters are τ, θ_-, θ_+, and bias of θ_+.

repeated inspections increases from 7 to 30. This is in contrast to the case of not incorporating the baseline data.

Figure 2.9 shows the simulation results for the N_1 slice. As N_1 increases, all estimates get improve. In contrast to sampling from previously failed parts, sampling from previously passed parts is shown to increase the precision of the τ and θ_+ estimates. The large standard deviations of the credible intervals for τ and θ_+ when not incorporating baseline information are the result of very few non conforming parts making it into the sample. This results in a near identifiablity problem which is reflected in the increased uncertainty of the intervals. In practice, one should never

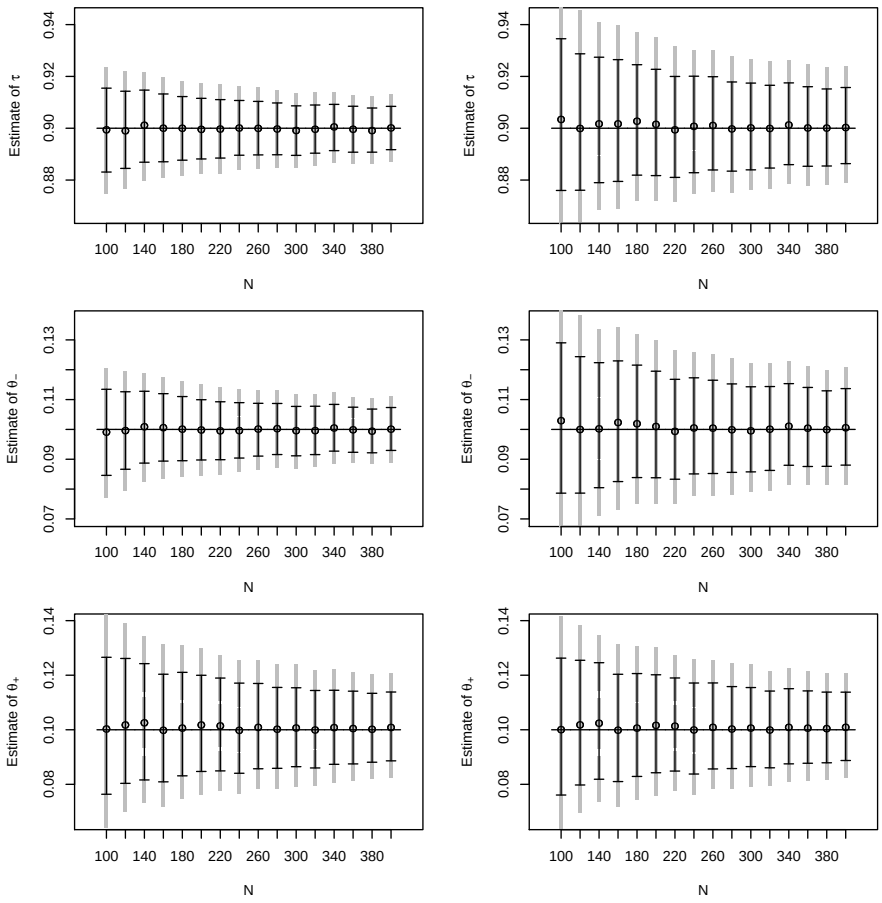

Figure 2.6: Plots summarizing the posterior results as N_0 changes from 100 to 400. On the left are the results incorporating the baseline data while results on the right do not incorporate baseline data. Top to bottom, the parameters are τ, θ_-, and θ_+.

sample from previously passed parts in this case, but it is nonetheless instructive to see that when so done it is the θ_+ parameter and not the θ_- parameter that benefits from the incorporation of baseline information.

2.5 Conclusion

In summary, we have constructed a Bayesian model capable of incorporating baseline data in the analysis of a BMS. We have included a worked example as well as considerable information on the operating characteristics of the model. The importance of the including baseline data when available has been shown repeatedly.

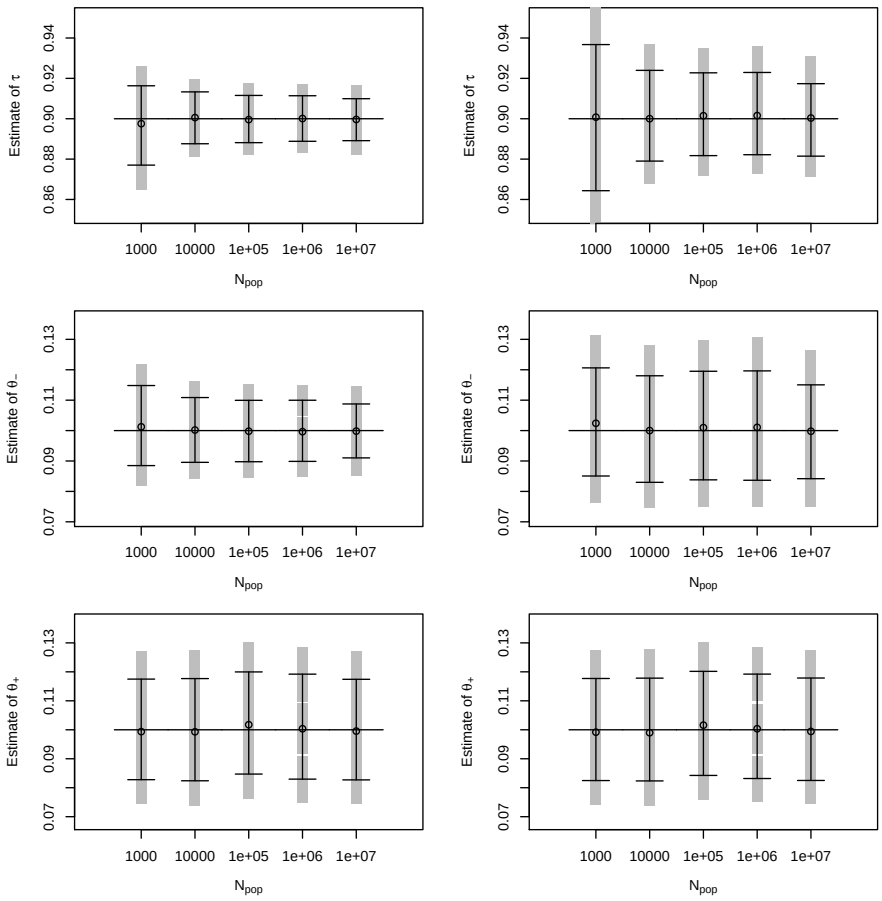

Figure 2.7: Plots summarizing the posterior results as N_{pop} changes from 1000 to 10000000. On the left are the results incorporating the baseline data while results on the right do not incorporate baseline data. Top to bottom, the parameters are τ, θ_-, and θ_+.

While several of the assumptions behind this model may not be met in practice, it is one of the simplest models to implement.

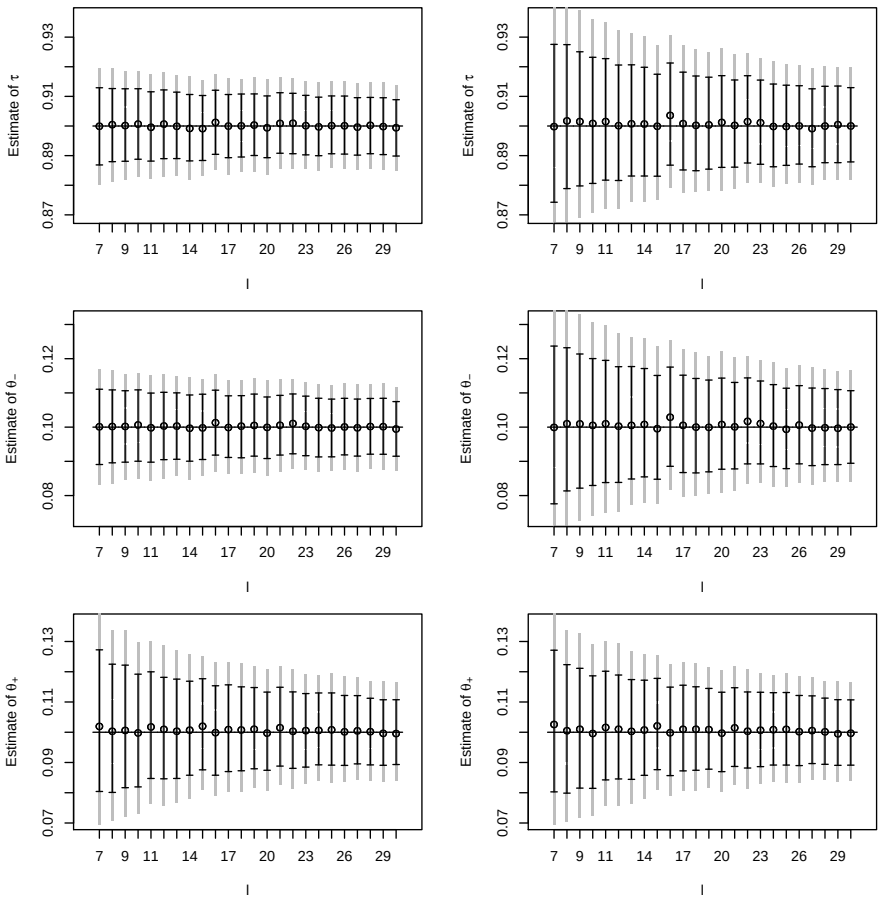

Figure 2.8: Plots summarizing the posterior results as the number of repeated inspections, l, changes from 7 to 30. On the left are the results incorporating the baseline data while results on the right do not incorporate baseline data. Top to bottom, the parameters are τ, θ_-, and θ_+.

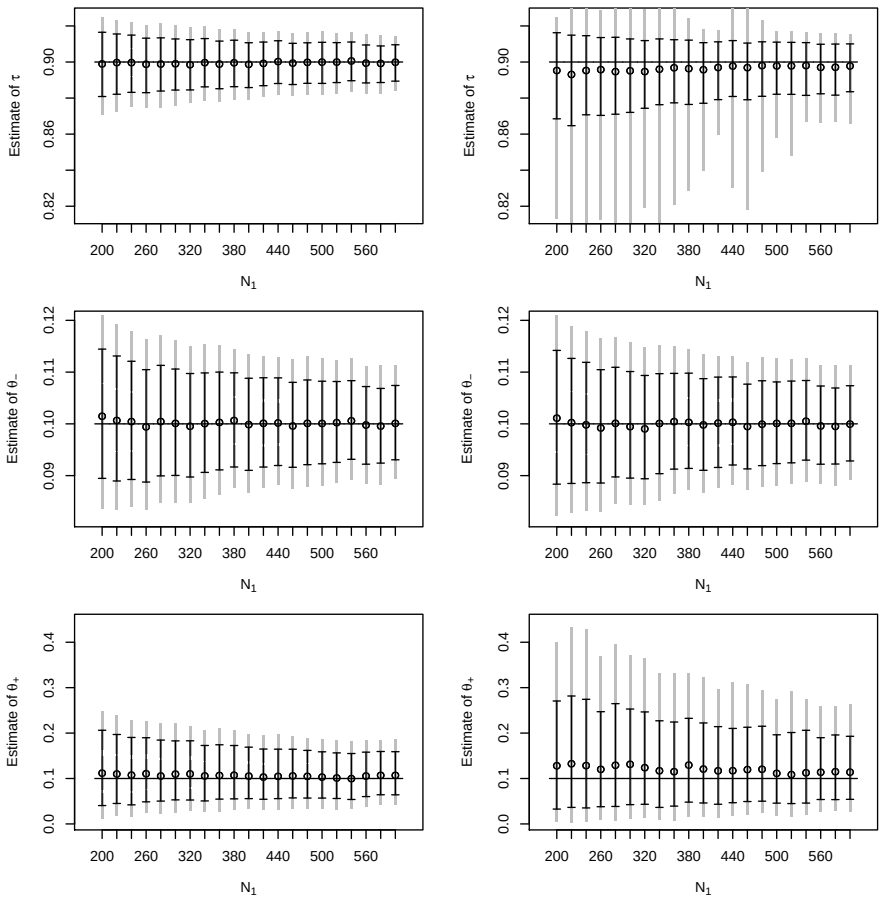

Figure 2.9: Plots summarizing the posterior results as N_1 changes from 200 to 600. On the left are the results incorporating the baseline data while results on the right do not incorporate baseline data. Top to bottom, the parameters are τ, θ_-, and θ_+.

CHAPTER THREE

Estimating Fixed Effects in Multiple Inspector BMS

3.1 Introduction

Repeated binary testing, often called a binary measurement system (BMS), is regularly used in quality control studies as a means of assessing the quality of the production units. When gold standard data is not available or is only available for some units, these inspection methods are highly dependent on the quality of the individual inspectors thus making the study of the inspection process itself an integral part of the quality control process.

One aspect of evaluating the inspection system is the process' repeatability and reproducibility. Its repeatability is how frequently a single inspector inspecting a single item will get the same result while reproducibility is how often different inspectors inspecting the same item will come to the same conclusion.

In the present chapter, the fixed effects model for multiple testers and multiple inspections per inspector is introduced from a Bayesian perspective. This is essentially an extension of Beavers and Stamey (2012). It is also related to the model presented by van Wieringen and de Mast (2008), but from a Bayesian perspective. In general, a fixed effects model treats the observed explanatory variables as if they were non-random. The idea behind using a fixed effects model (as opposed to a random effects model) for the case of multiple raters and multiple inspections is that assessment of individual inspectors is important. That is, we actually care about, for example, how inspector 1 compares to inspector 2. This is in contrast with the case, for instance, where there are so many inspectors that individual assessment is not a concern, but only assessing the population of inspectors is of interest.

There are multiple advantages to using a Bayesian approach. The first is that prior knowledge can be incorporated into the study in a natural way with the use of informative prior distributions. Another is that even in the case of no prior knowledge where the asymptotic dominance of the prior by the likelihood is necessary (or some form of sensitivity analysis is required), the interval estimates generated from the Bayesian paradigm are based largely on the likelihood which has been shown superior to other interval estimation methods for this problem type (Boyles 2001).

The parameter estimates have no known closed form solution in general, and thus must be estimated numerically. Markov Chain Monte Carlo methods are used to sample from the model's posterior distribution. The programs OpenBUGS and R were used to do the MCMC computations necessary for the model implementation.The R package R2WinBUGS was used to interface between them.

The rest of the chapter is organized as follows. Section 3.2 introduces the fixed effects model used to assess a binary measurement system. The operating characteristics of this model are examined through a simulation study in Section 3.3. A particular application, sample size determination, is developed in Section 3.4. We conclude with some comments in Section 3.5.

3.2 The Model

Assume that there are N randomly selected items to be inspected. Let the true quality state of an item to be inspected be denoted by T where $T = 1$ indicates a good item while $T = 0$ denotes an item that fails to meet the quality specifications. The probability τ denotes the overall conforming rate. We assume that no gold standard is used. Thus, T is a latent variable, and we further assume that T is distributed Bernoulli(τ).

Repeated independent and fallible observations are then made by m different inspectors on the i^{th} unit to indirectly assess the true state of the i^{th} unit where $i \in \{1, \cdots, N\}$. Let $Y_{i,j,k}$ be the result of the k^{th} inspection on the i^{th} item by the j^{th} inspector where $k \in \{1, \cdots, n_{i,j}\}$. Also let $Y_{i,j,k} = 1$ be a passed inspection and $Y_{i,j,k} = 0$ be a failed inspection. For each $Y_{i,j,k}$ and inspector j, further define the conditional probabilities $\theta_{j,+} = P(Y_{i,j,k} = 1|T_i = 0)$ (false positive rate) and $\theta_{j,-} = P(Y_{i,j,k} = 0|T_i = 1)$ (false negative rate) with respect to the true state of the item, T_i. Further, assume that the conditional distribution of $Y_{i,j,k}$ is

$$Y_{i,j,k}|T_i = 0 \sim \text{Bernoulli}(\theta_{j,+}),$$
$$Y_{i,j,k}|T_i = 1 \sim \text{Bernoulli}(1 - \theta_{j,-}).$$

Let $X_{i,j} = \sum_{k=1}^{n_{i,j}} Y_{i,j,k}$. Due to the independence of the inspections we have that the conditional distributions $X_{i,j}|T_i = 0 \sim \text{Binomial}(n_{i,j}, \theta_{j,+})$ and similarly $X_{i,j}|T_i = 1 \sim \text{Binomial}(n_{i,j}, 1 - \theta_{j,-})$. Thus the likelihood of the latent vector $\mathbf{t} = \{t_1, \cdots, t_N\}'$ and the observed data matrix $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ where $\mathbf{x}_i = \{x_{i,1}, \cdots, x_{i,m}\}'$ is

$$\begin{aligned} L(\theta_-, \theta_+, \tau|\mathbf{x}, \mathbf{t}) \propto & \tau^{\sum_{i=1}^N t_i}(1-\tau)^{N-\sum_{i=1}^N t_i} \\ & \times \left(\prod_{j=1}^m (1-\theta_{j,-})^{\sum_{i=1}^N t_i x_{ij}} (\theta_{j,-})^{\sum_{i=1}^N t_i (n_{ij}-x_{ij})} \right) \\ & \times \left(\prod_{j=1}^m (\theta_{j,+})^{\sum_{i=1}^N (1-t_i) x_{ij}} (1-\theta_{j,+})^{\sum_{i=1}^N (1-t_i)(n_{ij}-x_{ij})} \right) \end{aligned} \tag{3.1}$$

This is the same likelihood as van Wieringen and de Mast (2008) with the change that the number of repeated inspections is now $n_{i,j}$ instead of the same fixed number.

3.2.1 Identifiability and Additional Model Assumptions

There are several additional model assumptions required for inference, most notably those pertaining to the identifiability of the model. The first is that

$$\theta_{j,+} + \theta_{j,-} < 1. \tag{3.2}$$

This condition essentially says that the chance of correctly classifying an item is greater than the chance of misclassifying it. It is necessary for identifiability due to the bimodal nature of the likelihood (Boyles 2001). If this condition were to not hold it would mean that $\theta_{j,+}$ and $\theta_{j,-}$ were classification rates instead of misclassification rates.

The second identifiability assumption ensures that there are enough degrees of freedom to estimate all the parameters. Letting $l_j = \min(n_{1,j}, \cdots, n_{N,j})$ we have that a sufficient condition for identifiability is that

$$-1 + \prod_{j=1}^{m}(l_j + 1) \geq 2m + 1. \tag{3.3}$$

It is shown in van Wieringen (2005) that when $n_{i,j'} = n_{i,j}$ for all $j \neq j'$ that this is a necessary and sufficient condition for identifiability while in the present model it is sufficient because additional inspections do not hurt the model identifiability.

The third identifiability assumption is that both true negatives ($T_i = 0$) and true positives ($T_i = 1$) exist in the sample. This is because the absence, for instance, of any true negatives means there is no data from which to draw inferences about the false negative rates. Thus, Danila et al. (2012) demonstrate that this absence of either true negatives or true positives essentially means that there is only data to estimate one half of the parameters (θ_+ or θ_-).

An additional note on model assumptions is that the conditional independence assumption has been questioned by several authors as often being unrealistic. In particular, if the underlying measurand is actually a continuous variable that is

dichotomized then de Mast et al. (2011) show that the conditional independence assumption does not hold.

3.2.2 The Bayesian Approach

We employ the Bayesian approach to parameter estimation, which is fundamentally different from the frequentist approach. Information concerning the model parameters or the lack thereof is incorporated into a prior distribution which then describes our uncertainty concerning the model parameters prior to data analysis. This prior is then updated via Bayes rule with the likelihood to produce a posterior distribution that is then used for inference. For more information on this, and on the elicitation of priors see Robert (2001). Also the last two identifiability requirements can be alleviated through the use of informative priors as discussed in the next subsection.

Due to the flexibility of the beta distribution and its conjugacy with the binomial distribution, independent beta priors will be placed on τ, θ_+, and θ_-. Thus, the prior distribution is

$$\begin{aligned} p(\theta_-, \theta_+, \tau) =& p(\tau) \prod_{i=1}^{m} p(\theta_{i-}) p(\theta_{i+}) \\ \propto& \tau^{\alpha_\tau - 1}(1-\tau)^{\beta_\tau - 1} \prod_{i=1}^{m} \theta_{i-}^{\alpha_{i-}-1}(1-\theta_{i-})^{\beta_{i-}-1} \\ & \times \theta_{i+}^{\alpha_{i+}-1}(1-\theta_{i+})^{\beta_{i+}-1}. \end{aligned} \tag{3.4}$$

This results in the joint posterior distribution below.

$$\begin{aligned} f(\theta_-, \theta_+, \tau, \mathbf{t}|\mathbf{x}) \propto& \tau^{\sum_{i=1}^{N} t_i + \alpha_\tau - 1}(1-\tau)^{N - \sum_{i=1}^{N} t_i + \beta_\tau - 1} \\ & \times \left(\prod_{j=1}^{m} (\theta_{j-})^{\sum_{i=1}^{N} t_i (n_{ij} - x_{ij}) + \alpha_{j-} - 1} (1-\theta_{j-})^{\sum_{i=1}^{N} t_i x_{ij} + \beta_{j-} - 1} \right) \\ & \times \left(\prod_{j=1}^{m} (\theta_{j+})^{\sum_{i=1}^{N} (1-t_i) x_{ij} + \alpha_{j+} - 1} (1-\theta_{j+})^{\sum_{i=1}^{N} (1-t_i)(n_{ij} - x_{ij}) + \beta_{j+} - 1} \right). \end{aligned} \tag{3.5}$$

It is now standard practice in Bayesian analysis for many problems especially high dimensional ones to turn to Markov Chain Monte Carlo methods to calculate posterior distributions. This particular posterior distribution can easily be sampled from with a Gibbs sampler, itself a special case of the Metropolis-Hasting algorithm. See Carlin and Louis (2008) for further details on MCMC methods as applied to Bayesian problems. The full conditionals needed to construct a Gibbs sampler are below.

$$f(\theta_{j+}|\theta_-, \mathbf{t}, \mathbf{x}, \tau, \theta_{j\neq k+}) \propto \text{Beta}\left(\sum_{i=1}^{N}(1-t_i)x_{ij} + \alpha_{j+}, \sum_{i=1}^{N}(1-t_i)(n_{ij}-x_{ij}) + \beta_{j+}\right) \tag{3.6}$$

$$f(\theta_{j-}|\theta_+, \mathbf{t}, \mathbf{x}, \tau, \theta_{j\neq k-}) \propto \text{Beta}\left(\sum_{i=1}^{N}t_i(n_{ij}-x_{ij}) + \alpha_{j-}, \sum_{i=1}^{N}t_i x_{ij} + \beta_{j-}\right) \tag{3.7}$$

$$f(\tau|\theta_+, \mathbf{t}, \mathbf{x}, \theta_-) \propto \text{Beta}\left(\sum_{i=1}^{N}t_i + \alpha_\tau, N - \sum_{i=1}^{N}t_i + \beta_\tau\right) \tag{3.8}$$

$$f(t_i|\tau, \theta_+, \mathbf{x}, \theta_-) \propto \text{Bernoulli}\left(\frac{p}{p+q}\right), \tag{3.9}$$

where $p = \tau\prod_{j=1}^{m}(\theta_{j-})^{(n_{ij}-x_{ij})}(1-\theta_{j-})^{x_{ij}}$ and $q = (1-\tau)\prod_{j=1}^{m}(\theta_{j+})^{x_{ij}}(1-\theta_{j+})^{(n_{ij}-x_{ij})}$.

3.3 Simulation Study

A simulation study was conducted to assess the operating characteristics of the model. It is assumed that τ, the proportion of good parts, is not of interest (the interest instead being on repeatability and reproducibility). Thus, by utilizing certain sampling techniques (Danila et al. 2010) it is generally possible to have a sample with a roughly equal number of good and bad parts and so all simulations were done with $\tau = .5$.

The rest of the design parameters utilized in a full factorial design were as follows: $N = 100$, $m \in \{2, 5, 11\}$, $n_{i,j} = n \in \{3, 5, 11\}$, and $\theta_{j-} = \theta_{j+} \in \{.05, .15\}$ for all j. All prior distibutions were beta$(1, 1)$ distributions, both on τ and on all the misclassification parameters. The simulations were done in R using R2OpenBUGS

to interface with OpenBUGS. The simulations had a sample size of 1500 following a burn in of 500. Convergence was checked with the convergence diagnostics available in OpenBUGS (Gelman and Rubin 1992). It was generally found that the fixed effects model converged quite quickly, and thus far fewer MCMC iterations were necessary compared to more complicated models. This was probably due to the asymptotic independence of the parameters (Boyles 2001). 200 simulations were run per design point.

Two smaller scale simulations were also done to investigate the effect that changing the number of items to be inspected, inspectors, and inspections per inspector had on the estimation of a single misclassification rate at low sample sizes. These were 2 factor simulations where m and N were varied. The first simulation was done with $l = 1$, $N \in 20, 30, 40, 60, 100$, and $m \in 3, 4, 5, 6, 8, 11$. The second simulation was done with $l = 3$, $N \in 10, 15, 20, 30, 40, 60$, and $m \in 1, 2, 3, 4, 5, 8$. The same beta$(1, 1)$ prior distributions were used as in the previous simulation.

3.3.1 Simulation Results

All the simulation coverages were good, and are not, therefore, recorded here. Likewise, the MCMC convergence diagnostics looked good for all simulation runs. An example, randomly chosen, of the MCMC simulation diagnostics for these simulation runs is found in Appendix B.2.

We see that when the number of inspectors is increased with $\theta_{j-} = \theta_{j+} = .15$ the posterior credible set width remains nearly the same for all parameters of interest at $N = 100$ in figure 3.1. Only a very slight reduction of the average credible set widths is visible.

As expected, increasing the number of inspections per inspector makes the misclassification estimates more precise as seen in figures 3.2 and 3.3. Interestingly,

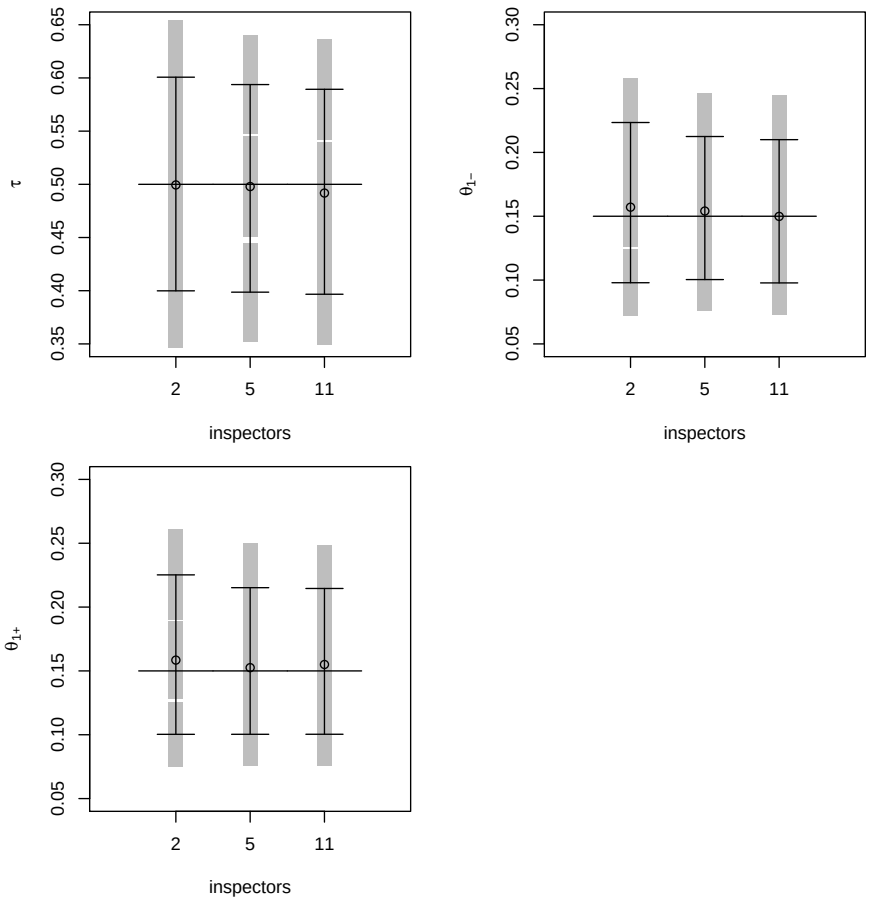

Figure 3.1: Summary of the posterior results for τ and one instance each of θ_{j+} and θ_{j-} parameters as the number of inspectors changes when $\theta_{j+} = \theta_{j-} = .15$.

it's shown that there is a little bias in these estimates at $\theta_{j+} = \theta_{j-} = .05$ which goes away as the number of inspections increases.

At $N = 100$ there was no discernible effect on the precision in the estimates either of the proportion of good items, τ, or of a single inspector's misclassification rates, θ_{j+} or θ_{j-}. The results of two smaller scale simulations to further investigate the effect of increasing the number of inspectors with low sample sizes are shown in Figure 3.4. As can be seen in the contour plots, the effect on the estimation of a single inspector obtained by increasing the total number of inspectors is always minimal.

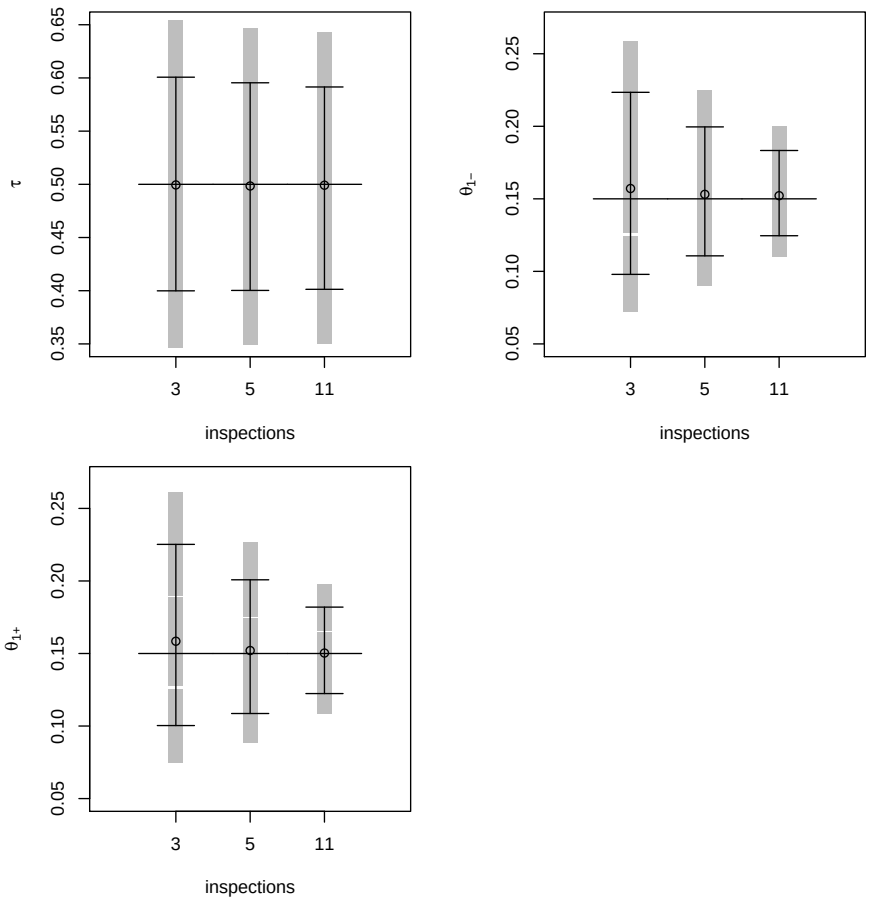

Figure 3.2: Summary of the posterior results for τ and one instance each of θ_{j+} and θ_{j-} parameters as the number of inspections changes when $\theta_{j+} = \theta_{j-} = .15$.

3.4 Sample Size Determination

Appropriately designed experiments are an important aspect of statistics in all fields, and no less so in binary measurement system (BMS) assessment. The current section focuses on a fixed effects model to do this assessment, which implies or should imply that the individual effects (inspectors) are of interest. At the design stage, the knowledge of which comparisons among inspectors are of interest coupled with at least some information about what to expect can be used to estimate an appropriate sample size for the experiment. There is significant literature on determining the appropriate sample size from the Bayesian perspective for binary data. See, for instance, Beavers and Stamey (2012) and Joseph et al. (1997).

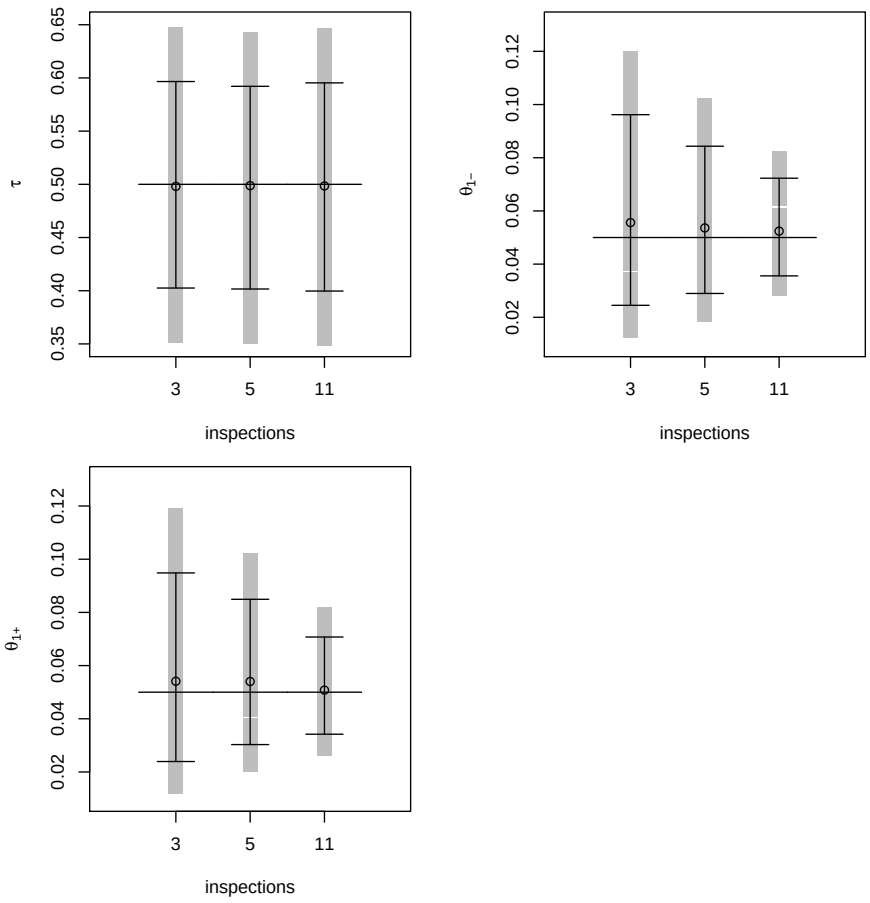

Figure 3.3: Summary of the posterior results for τ and one instance each of θ_{j+} and θ_{j-} parameters as the number of inspections changes when $\theta_{j+} = \theta_{j-} = .05$.

The sample size determination procedure proposed here is a simulation based approach similar to Beavers and Stamey (2012). The basic idea, is that you begin by taking your best guess concerning the results of the experiment and also by providing a set of design points (sample size and number of inspections combinations) to test. You then use this guess to construct "design priors" which are probability distributions that allow for uncertainty at the design phase for each parameter in the model. A simulation is then done at each design point by repeatedly taking samples from the design priors, generating data, fitting the model, and seeing how many times the desired experimental result is met. These results are summarized per design point, and then the optimal design point is picked.

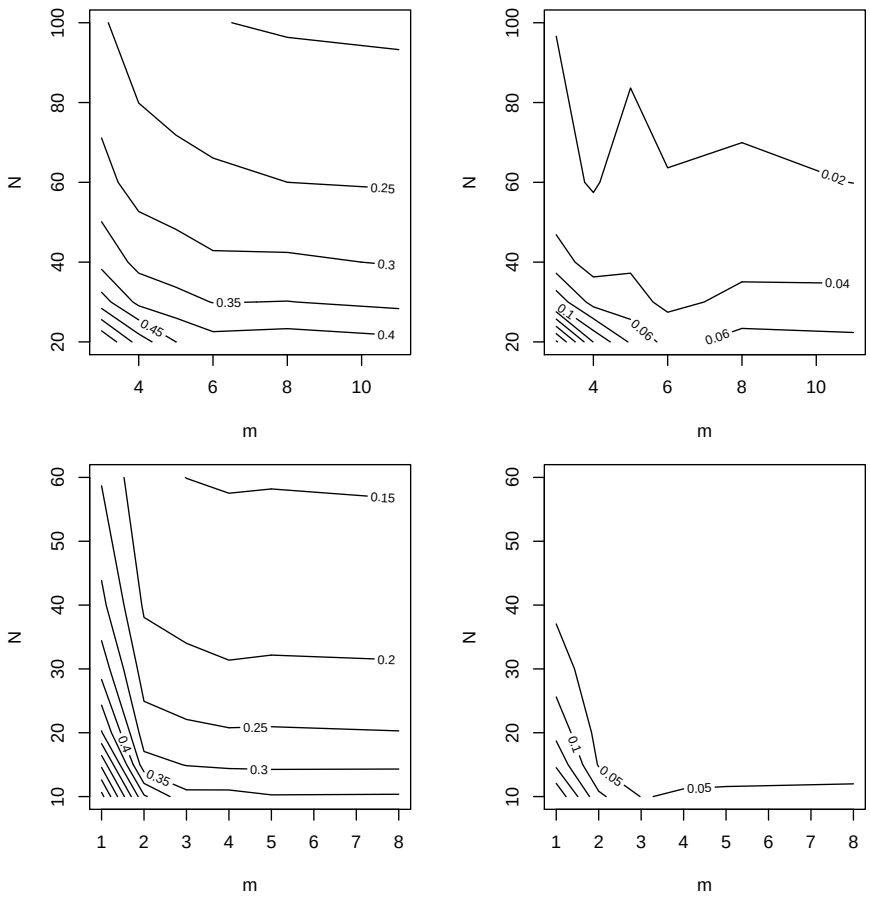

Figure 3.4: Contour plots of the average bias and credible set width of a single misclassification rate for varying values of m and N. Plots on top are of $l = 1$ while those on the bottom are of $l = 3$. Plots on the right are of average bias while those on the those on the left are of average 95% credible interval width.

Three inputs are required from the practitioner: a design prior, an analysis prior, and a grid of potential experimental design conditions (e.g. sample size). The design priors and analysis priors share similarities in that both are probability distributions that describe the a priori knowledge of the parameters. However, the design priors tend to be informative distributions often entailing the eliciting of expert opinion or derived using regulatory requirements, while frequently the analysis priors are more objective (often relatively uninformative priors). The experimental design conditions considered here are variations in the sample size, N, and the number of inspections, l. Conceivably the number of inspectors, m, could also be

variable, but that case is not considered here because it is assumed that using a fixed effects model implies an interest in particular inspectors, and thus probably a fixed number of them.

The simulation begins by generating parameter values of τ, θ_+, and θ_- from the design priors. Denote the total number of sets of parameters generated this way as B. For each parameter set, data are generated and posterior quantities of interest calculated using the model with the analysis priors. These quantities of interest are recorded for each set of parameters used. They can then be averaged over the B data sets to get a summary statistic for the design point.

There are a number of potential quantities of interest for the simulation Wang and Gelfand (2002). We utilized power as a criteria. However, the interest could also be in, for instance, the average posterior variance criterion (APVC) or the average coverage criterion (ACC). There are other quantities that may be of interest also like posterior probabilities of various sorts.

The last step is to use the simulation output to determine the sample size and number of inspections required for the criterion of interest. This can be done graphically via a contour plot in the case that both the sample size N and the number of inspections l were allowed to vary. It can also be done numerically, after defining an appropriate loss function to describe the preference for additional items versus additional inspections.

We summarize the method in the following algorithm:

(1) Determine the quantity or comparison of interest (e.g. is the posterior probability that inspector 1 has a lower false positive misclassification rate than inspector 2 at least 95 percent.)

(2) Elicit design priors for all parameters in the model.

(3) Create a grid of design points (i.e. N and l combinations) to be considered.

(4) For each design point, simulate parameters from the design priors.

(5) From these parameters, simulate data for the model.

(6) Fit the model with this data using the analysis priors via MCMC.

(7) Record the quantity or comparison of interest.

(8) Repeat steps 4 through 7 B times to ensure a sufficient sampling of the quantity of interest has been made at each design point (e.g. a couple hundred samples of the quantity of interest).

(9) Compute the relevant summary statistic(s) for each design point.

(10) Perform a search through the design point grid and find the optimal sample sizes and number of inspectors.

For each design point, care must be taken to carefully consider both the number of MCMC iterations to be performed in fitting the model and the number datasets generated at that design point. The number of MCMC iterations must be large enough to ensure convergence during the burn-in, and be a large enough sample of the posterior during the sampling phase. There are a number of built in tools in both OpenBUGS and the R2WinBUGS program (Gelman and Rubin 1992) to help determine this.

The choice of B is more subjective and is often a balancing act between accuracy and time to complete the simulation. In general, for continuous posterior quantities fewer simulations are required than for categorical outcomes Beavers et al. (2011). For example, interval widths do not need as many iterations as power in order to obtain a reasonable approximation.

3.4.1 Example

As an example, consider the situation in an industrial setting where you have 3 inspectors to evaluate a manufacturing process. We are interested in comparing the accuracy of the three inspectors.

It is believed that the mean false positive misclassification rates of the three inspectors are .05, .1, and .2 respectively. It is also believed that the 75^{th} percentiles for the three inspectors' misclassification rates are 0.069, 0.1251, and 0.2287 respectively. This information can be used to obtain a beta distribution that corresponds to the expert beliefs. See Robert (2001) for more information on prior elicitation. Using beta distributions for these priors, this information is sufficient to find that the design priors are

$$\theta_{1-} \sim \text{Beta}(1.5, 28.5)$$

$$\theta_{2-} \sim \text{Beta}(5, 45)$$

$$\theta_{3-} \sim \text{Beta}(16, 64).$$

The false negative misclassification rates were thought to be the same as the false positive misclassification rates, and thus had the same prior distributions. A plot of the false positive misclassification design priors can be seen in Figure 3.5.

It is assumed that half of the parts were taken from previously rejected parts and the other half from passed parts. Thus, it is thought that for the sample the proportion of good parts should be about 0.5. It is also thought with 95% probability that the proportion should be between 0.2886 and 0.7114. This information is enough to determine that an appropriate prior for the τ parameter is a beta$(10, 10)$.

One implication of this set of design priors is that we think a priori that $P(\theta_{1+} < \theta_{3+}) = 0.9883$. We also think a priori that $P(\theta_{1+} < \theta_{2+}) = 0.827$. These a priori probabilities are important because as the sample size and number of in-

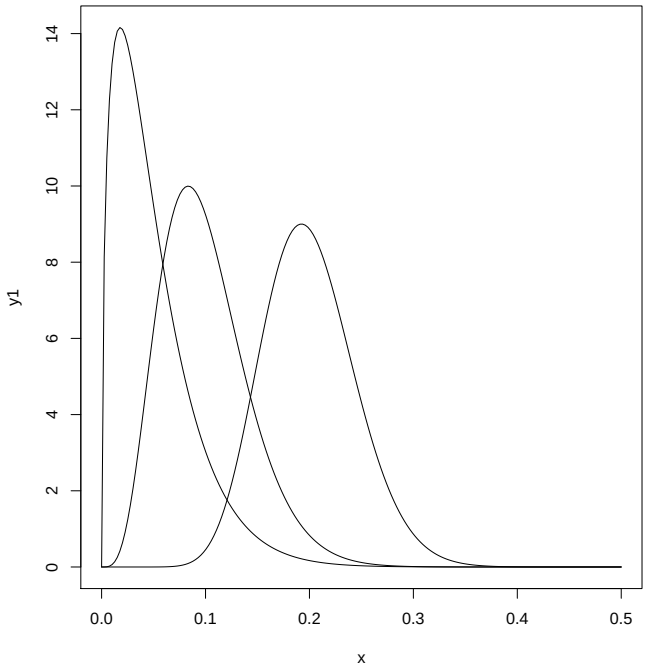

Figure 3.5: The Design Prior for the false positive rates.

spections increase, these are maximum values that the power can attain. See also O'Hagan et al. (2005) for more examples of Bayesian assurance.

Continuing this example, the design grid selected was $l \in 1, 2, 4, 6, 8, 10$ and $N \in 40, 80, 120, 160, 200$. Other simulation design parameters were $B = 200$, and a MCMC sample of size 2500 with a thinning of 5 following a burnin of 500. This model, like all the fixed effects models in this chapel, achieved MCMC convergence quite quickly.

Simulation results are in Figures 3.6 and 3.7. Figure 3.6 shows the posterior probability that $\theta_{1+} < \theta_{2+}$ while Figure 3.7 shows the posterior probability that $\theta_{1+} < \theta_{3+}$. In both figures, a strong increase in the desired posterior probability can especially be observed as l increases to 3.

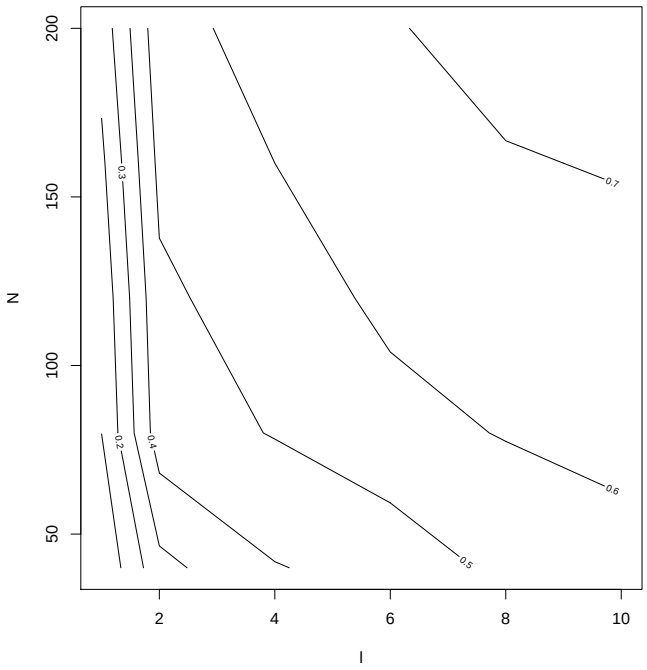

Figure 3.6: The 95% posterior probability that $\theta_{1+} < \theta_{2+}$.

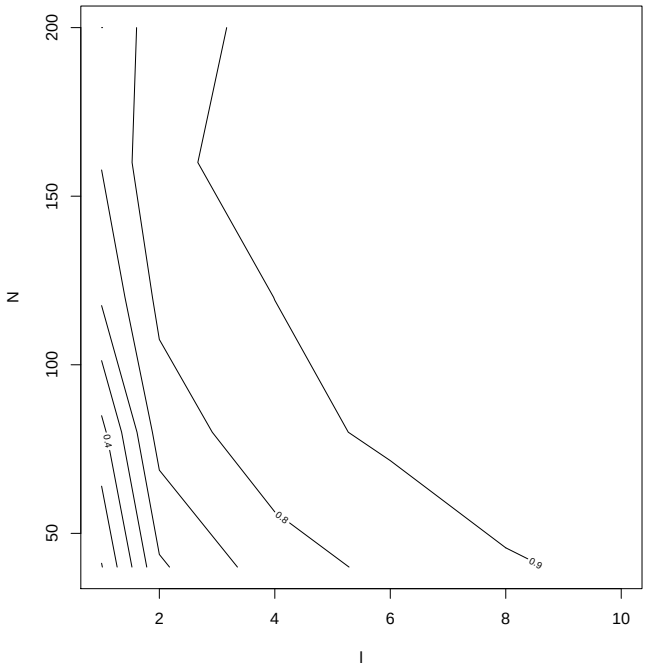

Figure 3.7: The 95% posterior probability that $\theta_{1+} < \theta_{3+}$.

3.5 Conclusion

We have constructed a Bayesian model capable of analyzing a R&R experiment with multiple testers and multiple inspections per inspector. While several of the assumptions behind this model may not be met in practice, it is one of the simplest models to implement. We have also provided and demonstrated a sample size determination algorithm for this model, which is very useful in an experimental design setting.

CHAPTER FOUR

Estimating Random Effects in Multiple Inspector BMS

4.1 Introduction

Repeated binary testing often called a binary measurement system (BMS) is regularly used in quality control studies as a means of assessing the quality of the production units. These inspection methods are, however, highly dependent on the quality of the individual inspectors thus making the study of the inspection process itself an integral part of the quality control process.

One aspect of evaluating the inspection process is the repeatability and reproducibility. Its repeatability is how frequently a single inspector inspecting a single item will get the same result while reproducibility is how often different inspectors inspecting the same item will come to the same conclusion.

In the present chapter, the random effects model for multiple testers and multiple inspections per inspector is introduced from a Bayesian perspective. This is an extension of Beavers and Stamey (2012). The idea behind using a random effects model (as opposed to a fixed effects model) for the case of multiple raters and multiple inspections is that assessment of individual inspectors is unimportant. In general, this requires a fairly large number of different inspectors, both from a conceptual standpoint so that we don't care about the individual inspectors and also from an estimation standpoint as more inspectors are needed to estimate parameters in a random effects model.

There are multiple advantages to using a Bayesian approach. The first is that prior knowledge can be incorporated into the study in a natural way with the use of informative prior distributions. Another is that even in the case of no prior knowledge where the asymptotic dominance of the prior by the likelihood is

necessary (or some form of sensitivity analysis is required), the interval estimates generated from the Bayesian paradigm are based largely on the likelihood which has been shown superior to other interval estimation methods for this problem type (Boyles 2001).

The estimates have no known closed form solution in general, and thus must be solved approximately. Markov Chain Monte Carlo simulations were done to sample from the model's posterior distribution. The programs JAGS and R together with the R package RJAGS to interface between them were used for the MCMC computations necessary for the model implementation.

Two slightly different models will be developed in this chapter. The first assumes that the sampling is from the general population while the second allows for sampling from previously inspected parts (either passed or failed). After each model is presented a simulation study will follow in the subsection.

555543

4.2 The Model

Assume that there are N randomly selected items to be inspected sampled from the general population of items. Let the true quality state of an item to be inspected be denoted by T where $T = 1$ indicates a good item while $T = 0$ denotes an item that fails to meet the quality specifications. The probability τ denotes this overall conforming rate. We assume that no gold standard is used. Thus, T is a latent variable, and we further assume that T is distributed Bernoulli(τ).

Repeated independent and fallible observations are then made by m different inspectors on the i^{th} unit to indirectly assess the true state of the i^{th} unit where $i \in \{1, \cdots, N\}$. Let $Y_{i,j,k}$ be the result of the k^{th} inspection on the i^{th} item by the j^{th} inspector where $k \in \{1, \cdots, n_{i,j}\}$. Also let $Y_{i,j,k} = 1$ be a passed inspection and $Y_{i,j,k} = 0$ be a failed inspection. For each $Y_{i,j,k}$ and inspector j, further define

the conditional probabilities $\theta_{j,+} = P(Y_{i,j,k} = 1|T_i = 0)$ (false positive rate) and $\theta_{j,-} = P(Y_{i,j,k} = 0|T_i = 1)$ (false negative rate) with respect to the true state of the item, T_i. Further, assume that the conditional distribution of $Y_{i,j,k}$ is

$$Y_{i,j,k}|T_i = 0 \sim \text{Bernoulli}(\theta_{j,+}),$$
$$Y_{i,j,k}|T_i = 1 \sim \text{Bernoulli}(1 - \theta_{j,-}).$$

We are assuming here that inspections are independent given the true latent state of the part. Thus, for instance, $\sum_{k=1}^{l} Y_{i,j,k}|T_i = 0 \sim \text{Binomial}(l, \theta_{j,+})$. This is known as the conditional independence assumption, and has been questioned by several authors as an often unrealistic assumption. In particular, if the underlying measurand is actually a continuous variable that is dichotomized then de Mast et al. (2011) show that the conditional independence assumption does not hold.

We also assume for the random effects model that

$$\theta_{j,+} \sim \text{Beta}(\mu_+, \gamma_+),$$
$$\theta_{j,-} \sim \text{Beta}(\mu_-, \gamma_-),$$

where the Beta distribution has been reparameterized as $\mu = \alpha/(\alpha + \beta)$ and $\gamma = \alpha + \beta$. Thus, the pdf of the reparameterized beta distribution is

$$f(x) = \frac{x^{\mu\gamma-1}(1-x)^{\gamma-\mu\gamma-1}}{\text{B}(\mu\gamma,\gamma\text{-}\mu\gamma)}. \tag{4.1}$$

A beta distribution has been chosen here to model the random effects because of its flexibility, and because, under this reparameterization, of its interpretability with the mean being μ and γ being approximately inversely proportional to the variance. More precisely,

$$\sigma^2 = \frac{\mu(1-\mu)}{1+\gamma}. \tag{4.2}$$

Noting that the maximum possible variance of a beta distribution given its mean is $\mu(1 - \mu)$ (also the variance of a Bernoulli trial) gives the interpretation to the

γ parameter, that $\gamma + 1$ is the reciprocal of the variance divided by the maximum variance given the mean.

The usual alternative model structure to what is given here is

$$\text{logit}(\theta) \sim \text{N}(\mu, \sigma). \tag{4.3}$$

Equation 4.3 is not used in this chapter due to the interpretability possible from the reparameterized beta distribution (Equation 4.1).

Let $X_i = \sum_{j=1}^{l_i} Y_{i,j}$. On account of the conditional independence of the inspections we have that the conditional distributions $X_i|T_i = 0 \sim \text{Binomial}(l_i, \theta_+)$ and similarly $X_i|T_i = 1 \sim \text{Binomial}(l_i, 1 - \theta_-)$. Thus the likelihood of the latent vector $\mathbf{t} = \{t_1, \cdots, t_N\}'$ the observed data matrix $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ where $\mathbf{x}_i = \{x_{i,1}, \cdots, x_{i,m}\}'$, and the random effects parameters μ and γ is

$$\begin{aligned}
L(\mathbf{x}, \mathbf{t}, \theta_-, \theta_+|\mu_+, \gamma_+, \mu_-, \gamma_-, \tau) =& L(\mathbf{x}|\theta_-, \theta_+, \mathbf{t}) L(\theta_+|\mu_+, \gamma_+) L(\theta_-|\mu_-, \gamma_-) L(\mathbf{t}|\tau) \\
\propto& \tau^{\sum_{i=1}^N t_i} (1-\tau)^{N - \sum_{i=1}^N t_i} \\
&\times \left(\prod_{j=1}^m (1-\theta_{j-})^{\sum_{i=1}^N t_i x_{ij} + \mu_- \gamma_- - 1} \right. \\
&\left. \times (\theta_{j-})^{\sum_{i=1}^N t_i (n_{ij} - x_{ij}) + \gamma_- - \mu_- \gamma_- - 1} \right) \\
&\times \left(\prod_{j=1}^m (\theta_{j+})^{\sum_{i=1}^N (1-t_i) x_{ij} + \mu_+ \gamma_+ - 1} \right. \\
&\left. \times (1-\theta_{j+})^{\sum_{i=1}^N (1-t_i)(n_{ij} - x_{ij}) + \gamma_+ - \mu_+ \gamma_+ - 1} \right).
\end{aligned} \tag{4.4}$$

4.2.1 Identifiability

For the random effects model, the first assumption necessary for identifiability (Danila et al. 2012) is that

$$\mu_+ + \mu_- < 1. \tag{4.5}$$

The interpretation of this condition is that it says that the overall average chance of correctly classifying an item is greater than the chance of misclassifying it. It is

necessary for identifiability due to the bimodal nature of the likelihood (Boyles 2001). If this condition were to not hold it would mean that what are being modeled as misclassification rates should instead be reversed and modeled as classification rates.

The second identifiability assumption ensures that there are enough degrees of freedom to estimate all the variables. This requires two things. It requires enough inspectors and inspections per inspector to estimate the status of each item, and it requires enough inspectors to estimate the inspectors' random effects parameters. The second condition requires at least two inspectors while letting $l_j = \min(n_{1,j}, \cdots, n_{N,j})$, a sufficient condition to meet the first requirement is that we have

$$-1 + \prod_{j=1}^{m}(l_j + 1) \geq 2m + 1. \tag{4.6}$$

It is shown in van Wieringen (2005) that when $n_{i,j'} = n_{i,j}$ for all $j \neq j'$ that this is a necessary and sufficient condition for identifiability while in the present model it is sufficient because additional inspections do not hurt the model identifiability.

The third identifiability assumption is that both true negatives ($T_i = 0$) and true positives ($T_i = 1$) exist in the sample. This is because the absence, for instance, of any true negatives means there is no data from which to draw inferences about the false negative rates. Thus, Danila et al. (2012) demonstrate that this absence of either true negatives or true positives essentially means that there is only data to estimate one half of the variables (θ_+, μ_+, and γ_+ or θ_-, μ_-, and γ_-).

4.2.2 Independence Assumptions and the Bayesian Approach

We employ a Bayesian approach for inference. Information concerning the model parameters or the lack thereof is incorporated into a prior distribution which describes our uncertainty concerning the model parameters prior to data analysis. The prior distribution is then updated via Bayes rule with the likelihood to produce what is called the posterior distribution, that is then used for inference. For more

information on this, and on the elicitation of priors see Robert (2001). Also the last two identifiability requirements can be alleviated through the use of informative priors as discussed in the next subsection.

The Bayesian approach requires prior distributions for all unknown parameters. This necessitates a determination of the independence assumptions to justify and identify where independent priors can be used and where they can not be used.

The first independence assumption is that the underlying latent goodness of the items to be inspected is independent of the inspection methodology. This implies that the prior on τ will be independent from all the other prior distributions.

The second independence assumption is that the misclassification rates given a bad part are independent of the misclassification rates given a good part. This assumption may be justified by the claim that there exists a true dichotomy between the good and bad parts (the classic example is a pregnancy test: a woman is either pregnant or she's not). This true dichotomy is implied by the conditional independence assumption (introduced in the model section). Mathematically, this second assumption implies that the priors on μ_+ and γ_+ will be independent of the prior distributions placed on μ_- and γ_-.

Lastly, independent priors will be placed on μ_+ and γ_+ and likewise on μ_- and γ_-. This is done because the μ parameter is essentially a location parameter while the γ parameter in the reparameterized beta distribution is essentially a measure of variability. Thus, the general form of the prior distribution will be

$$p(\mu_+, \gamma_+, \mu_-, \gamma_-, \tau) = p(\mu_+)p(\gamma_+)p(\mu_-)p(\gamma_-)p(\tau).$$

We will use uniform$(0, 1)$ prior distributions on τ and on both μ parameters while using uniform$(0, 50)$ distributions on the γ parameters. Thus, the prior distribution we will be using is

$$p(\mu_+, \gamma_+, \mu_-, \gamma_-, \tau) = \frac{1}{2500}.$$

This prior together with the likelihood previously developed yields the following posterior distribution.

$$\begin{aligned} p(\mathbf{t}, \theta_-, \theta_+, \mu_+, \gamma_+, \mu_-, \gamma_-, \tau|\mathbf{x}) \propto & L(\mu_+, \gamma_+|\theta_+) L(\mu_-, \gamma_-|\theta_-) L(\mathbf{t}|\tau) L(\theta_-, \theta_+, \mathbf{t}|\mathbf{x}) \\ & \times p(\mu_+, \gamma_+, \mu_-, \gamma_-, \tau)/f(\mathbf{x}) \qquad (4.7) \\ \propto & \tau^{\sum_{i=1}^N t_i} (1-\tau)^{N-\sum_{i=1}^N t_i} \\ & \times \left(\prod_{j=1}^m (1-\theta_{j-})^{\sum_{i=1}^N t_i x_{ij} + \mu_- \gamma_- - 1} \right. \\ & \left. \times (\theta_{j-})^{\sum_{i=1}^N t_i (n_{ij} - x_{ij}) + \gamma_- - \mu_- \gamma_- - 1} \right) \\ & \times \left(\prod_{j=1}^m (\theta_{j+})^{\sum_{i=1}^N (1-t_i) x_{ij} + \mu_+ \gamma_+ - 1} \right. \\ & \left. \times (1-\theta_{j+})^{\sum_{i=1}^N (1-t_i)(n_{ij} - x_{ij}) + \gamma_+ - \mu_+ \gamma_+ - 1} \right). \qquad (4.8) \end{aligned}$$

4.3 Example

We now consider an example analysis of this model complete with the diagnostic plots that indicate convergence of the MCMC simulation. Suppose that 200 electronic items are sampled from previously failed items from a manufacturing process. No previous assessment of the inspection process has been done, so it is decided to use relatively uninformative uniform priors on γ_-, γ_+, μ_-, μ_+, and τ. Each part is inspected $l = 1$ time by $m = 20$ different inspectors in the experiment.

Simulated data is used for the experiment. The true values are $\gamma_- = \gamma_+ = 15$, $\mu_- = \mu_+ = .1$ and $\tau = .8$. The observations were the number of instances that that particular number of passes was observed. Data generation is done with the R code in Appendix C.2, while the R code for this experiment including the random seed used can be found in Appendix C.3. The program JAGS was used in conjuction with R and the R package Rjags to do the MCMC computations. A burn-in of 5000 samples was used followed by a further sampling of 25000 with a thinning

of 25. Four additional variables were are kept track of, called $thetan[1:2]$ and $thetap[1:2]$. These were arbitrarily chosen inspector misclassification rates.

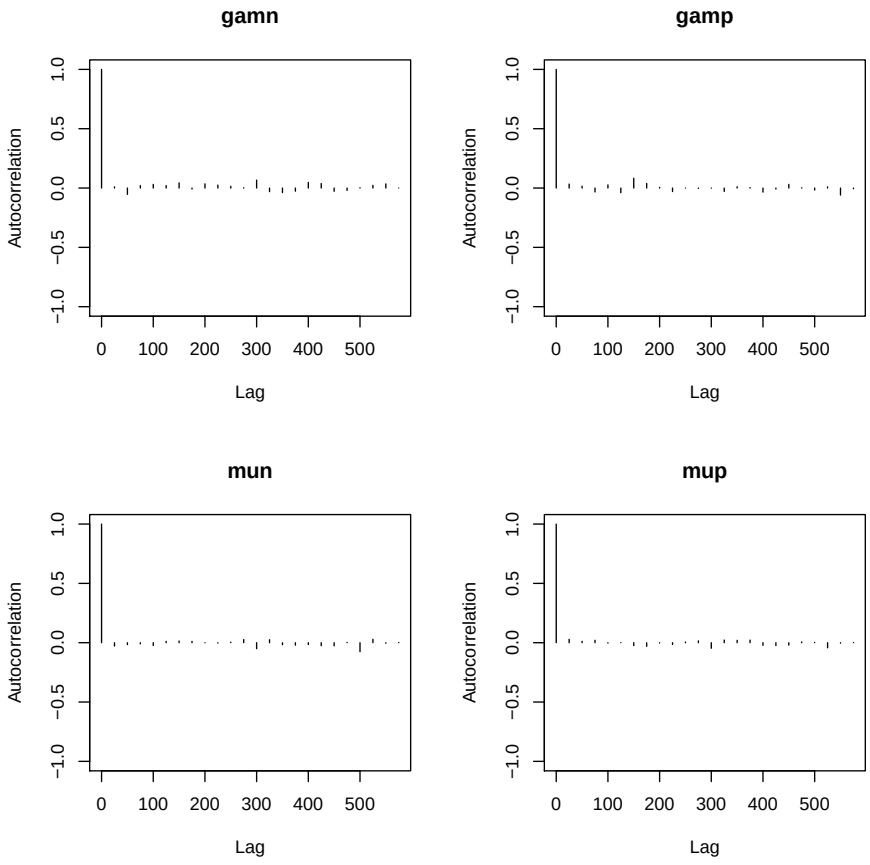

Figure 4.1: Autocorrelation plots of γ_-, γ_+, μ_-, and μ_+.

Figures 4.1 and 4.2 show the autocorrelation plots. Figures 4.3 and 4.4 show the trace and density plots from the MCMC iterations. Both sets of plots provide evidence that the MCMC process has converged.

A summary of the posterior results is contained in Table 4.1. We see that the true value of the parameter is covered by the 95% credible set in all cases.

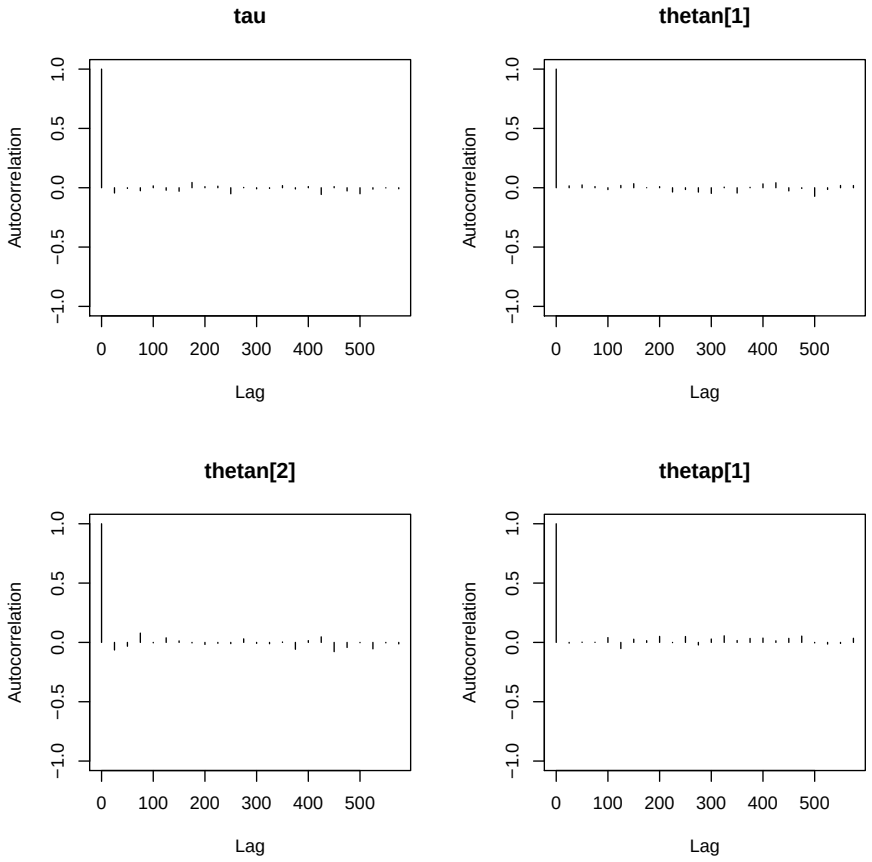

Figure 4.2: Autocorrelation plots of τ and three arbitrarily chosen misclassification rates.

4.4 Simulation Study

A simulation study was conducted to assess the operating characteristics of the model. Because the sample is assumed to be drawn from the general population for this model, τ, the proportion of good parts is of interest along with the parameters for the repeatability and reproducibility.

The parameters varied in the simulation study were τ, μ_+, μ_-, γ_+, γ_-, m, l, and N. Slices (the change of one parameter of interest at a time) of the design space were investigated rather than doing a full factorial design. What might be thought of as the base design point was $\tau = .8$, $\mu_+ = \mu_- = .1$, $\gamma_+ = \gamma_- = 15$, $m = 20$,

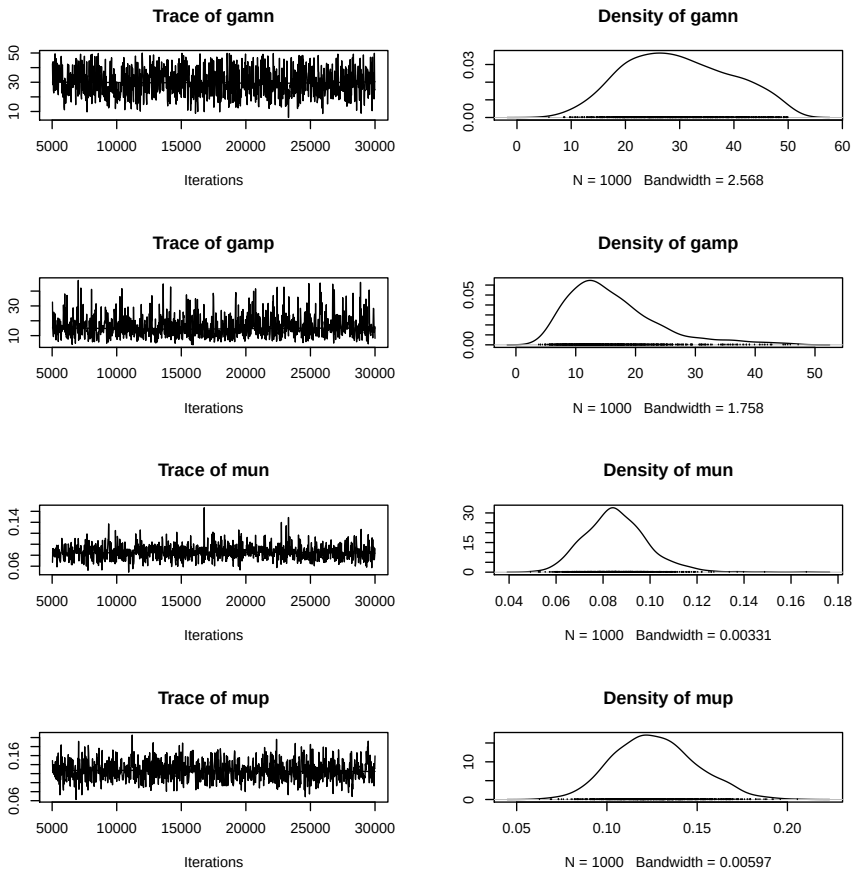

Figure 4.3: Trace and Density plots of γ_-, γ_+, μ_-, and μ_+.

$l = 1$, and $N = 200$. All the slices varied one of the parameters while the rest of the parameters were set at their base design point value.

The slices were $\tau \in .2, .21, \cdots, .8$, $\mu_+ = \mu_- \in .01, .02, \cdots, .3$, $\gamma_+ = \gamma_- \in 5, 6, \cdots, 15$, $m \in 7, 30$, $l \in 1, 2, 3$, and $N \in 100, 120, \cdots, 400$.

The prior distributions for all simulation runs were

$$\mu_+ \sim \mathrm{U}(0, 1)$$

$$\mu_- \sim \mathrm{U}(0, 1)$$

$$\gamma_+ \sim \mathrm{U}(0, 50)$$

$$\gamma_- \sim \mathrm{U}(0, 50)$$

$$\tau \sim \mathrm{U}(0, 1).$$

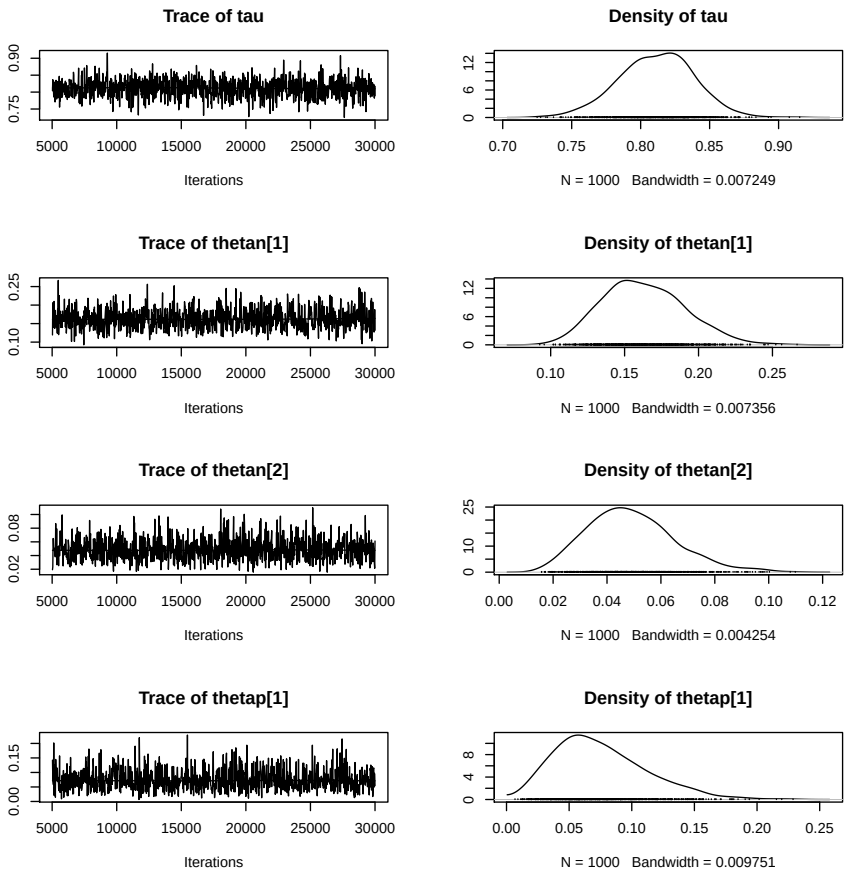

Figure 4.4: Trace and Density plots plots of τ and three arbitrarily chosen misclassification rates.

It was discovered that occasionally, even with good starting values the misclassification rate parameters would instead converge to the classification rates. This could have been corrected with a change to the prior distributions given above. For instance, enforcing $\mu_- + \mu_+ < 1$ could have been done with $\mu_+ \sim \mathrm{U}(0,1)$ and $\mu_+ \sim \mathrm{U}(0, 1-\mu_+)$. But it happened seldom enough, and was only noticed afterwards so the decision was made to instead just throw out the simulation runs in which this switch occurred.

The simulations were done in R using RJAGS to interface with JAGS. JAGS was used instead of OpenBUGS because of an error that occurs when using this model in OpenBUGS. It was found that using a beta distribution as the highest

Table 4.1: Posterior Summary

Parameters	Mean	SD	2.5%	50%	97.5%
an	2.51279	0.80152	1.15610	2.39566	4.16008
ap	2.00563	0.93477	0.78001	1.83841	4.34154
bn	27.30847	8.93456	11.49686	26.75044	44.18924
bp	14.14948	6.80605	5.11596	12.67646	32.17866
gamn	29.82126	9.64504	12.90818	29.14211	47.98427
gamp	16.15511	7.66225	6.05156	14.47033	36.88565
mun	0.08564	0.01345	0.06254	0.08489	0.11521
mup	0.12688	0.02242	0.08647	0.12582	0.17178
tau	0.81145	0.02723	0.75568	0.81266	0.86213
thetan[1]	0.16305	0.02763	0.11577	0.16134	0.22004
thetan[2]	0.04901	0.01611	0.02210	0.04735	0.08504
thetap[1]	0.07530	0.03662	0.01985	0.06949	0.15613
thetap[2]	0.30560	0.06899	0.18356	0.29896	0.44495

distribution in a hierarchical model in OpenBUGS resulted in convergence to nonsensical beta distributions when using R2OpenBUGS. The error did not occur when just using OpenBUGS. However, the need for simulations to find the model's operating characteristics necessitated some sort of rjags or R2OpenBUGS interface, and so JAGS was used for these simulations. The simulations had a sample size of 25000 with a thinning of 25 following a burn in of 5000. 200 simulations were run per design point.

4.4.1 Simulation Results

In Figure 4.5 we see the effect on parameter estimation of changing values of μ_+ and μ_-. Several things are noteworthy as the misclassification parameters are increased from $\mu_- = \mu_+ = .01$ to $\mu_- = \mu_+ = .3$. For small misclassification values, the box created by the upper 97.5 percentile and lower 2.5 percentile of the misclassification rates show noticeable bias, although the median of the estimates is approximately unbiased. The uncertainty in the estimates of the misclassification parameters appear to be increasing throughout the range. Estimates of the overall quality of the production process, τ, remain relatively constant which is because the

conditional indepedence assumption combined with the low misclassification rates results in the vast majority of the parts being ultimately correctly classified. Also, increasing misclassification rates lead, at least initially, to decreasing credible set width of the γ parameters all of which have upwardly biased estimates.

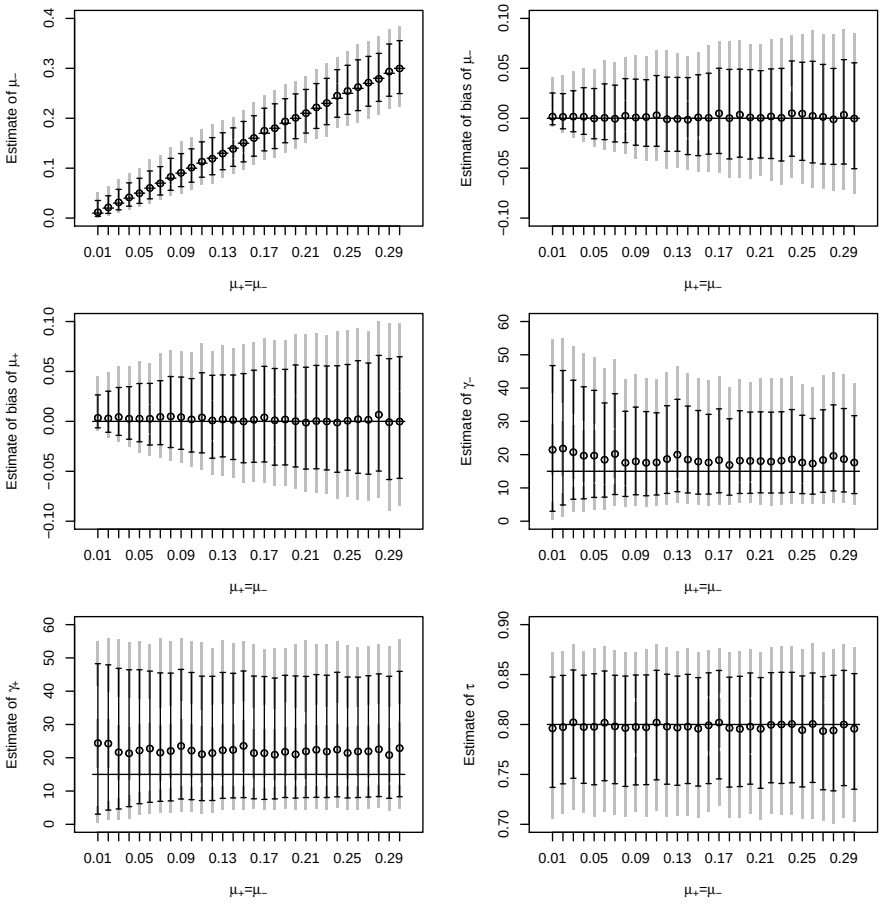

Figure 4.5: From left to right and top to bottom, the summary of the posterior results for μ_-, the bias of μ_-, the bias of μ_+, γ_+, γ_-, and τ as $\mu_- = \mu_+$ changes from .01 to .3.

Figure 4.6 shows the effect that increasing γ_+ and γ_- has. The bias steadily increases as γ increases as does the variability of the estimate of γ. Because $\tau = .8$ there are more good parts than bad parts in the sample, and this results in more precise estimates of γ_- than of γ_+. The estimates of the mean misclassification rates, μ_+ and μ_- appear to improve slightly as the γ parameters get larger. This

essentially, means that as the variability of the inspectors decreases the estimate of their mean effectiveness improves. Lastly, the estimate of τ appears unchanged as the γ parameters vary.

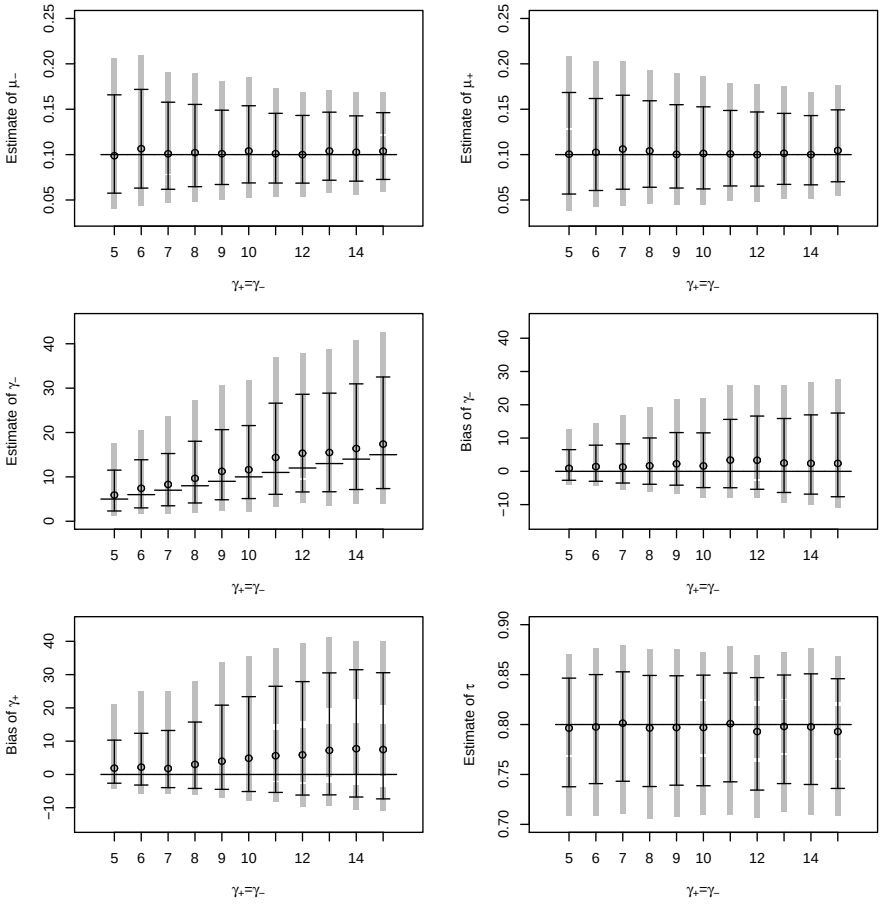

Figure 4.6: From left to right and top to bottom, the summary of the posterior results for μ_-, μ_+, γ_-, the bias of γ_-, the bias of γ_+, and τ as $\gamma_- = \gamma_+$ changes from 5 to 15.

Figure 4.7 shows the effect that increasing the number of inspectors, m, has. Parameter estimates of the mean parameter of the misclassification rates improves substantially as the number of inspectors is increased and is unbiased through the simulation range. Estimation of the γ parameters improves also as the number of inspectors increases. Notably, the increase in the precision is much more apparent for the γ_- parameter than for the γ_+ parameter. What this is probably showing is that

it takes both a decent number of inspectors and a relatively large number of either true positives (for γ_-) or true negatives (for γ_+) for the likelihood associated with the respective γ parameter to begin dominating its prior. That τ remains relatively constant throughout the range is largely indicative that even 7 conditionally independent inspections can determine the true value of an item with little uncertainty at the given misclassification rate.

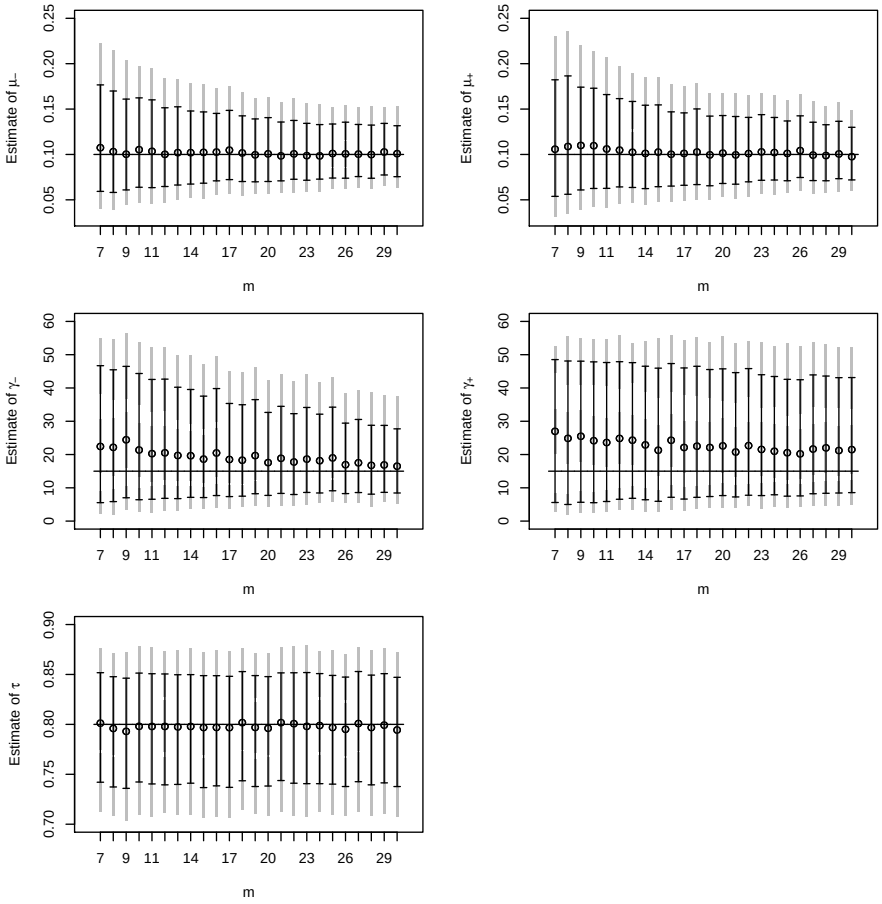

Figure 4.7: From left to right and top to bottom, the summary of the posterior results for μ_-, μ_+, γ_-, γ_+, and τ as the number of inspectors, m, changes from 7 to 30.

Figure 4.8 shows the effect of increasing the number of items inspected, N. Parameter estimates of the mean parameter of the misclassification rates remain relatively the same, while estimation of the γ parameters improves slightly, becoming

both less biased and having less variability. Again, this effect is much more apparent for the γ_- parameter than for the γ_+ parameter. The biggest estimation improvement that came with increasing the sample size, however, is the estimation of the proportion of good items in the population, τ.

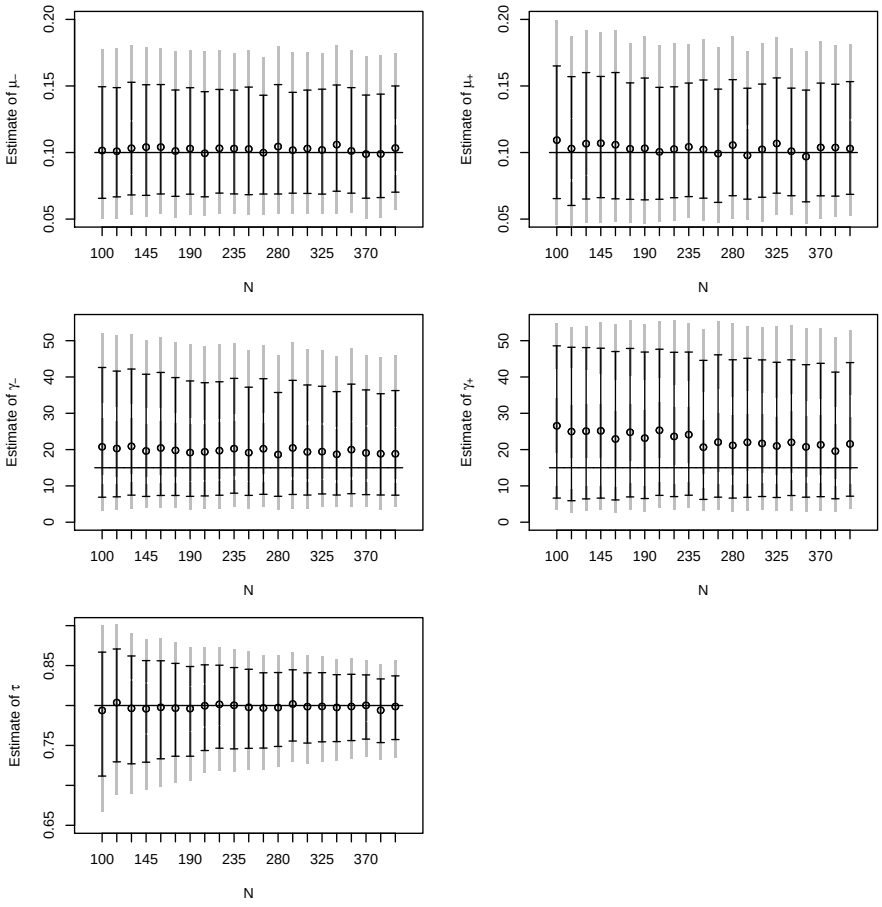

Figure 4.8: From left to right and top to bottom, the summary of the posterior results for μ_-, μ_+, γ_-, γ_+, and τ as the number of items inspected, N, changes from 100 to 400.

Figure 4.9 shows the effect of increasing the proportion of good items, τ. The primary effect that changing τ appears to have over this range is on the estimation of the γ parameters. The variability of the estimate of γ_- decreases as τ increases, while the variability of the estimate of γ_+ increases as τ increases. This is expected because changing the value of τ is changing the number of good and bad parts in the

sample which in turn effects estimates that are highly dependent on these numbers. Interestingly, it was found that τ values near .5 resulted in a much higher percentage of the misclassification rates becoming the classification rates. This is where this issue was discovered as it had a substantial impact on the results here.

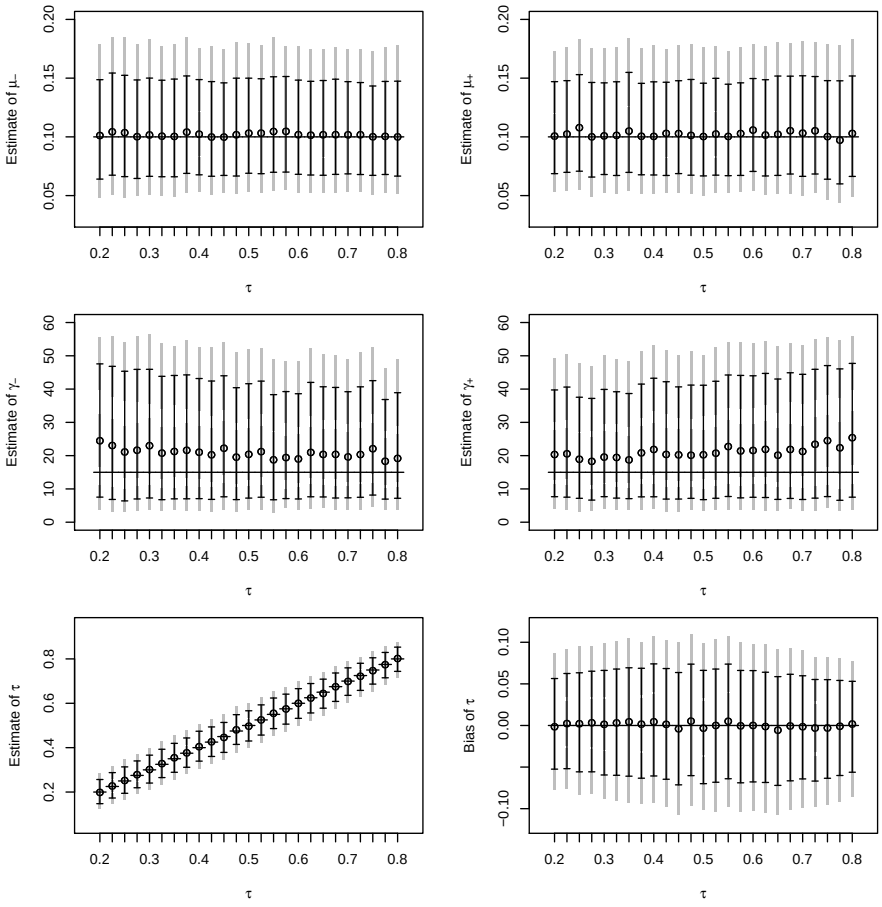

Figure 4.9: From left to right and top to bottom, the summary of the posterior results for μ_-, μ_+, γ_-, γ_+, τ, and the bias of τ as the proportion of good items, τ, changes from .2 to .8.

Figure 4.10 shows the effect of increasing the number of inspections per inspector of each item, l. Its effect appears to be minimal with only some slight improvement in the estimation of the γ parameters.

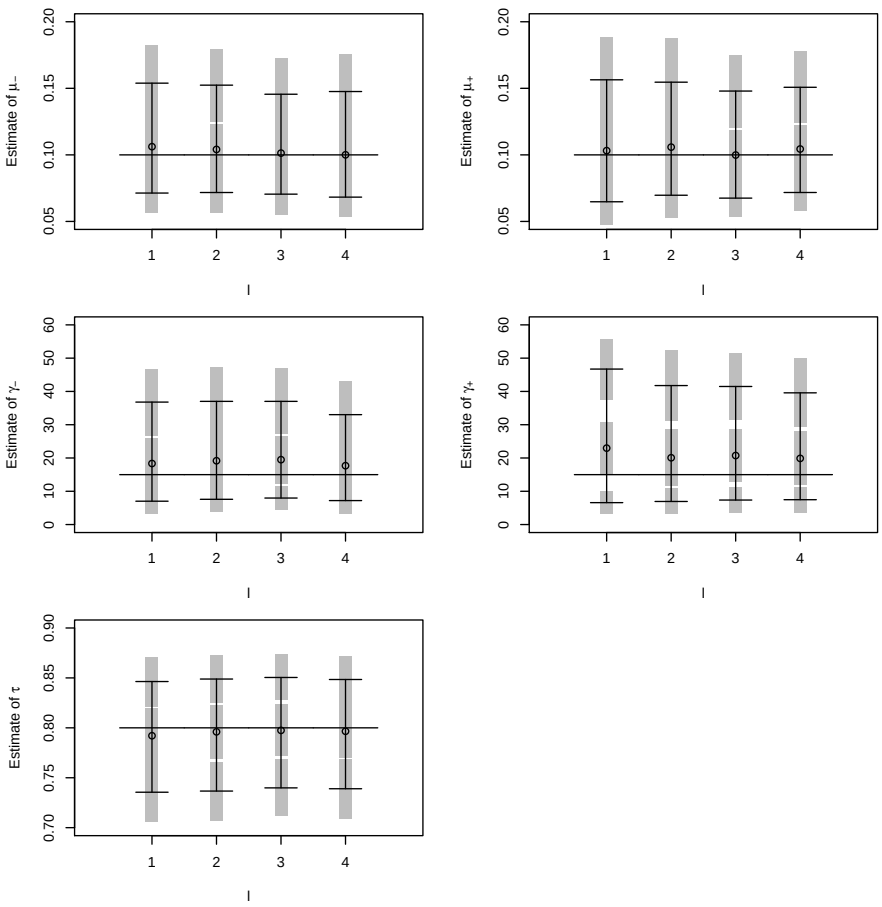

Figure 4.10: From left to right and top to bottom, the summary of the posterior results for μ_-, μ_+, γ_-, γ_+, and τ as the number of inspections per inspector of each item, l, changes from 1 to 4.

4.5 Model II

The first model introduced in Section 4.2 utilizes a population based sampling scheme. However, Danila et al. (2008) introduce a sampling scheme that samples some predetermined combination of previously passed or failed items, and utilizes information concerning the overall inspection pass rate to estimate all model parameters. She shows that this method can be more efficient by selecting the sample such that there is a more nearly half and half split of good and bad items.

Model II incorporates design elements from Danila et al. (2008) into the model, and, additionally, records and uses the information of whether or not the first inspection was a pass or a fail.

Assume that there are N_0 and N_1 randomly selected items to be inspected sampled from the previously failed and passed parts respectively. Define $N = N_0 + N_1$. T and τ are as defined in Section 4.2.

The result of the initial inspection is recorded as ξ_i, where $\xi_i = 1$ denotes that the i^th item passed its first inspection and $xi_i = 0$ denotes an item that failed its first inspection. Repeated independent and fallible observations are then made just like for Model I, and using the same notation. The same distributional assumptions with regards to $\theta_{j,+}$ and $\theta_{j,-}$ are made. Conditional independence is also assumed.

To incorporate information on whether the part comes from the previously passed parts or previously failed parts requires an additional assumption. Specifically, is information on who the inspector was for the first inspections available or is it not available. If this information is available then the two conditional probabilities needed are

$$P(T_i = 1|\xi_i = 1) = \frac{\tau(1 - \theta_{j(i),-})}{\tau(1 - \theta_{j(i),-}) + (1 - \tau)\theta_{j(i),+}}, \tag{4.9}$$

$$P(T_i = 1|\xi_i = 0) = \frac{\tau\theta_{j(i),-}}{\tau\theta_{j(i),-} + (1 - \tau)(1 - \theta_{j(i),+})}, \tag{4.10}$$

where $j(i)$ is the initial inspector of the i^{th} part. This information can then be incorporated into the likelihood as

$$
\begin{aligned}
L(\mathbf{t}|\tau,\theta_-,\theta_+,\xi) \propto & \prod_{i=1}^{N_0}\left(\frac{\tau\theta_{j(i),-}}{\tau\theta_{j(i),-}+(1-\tau)(1-\theta_{j(i),+})}\right)^{t_i} \\
& \times\left(1-\frac{\tau\theta_{j(i),-}}{\tau\theta_{j(i),-}+(1-\tau)(1-\theta_{j(i),+})}\right)^{1-t_i} \\
& \times\prod_{i=N_0+1}^{N}\left(\frac{\tau(1-\theta_{j(i),-})}{\tau(1-\theta_{j(i),-})+(1-\tau)\theta_{j(i),+}}\right)^{t_i} \\
& \times\left(1-\frac{\tau(1-\theta_{j(i),-})}{\tau(1-\theta_{j(i),-})+(1-\tau)\theta_{j(i),+}}\right)^{1-t_i} \\
= & \left(\frac{\tau\theta_{j(i),-}}{\tau\theta_{j(i),-}+(1-\tau)(1-\theta_{j(i),+})}\right)^{\sum_{i=1}^{N_0}t_i} \\
& \times\left(1-\frac{\tau\theta_{j(i),-}}{\tau\theta_{j(i),-}+(1-\tau)(1-\theta_{j(i),+})}\right)^{N_0-\sum_{i=1}^{N_0}t_i} \\
& \times\left(\frac{\tau(1-\theta_{j(i),-})}{\tau(1-\theta_{j(i),-})+(1-\tau)\theta_{j(i),+}}\right)^{\sum_{i=N_0+1}^{N}t_i} \\
& \times\left(1-\frac{\tau(1-\theta_{j(i),-})}{\tau(1-\theta_{j(i),-})+(1-\tau)\theta_{j(i),+}}\right)^{N_1-\sum_{i=N_0+1}^{N}t_i}. \qquad (4.11)
\end{aligned}
$$

If we do not assume that we know which particular inspector made the first inspection, then there are two different conditional probabilities needed to incorporate the data on first inspections into the model, namely $P(T=1|\xi=0)$ and $P(T=1|\xi=1)$. We have that

$$P(T=1|\xi=1)=\frac{\tau(1-\theta_-)}{\tau(1-\theta_-)+(1-\tau)\theta_+}, \qquad (4.12)$$

$$P(T=1|\xi=0)=\frac{\tau\theta_-}{\tau\theta_-+(1-\tau)(1-\theta_+)}, \qquad (4.13)$$

where θ_+ is the overall false positive rate which is approximately μ_+. Likewise, θ_- is the overall false positive rate which is approximately μ_-. These approximations are what will be used in this model. In the simulation study that follows, it will be shown that for the cases studied, that this approximation works well. Thus, we have

that

$$L(\mathbf{t}|\tau,\mu_-,\mu_+,\xi) \propto \left(\frac{\tau\mu_-}{\tau\mu_- + (1-\tau)(1-\mu_+)}\right)^{\sum_{i=1}^{N_0} t_i} \times \left(1 - \frac{\tau\mu_-}{\tau\mu_- + (1-\tau)(1-\mu_+)}\right)^{N_0 - \sum_{i=1}^{N_0} t_i} \times \left(\frac{\tau(1-\mu_-)}{\tau(1-\mu_-) + (1-\tau)\mu_+}\right)^{\sum_{i=N_0+1}^{N} t_i} \times \left(1 - \frac{\tau(1-\mu_-)}{\tau(1-\mu_-) + (1-\tau)\mu_+}\right)^{N_1 - \sum_{i=N_0+1}^{N} t_i}. \quad (4.14)$$

Also incorporated in this model for the first time is information about the overall pass rate. Specifically, we use the fact that

$$P(Y=1) = \tau(1-\theta_-) + (1-\tau)\theta_+, \quad (4.15)$$

where τ, θ_-, and θ_+ are defined as before.

Let N_{pop} be the total number of items inspected by an inspection process, and N_{pass} be the total number of items passed by an inspection process. Again, because $\theta_+ \approx \mu_+$ and $\theta_- \approx \mu_-$, these substitutions will be made. Then the additional contribution this information makes to the likelihood is

$$L(N_{pass}|\mu_-,\mu_+,\tau) \propto (\tau(1-\theta_-) + (1-\tau)\theta_+)^{N_{pass}} \times (1 - \tau(1-\theta_-) + (1-\tau)\theta_+)^{N_{pop} - N_{pass}}. \quad (4.16)$$

The full likelihood is the product of all the previously introduced pieces of the likelihood. Specifically, it is the multiplication of Equation 4.4 without τ, Equation 4.11 or 4.14 (whichever is appropriate), and Equation 4.16. We will assume here that we don't know which inspector performed the first inspection (thus, Equation 4.14), and will be using previously defined notation. Thus, the likelihood given the observed data matrix $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ where $\mathbf{x}_i = \{x_{i,1}, \cdots, x_{i,m}\}'$ and the first

inspections ξ is

$$
\begin{aligned}
&L(\mathbf{x}, \mathbf{t}, \theta_-, \theta_+, N_{pass} | \mu_+, \gamma_+, \mu_-, \gamma_-, \tau, \xi) \propto \\
&\left(\frac{\tau\mu_-}{\tau\mu_- + (1-\tau)(1-\mu_+)}\right)^{\sum_{i=1}^{N_0} t_i} \left(1 - \frac{\tau\mu_-}{\tau\mu_- + (1-\tau)(1-\mu_+)}\right)^{N_0 - \sum_{i=1}^{N_0} t_i} \\
&\times \left(\frac{\tau(1-\mu_-)}{\tau(1-\mu_-) + (1-\tau)\mu_+}\right)^{\sum_{i=N_0+1}^{N} t_i} \left(1 - \frac{\tau(1-\mu_-)}{\tau(1-\mu_-) + (1-\tau)\mu_+}\right)^{N_1 - \sum_{i=N_0+1}^{N} t_i} \\
&\times \left(\prod_{j=1}^{m} (1-\theta_{j-})^{\sum_{i=1}^{N} t_i x_{ij} + \mu_- \gamma_- - 1} (\theta_{j-})^{\sum_{i=1}^{N} t_i (n_{ij} - x_{ij}) + \gamma_- - \mu_- \gamma_- - 1}\right) \\
&\times \left(\prod_{j=1}^{m} (\theta_{j+})^{\sum_{i=1}^{N} (1-t_i) x_{ij} + \mu_+ \gamma_+ - 1} (1-\theta_{j+})^{\sum_{i=1}^{N} (1-t_i)(n_{ij} - x_{ij}) + \gamma_+ - \mu_+ \gamma_+ - 1}\right) \\
&\times \left(\tau(1-\mu_-) + (1-\tau)\mu_+\right)^{N_{pass}} \left(1 - \tau(1-\mu_-) + (1-\tau)\mu_+\right)^{N_{pop} - N_{pass}} . \qquad (4.17)
\end{aligned}
$$

Independent uniform priors are again used, so the posterior is proportional to the likelihood.

4.5.1 Model II Simulation Study

A simulation study was conducted to assess the operating characteristics of the model. The parameters varied in the simulation study were τ, μ_+, μ_-, γ_+, γ_-, m, l, and N. Slices (the change of one parameter of interest at a time) of the design space were investigated rather than doing a full factorial design. What might be thought of as the base design point was $\tau = .9$, $\mu_+ = \mu_- = .1$, $\gamma_+ = \gamma_- = 15$, $m = 15$, $l = 1$, $N_0 = 200$, $N_1 = 0$, and $N_{pop} = 100000$. All the slices varied one of the parameters while the rest of the parameters were set at their base design point value.

The slices were $\tau \in .6, .61, \cdots, .95$, $\mu_+ = \mu_- \in .01, .02, \cdots, .3$, $\gamma_+ = \gamma_- \in 5, 6, \cdots, 15$, $m \in 7, 30$, $l \in 1, \cdots, 4$, $N_0 \in 100, 120, \cdots, 400$, and $N_{pop} \in 1000, 10000, \cdots, 10000000$. Additionally, while holding $N = 200$ constant, the fraction of $N_0/(N_0 + N_1)$ was varied from 0 to 1 in increments of .1.

The prior distributions for all simulation runs were

$$\mu_+ \sim \mathrm{U}(0,1)$$

$$\mu_- \sim \mathrm{U}(0,1)$$

$$\gamma_+ \sim \mathrm{U}(0,50)$$

$$\gamma_- \sim \mathrm{U}(0,50)$$

$$\tau \sim \mathrm{U}(0,1).$$

It was discovered that occasionally, even with good starting values the misclassification rate parameters would instead converge to the classification rates. This could have been corrected with a change to the prior distributions given above, but it happened seldom enough and was only noticed afterwards and so the decision was made to instead just throw out the simulation runs in which this switch occurred.

The simulations were done in R using RJAGS to interface with JAGS. JAGS was used instead of OpenBUGS because of a strange error that was occurring when using this model in OpenBUGS. It was found that using a beta distribution as the highest distribution in a hierarchical model in OpenBUGS resulted in convergence to nonsensical beta distributions when using R2OpenBUGS. The error did not occur when just using OpenBUGS. However, the need to simulations to find the model's operating characteristics necessitated some sort of rjags or R2OpenBUGS interface, and so JAGS was used for these simulations. The simulations had a sample size of 25000 with a thinning of 25 following a burn in of 5000. 200 simulations were run per design point.

4.5.2 Model II Simulation Results

In Figure 4.11 we see the effect on parameter estimation of changing values of μ_+ and μ_-. Several things are noteworthy as the misclassification parameters are increased from $\mu_- = \mu_+ = .01$ to $\mu_- = \mu_+ = .3$. For small misclassification values, the plots of the false positive misclassification rate is noticeably skewed, although

the median of the estimates is approximately unbiased. The incorporation of the first inspection results in a greatly decreased credible set width for the false negative rates. The uncertainty in the estimates of the misclassification parameters appear to be increasing throughout the range. Worse misclassification rates predictably lead to less certainty about the overall quality of the production process, τ. Also, increasing misclassification rates lead, at least initially, to decreasing credible set width of the γ parameters all of which have upwardly biased estimates.

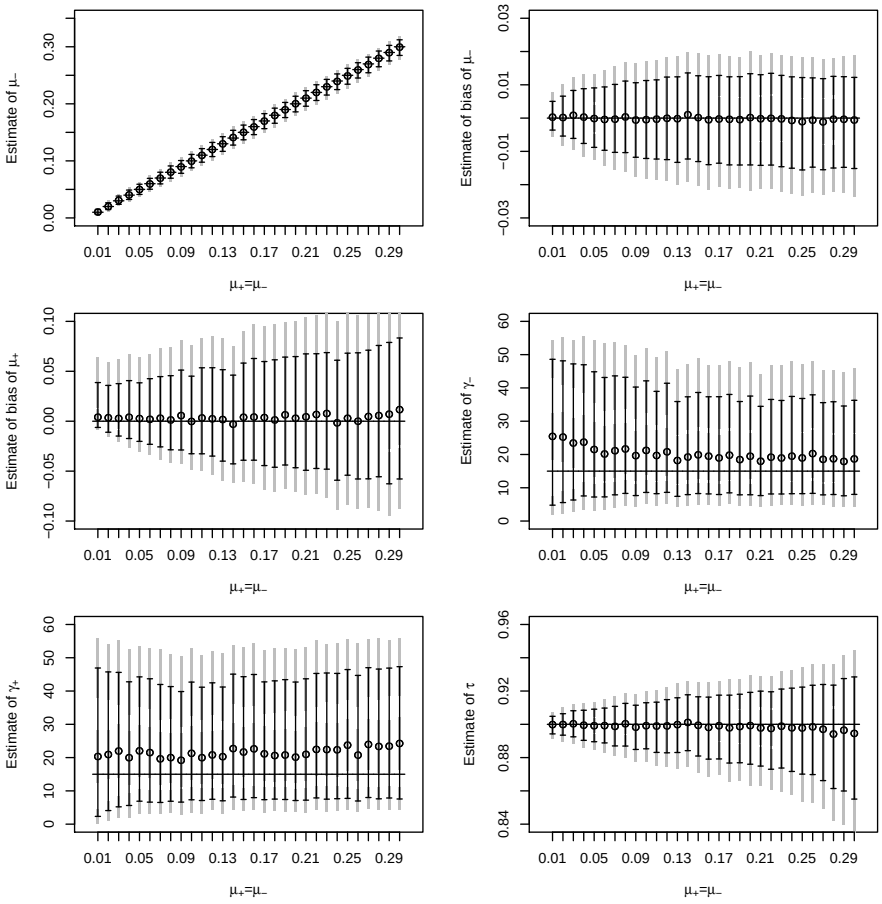

Figure 4.11: From left to right and top to bottom, the summary of the posterior results for μ_-, the bias of μ_-, the bias of μ_+, γ_+, γ_-, and τ as $\mu_- = \mu_+$ changes from .01 to .3.

Figure 4.12 shows the effect that increasing γ_+ and γ_- has. The skewness steadily increases as γ increases as does the variability of the estimate of γ. Because

$\tau = .9$ there are more good parts than bad parts in the sample, and this results in more precise estimates of γ_- than of γ_+. Again the incorporation of the first inspection results in a greatly decreased credible set width for the false negative rates. The estimates of the mean misclassification rate, μ_+ appears to improve slightly as the γ parameters get larger. This essentially, means that as the variability of the inspectors decreases the estimate of their mean effectiveness improves. Lastly, the estimate of τ appears unchanged as the γ parameters vary.

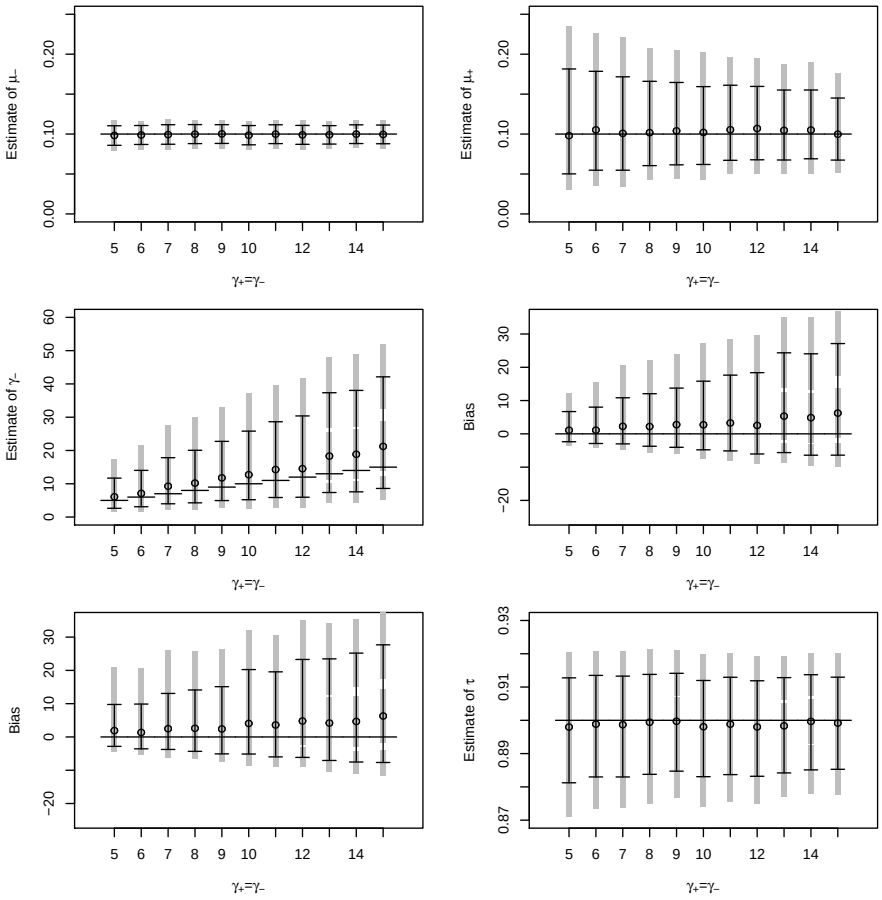

Figure 4.12: From left to right and top to bottom, the summary of the posterior results for μ_-, μ_+, γ_-, the bias of γ_-, the bias of γ_+, and τ as $\gamma_- = \gamma_+$ changes from 5 to 15.

Figure 4.13 shows the effect that increasing the number of inspectors, m, has. Parameter estimates of both mean parameters are unbiased. Notably, the misclassi-

fication rate μ_+ improves substantially as the number of inspectors is increased while the estimate of μ_- does not. This is because the replication is being done within the portion of the likelihood in Equation 4.4 and not within Equation 4.14. Estimation of the γ parameters improves also as the number of inspectors increases. That τ remains relatively constant throughout the range is largely indicative that even 7 conditionally independent inspections can determine the true value of an item with little uncertainty.

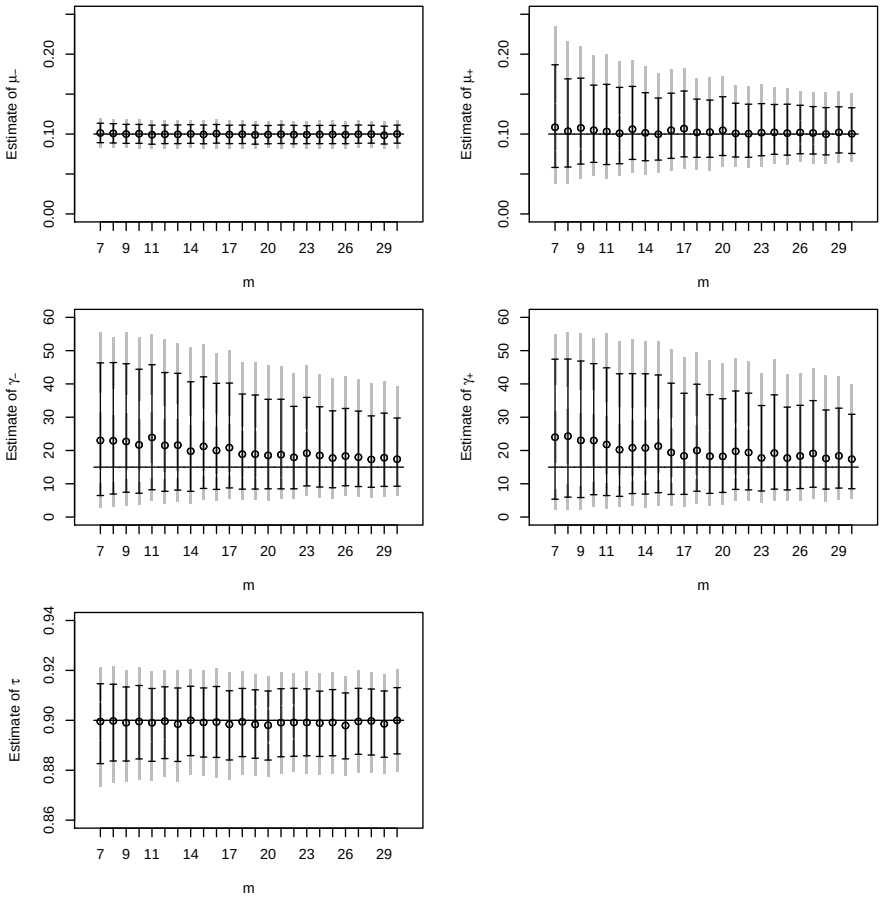

Figure 4.13: From left to right and top to bottom, the summary of the posterior results for μ_-, μ_+, γ_-, γ_+, and τ as the number of inspectors, m, changes from 7 to 30.

Figure 4.14 shows the effect that increasing the number of items inspected, N, has. Parameter estimates of the μ_- improve while the estimate of μ_- remains

fairly constant. Estimation of the γ parameters improve slightly, becoming both less biased and having less variability. The biggest estimation improvement that came with increasing the sample size, however, is the estimation of the proportion of good items in the population, τ.

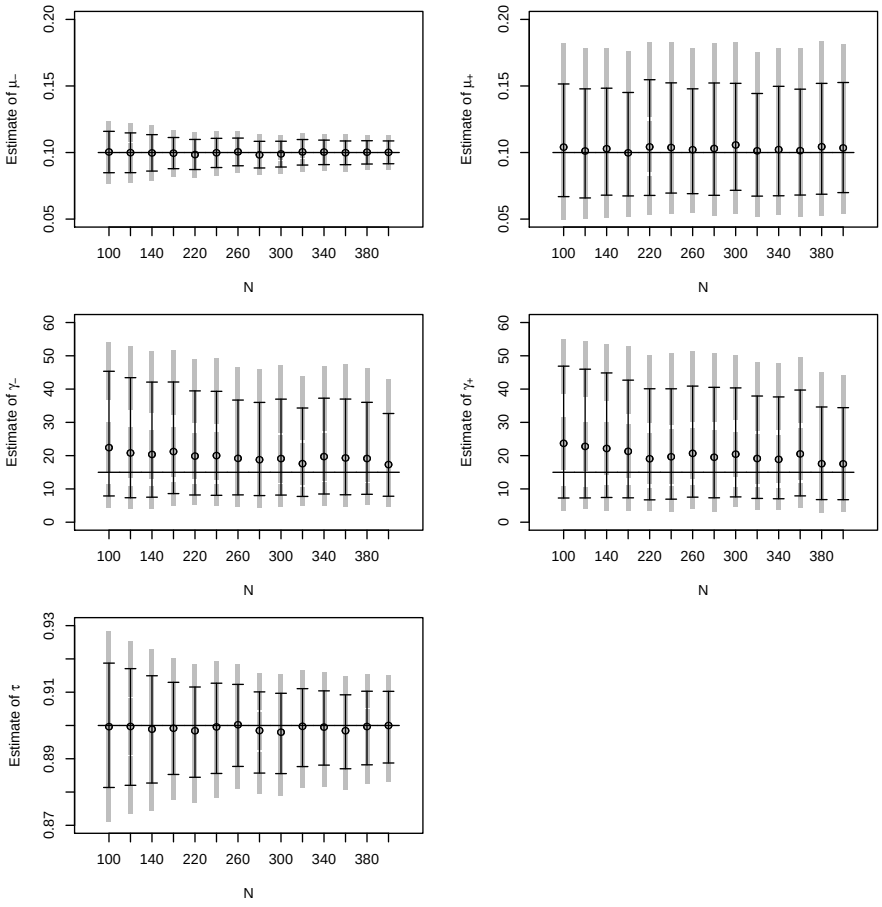

Figure 4.14: From left to right and top to bottom, the summary of the posterior results for μ_-, μ_+, γ_-, γ_+, and τ as the number of items inspected, N, changes from 100 to 400.

Figure 4.15 shows the effect that increasing the proportion of good items, τ, has. The primary effect that changing τ appears to have over this range is on the estimation of the τ and μ_-parameters. Both showing decreasing variability as the parameter τ increases. The variability of the estimate of γ_- decreases as τ increases, while the variability of the estimate of γ_+ increases as τ increases. This is expected

because changing the value of τ is changing the number of good and bad parts in the sample which in turn effects estimates that are highly dependent on these numbers.

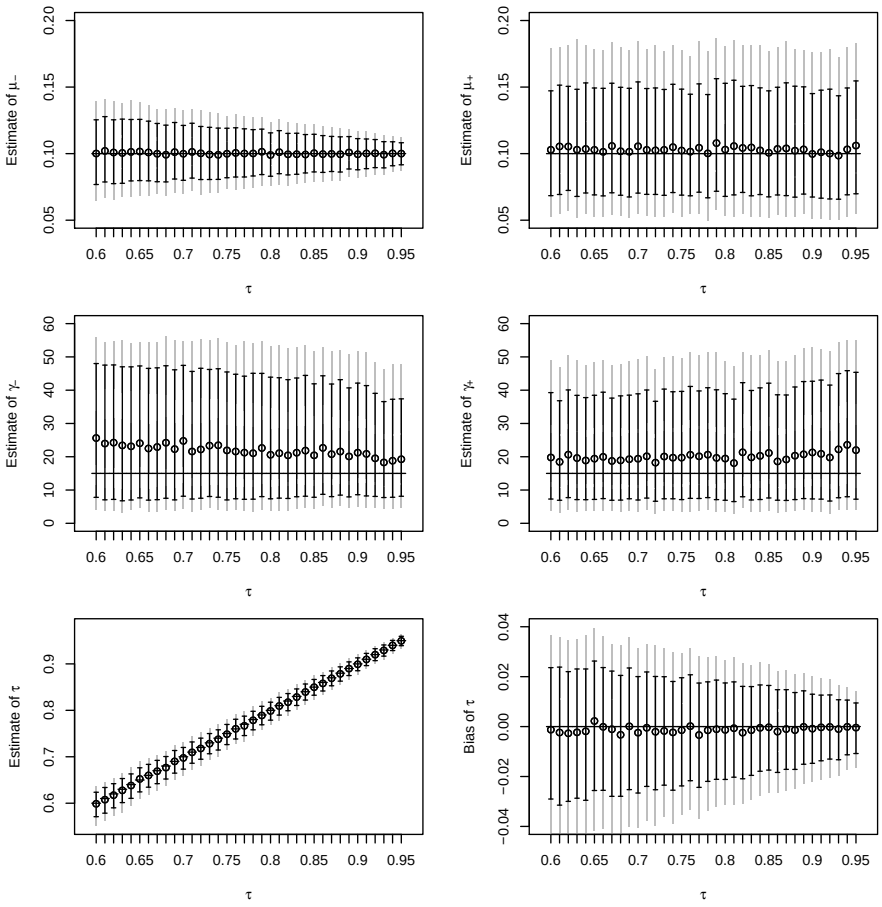

Figure 4.15: From left to right and top to bottom, the summary of the posterior results for μ_-, μ_+, γ_-, γ_+, τ, and the bias of τ as the proportion of good items, τ, changes from .6 to .95.

Figure 4.16 shows the effect that increasing the number of inspections per inspector of each item, l, has. Its effect appears to be minimal with only some slight improvement in the estimation of the γ parameters.

4.6 An Application: Ranking

It could be the case that rather than all the raters or all of the parts coming from a single population, that there are instead multiple populations being sampled

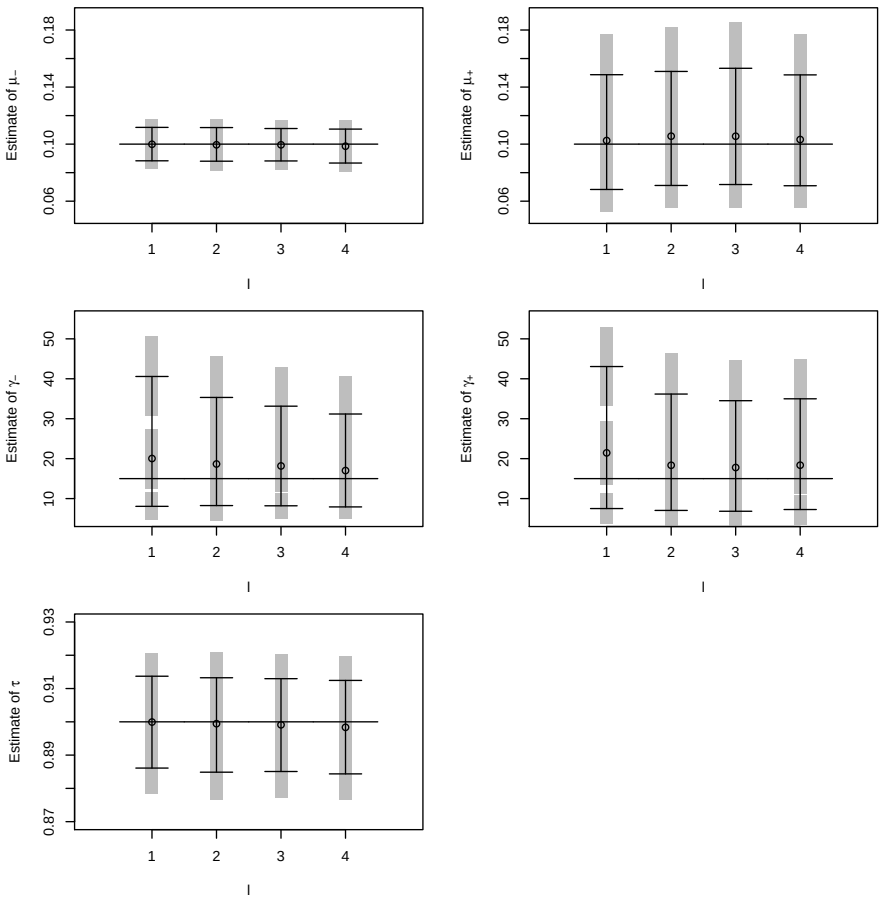

Figure 4.16: From left to right and top to bottom, the summary of the posterior results for μ_-, μ_+, γ_-, γ_+, and τ as the number of inspections per inspector of each item, l, changes from 1 to 4.

from. This could easily be the case, for instance, if a production process spanned several different plants, and there was a difference between the plants. In this case, while the misclassification rates at each plant are of interest it might also be of interest to know both what the ranking between plants is.

What is presented in this section, is an extension of the first model introduced in the chapter, the random effects model for multiple inspectors based on a population based standard sampling procedure. The model has been extended to allow for multiple populations of inspectors and production processes, and ranking procedures and best subset selection methods are performed on these intermediate results.

More precisely, the model in 4.2 is fit to each plant independently, and parameter(s) of interest at each plant are then compared to the same parameter at the other plants. In an MCMC simulation it is straightforward to obtain ranking information about a set of parameters, and this is what is then done.

Let N_p be the number of plants to be ranked, and let the index $k \in 1, \cdots, N_p$ represent all parameters particular to the k^{th} plant. Thus, the likelihood for this model is

$$\begin{aligned} L(\mathbf{x}, \mathbf{t}, \theta_-, \theta_+ | \mu_+, \gamma_+, \mu_-, \gamma_-, \tau) =& L(\mathbf{x}|\mathbf{t}, \theta_-, \theta_+) L(\theta_-|\mu_-, \gamma_-) \\ & \times L(\mathbf{t}|\tau) L(\theta_+|\mu_+, \gamma_+) \\ =& \prod_{k=1}^{N_p} \Bigg[\tau_k^{\sum_{i=1}^{N} t_{ik}} (1-\tau)^{N_k - \sum_{i=1}^{N_k} t_{ik}} \\ & \times \left(\prod_{j=1}^{m} (1-\theta_{jk-})^{\sum_{i=1}^{N_k} t_{ik} x_{ijk} + \mu_{k-}\gamma_{k-} - 1} \right. \\ & \left. \times (\theta_{jk-})^{\sum_{i=1}^{N_k} t_{ik}(n_{ijk} - x_{ijk}) + \gamma_{k-} - \mu_{k-}\gamma_{k-} - 1} \right) \\ & \times \left(\prod_{j=1}^{m} (\theta_{jk+})^{\sum_{i=1}^{N_k} (1-t_{ik}) x_{ijk} + \mu_{k+}\gamma_{k+} - 1} \right. \\ & \left. \times (1-\theta_{jk+})^{\sum_{i=1}^{N_k} (1-t_{ik})(n_{ijk} - x_{ijk}) + \gamma_{k+} - \mu_{k+}\gamma_{k+} - 1} \right) \Bigg]. \end{aligned} \tag{4.18}$$

For this model to be identifiable, the parameters for each plant must be identifiable (see Section 4.2).

It is quite easy in an MCMC simulation to record ranking information for the misclassification μ_{k+}'s and μ_{k-}'s. Additionally, it is quite easy to record this same ranking information for the τ_k's.

4.6.1 Best Subset Selection

Sometimes rather than just a raw ranking, what is really desired is to find the best plant(s). This can, perhaps, best be accoplished using a combination of deci-

sion theory and subsetting. Bratcher and Phalla (1974) derive a decision-theoretic approach to partition parameters into two sets based upon an ordering of the parameters of interest. Stamey et al. (2004) extend their work to include Poisson rates. In this subsection, we extend this work further to consider subsetting of the population of manufacturing plants into a superior set, S, and an inferior one, S^C.

As a brief review, we have assumed a true dichotomy between good and false parts. This in turns implies an independence between the two different misclassification rates for any given inspector, the false positive and false negative misclassification rates. Note that the γ parameters effect the variance of the inspectors only given the respective μ parameter. Thus, there are only three parameters that it makes sense to rank, the μ_+'s, the μ_-'s, and the τ's. In all three cases, the mathematical derivation of the superior set is the same and is given below.

Let $\eta_k \in \mu_{k+}, \mu_{k-}, \tau_k$. In the creation of a best subset, there are really N_p separate two-state decision problems. These are whether or not to place a plant's misclassification rate in the superior set, $d_+^k : \eta_k \in S$, or not. The following constant loss functions are assumed:

$$L_+^k(\eta) = \begin{cases} 0 & \text{if } \eta_k = \eta_{best}, \\ c_1 & \text{if } \eta_k \neq \eta_{best} \end{cases} \quad \text{and}$$

$$L_-^k(\eta) = \begin{cases} c_2 & \text{if } \eta_k = \eta_{best}, \\ 0 & \text{if } \eta_k \neq \eta_{best}, \end{cases} \quad i = 1, 2, \ldots, N_p,$$

where L_+^k is the loss function for d_+^k and L_-^k is the loss function for d_-^k. Note that the word *best* is used in the definition of the loss function above because best η_k for the μ's is the smallest one, while best for the τ's is the largest one. To make a decision, only $c = c_1/c_2$ is utilized. These loss functions determine the decision criteria: Take action d_+^k and include η_k as a candidate for the largest parameter if

$\Pr(\eta_k = \eta_{best}|\mathbf{x}) \geq 1/(c+1)$. Generally, c_2 will be greater than c_1 in this problem because generally the more serious error is failing to place the best η_k in S.

The probability that η_k is the best of the η parameters can be found by integrating either

$$P(\eta_i = \eta_{best}|\mathbf{x}) = \int_0^1 \int_0^{\eta_i} \cdots \int_0^{\eta_i} L(\eta|\mathbf{x}) \mathrm{d}\eta_1, \cdots \mathrm{d}\eta_{i-1}, \mathrm{d}\eta_{i+1}, \cdots, \mathrm{d}\eta_{N_p}, \mathrm{d}\eta_i, \quad (4.19)$$

for the τ's and

$$P(\eta_i = \eta_{best}|\mathbf{x}) = \int_0^1 \int_{\eta_i}^1 \cdots \int_{\eta_i}^1 L(\eta|\mathbf{x}) \mathrm{d}\eta_1, \cdots \mathrm{d}\eta_{i-1}, \mathrm{d}\eta_{i+1}, \cdots, \mathrm{d}\eta_{N_p}, \mathrm{d}\eta_i, \quad (4.20)$$

for the μ's. Because of the large computational expense of calculating these (4.19 and 4.20) numerically, one can approximate them using Gibbs sampling methods. To do this, one generates a sample $(\eta_{k1}, \eta_{k2}, \cdots, \eta_{kn})'$, for $k = 1, 2, \cdots, N_p$ of size n from the posterior distribution, and then approximates the posterior probability that η_k is the best parameter by

$$\hat{P}(\eta_k = best(\eta_1, \cdots, \eta_{N_p})|\mathbf{x}) = \frac{\#(\eta_{ki} = best(\eta_{1i}, \cdots, \eta_{N_p i})}{n},$$

where $k = 1, \cdots, N_p$,$i = 1, 2, \cdots, n$, and *best* is either the minimum or the maximum.

The parameter η_k is then included in the superior set S if

$$\hat{P}(\eta_k = best(\eta_1, \cdots, \eta_{N_p})|\mathbf{x}) \geq 1/(c+1), k = 1, \cdots, N_p.$$

4.6.2 Subset Selection Simulation Study

A simulation study was conducted to assess the operating characteristics of a few instances of this model. Operating characteristics of both the ranking and the subset selection are studied. Also, because the sample is assumed to be drawn from the general population for this model, τ, the proportion of good parts is of interest along with the parameters for the repeatability and reproducibility.

We chose to do two slices of the simulation space. We let $N_p = 5$ for both. For the first, we held τ constant across all plants and varied $\mu_+ = \mu_-$. Let

$$\begin{aligned} Dmu =& \mu_{k,+} - \mu_{k+1,+} \\ =& \mu_{k,-} - \mu_{k+1,-}, \ k \in 1, 2, \cdots, N_p - 1. \end{aligned}$$

We changed Dmu from .015 to .05. Implied in this notation, is the fact that we chose to uniformly vary the distance between the μ's. We anchored $\mu_{3,+} = \mu_{3,-} = .15$ for this simulation. Other parameters were, $\gamma_{k,+} = \gamma_{k,-} = 15$, $m_k = 15$, $l_k = 1$, $\tau_k = .8$, and $N_k = 200$.

In the second simulation slice, we varied τ while holding the μ's constant at $\mu_{k,+} = \mu_{k,-} = 0.15$. Let

$$Dtau = \tau_k - \tau_{k+1} \ k \in 1, 2, \cdots, N_p - 1.$$

We changed $Dtau$ from 0 to .05 while holding $\tau_3 = .8$.

The prior distributions for all simulation runs were

$$\begin{aligned} \mu_{k,+} &\sim \mathrm{U}(0, 1) \\ \mu_{k,-} &\sim \mathrm{U}(0, 1) \\ \gamma_{k,+} &\sim \mathrm{U}(0, 50) \\ \gamma_{k,-} &\sim \mathrm{U}(0, 50) \\ \tau_k &\sim \mathrm{U}(0, 1). \end{aligned}$$

Occasionally, even with good starting values the misclassification rate parameters would instead converge to the classification rates as noted in Section 4.4. Again the decision was made to just throw out the simulation runs in which this switch occurred.

The simulations were done in R using RJAGS to interface with JAGS. JAGS needed to be used as for reason stated in Section 4.4. And as in that simulation

study, the simulations here had a sample size of 25000 with a thinning of 25 following a burn in of 5000. 200 simulations were run per design point.

4.6.3 Subset Selection Simulation Results

In Figure 4.17 we see the effect on expected size of the superior set and the probability that the actual best parameter is in the superior subset as Dmu changes. As separation between the μ parameters increases, so also does the probability that the best misclassification rate is placed in the superior set. Also the expected size of the superior set decreases as Dmu increases. There are plots for $c \in 4, 9, 19$ that clearly show that higher values of c result in a larger expected superior set size and a higher probability that the true best is in the superior set.

In Figure 4.18 we see the effect on estimated rank of $\mu_{1,-}, \cdots, \mu_{4,-}$ as Dmu changes. As separation between the μ parameters increases, the rank estimates increase in precision. Figure 4.19 shows a posterior summary of the estimated ranks of $\mu_{1,-}, \cdots, \mu_{5,-}$ at a particular Dmu.

In Figure 4.20 we see the effect on expected size of the superior set and the probability that the actual best parameter is in the superior subset as $Dtau$ changes. As separation between the τ parameters increases, so also does the probability that the best misclassification rate is placed in the superior set. Also the expected size of the superior set decreases as $Dtau$ increases. There are plots for $c \in 4, 9, 19$ that clearly show that higher values of c result in a larger expected superior set size and a higher probability that the true best is in the superior set.

In Figure 4.21 we see the effect on estimated rank of $\tau_1, \cdots, \tau_4$ as $Dtau$ changes. As separation between the τ parameters increases, the rank estimates increase in precision. Figure 4.22 shows the posterior summary of the estimated ranks of $\tau_1, \cdots, \tau_5$ at a particular $Dtau$.

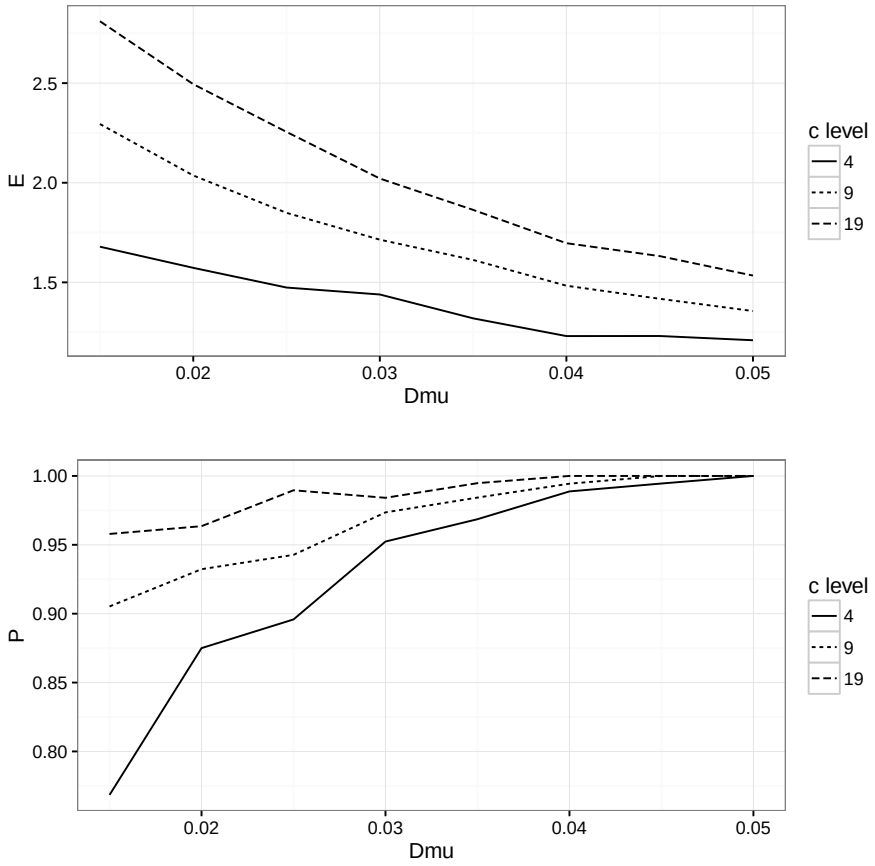

Figure 4.17: Summary of the posterior results, above, the Expected value of the best selection subset as the spacing of the μs, Dmu, changes from .01 to .05. Below, the probability that the best μ is included in the superior subset as Dmu changes.

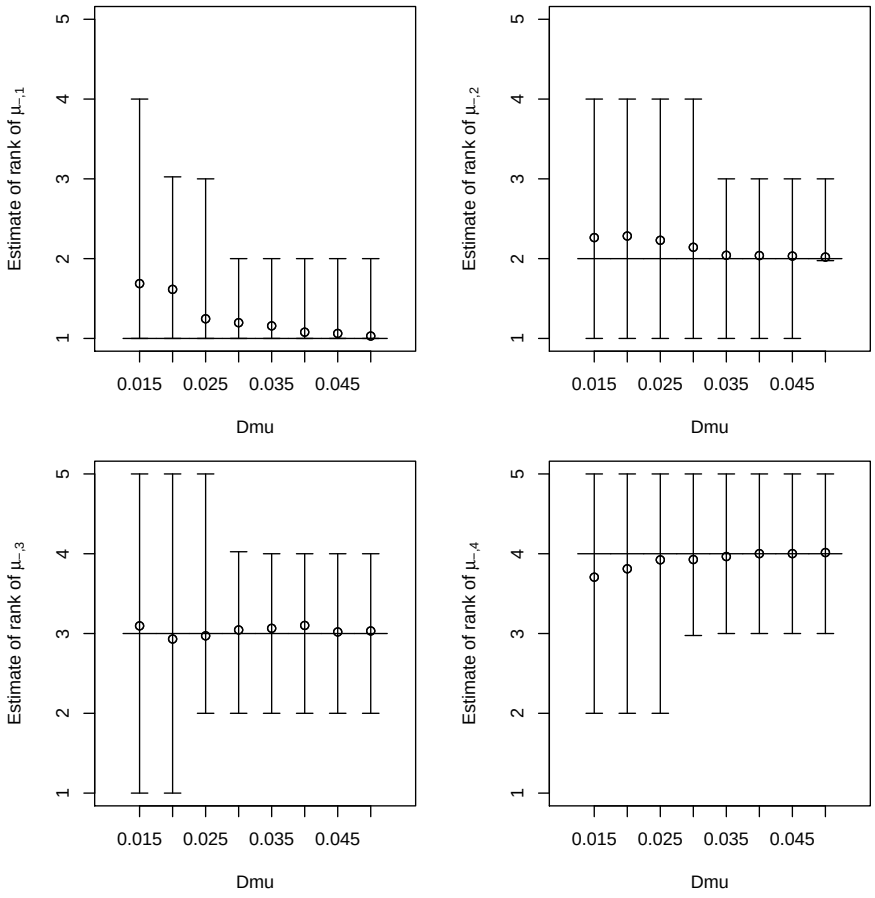

Figure 4.18: Left to right, and then top to bottom, summaries of the posterior results of the rank of $\mu_{1,-}, \cdots, \mu_{4,-}$ as Dmu changes from .01 to .05.

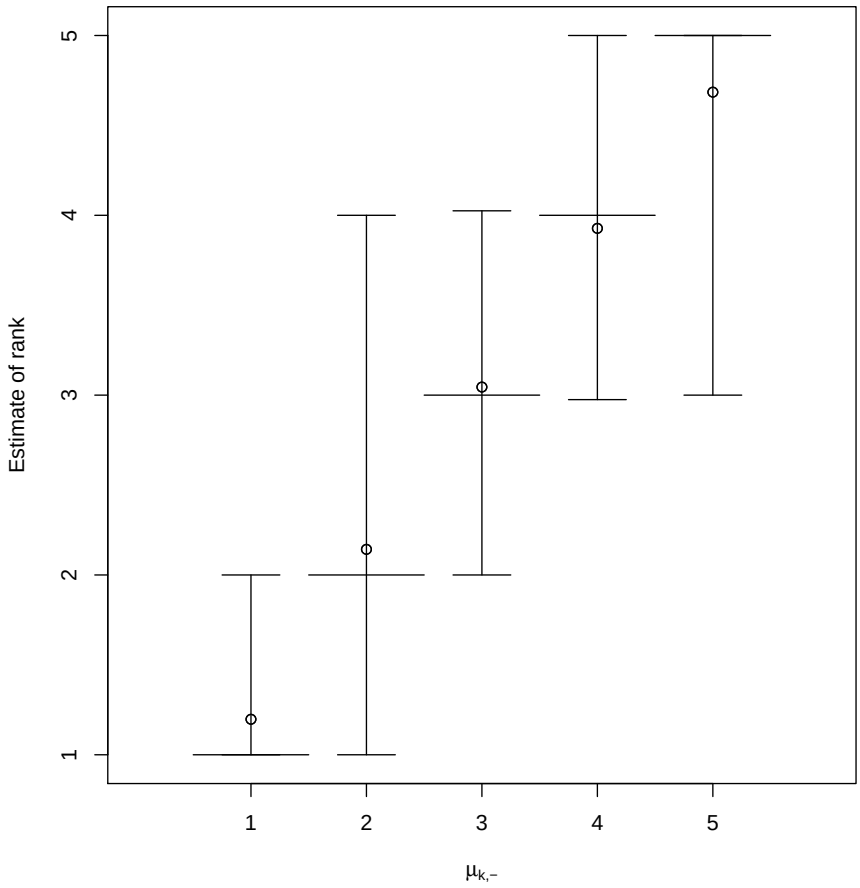

Figure 4.19: Summary of the posterior results of the ranks of $\mu_{1,-}, \cdots, \mu_{5,-}$ at $Dmu = 0.03$.

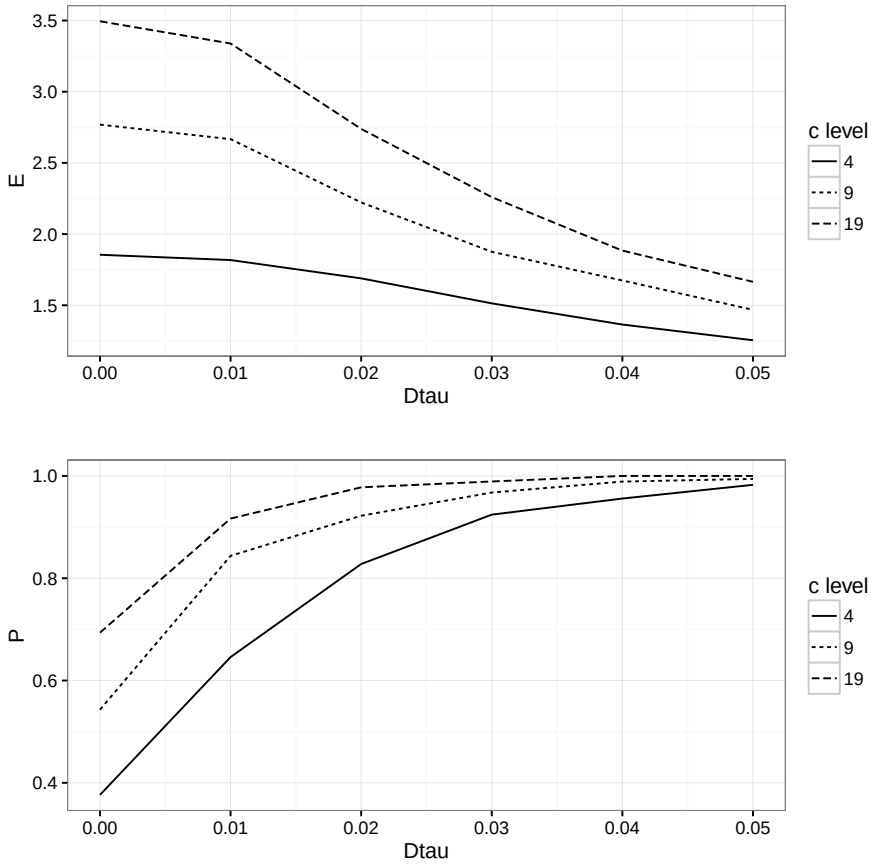

Figure 4.20: Summary of the posterior results, above, the Expected value of the best selection subset as the spacing of the τs, *Dtau*, changes from 0 to .05. Below, the probability that the best τ is included in the superior subset as *Dtau* changes.

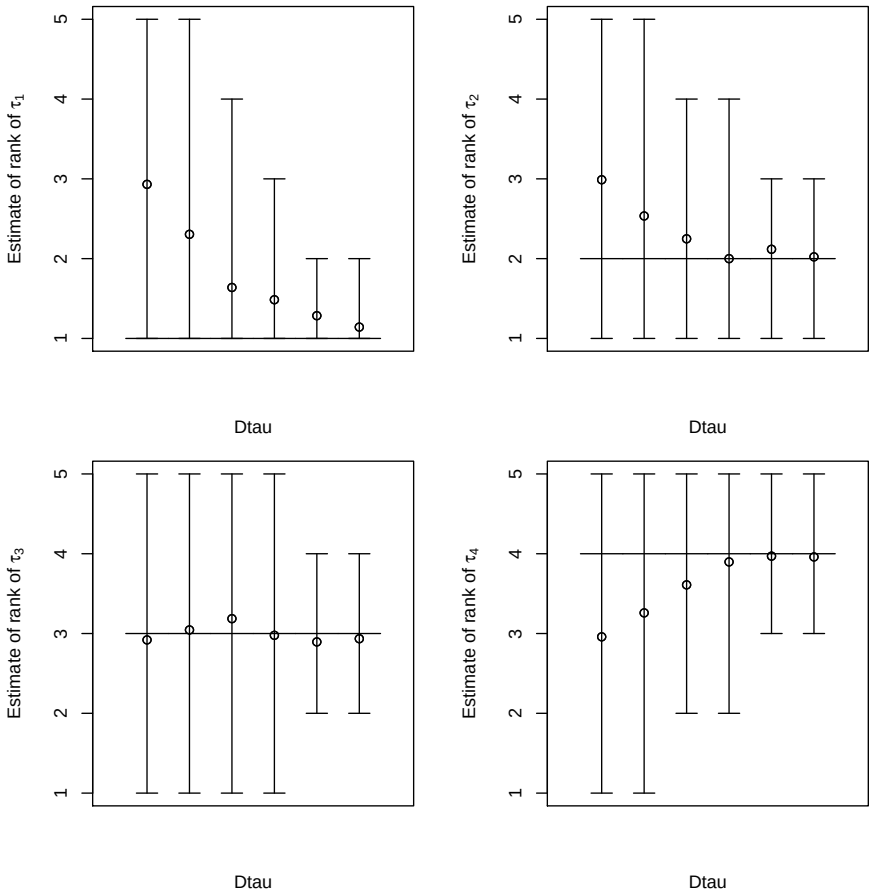

Figure 4.21: Left to right, and then top to bottom, summaries of the posterior results of the rank of $\tau_-, \cdots, \tau_4$ as $Dtau$ changes from 0 to .05.

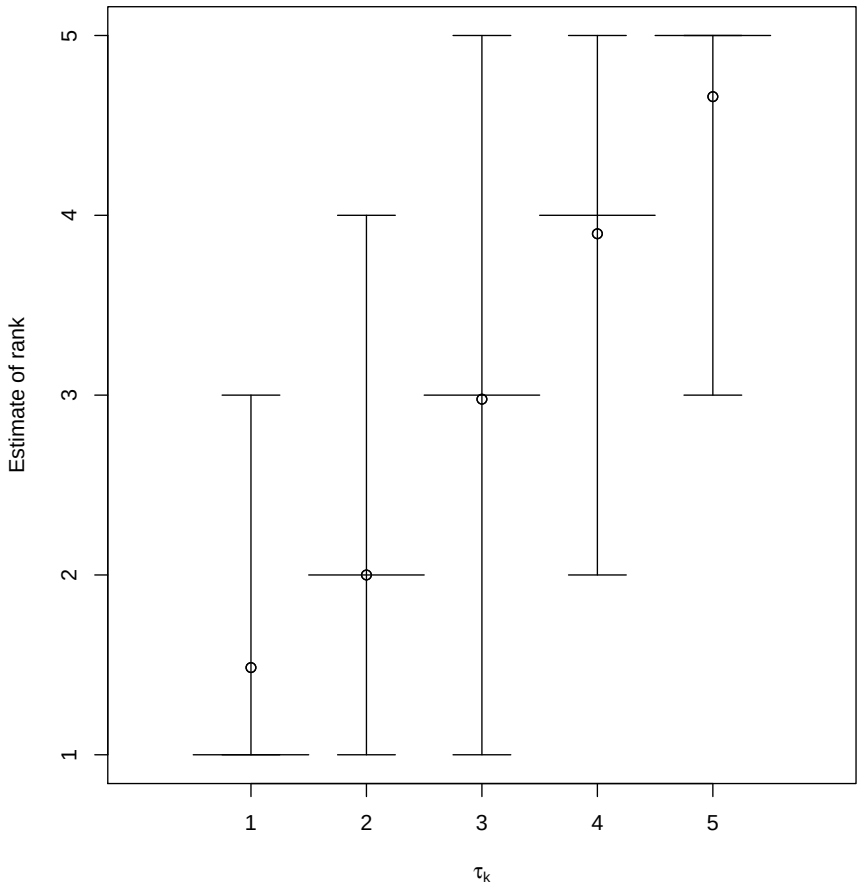

Figure 4.22: Summary of the posterior results of the ranks of $\tau_1, \cdots, \tau_5$ at $Dtau = 0.03$.

CHAPTER FIVE

Conclusion

In Chapter Two, we developed a Bayesian single inspector model that incorporates baseline data. An example analysis was shown in Section 2.3 and a simulation study was done in Section 2.4. The simulation study demonstrates the superiority of the model over one in which the baseline data is not incorporated.

In Chapter Three, we developed a Bayesian fixed effects model for multiple inspectors. A simulation study was performed in Section 3.3 which demonstrated the performance characteristics of the model. In Section 3.4, an algorithm for sample size determination for the model was presented, and an example was worked out.

In Chapter Four, we developed a Bayesian random effects model for multiple inspectors. An example analysis is presented in Section 4.3, and a simulation study was performed in Section 4.4. The simulation study demonstrated good performance characteristics of the model. In Section 4.5, a second model for random effects was presented which incorporates baseline in a way similar to Chapter Two. We showed with a simulation study that this model outperforms a model without the baseline data. Lastly, a ranking and best subset selection model was presented in Section 4.6. It was accompanied by a small simulation study showing the the model works.

Future work includes relaxing the conditional indepedence assumption so criticized by de Mast et al. (2011) among others. This has been attempted by Danila et al. (2012), but it is unclear if it is better to gradually relax the conditional independence assumption while working within the context of a BMS, or if it is better to start with a latent trait model with continuous "goodness" measurements and gradually work towards a BMS.

APPENDICES

APPENDIX A

Additional Material for BMS Incorporating Existing Baseline Data

A.1 JAGS code

The JAGS code that includes the baseline data is below.

```
# parameters that are fixed before the experiment:
# Ngold: number of gold standard inspected items in sample
# N0: number of items examined that failed first inspection
# N1: the number of items examined that passed first inspection
# l: the number of inspections per inspecter
#
# data
# Gstdpart: goodness of item, result of gold standard inspection,
#           1=good item & 0=bad item
# firstI: results of first inspection:
#         1=passed inspection, 0=failed inspection
# obs: vector of experimental results: obs[ i ] is the number
#      of times ith part passed inspection
#
# numpassed: number of items that passed overall inspection system
# N: the number of items inspected by overall inspection system
#
# unknown variables
# unkpart: latent variable, goodness of item,
#          1=good item & 0=bad item
# stau: probability that a sample item is a good item
```

```
# tau: proportion of good items generated by manufacturing process
# thetan: false negative rate
# thetap: false positive rate
#
model{
  # first for gold standard data
  for(i in 1:Ngold){
    Gstdpart[i] ~ dbern(GI[i])
    GI[i] <- firstI[i]* tau*(1-thetan)/(tau*(1-thetan)+
             (1-tau)*thetap) +(1-firstI[i])*tau*thetan/
             ((1-tau)*(1-thetap)+tau*thetan)
    # GI is the probability that the part is good.
    # It takes the intial inspection into account
    # if the initial inspection is good then
    # it is P(good|passed) if the initial inspection is
    # failed then it is P(good|failed)
#
  obs[ i ] ~ dbin( AR[ i ], 1 ) # acceptance rate
  AR[ i ] <- Gstdpart[i]*(1 - thetan) +
             (1-Gstdpart[i])*thetap
  }
# now for the truly random effects data
#
  for(i in (Ngold+1):(N0+N1)){
    unkpart[i - Ngold] ~ dbern(GI[i])
    GI[i] <- firstI[i]* tau*(1-thetan)/(tau*(1-thetan)+
             (1-tau)*thetap) + (1-firstI[i])*tau*thetan/
```

```
            ((1-tau)*(1-thetap)+tau*thetan)
    # GI is the probability that the part is good.
    # if the initial inspection is good then
    # it is P(good|passed) if the initial inspection is
    # failed then it is P(good|failed)

 obs[ i ] ~ dbin( AR[ i ], 1 ) # acceptance rate
 AR[ i ] <- unkpart[i - Ngold]*(1 - thetan) +
            (1-unkpart[i - Ngold])*thetap
  }
#
# taking known overall system pass rate into account
#
  p <- tau*(1-thetan)+(1-tau)*thetap
  numpassed ~ dbin( p, Npop)
#
# priors
#
  thetan ~ dbeta(1,1)
  thetap ~ dbeta(1,1)
  tau ~ dunif(0,1)
#
# additional parameter(s) of interest
  stau <- mean(GI)
}
```

The JAGS code that does not include the baseline data is the same as the one above with the exceptions below.

```
#in first for loop
Gstdpart[i] ~ dbern(stau)
#
#in second for loop
unkpart[i - Ngold] ~ dbern(stau)
#
#lastly
stau ~ dunif(0,1)
```

A.2 Full Conditional Distributions

The full conditional distributions necessary for the Metropolis-Hastings algorithm are below.

$$
\begin{aligned}
f(\theta_-|\theta_+,\tau,\mathbf{t},\xi,\mathbf{x},N_{pass},N_{pop}) \propto & (1-\theta_-)^{\sum_{i=1}^{N} t_i x_i+\beta_- -1}(\theta_-)^{\sum_{i=1}^{N} t_i(l_i-x_i)+\alpha_- -1} \\
& \times \left(\frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)}\right)^{\sum_{i=1}^{N_0} t_i} \\
& \times \left(1 - \frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)}\right)^{N_0 - \sum_{i=1}^{N_0} t_i} \\
& \times \left(\frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+}\right)^{\sum_{i=N_0+1}^{N} t_i} \\
& \times \left(1 - \frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+}\right)^{N_1 - \sum_{i=N_0+1}^{N} t_i} \\
& \times (\tau(1-\theta_-) + (1-\tau)\theta_+)^{N_{pass}} \\
& \times (1 - \tau(1-\theta_-) - (1-\tau)\theta_+)^{N_{pop}-N_{pass}} .
\end{aligned}
$$

$$
\begin{aligned}
f(\theta_+|\theta_-,\tau,\mathbf{t},\xi,\mathbf{x},N_{pass},N_{pop}) \propto & (\theta_+)^{\sum_{i=1}^{N}(1-t_i)x_i+\alpha_+-1}(1-\theta_+)^{\sum_{i=1}^{N}(1-t_i)(l_i-x_i)+\beta_+-1} \\
& \times \left(\frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)}\right)^{\sum_{i=1}^{N_0} t_i} \\
& \times \left(1 - \frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)}\right)^{N_0-\sum_{i=1}^{N_0} t_i} \\
& \times \left(\frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+}\right)^{\sum_{i=N_0+1}^{N} t_i} \\
& \times \left(1 - \frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+}\right)^{N_1-\sum_{i=N_0+1}^{N} t_i} \\
& \times (\tau(1-\theta_-) + (1-\tau)\theta_+)^{N_{pass}} \\
& \times (1-\tau(1-\theta_-) - (1-\tau)\theta_+)^{N_{pop}-N_{pass}} .
\end{aligned}
$$

$$
\begin{aligned}
f(\tau|\theta_-,\theta_+,\mathbf{t},\xi,\mathbf{x},N_{pass},N_{pop}) \propto & \tau^{\alpha_\tau-1}(1-\tau)^{\beta_\tau-1} \\
& \times \left(\frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)}\right)^{\sum_{i=1}^{N_0} t_i} \\
& \times \left(1 - \frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)}\right)^{N_0-\sum_{i=1}^{N_0} t_i} \\
& \times \left(\frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+}\right)^{\sum_{i=N_0+1}^{N} t_i} \\
& \times \left(1 - \frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+}\right)^{N_1-\sum_{i=N_0+1}^{N} t_i} \\
& \times (\tau(1-\theta_-) + (1-\tau)\theta_+)^{N_{pass}} \\
& \times (1-\tau(1-\theta_-) - (1-\tau)\theta_+)^{N_{pop}-N_{pass}} .
\end{aligned}
$$

$$
\begin{aligned}
f(t_i|\theta_-,\theta_+,\tau,\xi,\mathbf{x},N_{pass},N_{pop}) \propto & (1-\theta_-)^{t_i x_i}(\theta_-)^{t_i(l_i-x_i)} \\
& \times (\theta_+)^{(1-t_i)x_i}(1-\theta_+)^{(1-t_i)(l_i-x_i)} \\
& \times \left(\frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)}\right)^{(1-\xi_i)t_i} \\
& \times \left(1-\frac{\tau\theta_-}{\tau\theta_- + (1-\tau)(1-\theta_+)}\right)^{(1-\xi_i)(1-t_i)} \\
& \times \left(\frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+}\right)^{\xi_i t_i} \\
& \times \left(1-\frac{\tau(1-\theta_-)}{\tau(1-\theta_-) + (1-\tau)\theta_+}\right)^{\xi_i(1-t_i)}.
\end{aligned}
$$

A.3 R Code to Simulate BMS incorporating existing baseline data

The random sample is generated via the code below.

```
###  gensamplefirstI  ###
# Note for below: if the part is good then
# the worry is the false negative rate while if
# the part is bad the concern is the false positive
# rate. Hence for good parts the _n's
# for false negatives are used and vice versa
# for bad parts.  Btw obs is the number
# of observed passes. N1 is the number of parts to sample
# from parts that passed first inspection,
# N0 is the number to sample that failed first inspection,
# tau is the proportion of good parts,
# an and bn are the parameters for the thetan beta distribution,
# ap and bp are the parameters for the thetap beta distribution,
# m is the number of inspecter in experiment,
# l is the number of inspections per inspecter,
# inspectorerrorn is a vector of the false negative rates
```

```
# for the m inspectors (gened from beta),
# and inspectorerrorp " " positive rates for the m
# inspectors (generated from beta).
#
# example of a run
# N0=100; N1=0; tau=.95;  l=11;
# thetan = .1; thetap = .1
# gensamplefirstI(N0,N1,tau,l,thetan,thetap)
#
gensamplefirstI <-function(N0,N1,tau,l,thetan,thetap){
  parts=mat.or.vec(nr=(N0+N1),nc=2)  #for final output
  obs  =mat.or.vec(nr=(N0+N1),nc=1)
  colnames(parts) <- c("itemqual","inspect")
  n0=0
  n1=0
  Nsamp=0
  while(n0<N0 | n1<N1){
    Nsamp=Nsamp+1
    contpart=rnorm(1)
    binpart=ifelse(contpart<qnorm(tau),1,0)
    pass=binpart*rbinom(1,size=1,prob=1 - thetan)+
      (1-binpart)*rbinom(1,size=1,prob=thetap)
    #prob of passing inspection
    if(pass & n1<N1){
      n1=n1+1
      parts[(n0+n1),1]=binpart
      parts[(n0+n1),2]=pass
```

```
    } else if(!pass & n0<N0){
      n0=n0+1
      parts[(n0+n1),1]=binpart
      parts[(n0+n1),2]=pass
    }
  }
  #
  for(i in 1:(N0+N1)){

    if(parts[i,1]){
      obs[ i ] = rbinom(1, l, 1 - thetan)
    }else{
      obs[ i ] = rbinom(1, l, thetap)
    }
  }
  list(parts=parts,obs=obs)
}
```

JAGS is run with the following code.

```
require("rjags")
basedir=getwd()
#data directory is where I'm storing the data from the simulations:
datadir= getwd()
#the model files:
modelfile.wFirstI      = paste(basedir,"/FirstIjags.txt",sep="")
modelfile.noFirstI     = paste(basedir,"/noFirstIjags.txt",sep="")
#
setwd(datadir)
```

```
#
#Creating a matrix of all design pt combinations
#first set of design pts
tau=c(.90)
thetan=.1
thetap=.1
l=11         # number of inspections
N0=c(200)    # number of items to be inspected that failed FI
N1=c(0)      # number of items to be inspected that passed FI
Npop=100000  # total number of items inspected at plant
Gstd=c(0)    # proportion of gold standard data available
#
simcomb=expand.grid(tau,thetan,thetap,l,N0,N1,Npop,Gstd)
#
#
#for MCMC
REthinning=25
REadapt.n = 5000
REiter.n = 25000
#
numcombinations=dim(simcomb)[1]
simcomb=cbind(simcomb,rep(REiter.n,numcombinations),
              rep(REthinning,numcombinations),
              rep(REadapt.n,numcombinations))
names(simcomb)<-c("tau","thetan","thetap","l",
                  "N0","N1","Npop","Gstd",
                  "mciterations","thinning","burnin")
```

```
#
for(designpt in 1:numcombinations){
  rm(mystats.FI)
  rm(mystats.noFI)
  mystats.FI=list()
  mystats.noFI=list()
  tau       = simcomb$tau[designpt]
  thetan    = simcomb$thetan[designpt]
  thetap    = simcomb$thetap[designpt] e
  #
  #first generate some random data
  simruns = 2
  l       = simcomb$l[designpt]
  N0      = simcomb$N0[designpt]
  N1      = simcomb$N1[designpt]
  Gstd    = simcomb$Gstd[designpt]
  Npop    = simcomb$Npop[designpt]
  Ngold   = floor(Gstd*(N0+N1))
  #
  obs= mat.or.vec(nr=(N0+N1),nc=1)
  basename=paste("t",simcomb$tau[designpt],"thetan",
                 simcomb$thetan[designpt],
                 "thetap",simcomb$thetap[designpt],
                 "l",simcomb$l[designpt],"N0",
                 simcomb$N0[designpt],"N1",
                 simcomb$N1[designpt],"Npk",
                 simcomb$Npop[designpt]/1000,
```

```
                "runs",simruns,"MCiter",
                simcomb$mciterations[designpt],
                "thin",simcomb$thinning[designpt],
                sep="")
#
for(iterations in 1:simruns){
  #printing status
  print(paste("Design point: ", designpt, " Iteration: ",
              iterations, " thetan: ",thetan,
              " thetap: ", thetap," l: ", l," N0: ",N0,
              " N1: ",N1," Time: ",Sys.time()))
  #
  ###   Generation of random data  ###
  #
  # obs is the number of observed passes.
  sample=gensamplefirstI(N0,N1,tau,l,thetan,thetap)
  #
  obs=sample$obs
  firstinspect=sample$parts[,2]
  Gstdpart=sample$parts[1:Ngold,1]
  unkpart=rep("NA",(N0+N1)-Ngold)
  numpassed = rbinom(1,Npop,tau*(1-thetan)+(1-tau)*thetap)
  #
  ### part 1: considering initial inspection  ###
  #
  data = list(numpassed=numpassed,Npop=Npop,N0=N0,N1=N1,Ngold=0,
              l=l,obs=obs,Gstdpart=0,firstI=firstinspect)
```

```
#
inits = list( tau=runif(1,.75,.95),thetan=.1,
              unkpart=  rbinom((N0+N1),1,.5), thetap=.1)
#
#simulation parameters
thinning = REthinning
adapt.n  = REadapt.n
iter.n   = REiter.n
paramofint=c("tau","stau","thetan","thetap")
#
# using while loop with tryCatch to repeat the
# simulation attempt up to 25 times; the idea is that
# occasionally the MCMC gets fouled up and needs a restart
# with the same data
repeats=0
while(repeats<25){
  tryCatch({
    repeats=repeats+1
    Jag.obj=jags.model(file=modelfile.wFirstI, data=data,
                       inits=inits, n.chains = 1,
                       n.adapt=adapt.n, quiet=FALSE)
    output.coda.FI=coda.samples(Jag.obj, paramofint,
                     n.iter=iter.n, thin = thinning)
    repeats=100
    #if it makes it to here then break out of while loop
  },
  error=function(e){
```

```
    repeats=repeats+1
  })
}
#
### part 2: not considering initial inspection ###
#
data = list(numpassed=numpassed,Npop=Npop,N0=N0,N1=N1,
            Ngold=0,l=l,obs=obs,Gstdpart=0)
inits =     list( tau=runif(1,.75,.95), thetan=.1,
                  unkpart=  rbinom((N0+N1),1,.5),
                  thetap=.1, stau=runif(1,.5,.95))
#
#simulation parameters
paramofint=c("tau","stau","thetan","thetap")
thinning = REthinning
adapt.n  = REadapt.n
iter.n   = REiter.n *2
#
# using while loop with tryCatch to repeat the
# simulation attempt up to 25 times
repeats=0
while(repeats<25){
  tryCatch({
    repeats=repeats+1
    Jag.obj=jags.model(file=modelfile.noFirstI, data=data,
                       inits=inits, n.chains = 1,
                       n.adapt=adapt.n, quiet=FALSE)
```

```
        output.coda.noFI=coda.samples(Jag.obj, paramofint,
                                      n.iter=iter.n, thin = thinning)
        repeats=100
        #if it makes it to here then break out of while loop
      },
      error=function(e){
        repeats=repeats+1
      })
    }
    #
    sum1=summary(output.coda.FI)
    sum2=summary(output.coda.noFI)
    mystats.FI[[iterations]] = cbind(sum1[[1]],sum1[[2]])
    mystats.noFI[[iterations]]  = cbind(sum2[[1]],sum2[[2]])
  }
  #
  lencoda=length(mystats.FI)
  mystats.FI[[lencoda+1]]=simcomb[designpt,]
  summarystatsname=paste("wFI",basename,".txt",sep="")
  dput(mystats.FI,summarystatsname)
  lencoda=length(mystats.noFI)
  mystats.noFI[[lencoda+1]]=simcomb[designpt,]
  summarystatsname=paste("noFI",basename,".txt",sep="")
  dput(mystats.noFI,summarystatsname)
}
```

```
require("magrittr")
#drive
drive="F:/"
produceplots=FALSE
errorcode=-10000
#
#data directory is the data from the simulations is stored
datadir=getwd()
#
#plots directory
plotsdir=getwd()
#
###    plotsim   ###
#code originally written by Johnny Seaman later modified
#by Ross Bray and myself. Below you will see the code for
#an R function to produce summary graphics for simulations
#input takes comprises posterior means, medians, .025
#and .975 percentiles from each simulation replication.
#errorcode variable was added to allow the omission of
#certain data points if a data point = errorcode than
#it is omitted from median and sd calculations
#
plotsim=function(data, param, label, ylims, error=TRUE,
                 color="grey", trans=.75,xlab="",ylab="",
                 cparam=0,errorcode=-10000)
{
```

```
total=ncol(data)/3;#Counts number of simulations to plot
# storage matrices for medians and sd's
meds=matrix(NA,nrow=total,ncol=3);
stdev=matrix(NA,nrow=total,ncol=3);
#
#Loop to calc medians and standard deviations from sims
for(i in 1:total)
{
  meds[i,1]=median(data[,1+(i-1)*3][which(
            data[,1+(i-1)*3] != errorcode)])
  meds[i,2]=median(data[,2+(i-1)*3][which(
            data[,1+(i-1)*3] != errorcode)])
  meds[i,3]=median(data[,3+(i-1)*3][which(
            data[,1+(i-1)*3] != errorcode)])
  stdev[i,1]=sd(data[,1+(i-1)*3][which(
             data[,1+(i-1)*3] != errorcode)])
  stdev[i,2]=sd(data[,2+(i-1)*3][which(
             data[,1+(i-1)*3] != errorcode)])
  stdev[i,3]=sd(data[,3+(i-1)*3][which(
             data[,1+(i-1)*3] != errorcode)])
}
upper=meds+stdev  #+1 sd above the mean
lower=meds-stdev  #-1 sd below the mean
plot(c(1:total),meds[,1],xlim=c(0,total+1), ylim=ylims,
    xlab=xlab,ylab=ylab,xaxt='n',main=label)
for(i in 1:total)
{
```

```
  if(error==TRUE)
  {#Rectangle around median
    rect(xleft=i-.1,xright=i+.1,ytop= upper[i,1],
         ybottom=lower[i,1] , col="grey",border=FALSE);
    rect(xleft=i-.1,xright=i+.1,ytop= upper[i,2],
         ybottom=lower[i,2] , col="grey", border=FALSE);

    #Rectangle around upper bound
    rect(xleft=i-.1,xright=i+.1,ytop= upper[i,3],
         ybottom=lower[i,3] , col="grey", border=FALSE);
  }
  lines(x=c(i,i),y=c(meds[i,2],meds[i,3]),lty=1)
  lines(x=c(i-.25,i+.25),y=c(meds[i,2],meds[i,2]),lty=1)
  lines(x=c(i-.25,i+.25),y=c(meds[i,3],meds[i,3]),lty=1)
}
points(c(1:total),meds[,1])    #Replot medians
#Plot true value of parameter
if(length(param) == 1){
  lines(y=c(param,param),x=c(0,15))
}
if(length(param) > 1){
  for(i in 1:total){
    lines(x=c(i-.5,i+.5),y=c(param[i],param[i]),lty=1)
  }
}
if(cparam[1]!=0){
  axis(1,at=1:total,labels=cparam)
```

```
  }
}
#
#
###      fileextraction    ###
# loading up simulation data
# the simulation parameters for each simulation run are
# appended to the list of simulation summaries.  Thus,
# to get the simulation parameters a list is generated
# of the simulation files in the dir followed by
# reading each in one by one via dget() and the last
# element in the list i.e. the simulation parameters
# is put into a simparam list.  The purpose of this is
# so as not to overload the memory by loading up
# everything at once.  Only the simulations desired for
# a particular plot need to be loaded up. possible
# paramslices: tau thetan thetap  l  N0 N1  Npop Gstd
# mciterations thinning burnin
#
fileextraction<-function(dir,pattern="wFIt0.*.txt",
                         paramslice="tau"){
  setwd(dir)
  data.files=list.files(pattern = pattern)
  numfiles=length(data.files)
  #
  # extracting the simulation parameters
  rm(simparam)
```

```
  simparam=list()
  numsims=0
  for(i in 1:numfiles){
    temp=dget(data.files[i])
    lendatafile=length(temp)
    simparam[[i]]=temp[[lendatafile]]
    newnumsims=lendatafile-1
    if(numsims!=0 & numsims!=newnumsims){ stop }
    numsims=newnumsims
  }
#
# here is where I select what points I want for
# a contrast/comparison or whatever note that it's
# possible for not all pts to actually exist in
# which case only the pts that do exist will be used.
# Points that don't actually exist will have the
# value of 0 in the mapping
#
  simparammat=Reduce(rbind,simparam)
  baselevel <- apply(simparammat,2,function(x){
    ux <- unique(x)
    ux[which.max(tabulate(match(x, ux)))]
    })
  #having to add this for the N1 slice
  if(paramslice=="N1"){
    baselevel["N0"]=0
    baselevel["N1"]=200
```

```
  }
  uniquesliceparam=sort(unique(simparammat[,paramslice]))
  numdesired.pts=length(uniquesliceparam)
  desired.pts = matrix(rep(baselevel,numdesired.pts),
                       byrow=TRUE,nrow=numdesired.pts)
  colnames(desired.pts) <- names(baselevel)
  desired.pts[,paramslice]=uniquesliceparam
  #
  mapping=mat.or.vec(nr=numdesired.pts,nc=1)
  for( i in 1:numdesired.pts){
    for( j in 1:numfiles){
      if( desired.pts[i,1] == simparam[[j]][1] &
          desired.pts[i,2] == simparam[[j]][2] &
          desired.pts[i,3] == simparam[[j]][3] &
          desired.pts[i,4] == simparam[[j]][4] &
          desired.pts[i,5] == simparam[[j]][5] &
          desired.pts[i,6] == simparam[[j]][6] &
          desired.pts[i,7] == simparam[[j]][7])
      { mapping[i]=j }
    }
  }
  desired.pts=as.data.frame(cbind(desired.pts,mapping))
#
# now to wipe out point that don't exist from list
  desired.pts=desired.pts[which(!desired.pts$mapping==0),]
  numdesired.pts=dim(desired.pts)[1]
  list(desired.pts=desired.pts,numdesired.pts=
```

```
       numdesired.pts,numsims=numsims,
       data.files=data.files,numfiles=numfiles)
}
#
###   prepareplots    ###
#This function calculates coverage and generates output
#compatible with the plotsim function above.
#
prepareplots<-function(dataframe){
  desired.pts=dataframe$desired.pts
  numdesired.pts=dataframe$numdesired.pts
  numsims=dataframe$numsims
  data.files=dataframe$data.files
  numfiles=dataframe$numfiles
  desired.pts=cbind(desired.pts,mat.or.vec(
                   nr=dim(desired.pts)[1],nc=4))
  plotbiasidxkeep=2:4
  for( i in 1:numdesired.pts ){
    #First pull in relavent data if it exists
    if(desired.pts[i,]$mapping != 0){
      mystats = dget(data.files[desired.pts[i,]$mapping])
      myrownames<-mystats[[1]]   %>% rownames %>%
        gsub("]","",.) %>%
        gsub("\\[","",.)  %>% gsub(",","",.)
      lendata = length(mystats) - 1
      N0 = desired.pts[i,]$N0
      N1 = desired.pts[i,]$N1
```

```
thetan =  desired.pts[i,]$thetan
thetap =  desired.pts[i,]$thetap
tau = desired.pts[i,]$tau
stau = (N1* tau*(1-thetan)/(tau*(1-thetan)+
       (1-tau)*thetap) + N0*tau*thetan/
       ((1-tau)*(1-thetap)+tau*thetan))/(N1+N0)
#
#checking for inversion of misclassification/
# classification rates first
goodruns=c(unlist(lapply(mystats[-(numsims+1)],
         function(x){ifelse(max(x[3:4,1])>.5, 0, 1)})),0)
goodidx=which(goodruns==1)
#
#throwing out inverted data points
mystats=mystats[goodidx]
#
#calculating coverages
#first putting true values into a vector
truevec=c(stau,tau,thetan,thetap)
#
# now using lapply and ifelse to compare the
# 95% credible sets to the true values
cover=lapply(mystats,function(x){
  ifelse(x[,5] <= truevec & x[,9] >= truevec, 1, 0)
})
#
#and lastly averaging the lists to find the coverages
```

```
coverages=(Reduce('+',cover)/length(mystats))
desired.pts[i,13:16]=coverages
#
# now to loop through list and extract upper
# and lower bounds plus means
myrossdata=Reduce(cbind,lapply(mystats,
                  function(x){x[,c(1,5,9)]}))
# array manipulation to create matrices
# for ross' plot function
rossplot=lapply(as.list(rownames(mystats[[1]])),
  function(x){
  mat=matrix(myrossdata[x,],byrow=TRUE,ncol=3)
  if(dim(mat)[1]<numsims){
    mat=rbind(mat,matrix(rep(errorcode,
              3*(numsims-dim(mat)[1])),ncol=3))
  }
  colnames(mat)<-colnames(myrossdata[,1:3])
  mat
})
names(rossplot) <- paste(myrownames,"ross",sep="")
#
#data for the bias plots
myrossbiasdata=myrossdata - truevec
#
#array manipulation to create matrices for
# ross' plot function
rossbiasplot=lapply(as.list(rownames(mystats[[1]])),
```

```
                                function(x){
      mat=matrix(myrossbiasdata[x,],byrow=TRUE,ncol=3)
      if(dim(mat)[1]<numsims){
        mat=rbind(mat,matrix(rep(errorcode,
                  3*(numsims-dim(mat)[1])),ncol=3))
      }
      colnames(mat)<-colnames(myrossbiasdata[,1:3])
      mat
    })
    names(rossbiasplot) <- paste(myrownames,
                                 "rossbias",sep="")
    #
    # combining the list of matrices and deleting
    # uninteresting ones
    rossbiasplot = rossbiasplot[plotbiasidxkeep]
    allrossplot=c(rossplot,rossbiasplot)
    #appending results to a larger list object
    if(i==1){
      allrossplots=allrossplot
    }  else {
      allrossplots=mapply(cbind,allrossplots,
                          allrossplot,SIMPLIFY=FALSE)
    }
  }
}
names(desired.pts)[13:16]<-paste(myrownames,"cov",sep="")
list(desired.pts.results=cbind(desired.pts,0),
```

```
                                allrossplots=allrossplots)
}
#
wFI.names=list("stauross",expression(paste("FI ",tau)),
               expression(paste("FI ",theta["-"])),
               expression(paste("FI ",theta["+"])),
               expression(paste("FI ",tau," bias")),
               expression(paste("FI ",theta["-"])," bias"),
               expression(paste("FI ",theta["+"])," bias"))
noFI.names=list("stauross",expression(paste("No FI ",tau)),
               expression(paste("No FI ",theta["-"])),
               expression(paste("No FI ",theta["+"])),
               expression(paste("No FI ",tau," bias")),
               expression(paste("No FI ",theta["-"])," bias"),
               expression(paste("No FI ",theta["+"])," bias"))
ylab.list=list("stauross",expression(paste("Estimate of ",tau)),
              expression(paste("Estimate of ",theta["-"])),
              expression(paste("Estimate of ",theta["+"])),
              expression(paste("Estimate of ",tau," bias")),
              expression(paste("Estimate of ",theta["-"]," bias")),
              expression(paste("Estimate of ",theta["+"]," bias")))
#
# Now an example call:
#
filedata.wFI=fileextraction(datadir,paramslice="tau")
results.wFI=prepareplots(filedata.wFI)
filedata.noFI=fileextraction(datadir,pattern="noFIt0.*.txt",
```

```
              paramslice="tau")
results.noFI=prepareplots(filedata.noFI)
idx=c(17,1,2,3,17,17,17)
#
temp=2
plotsim(results.wFI$allrossplots[[temp]],
        param=results.wFI$desired.pts.results[,idx[temp]],
        label=wFI.names[[temp]],
        ylims=c(quantile(c(results.wFI$allrossplots[[temp
                      ]][which(results.wFI$allrossplots[[
                      temp]]!=-10000)],
                      results.noFI$allrossplots[[temp
                      ]][which(results.noFI$allrossplots[[
                      temp]]!=-10000)]),probs=.01),
                quantile(c(results.wFI$allrossplots[[temp]],
                           results.noFI$allrossplots[[temp]])
                           ,probs=.99)),
        cparam=results.wFI$desired.pts.results$tau,
        xlab=expression(paste(tau)),
        ylab=ylab.list[[temp]])
```

APPENDIX B

Additional Material for Bayesian BMS Fixed Effects Model

B.1 WinBUGS Code for Model with Uniform Priors

```
model{
  for(i in 1:N){
    T[i] ~ dbern(tau)  #T[i]=0 is a defective item
for(j in 1:m){
  obs[ (i-1)*m+ j ] ~ dbin( AR[ i, j ], l ) # acceptance rate
  AR[ i, j ] <- T[i]*(1-thetan[ j ]) + (1-T[i])*thetap[ j ]
}
  }
#
#priors
  for(j in 1:m){
    thetap[ j ] ~ dbeta(1,1)          # false positive rate
    thetan[ j ] ~ dbeta(1,1)         # false negative rate
  }
  tau ~ dunif(0,1)
#
#quantities of interest
thetan13 <- (thetan[1] - thetan[3]) > 0
thetan12 <- (thetan[1] - thetan[2]) > 0
}
```

B.2 Plots for MCMC Assumptions, Fixed Effects Model

As our model was implemented using OpenBUGS, which implements Markov chain Monte Carlo (MCMC) methods, we need to check the convergence of our model.

A series of plots is used to investigate model convergence. These were randomly obtained from the simulation runs. We first obtain history plots, which show the chain of iterations for each parameter. Ideally, these plots vary randomly around the point of convergence for the parameter. Otherwise, the chain has not yet converged. The history plots are shown in Figure B.1. Secondly, we generate density plots, provided in Figure B.2. We hope that these plots are relatively smooth. Lastly, we generate autocorrelation plots, provided in Figure B.3. These plots show the correlation between current chain values and future chain values. We hope that that the autocorrelation quickly goes to zero. The plots in the appendix all behave as expected.

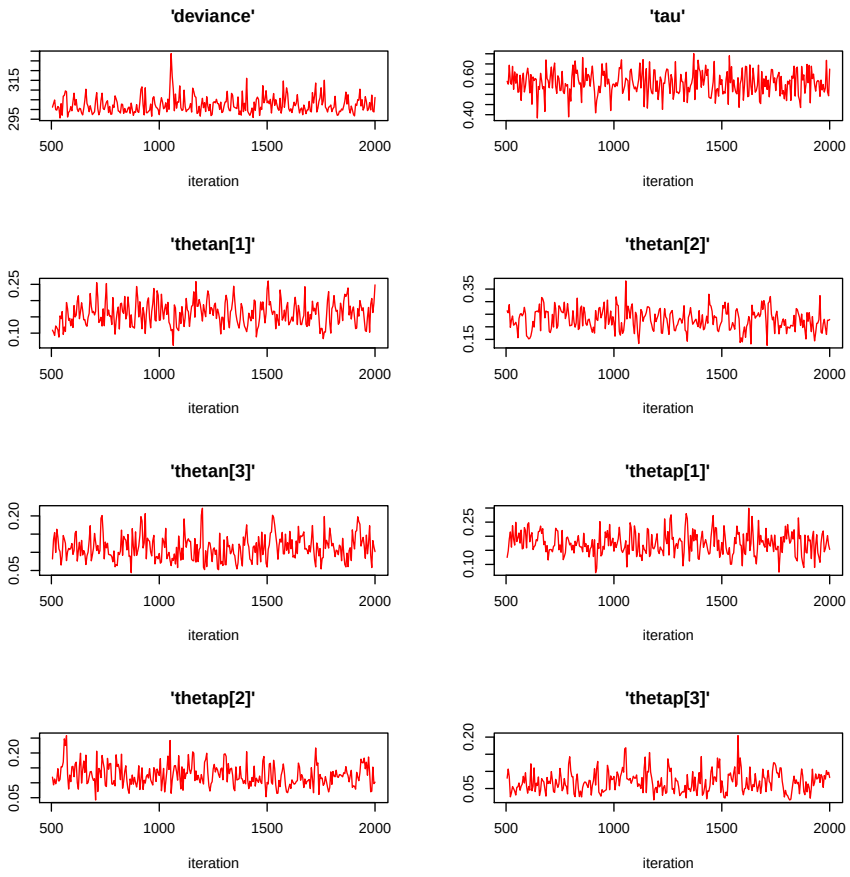

Figure B.1: History plots from a simulation run in which $n = 60$, $\tau = .5$, $m = 3$, $l = 3$, and $\theta_{j+} = \theta_{j-} = .15$.

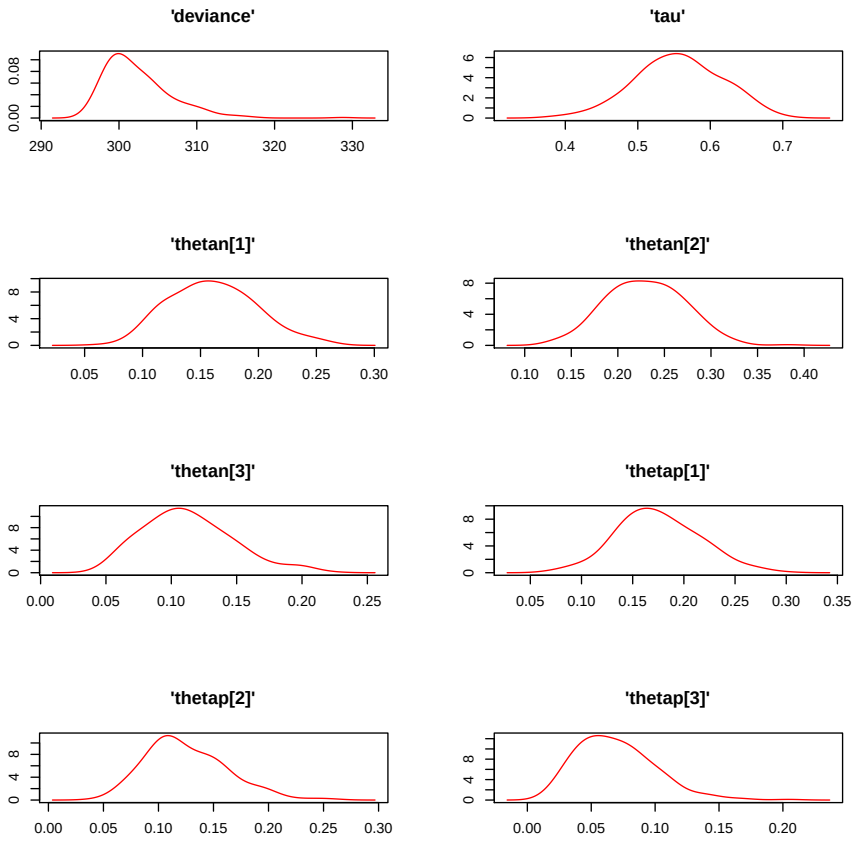

Figure B.2: Density plots from a simulation run in which $n = 60$, $\tau = .5$, $m = 3$, $l = 3$, and $\theta_{j+} = \theta_{j-} = .15$.

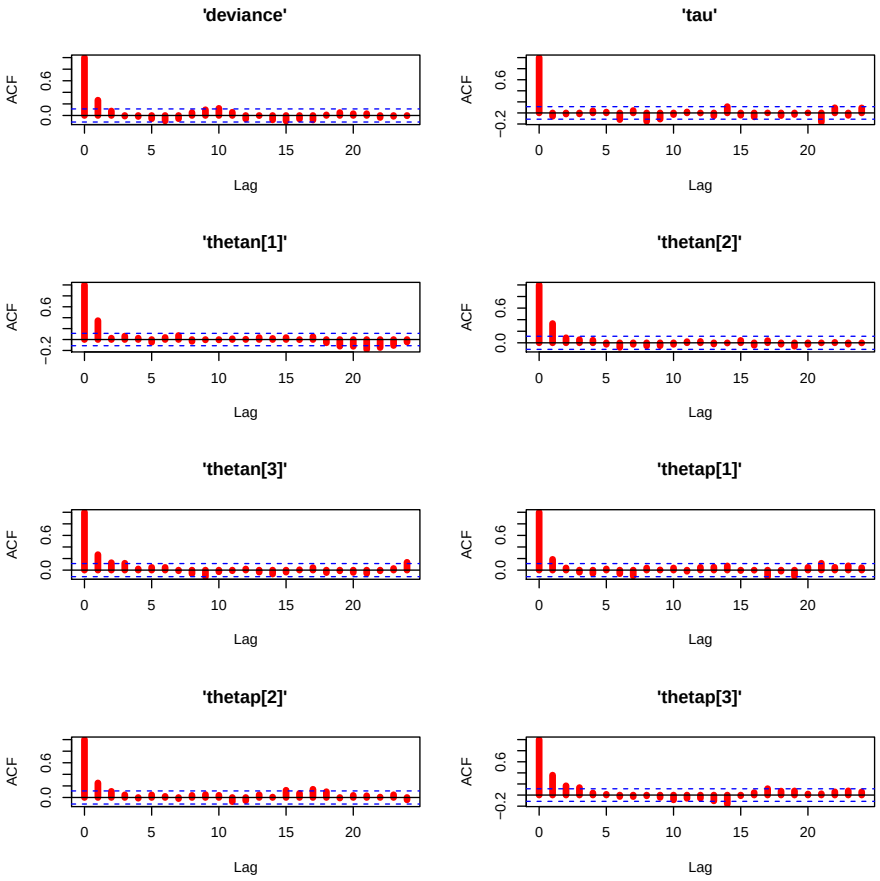

Figure B.3: Autocorrelation plots from a simulation run in which $n = 60$, $\tau = .5$, $m = 3$, $l = 3$, and $\theta_{j+} = \theta_{j-} = .15$.

APPENDIX C

Additional Material for Bayesian BMS Random Effects Model

C.1 JAGS code to implement BMS Random Effects Model

3.1.1 Population based model with uniform priors

```
# variables
# parameters that are fixed before the experiment:
# Ngold: the number of gold standard inspected items in sample
# N: the number of items taken for further examination
# m: the number of inspecters
# l: the number of inspections per inspecter
#
# data
# Gstdpart: goodness of item, result of gold standard
#           inspection, 1=good item & 0=bad item
# obs: vector of experimental results: obs[ (i-1)*m+ j ]
#          is the number of times
#          that the jth inspector passes the ith item
# numpassed: number of items that passed overall inspection system
# Npop: the number of items inspected by overall inspection system
#
# unknown variables
# unkpart: latent variable, goodness of item,
#          1=good item & 0=bad item
# tau: proportion of good items generated by manufacturing process
# thetan[j]: false negative rate of the jth inspecter
```

```
# thetap[j]: false positive rate of the jth inspecter
#
# heirarchical model parameters
# these parameters come from a reparameterized
# beta distribution where
# thetan~beta(mun,gamn), thetap~beta(mup,gamp)
# mup:  mean of inspecters false positive rates
# gamp: roughly inversly proportional to variance
# of inspecter false positive rates
# mun:  mean of inspecters false negative rates
# gamn: roughly inversly proportional to variance of
# inspecter false negative rates
#
model{
  # first for gold standard data
  for(i in 1:Ngold){
    Gstdpart[i] ~ dbern(tau)
  for(j in 1:m){
    obs[ (i-1)*m+ j ] ~ dbin( AR[ i, j ], 1 )
    AR[ i, j ] <- Gstdpart[i]*(1 - thetan[ j ]) +
              (1-Gstdpart[i])*thetap[ j ]
   }
  }
#
# now for the truly random effects data
  for(i in (Ngold+1):N){
    unkpart[i - Ngold] ~ dbern(tau)
```

```
    for(j in 1:m){
    obs[ (i-1)*m+ j ] ~ dbin( AR[ i, j ], l )
    AR[ i, j ] <- unkpart[i - Ngold]*(1 - thetan[ j ])
                 + (1-unkpart[i - Ngold])*thetap[ j ]
   }
  }
  #
  for(j in 1:m){
    thetan[ j ] ~ dbeta(an,bn)
    thetap[ j ] ~ dbeta(ap,bp)
  }
  mup~dunif(0,1)          #mu plus
  gamp~dunif(0,50)        #gamma plus
  mun~dunif(0,1)          #mu negative
  gamn~dunif(0,50)        #gamma negative
  ap<-mup*gamp
  bp<-gamp-mup*gamp
  an<-mun*gamn
  bn<-gamn-mun*gamn
  tau ~ dunif(0,1)
}
}
```

3.1.2 Incorporating baseline data with uniform priors

```
# variables
# parameters that are fixed before the experiment:
# Ngold: the number of gold standard inspected items in sample
```

```
# N0: the number of items taken for further examination
#     that failed first inspection
# N1: the number of items taken for further examination
#     that passed first inspection
# m: the number of inspecters
# l: the number of inspections per inspecter
#
# data
# Gstdpart: goodness of item, result of gold standard
#           inspection, 1=good item & 0=bad item
# firstI: results of first inspection:
#         1=passed inspection, 0=failed inspection
# obs: vector of experimental results: obs[ (i-1)*m+ j ]
#      is the number of times that the jth inspector
#      passes the ith item
# numpassed: number of items that passed overall inspection system
# N: the number of items inspected by overall inspection system
#
# unknown variables
# unkpart: latent variable, goodness of item,
#          1=good item & 0=bad item
# stau: probability that a sample item is a good item
#      (based on intial inspection)
# tau: proportion of good items generated by
#      manufacturing process
# thetan[j]: false negative rate of the jth inspecter
# thetap[j]: false positive rate of the jth inspecter
```

```
#
# heirarchical model parameters
# these parameters come from a reparameterized
# beta distribution where
# thetan~beta(mun,gamn), thetap~beta(mup,gamp)
# mup:  mean of inspecters false positive rates
# gamp: roughly inversly proportional to variance of
#       inspecter false positive rates
# mun:  mean of inspecters false negative rates
# gamn: roughly inversly proportional to variance of
#       inspecter false negative rates
#
model{
  # first for gold standard data
  for(i in 1:Ngold){
    Gstdpart[i] ~ dbern(GI[i])
    GI[i] <- firstI[i]* tau*(1-mun)/(tau*(1-mun)+(1-tau)*mup) +
             (1-firstI[i])*tau*mun/((1-tau)*(1-mup)+tau*mun)
    # GI is the probability that the part is good.
    # It takes the intial inspection into account
    # if the initial inspection is good then it is
    # P(good|passed) if the initial inspection is
    # failed then it is P(good|failed) 1 inspection
    #
  for(j in 1:m){
    obs[ (i-1)*m+ j ] ~ dbin( AR[ i, j ], 1 )
    AR[ i, j ] <- Gstdpart[i]*(1 - thetan[ j ]) +
```

```
                (1-Gstdpart[i])*thetap[ j ]
   }
  }
#
# now for the truly random effects data
  for(i in (Ngold+1):(NO+N1)){
    unkpart[i - Ngold] ~ dbern(GI[i])
    GI[i] <- firstI[i]* tau*(1-mun)/(tau*(1-mun)+(1-tau)*mup) +
             (1-firstI[i])*tau*mun/((1-tau)*(1-mup)+tau*mun)
    # GI is the probability that the part is good.
    # It takes the intial inspection into account
    # if the initial inspection is good then it is
    # P(good|passed) if the initial inspection is
    # failed then it is P(good|failed) 1 inspection
    #
    for(j in 1:m){
    obs[ (i-1)*m+ j ] ~ dbin( AR[ i, j ], 1 )
    AR[ i, j ] <- unkpart[i - Ngold]*(1 - thetan[ j ]) +
                (1-unkpart[i - Ngold])*thetap[ j ]
   }
  }
#
# taking known overall system pass rate into account
  p <- tau*(1-mun)+(1-tau)*mup
  numpassed ~ dbin( p, Npop)
  #
  for(j in 1:m){
```

```
    thetan[ j ] ~ dbeta(an,bn)
    thetap[ j ] ~ dbeta(ap,bp)
  }
  mup~dunif(0,1)          #mu plus
  gamp~dunif(0,50)         #gamma plus
  mun~dunif(0,1)          #mu negative
  gamn~dunif(0,50)         #gamma negative
  ap<-mup*gamp
  bp<-gamp-mup*gamp
  an<-mun*gamn
  bn<-gamn-mun*gamn
  tau ~ dunif(0,1)
  stau <- mean(GI)
}
```

3.1.3 Implementing BMS ranking and subset selection with uniform priors

```
# variables
# parameters that are fixed before the experiment:
# Ngold: the number of gold standard inspected items in sample
# N: the number of items taken for further examination
# m: the number of inspecters
# l: the number of inspections per inspecter
#
# data
# Gstdpart: goodness of item, result of gold standard inspection,
#           1=good item & 0=bad item
# obs: vector of experimental results: obs[ (i-1)*m+ j ] is the
#      number of times that the jth inspector passes the ith item
# numpassed: number of items that passed overall inspection system
# Npop: the number of items inspected by overall inspection system
#
# unknown variables
# unkpart: latent variable, goodness of item,
#          1=good item & 0=bad item
# stau: probability that a sample item is a good item
#       (based on intial inspection)
# tau: proportion of good items generated by manufacturing process
# thetan[j]: false negative rate of the jth inspecter
# thetap[j]: false positive rate of the jth inspecter
#
# heirarchical model parameters
# these parameters come from a reparameterized
```

```
# beta distribution where
# thetan~beta(mun,gamn), thetap~beta(mup,gamp)
# mup:  mean of inspecters false positive rates
# gamp: roughly inversly proportional to variance of
#       inspecter false positive rates
# mun:  mean of inspecters false negative rates
# gamn: roughly inversly proportional to variance of
#       inspecter false negative rates
#
model{
  # first for gold standard data
  for(i in 1:numplants){
    for(j in 1:Ngold){
      Gstdpart[i,j] ~ dbern(tau[i])
    for(k in 1:m[i]){
      obs[ offsets[i] + (j-1)*m[i] + k ] ~
           dbin( AR[ i, j, k ], l )
      AR[ i, j, k ] <- Gstdpart[i,j]*(1 - thetan[ i, k ]) +
                       (1-Gstdpart[i,j])*thetap[ i, k ]
     }
    }
#
# now for the truly random effects data
    for(j in (Ngold+1):N){
      unkpart[i, j - Ngold] ~ dbern( tau[i] )
      for(k in 1:m[i]){
      obs[ offsets[i] + (j-1)*m[i] + k ] ~
```

```
                    dbin( AR[ i, j, k ], 1 )
      AR[ i, j, k ] <- unkpart[i,j - Ngold]*(1 - thetan[ i, k ]) +
                       (1-unkpart[i,j - Ngold])*thetap[ i, k ]
      }
    }
  }
  #
  for(i in 1:numplants){
    for(k in 1:m[i]){
      thetan[ i, k ] ~ dbeta(an[i],bn[i])
      thetap[ i, k ] ~ dbeta(ap[i],bp[i])
    }
    #
    mup[i]  ~ dunif(0,1)          #mu plus
    gamp[i] ~ dunif(0,50)         #gamma plus
    mun[i]  ~ dunif(0,1)          #mu negative
    gamn[i] ~ dunif(0,50)         #gamma negative
    tau[i]  ~ dunif(0,1)
    ap[i]<-mup[i]*gamp[i]
    bp[i]<-gamp[i]-mup[i]*gamp[i]
    an[i]<-mun[i]*gamn[i]
    bn[i]<-gamn[i]-mun[i]*gamn[i]
  }
  muprank <- rank(mup)
  munrank <- rank(mun)
  taurank <- rank(tau)
}
```

```
###  gensamplepop  ###
# Note for below: if the part is good then the worry
# is the false negative rate while if
# the part is bad the concern is the false positive
# rate. Hence for good parts the _n's
# for false negatives are used and vice versa
# for bad parts.  Btw obs is the number
# of observed passes. N is the number of parts to sample
# from the general population,
# tau is the proportion of good parts,
# an and bn are the parameters for the false negative
#    beta distribution,
# ap and bp are the parameters for the false positive
#    beta distribution,
# m is the number of inspecter in experiment, l is the
#    number of inspections per inspecter,
# inspectorerrorn is a vector of the false negative rates
#    for the m inspectors (gened from beta),
# and inspectorerrorp " " positive rates for the m
#    inspectors (gened from beta).
#
# example of a run
# N=100;  tau=.75; an=1.5; bn= 13.5; ap=1.5; bp= 13.5;
# m=7; l=3; inspectorerrorn = c(0.041288, 0.054843,
#    0.044206, 0.032168, 0.02787, 0.033028, 0.060692)
# inspectorerrorp = c(0.042385, 0.21098, 0.11959, 0.019462,
```

```
#    0.016764, 0.025344, 0.090936)
# gensamplepop(N,tau,an,bn,ap,bp,m,l,inspectorerrorn,inspectorerrorp)
#
gensamplepop <-function(N,tau,an,bn,ap,bp,m,l,
                        inspectorerrorn,inspectorerrorp){
  parts=mat.or.vec(nr=N,nc=1)  #to record quality of part
  obs  =mat.or.vec(nr=N*m,nc=1)
  parts=rbinom(N,1,tau)
  #
  for(i in 1:N){
    for(j in 1:m){
      if(parts[i]){
        obs[m*(i-1)+j] = rbinom(1, l, 1 - inspectorerrorn[j])
      }else{
        obs[m*(i-1)+j] = rbinom(1, l, inspectorerrorp[j])
      }
    }
  }
  list(parts=parts,obs=obs)
}
```

C.3 R Code for Random Effects Population Based BMS Example

```
require("rjags")
basedir=getwd()
modelfile.pop       = paste(basedir,"/RandomEFpopjags.txt",sep="")
#
#Parameter values
tau=.8
```

```
mup=.1

mun=.1

gamp=15

gamn=15

m=20          # number of inspectors

l=1           # number of inspections

N=200         # number of items to be inspected

Npop=100000   # total number of items inspected at plant

Gstd=c(0)     # proportion of gold standard data available

Ngold   = floor(Gstd*N)

#

ap     = mup*gamp

bp     = gamp - mup*gamp

an     = mun*gamn

bn     = gamn - mun*gamn

#

#first generate some random data

obs= mat.or.vec(nr=N*m,nc=1)

#

set.seed(1234567)

#

###   Generation of random data  ###

#

inspectorerrorp=rbeta(m,ap,bp)

inspectorerrorn=rbeta(m,an,bn)

    #

    # Note for below: if the part is good then the
```

```
    # worry is the false negative rate while if
    # the part is bad the concern is the false positive
    # rate. Hence for good parts the _n's
    # for false negatives are used and vice versa for bad parts.
    # Btw obs is the number
    # of observed passes.
    #
sample=gensamplepop(N,tau,an,bn,ap,bp,m,l,inspectorerrorn,
                    inspectorerrorp)
obs=sample$obs
Gstdpart=sample$parts[1:Ngold]
unkpart=rep("NA",N-Ngold)
numpassed = rbinom(1,Npop,tau*(1-mun)+(1-tau)*mup)
#
### random effects no gold standard population based sample  ###
#
data =       list(N=N,Ngold=0,m=m,l=l,obs=obs,Gstdpart=0)

inits =     list( tau=runif(1,.5,.95),stau=runif(1,.25,.75),
                  thetan=rep(.1,m),unkpart=  rbinom(N,1,.5),
                  thetap=rep(.1,m), mup=.1, gamp=20, mun=.1, gamn=20)
#
#simulation parameters
#
thinning=25
paramofint=c("mup","gamp","mun","gamn","ap","an","bp","bn","tau",
                "thetan[1]","thetan[2]","thetap[1]","thetap[2]")
```

```
adapt.n = 5000
iter.n = 25000
    #
    # using while loop with tryCatch to repeat the
    # simulation attempt up to 25 times
repeats=0
while(repeats<25){
  tryCatch({
    Jag.obj=jags.model(file=modelfile.pop, data=data, inits=inits,
                       n.chains = 1, n.adapt=adapt.n, quiet=FALSE)
    output.coda.reg=coda.samples(Jag.obj, paramofint,
                                 n.iter=iter.n, thin = thinning)
    #if it makes it to here then break out of while loop
    repeats=100
  },
  error=function(e){
    repeats=repeats+1
  })
}
summary(output.coda.reg)
autocorr.plot(output.coda.reg)
plot(output.coda.reg)
```

BIBLIOGRAPHY

Altman, D. G., and Bland, J. M. (1993), "Statistics notes," *Quality in Health Care*, 2, 232-8.

Beavers, D. P., and Stamey, J. D. (2012), "Bayesian sample size determination for binary regression with a misclassified covariate and no gold standard," *Computational Statistics and Data Analysis*, 56(8), 2574-2582.

Beavers, D. P., Stamey, J. D., and Bekele, B. N. (2011), "A Bayesian Model to Assess a Binary Measurement System When No Gold Standard System Is Available," *Journal of Quality Technology*, 43(1), 16-27.

Boyles, R. A. (2001), "Gauge Capability for Pass-Fail Inspection," emphTechnometrics, 43(2), 223-229.

Bratcher, T. L., and Phalla, P. (1974), "On the Properties of an Optimal Selection Procedure," *Communications in Statistics. Simulation and Computation*, 3(2), 191 - 196.

Burdick, R. K., Borror, C. M., and Montgomery, D. C. (2003), "A review of methods for measurement systems capability analysis," *Journal of Quality Technology*, 35(4), 342-354.

Carlin, B. P., and Louis, T. A. (2008), *Bayesian Methods for Data Analysis* (3rd Edition), CRC Press.

Danila, O., Steiner, S. H., and MacKay, R. J. (2008), "Assessing a Binary Measurement System," *Journal of Quality Technology*, 40(3), 310-318.

Danila, O., Steiner, S. H., and Mackay, R. J. (2010), "Assessment of a Binary Measurement System in Current Use," *Journal of Quality Technology*, 42(2), 152-164.

Danila, O., Steiner, S. H., and MacKay, R. J. (2012), "Assessing a Binary Measurement System with Varying Misclassification Rates Using a Latent Class Random Effects Model," *Journal of Quality Technology*, 44(3), 179-191.

de Mast, J., Erdmann, T. P., and van Wieringen, W. N. (2011), "Measurement System Analysis for Binary Inspection: Continuous Versus Dichotomous Measurands," *Journal of Quality Technology*, 43(2), 99-112.

Dendukuri, N., and Joseph, L. (2001), "Bayesian Approaches to Modeling the Conditional Dependence Between Multiple Diagnostic Tests," *Biometrics*, 57(1), 158-167.

Erdmann, T. P., and de Mast, J. (2012), "Assessment of Binary Inspection with a Hybrid Measurand," *Quality and Reliability Engineering International*, 28(1), 47-57.

Farnum, N. R. (1994), *Modern statistical quality control and improvement*, Belmont, Calif: Duxbury Press.

Gelman, A., and Rubin, D. B. (1992), "Inference from Iterative Simulation Using Multiple Sequences," *Statistical Science*, 7(4), 457-472.

Joseph, L., Du Berger, R., and Blisle, P. (1997), "Bayesian and Mixed Bayesian/Likelihood Criteria for Sample Size Determination," *Statistics in Medicine*, 16(7), 769-781.

O'Hagan, A., Stevens, J. W., and Campbell, M. J. (2005), "Assurance in clinical trial design," *Pharmaceutical Statistics*, 4(3), 187-201.

Robert, C. P. (2001), *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation* (2nd Edition), New York: Springer-Verlag.

Stamey, J. D., Bratcher, T. L., and Young, D. M. (2004), "Parameter subset selection and multiple comparisons of Poisson rate parameters with misclassification," *Computational Statistics and Data Analysis*, 45(3), 467-479.

van Wieringen, W. N. (2005), "On identifiability of certain latent class models," *Statistics and Probability Letters*, 75(3), 211-218.

van Wieringen, W. N., and de Mast, J. (2008), "Measurement System Analysis for Binary Data," *Technometrics*, 50(4), 468-478.

Wang, F., and Gelfand, A. E. (2002), "A Simulation-based Approach to Bayesian Sample Size Determination for Performance under a Given Model and for Separating Models," *Statistical Science*, 193-208.