uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Achin Jain

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Master of Computer Science

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Software Defect Content Estimation – A Bayesian Approach

TITRE DE LA THÈSE / TITLE OF THESIS

Amiya Nayak

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Paola Flocchini

Nicola Santoro

Gary W. Slater

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

Software Defect Content Estimation - A Bayesian Approach

Achin Jain

A thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

For MSc degree in Computer Science

School of Information Technology and Engineering

University of Ottawa

Ottawa, Ontario

Your file Votre référence
ISBN: 0-494-11301-4
Our file Notre référence
ISBN: 0-494-11301-4

The undersigned hereby recommend to

The Faculty of Graduate Studies and Research

acceptance of the thesis.

Software Defect Content Estimation - A Bayesian Approach

submitted by

Achin Jain

In partial fulfillment of the requirements

For the degree of Master of Computer Science

Thesis Supervisor

University of Ottawa

August, 2005

Abstract

Software inspection is a method to detect errors in software artefacts early in the development cycle. At the end of the inspection process the inspectors need to make a decision whether the inspected artefact is of sufficient quality or not. Several methods have been proposed to assist in making this decision like capture recapture methods and Bayesian approach. In this study these methods have been analyzed and compared and a new Bayesian approach for software inspection is proposed.

All of the estimation models rely on an underlying assumption that the inspectors are independent. However, this assumption of independence is not necessarily true in practical sense, as most of the inspection teams interact with each other and share their findings. We, therefore, studied a new Bayesian model where the inspectors share their findings, for defect estimate and compared it with the Bayesian model (Gupta et al. 2003), where inspectors examine the artefact independently. The simulations were carried out under realistic software conditions with a small number of difficult defects and a few inspectors. The models were evaluated on the basis of decision accuracy and median relative error and our results suggest that the dependent inspector assumption improves the decision accuracy (DA) over the previous Bayesian model and CR models.

Acknowledgements

I would like to thank **Prof. Amiya Nayak** of School of Information Technology & Engineering, University of Ottawa, and **Dr. Alok Patnaik**, Cistel Technologies, Ottawa, for their guidance and support during my thesis work.

I would like to thank **Dr. Ranjeeta Mallick** for her help with the Bayesian Statistics.

I am sincerely grateful to **Dr. Nishith Goel**, Cistel Technologies for his support and advice.

I would also like to thank **Dr. Sanjib Basu** for his help during my study.

At last I would like to thank my family and friends for giving me moral support during difficult times.

Table of Contents

List of Tables

List of Figures

Abbreviations and Symbols

CR	Capture-Recapture
DA	Decision Accuracy
DCET	Defect Content Estimation Technique
DDE	Defect Detection Effectiveness
DPM	Detection Profile Method
Er	Offset from the actual mean
E (p)	Prior Mean
IQR	Inter Quartile Range
Med (RE)	Median Relative Error
RDA	Relative Decision Accuracy
RE	Relative Error
M0	CR model: Maximum Likelihood Estimator
MtMLE	CR model: Maximum Likelihood Estimator
MtCh	CR model: Chao Estimator
MhJE	CR model: Jackknife Estimator
MhCh	CR model: Chao Estimator

MthCh	CR model: Chao Estimator
α	Prior mean for Dirichlet distribution
κ	Concentration parameter for Dirichlet distribution

Chapter 1

Introduction

1.1 Overview

Software inspection (Ebenau and Strauss 1994; Gilb and Graham 1993) is a method to detect faults in software artefacts early in the development cycle (Petersson *et al.* 2003). It was introduced by Fagan (1976) and after that it has become a vital part of the software development process. During software inspection a software artefact is examined for errors by a team of inspectors and all the errors detected are removed. A typical inspection team consists of 3-4 inspectors with varied abilities. It has been shown that a defect that leaks to the next step in a software development will cost at least 10 times more to detect and correct (O'Neill 1997). Therefore, it is crucial to detect and remove errors as early as possible in the software development life cycle. The usefulness of software inspections is now well established. Once an organization adopts software inspections training, it can expect to detect 50 percent of the defects present (O'Neill 1997). It can take from 12 to 18 months to achieve a detection rate of 60 to 90 percent. After 10 years of use, IBM reported 83 percent and AT&T reported 92 percent for defect detection resulting from software inspections practice (O'Neill

1997). Therefore, it is very important to minimize the number of escapes from design and code inspections to improve software quality. Escapes are defects found after the inspection that should have been found and removed during the inspection (Barnard *et al.* 2003). One way can be to improve the software inspection process itself, for example, by using a team of highly experienced inspectors or using a large team size. Another option can be to reinspect the artefact, depending upon the level of satisfaction. A study revealed that only 52 percent of the reinspection decisions were correct (Barnard *et al.* 2003). Therefore, a correct reinspection decision is of utmost importance. Reinspections can be considered a part of the general problem of when to stop inspections. The decision of whether to reinspect or not depends on the estimation of the total number of defects in the artefact. Several methods have been proposed to assist in making a correct reinspection decision such as capture recapture (CR) models. All of these methods fall into the category of Defect Content Estimation Techniques (DCETs) that try to estimate the total number of defects in a document based on the number of defects detected during inspections. Once total number of defects are estimated then the remaining defects in the documents can be calculated, and then the reinspection decision can be made.

1.2 Motivation

There are different kinds of defects present in a piece of software of varying degree of difficulty. The defects can be broadly classified into two categories, minor, defects that are easy to find and major, defects that are hard to find. For example, a defect that would cause the system to fail to satisfy a requirement would be classified as major; all others would be classified as minor (e.g., typographical errors, minor standards violation) (NASA, 1993). Though the number of major defects is less, they are the ones that cause the most of the damage. Therefore, we are primarily interested in estimating the number of major defects, which have a strong impact on product quality and functioning. Hence, the accuracy of the defect content estimation techniques (DCET) is an important issue for their practical application in the industry.

The defect estimation problem is similar to the problem of estimating animal abundance in biology and wildlife research. For example, knowing the population size of a particular kind of fish is one such problem encountered by the fisheries department. Though, the exact number is impossible to determine, a reliable estimation indicating the order of magnitude is often adequate. The solution to this problem in wildlife research is to estimate the population size by means of capture-recapture (CR) models (see Chapter 3).

The concept of capture-recapture was first applied to software inspections by Eick *et al.* (1992). They were the first to adopt CR models for the inspection process at AT&T. However, in these studies the true number of defects was unknown and therefore an evaluation of their true efficacy was not possible. Later work consisted of a Monte Carlo simulation to evaluate the robustness of different CR models (Weil and Votta 1993).

Objective empirical evaluation of CR models started with the study of Wohlin *et al.* (1995). However, this study was conducted with non-software engineering documents. Subsequent work used software engineering artefacts (Miller 1998). All the above work utilized models that were originally developed in wildlife research. Other researchers considered the incorporation of Bayesian methods to estimate defect content (Basu and Ebrahimi 1998 & Gupta, 2003), performed further evaluations of CR models (Thelin and Runeson 1999a) and evaluated their applicability to perspective-based reading (Thelin and Runeson 1999b).

It has already been proved that the CR models fail under realistic software conditions (El Emam *et al.* 1997, El Emam *et al.* 2000, El Emam *et al.* 2001, Gupta 2003) and several modifications have been suggested to improve the results. One such approach

is proposed by Gupta (2003) who uses subjective estimates to estimate the population size. Subjective estimates depend on the knowledge and capability of the individual inspector, who inspects the object carefully and reports the defects. The basic concept behind the subjective estimates is to ask inspectors after an inspection to estimate the percentage of defects in a document they believe they have actually found (El Emam *et al.* 1999). Combining this information with a Bayesian DCET, one can estimate the total number of defects in a document. In many cases of population studies, prior information is available about the population size. Gupta (2003) has incorporated the prior information using a Beta family and derived Bayesian estimators for the population size.

Although several models have been proposed to assist in making a correct reinspection decision, these models overlook the basic fact of dependence among inspectors. The models studied so far are based on the assumption of independence among inspectors (discussed in Chapter 3). As discussed later, this assumption is not necessarily true in practice where inspectors may exchange notes. Therefore, in our research we try to study the effect of dependence among inspectors on the defect estimate.

1.3 Main Contribution

The goal of our research was to study the effect of introducing dependence among inspectors on the accuracy of the estimates and compare it with the independent inspector Bayesian model described by Gupta (2003). One of the major tasks involved in introducing dependence was how to incorporate it statistically in a mathematical framework. We included a correlation factor for depicting dependence and the values used were 0.2 (weak dependence) and 0.4 (moderate dependence). Since we were interested in major defects, the degree of difficulty for the defects was chosen as 0.1 (very difficult) and 0.4 (moderately difficult).

1.4 Theses Outline

In Chapter 2 we explain the software inspection process in detail and highlight the problems associated with it. Discussion of CR models and the different estimators is presented in Chapter 3. In Chapter 4 we look into the Bayesian approach for software defect content estimation. In Chapter 5, the overview of the research method and the evaluation criteria of different estimation techniques is given. The simulations results with appropriate tables and graphs are discussed in Chapter 6. We compare the results of the two types of Bayesian models (dependent and independent) in Chapter 7. The theses ends with Chapter 8 where results are discussed and future prospects are given.

Chapter 2

Software Inspections

Inspections are a formal, efficient, and economical method of finding errors in design and code (Fagan 1976) and were introduced in the year 1976. Software inspection can improve the quality of the software by detecting and removing errors before it is released (O'Neill 1997, NASA 1993). In this chapter we explain what software inspections are, how can it be used to improve the software quality and their effectiveness.

2.1 Software Inspections – An Overview

The software inspection process as described by Fagan (1976), involves the following key elements:

- Software Inspection Team

  A software inspection team is a group of people working together to accomplish the task of inspection. The team is assigned a set of procedural roles as following (Fagan 1976):

Moderator – The moderator is the key person in the inspection process. He/She is responsible for ensuring that inspection procedures are correctly followed during the inspection process. He/She must be a skilled programmer but need not be a technical expert. Some of the responsibilities of the moderator are:

- Managing the inspection team
- Scheduling suitable meeting places
- Reporting inspection results
- Follow-up on rework

Designer – The designer is a programmer responsible for producing the program design.

Coder/Implementer – The programmer is responsible for translating the design into code.

Tester – The tester is responsible for writing and/or executing test cases or otherwise testing the product of designer and coder.

- Software Inspection Process Stages

  A software inspection process follows a set of pre-defined stages which are shown in the image below.

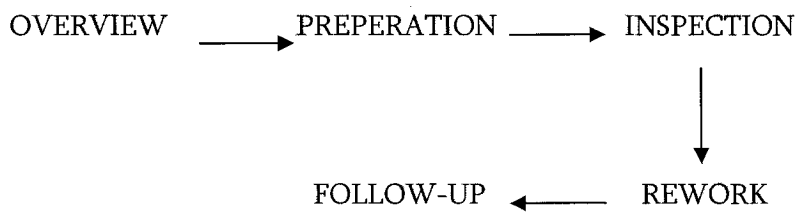

Fig. 2.1 Inspection Stages

Overview – This is done by the whole team. The designer explains the overall area being addressed and then the specific area he/she has designed in detail – logic, data paths, dependencies etc. Documentation of the design is distributed to all the members of the inspection team on conclusion of the overview.

Preparation – This is performed by each individual team member. Each individual studies the design document to understand the design, its intent and logic. This stage is like homework for the participants.

Inspection – After the preparation is done, the next step is inspection. It is performed by the whole team. A 'reader' is chosen by the moderator who describes the implementation of the design (as done by the designer). Every minute detail of the design like logic, branch etc is discussed. All higher level documentation, high level design specifications, logic specifications, etc. and macro and control block listings must be available and presented during

inspection. Once the design is understood, the objective is to detect errors. Errors are found during the implementer/coder's explanation of the code and noted down by the moderator along with its severity (major or minor) (see Section 1.2). Within one day of conclusion of the inspection, the moderator is expected to write a report describing the findings of the inspection process. This report will be used during rework and follow-up operations.

Rework - During rework all the errors encountered during inspection (noted in the report) are resolved by the designer or coder/implementer.

Follow-Up – It is required that all the errors, issues and concern be entirely resolved at this level. It is the duty of the moderator to ensure that all the errors and issues have been resolved.

2.2 When to Stop Inspections

At the end of the inspection process a decision has to be made whether the inspected artefact is of sufficient quality and if reinspection is required. This is vital as a defect that leaks into next phase are 10 to 100 times more expensive to fix at later stages (Fagan 1976). Several approaches have been suggested to make the decision of reinspection (Briand *et al.* 1997).

- The first approach is to let the inspection team make the reinspection decision at the end of the inspection process. Studies suggest that these kind of

subjective estimates have fairly good accuracy and can be applied to make the reinspection decision. However, these estimates can be politically motivated and can dilute the decision process and hence objective reinspection criterion is necessary (Briand *et al.* 1997).

- The second approach is to compare the inspection results with certain benchmarks. For example, a document is reinspected for the second time if the number of defects is significantly different from the historical average (Eick *et al.* 1992). Too many defects would indicate a poor document, and too few defects a poor inspection. There is a limitation with such an approach as a high-quality document may be re-inspected (due to low number of defects), and a poor-quality document may not be re-inspected if the inspection is performed poorly. Therefore, such an approach is reliable only when defect densities are roughly constant across inspected artefacts.
- Another approach is to use upper and lower thresholds on the number of defects found (Vander Wiel and Votta 1993). The lower limit is set to detect poor quality inspections and the upper limit for detecting low-quality documents. This, however, raises the risk that inspectors are tempted to find only a passing number of defects regardless of the document's quality.
- The last approach is to use the number of defects found to estimate the total number of defects remaining in the artefact. This estimate can then be used to make the reinspection decision. Since it is impossible to estimate the total

number of defects in a software before it is put in use, it is necessary to build estimation models of the number of defects in a software artefact. This estimation problem is similar to the problem of estimating animal abundance in biology and wildlife research. For example, knowing the population size of a certain type of animal is essential for deciding on the number to be released for shooting.

2.3 A Probable Solution

Now we know that the defect estimation problem is similar to animal abundance estimation problem in wildlife and biology. So how do wildlife researchers deal with animal abundance problem? The solution is the use of capture recapture (CR) estimation models (Otis *et al.* 1978, White *et al.* 1982). During CR studies animals are captured, marked, and released on several trapping occasions. If an animal bearing a mark is captured on a later trapping occasion, it is said to be recaptured. Based on the number of marked animals that are recaptured one can estimate the total population size using statistical models and their estimators. The capture-recapture principle in biology can easily be transferred to software inspections. The defects can be considered as equivalent to the animals. An inspector detects some number of defects from the population of defects in the software artefact. This is comparable to the purpose of a particular trapping occasion in biology. A defect found by one

inspector and later found by another is said to be recaptured (just as a marked animal is said to be recaptured if it is captured on a subsequent trapping occasion). Using the estimators similar to the ones used in biology, the total number of defects in the software artefact can be estimated statistically. Several estimators have been proposed based on the capture recapture principle (Darroch 1958, Burnham and Overton 1978, Chao 1987, Chao 1988, Chao 1992). During the past few years several Bayesian models have also been suggested to enhance the results of these CR models (Basu and Ebrahimi 1998, Basu and Ebrahimi 2001, Basu 2003, Gupta *et al.* 2003). Most of the Bayesian and capture recapture models work on a basic assumption of independent inspectors. This might not be a valid assumption in real life as the inspectors can share their findings. Therefore, in this work we studied a new dependent inspector Bayesian model for software defect content estimation and conducted simulations to evaluate different models to compare their performance. The capture recapture and Bayesian models are discussed in detail in Chapters 3 and 4, respectively.

Chapter 3

Capture Recapture Models

In biology and wildlife, CR method is used to estimate the size of an animal population like deer, fish etc. In this chapter the capture recapture models, their assumptions and methodology are discussed.

3.1 Basic Concepts

In capture recapture method to estimate animal abundance, animals are captured, marked, and then released on a number of trapping occasions. A marked animal that is caught at a subsequent trapping occasion is said to be recaptured. The number of marked animals that are recaptured allows one to estimate the total population size based on the overlap. As an example, suppose one wants to estimate the size N of a population. Let n_1 animals are captured on first day. These animals are marked and released into the population. After allowing some time for the marked and unmarked animals to mix, a second trapping occasion is performed on a second day. On this day, suppose n_2 animals are captured. Let this sample of n_2 animals consists of m_2 animals bearing a mark (animals captured on both days) and $n_2 - m_2$ animals without a mark

(newly captured animals). Assuming that the ratio of marked to total animals in the second sample is equal to the ratio of marked to total animals in the entire population, the so-called Lincoln-Peterson Estimator for the number of animals in the population can be derived as (Seber 1982 and White *et al.* 1982),

$$\widehat{N} = \frac{n_1 n_2}{m_2} \tag{3.1}$$

Applying the same principle to software inspections, each inspector is considered as a trapping occasion and the defects as the animals. Each inspector reads the software artefact and tries to find the defects, i.e. draw independent samples from the population of defects. Based on the overlap of defects amongst inspectors, one can estimate the total number of defects in a software artefact and hence the remaining defects can be calculated. Taking into account this number of remaining defects, one can decide on a more objective basis whether the software artefact has to be reinspected.

There are a number of estimation models available based on various assumptions made about the CR. These models and their assumptions are discussed in Section 3.7.

3.2 Closed and Open Population Models

All of the estimation models can be classified into two categories namely, open and closed population models. A closed population model is one in which the population remains unchanged during the capture recapture study period. In a closed population there is no birth, death, immigration or emigration. In contrast, an open population can have population change via birth, death, immigration or emigration.

3.2.1 Closure – An Important Assumption

The closure of population is an important assumption. It says is that there is no birth, death, immigration or emigration. This assumption is never completely true in biological systems but if the study is done carefully, it can be approximately true.

The closure property can be further subdivided into two components:

Geographical Closure - Closure by boundary.

Demographical Closure - Closure to birth, death, immigration and emigration.

The distinction between demographic closure and geographical closure is important because the open population models (as described before) are open to demographical closure only and not to geographical closure.

3.3 Assumptions

The closed population estimation models have certain assumptions that affect their estimation method used by those models. These assumptions are (White *et al.* 1982):

- The population is closed.
- Animals do not lose their marks during the experiment.
- All marks are correctly noted and recorded at each tapping occasion.

A fundamental assumption that affects all the models is the capture probabilities of various population members. The modelling of the capture probability is very basic problem faced during CR studies. A decade ago, it was assumed that all the animals have equal capture probabilities and that capturing and marking do not affect their subsequent catchability of the animal. This assumption is unrealistic in capture studies of animals and is generally not met (Young *et al.* 1952, Huber 1962, Swinebroad 1964). Therefore, the models must take these probabilities into account. Several models have been proposed based on the various capture probabilities that they take care of. These models are discussed later in the chapter in Section 3.7.

3.4 Applying CR Models for Software Inspections

Capture-recapture models make certain assumptions that may differ between biology and software inspections. Thus, before using the models it is necessary to check the

feasibility of using those assumptions in software inspections. These assumptions can easily be translated into software inspection perspective (Miller 1998):

- Once the document is issued for inspection, it must not be changed.
- Inspectors must not reveal their proposed defects to other inspectors, i.e. they work independently.
- Inspectors must ensure that they accurately record and document every defect they find.
- All inspectors must be provided with identical information, in terms of source materials, standards, inspection aids etc; and this material must be available to them at all times.

3.5 Sources of Variation

Various models and corresponding estimators have been developed and proposed in biology to alleviate the effects of these assumptions (Pollock 1991). The most important models that can be considered for inspections have been proposed by Otis *et al.* (1978) and White *et al.* (1982). They present a set of closed models that allow for a varying capture probability. Based on the source of variations different models can be classified into three broad categories:

- Behaviour
- Heterogeneity
- Time

3.5.1 Behaviour

These estimators were suggested with the idea that the probabilities of capture among animals differ with the behavioural response of these animals (Otis *et al.* 1978), i.e. animal behaviour is altered after capture, as the animals may get fascinated by traps and so these animals are more likely to be captured while others could be scared of capture. The capture probabilities of the animals change during the capture period. Initially, all animals have the same probability of capture. However, when an animal is caught its capture probability is changed. In software inspections, this variation may be used to model the fact that defects captured by more than one inspector have a higher probability of being detected. However, the estimators for this model depend on the order of inspectors and since there is as such no specified ordering of inspectors, these estimators are not considered adequate (Briand *et al.* 1997).

3.5.2 Heterogeneity

Variation by heterogeneity in biology models the fact that the animals vary in their capture probability, i.e. certain animals are more likely to be captured then the others. For example, older animals do not move around a lot and are less likely to be captured then young animals. This fact can be used to model

different detection probabilities of the defects in software inspections. Hard defects have low capture probability than the easy defects.

3.5.3 Time

In this family of estimators, the capture probability varies with the trapping occasion, i.e. at different time periods the animals may be more open to traps. For example, the weather on one day may be too cold or hot and so some animals may not move around a lot thus being less receptive to the traps. While on another day the weather could be ideal for the animals to move around thus being more likely to be captured by traps. The equivalent of time variation in software inspections can be the different detection abilities of the inspectors. An experienced inspector is more likely to find errors then an inexperienced inspector.

3.6 Statistics and Notation Used

The data obtained after capture recapture experiments is expressed in form of a matrix called X matrix. The rows denote the animals (defects) and the columns denote the trapping occasions (inspectors).

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{Nk} \end{pmatrix}$$

where,

$$x_{ij} = \begin{cases} 1 \text{ if inspector j detected defect i} \\ 0 \text{ otherwise} \end{cases}$$

It is to be kept in mind that the matrix may not be observable in its entirety as the total number of defects is not known. We observe only those rows that have at least one non zero entry. Hence, the matrix contains D rows, where D denotes the number of unique defects found during inspection.

3.6.1 Some Notations Used

$\widehat{N}$ - The estimate of the number of defects in the document.
N - The actual number of defects in the document.
Z_j - The number of defects found only by inspector j.
D - The number of unique defects found by the inspection team.
f_i - The number of defects found i times.
K - Number of inspectors.

n_j - The number of defects found by the j_{th} inspector.
n - Sum of the $n_j : n = \sum_{j=1}^{k} n_j$.
W - Subset of inspectors (1...k).
n_w - Number of defects that were found by the inspectors in W.

3.7 Capture Recapture Estimators

Based on the above categories different estimators have been proposed. Although the models have three broad categories, the estimators can overlap different categories. Various models that have been developed are:

- M_0 - no variation.
- M_t - variation by time.
- M_h - variation by heterogeneity.
- M_b - variation by behaviour.
- M_{th} - variation by time and heterogeneity.
- M_{tb} - variation by time and behaviour.
- M_{bh} - variation by behaviour and heterogeneity.
- M_{tbh} - all the variations.

Out of these, estimators with behavior variation are not studied as they are very difficult to implement. So the estimators M_0, M_t, M_h, M_{th} are studied.

Let, p_{ij} = probability that defect i is detected by inspector j (3.2)

We now define different models in terms of p_{ij}

- Model M_0 - No variation. $p_{ij} = p$. The probability of the defect being captured by an inspector is constant.

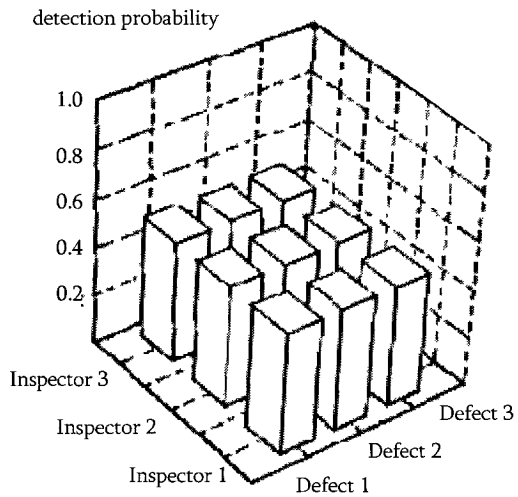

Fig. 3.1 Assumptions of model M0 (Briand *et al.* 1997)

Model M_t - variation with inspector ability $p_{ij} = p_j$.

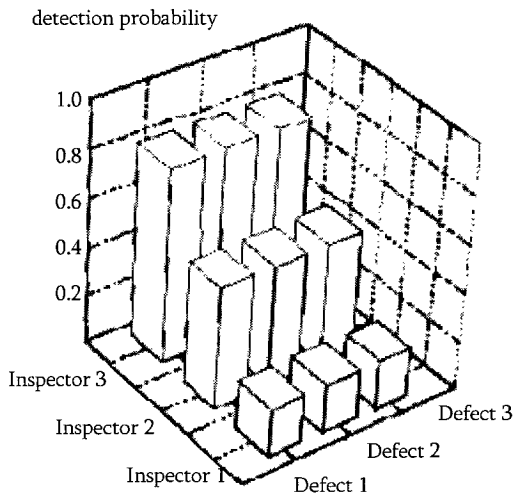

Fig. 3.2 Assumptions of model Mt (Briand *et al.* 1997)

- Model M_h - variation with defect detection probability $p_{ij} = p_i$. The probability depends on the difficulty of defect, all inspectors have same ability.

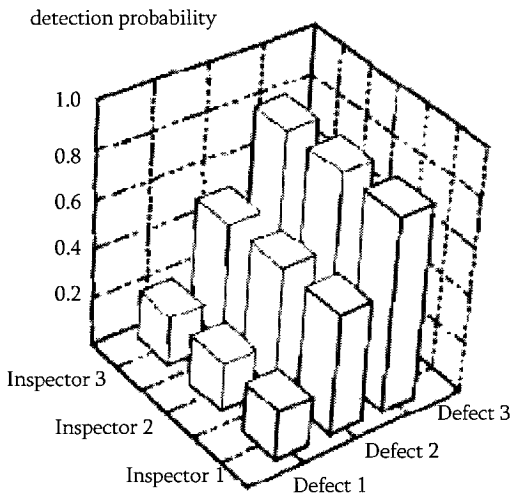

Fig. 3.3 Assumptions of model Mh (Briand *et al.* 1997)

- Model M_{th} - variation by defect difficulty and inspector ability $p_{ij} = p_i p_j$.

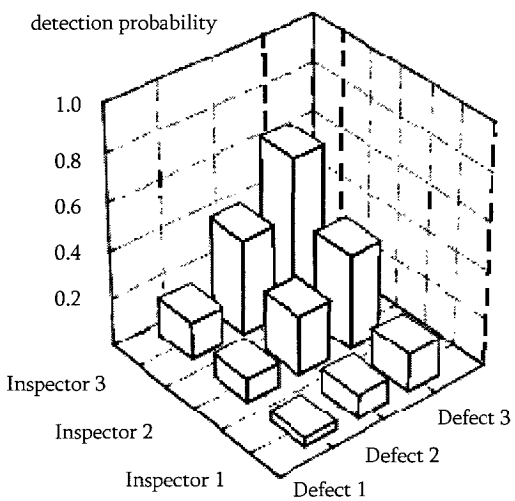

Fig. 3.4 Assumptions of model Mth (Briand *et al.* 1997)

3.7.1 Null Estimator - M_0 (Darroch 1958)

The derivation for this model involves the hypergeometric functions. The hypergeometric probability function is used to derive the generalized hypergeometric density, which itself is used to derive the following estimator:

$$\widehat{N} = \max_{N \in N} \left[\ln\left(\frac{N!}{(N-D)!} \right) + n\ln(n) + (tN-n)\ln(N-n) - tN\ln(tN) \right], \qquad (3.3)$$

where N is a member of I, the set of integers starting at D, i.e.{D, D+1, D+2,...}, D is the number of distinct animals found, t is the number of occasions, and n is the total number of animals caught during the experiment. The NULL estimator is described because it ignores the effect of time and heterogeneity. It would be worthwhile to see if these factors show great changes in the estimate by comparison with the NULL estimator.

3.7.2 Jackknife Estimator - $M_h(JE)$ (Burnham and Overton 1978)

For this estimator the theoretical assumptions on capture probability are:

$$p_{ij} = p_j$$

i.e.,

$$p_{ij} = p_1, \ldots\ldots\ldots, p_N$$

where $i = 1,..,t$ and $j = 1,\ldots,N$. (t is the number of days of the experiment and N is the number of animals in the wild)

The Jackknife requires the capture frequencies of the experiment. The capture frequencies are the frequencies associated with the number of times an animal was caught over the period of the experiment. Hence, the sum of the capture frequencies should equal the number of distinct animals caught, i.e.

$$D = \sum_{i=1}^{t} f_i \tag{3.4}$$

3.7.3 Chao Heterogeneity Estimator – $M_h(Ch)$

This is the first of three estimators that Chao developed. In her article (Chao 1987), Chao indicates that the Jackknife 'does not work well' when the captured animals were mostly caught once or twice., i.e. samples skewed right. Her estimator addresses this apparent shortcoming.

$$\widehat{N} = D + \frac{f_1^2}{2f_2} \tag{3.5}$$

3.7.4 Maximum Likelihood Estimator (time) - $M_t(MLE)$

This estimator is a derivative of the NULL estimator mentioned earlier. For this estimator the theoretical assumption on capture probability is

$$p_{ij} = p_i$$

i.e.,

$$p_{ij} = p_1, \ldots\ldots\ldots, p_t$$

where $i = 1,..,t$ and $j = 1,\ldots,N$ (Again, t is the number of occasions and N is the number of animals in the wild). Slightly more detailed capture data is

required. The algorithm is as follows:

$$\widehat{N} = \max_{N \in N} \left[\ln\left(\frac{N!}{(N-D)!}\right) + \sum_{i=1}^{t} n_i \ln(n_i) + \sum_{i=1}^{t} (N-n_i)\ln(N-n_i) - tN\ln(N) \right] \qquad (3.6)$$

The parameter n_i is the number of animals caught on occasion *i*. Other parameters are as in the NULL estimator.

3.7.5 Chao Time Estimator - $M_t(Ch)$

Chao (1988) discussed that the Maximum Likelihood Estimator overestimates very sparse data. To account for sparse data the model incorporates details regarding the frequency of animals caught once only.

$$N = D + \frac{f_1^2 - \sum_{i=1}^{t} Z_i^2}{2(f_2 + 1)}$$

$$= D + \frac{\sum_{i=1}^{t} \sum_{j=i+1}^{t} Z_i Z_j}{f_2 + 1} \qquad (3.7)$$

where Z_i are the components of f_i, i.e. Z_i is the number of animals caught on occasion *i* and no other.

3.7.6 Chao Heterogeneity – Time Estimator - $M_{th}(Ch)$

Chao (1992) formulated an estimator that would allow the probability to vary with the time and heterogeneity variables. The probability of capture will depend on both the animal and the occasion of the experiment, i.e.

$$p_{ij} = p_i p_j$$

The extra factor involved increases the details required. The formulation is reflective of this fact. There are three possible estimators in this model; the latter two versions are bias corrected. The inclusion of higher order frequency terms should reduce the dependence on f_1 and thus their bias. This C value is termed as the sample coverage and is defined as:

$$\widehat{C_1} = 1 - \frac{f_1}{\sum_{k=1}^{t} k f_k},\ \widehat{C_2} = 1 - \frac{f_1 - 2\frac{f_2}{t-1}}{\sum_{k=1}^{t} k f_k},\ \widehat{C_3} = 1 - \frac{f_1 - 2\frac{f_2}{t-1} + 6\frac{f_3}{(t-1)(t-2)}}{\sum_{k=1}^{t} k f_k} \quad (3.8)$$

These values are used in conjunction with:

$$\widehat{N}_{0,i} = \frac{D}{\widehat{C_i}}$$

$$\hat{\gamma}_i^2 = \max\left\{ \frac{\widehat{N}_{0,i} \sum_{k=1}^{t} k(k-1) f_k}{2 \sum_{j=1}^{t-1} \sum_{k=j-1}^{t} n_j n_k} - 1, 0 \right\}, i = 1,2,3 \quad (3.9)$$

Together, the estimates can be calculated as:

$$\widehat{N}_i = \frac{D}{\widehat{C}_i} + \frac{f_1}{\widehat{C}_i}\hat{\gamma}_t^2, i = 1,2,3 \tag{3.10}$$

3.8 Applying the Estimators To Software Inspection

All the models and their estimators described above can be applied to software inspections. The number of trapping occasion in biology is equivalent to the number of inspectors in an inspection team, i.e. *t*. The number of animals to be estimated is equivalent to the total number of defects in a software artefact, i.e. *N*. Therefore, the task of estimating total number of animals is equivalent to estimating total number of defects in a software document. Different assumptions associated with estimators have already been transformed in previous sections.

3.9 Performance of the CR Models

Previous simulation studies (El Emam *et al.* 1997, El Emam *et al.* 2000, El Emam *et al.* 2001, Gupta 2003) showed that CR models perform very poorly when the number of defects is small and they are difficult to find. Most of the models fail to estimate at all and if they estimate then the result is not satisfactory. The reason for the failures is the overlap parameter (Gupta, 2003). Overlap is the indication of identical defects found by different inspectors. The CR model estimates rely on the number of unique

defects found by the inspection team (given by D, see Section 3.6.1) and total number of defects found (given by n). In realistic scenario, these parameters tend to zero as the data matrix becomes sparse. As a result the models fail to estimate.

As the CR models fail to estimate under realistic software conditions we need to use some other model or combine the estimates with other variables to improve the estimates (Bernard *et al.* 2003). This led to the use of Bayesian technique for the CR models (Basu and Ebrahimi 1998, Gupta 2003, Gupta *et al.* 2003). The idea was to combine the prior knowledge about the population estimate with the calculated value to improve the result. In the next chapter this approach is discussed in detail.

Chapter 4

Bayesian Approach

The Bayesian approach to the defect population estimation in a software artefact is described in this chapter. The basic idea in using Bayesian method is to combine the prior knowledge about the population size with the number of defects found during inspection to get a better posterior estimate of the population. A general problem with most of the objective population estimators is that they fail to estimate when the X matrix (see Section 3.6) is sparse. Therefore, we need to use subjective estimates to get an improved population estimate.

4.1 Bayes' Theorem

Bayes' theorem is a result in probability theory, which gives the conditional probability distribution of a random variable A given B, in terms of the conditional probability distribution of variable B given A and the marginal probability distribution of A alone.

In the perspective of Bayesian probability theory and statistical inference, the marginal probability distribution of A alone is usually called the prior probability

distribution or simply the prior. The conditional distribution of A given the "data" B is called the posterior probability distribution or just the posterior.

4.2 Statement of Bayes' Theorem

Bayes' theorem is a relation among conditional and marginal probabilities. It can be viewed as a means of incorporating information, from an observation, for example, to produce a modified or updated probability distribution. To derive Bayes' theorem, note first that from the definition of conditional probability

$$P(A/B)P(B) = P(A,B) = P(B/A)P(A) \tag{4.1}$$

where $P(A, B)$ is the joint probability A and B.

It says that, the probability of A given B times the probability of B is equal to the probability of both event A and B occurring together and is also equal to the probability of B given A times the probability of A.

Rearranging Eqn. 4.1, we obtain

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)} \tag{4.2}$$

which is known as Bayes' theorem.

Each term in Bayes' theorem has a conventional name. The term *P(A)* is called the prior probability of *A*. It is "prior" in the sense that it precedes any information about *B*. *P(A)* is also the marginal probability of *A*. The term *P(A/B)* is called the posterior probability of *A*, given *B*. It is "posterior" in the sense that it is derived from the specified value of *B*. The term *P(B/A)*, for a specific value of *B*, is called the likelihood function for *A* given *B* and can also be written as *L(A/B)*. The term *P(B)* is the prior or marginal probability of *B*, and acts as the normalizing constant. With this terminology, the theorem may be paraphrased as

$$posterior = \frac{likelihood\ x\ prior}{normalizing\ constant} \tag{4.3}$$

4.3 Independent Inspector Bayesian Model

Ananda (1997) described a Bayesian model to estimate the population of mountain sheep and this model was used by Gupta (2003) for software inspections. As explained earlier, the defects in a software document are considered as animals and the inspector are equivalent to the trapping occasions in biological studies. So the task of estimating the total number of defects in the software artefact is similar to animal population estimation. The difference between a normal capture recapture and a Bayesian model is that the latter allows incorporating the prior knowledge directly

into statistical analysis. As given by Eqn. 4.3 the likelihood function is modified to give a posterior distribution as following:

$$posterior\ distribution = likelihood\ function\ x\ prior\ distribution\ , \qquad (4.4)$$

where *normalizing constant* is taken as unity.

4.3.1 Notations and Statistics Used

N - total number of defects in the software
n_0 - number of defects found by the most experienced inspector
s - number of inspectors
n_i - number of defects found by each inspector ($i = 1.....s$)
m_i - number of defects found by each inspector that overlapped with the most experienced inspector ($i = 1.....s$)
$\phi_\lambda(n)$ – density supplying n_i, s

4.3.2 Model Description

It is assumed that the distribution of m_i and n_i follows the hypergeometric distribution.

$$f(m_i / n_i) = \frac{\binom{n_0}{m_i}\binom{N-n_0}{n_i-m_i}}{\binom{N}{n_i}}, m_i = 0,1,.....n_i \qquad (4.5)$$

When n_i values are small and n_0 is large, this distribution can be approximated by the binomial distribution with parameters n_0 and $p = \frac{n_0}{N}$,

$$f(m_i / n_i) = \binom{n_i}{m_i} p_i^{m_i} (1-p)^{n_i - m_i}, m_i = 0,1,\ldots\ldots,n_i \tag{4.6}$$

Since the second stage samples are independent the likelihood function $L(p,\lambda)$ can be written as:

$$\begin{aligned} L(p,\lambda) &= \prod_{i=1}^{s} \binom{n_i}{m_i} p^{m_i} (1-p)^{n_i - m_i} \phi_\lambda(n_i) \\ &= \left[\prod_{i=1}^{s} \binom{n_i}{m_i} \phi_\lambda(n_i) \right] p^{\sum_{i=1}^{s} m_i} (1-p)^{\sum_{i=1}^{s} n_i - \sum_{i=1}^{s} m_i} \end{aligned} \tag{4.7}$$

To apply Bayesian methods, suppose that the prior density of p is a beta density function with parameters a and b,

$$f(p) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} p^{a-1}(1-p)^{b-1}, 0 < p < 1 \tag{4.8}$$

and the prior density of λ is $\psi(\lambda)$. Here all the prior information regarding the population size is incorporated into the inference process using the beta density and these parameters must be evaluated using the prior information. Then the posterior density p and λ is proportional to,

$$\begin{aligned} \pi(p,\lambda / data) &\propto p^{a-1}(1-p)^{b-1} \psi(\lambda) \left(\prod_{i=1}^{s} \binom{n_i}{m_i} p^{m_i} \phi_\lambda(n_i) \right) \\ &\propto \left\{ \psi(\lambda) \prod_{i=1}^{s} [\phi_\lambda(n_i)] \right\} p^{\sum_{i=1}^{s} m_i + a - 1} (1-p)^{\sum_{i=1}^{s} n_i - \sum_{i=1}^{s} m_i + b - 1} \end{aligned} \tag{4.9}$$

Therefore, the posterior density of p given data is,

$$\pi(p/data) = \Gamma\left(\sum_{i=1}^{s} n_i + a + b\right)\left[\Gamma\left(\sum_{i=1}^{s} m_i + a\right)\Gamma\left(\sum_{i=1}^{s} n_i - \sum_{i=1}^{s} m_i + b\right)\right]^{-1} \quad (4.10)$$
$$x\, p^{\sum_{i=1}^{s} m_i + a - 1} (1-p)^{\sum_{i=1}^{s} n_i - \sum_{i=1}^{s} m_i + b - 1}$$

which is a beta density with parameters $\sum_{i=1}^{s} m_i + a$ and $\sum_{i=1}^{s} n_i - \sum_{i=1}^{s} m_i + b$.

Assuming the quadratic loss, the Bayes estimate of the population size is expected value of n_0 / p with respect to the posterior distribution. Therefore, the Bayes estimate of the population size $N$is given by,

$$\widehat{N}_{ab} = \frac{n_0\left(a + b + \sum_{i=1}^{s} n_i - 1\right)}{\left(a + \sum_{i=1}^{s} m_i - 1\right)} \quad (4.11)$$

If the estimation problem has no prior information, then one can use a uniform prior on p corresponding the choice (*a=1, b=1*) or a generalized prior density on p, a choice such as (*a=1, b=0*). Notice that when (*a=1, b=0*), the generalized Bayes estimate coincides with the maximum likelihood estimate.

One approach to calculate a and b is to use the prior data. With the given prior data, estimates for mean $E(p)$ and standard deviation σ_p can be calculated and then the parameters a and b can be computed using the following formula,

$$E(p) = \frac{a}{a+b}, \text{ and } \sigma_p^2 = ab(a+b+1)^{-1}(a+b)^{-2} \quad (4.12)$$

4.4 Dependent Inspector Bayesian Model

The assumption of independent inspector is not necessarily true all the time and considering the dependence among inspectors might improve the defect estimate. We follow the model described by Basu (2003), Basu and Ebrahimi (2001) and Basu and Ebrahimi (1998), which incorporates the dependence among inspectors. This model also falls into the broad category of capture recapture methods where defects are considered as animals and the inspectors are considered as trapping occasions and the task of estimating the number of defects is similar to estimating the animal population. Just like the Bayesian model described in Section 4.3 this model too allows to combine the prior knowledge directly into statistical inference.

4.4.1 Notations Used

N - unknown defect population size

k - number of inspectors

X - the data matrix with entries x_{ij}

x_{ij} - $\begin{cases} 1 \text{ if } i^{th} \text{ member of population is included in the } j^{th} \text{ list} \\ 0 \text{ otherwise} \end{cases}$

i - 1.....N

j - 1.....k

n - number of distinct defects found

p_{ij} - $P(X_{ij}=1)$, it is the probability of an error being detected

n_j - number of members in the j^{th} list (the number of error detected by j^{th} inspector) $n_j = \sum_{i=1}^{N} x_{ij}, j = 1....k$

y_i - $\sum_{j=1}^{k} x_{ij}, i = 1.....N$. It is the i^{th} row sum of the X matrix.

f_i - $\#\{y_l : y_l = j\}, j = 0.....k$

f_0 - $N - n$

4.4.2 Model Description

For this dependence model, the probability of inclusion of a defect in a list depends on the other inspectors. For this, one needs to consider the joint probabilities, $p(x_1, x_2, ...x_k) = P(X_{i1} = x_1, ...X_{ik} = x_k),\ i = 1, ...N$ and the joint counts $n(x_1, x_2, ...x_k) = \#\{X_{i1} = x_1, ...X_{ik} = x_k,\ i = 1, ...N\}$,

where, $x_j = 0$ or 1, $j = 1, ...k$. As an example $p(0, ..., 0)$ is the probability that none of the inspectors detects the error, $p(1, ..., 0)$ gives the probability that the error is detected by only the first inspector only and so on. Since there are k inspectors, there will be 2^k $p(x_1, ..., x_k)$'s summing to 1. Similarly, there will be 2^k $n(x_1, ..., x_k)$'s, where $n(0, ..., 0) = N - \sum\{n(x_1, ...x_k) :$ at least one $x_k \neq 0\} = N - n$ is unknown as N is unknown. Let $s = 2^k$. Under these assumptions the likelihood is given by:

$$L_{DEP}(data \mid N, p(1), ...p(s)) = N! \prod_{j=1}^{s} p(j)^{n(j)} \Big/ \prod_{j=1}^{s} n(j)! \qquad (4.13)$$

In Bayesian method the nuisance parameters $\{p(1), ...p(s)\}$ can be integrated out to obtain the marginal likelihood of N. For further simplicity

let $\gamma(x_1,....x_k) = p(x_1,...x_k)/\{1-p(0,.....,0)\}$, where each x_j is 0 or 1 but at least one $x_j \neq 0$. For example, $\gamma(1,0,...0) = p(1,0,...0)/\{1-p(0,.....,0)\}$ is the conditional probability that the error is detected by only the first inspector and no body else given that it was found by at least one inspector. Using this notation the likelihood in Eqn. 4.13 can be written as,

$$L_{DEP}(data \mid N, p(s), \gamma) = \left[\binom{N}{n}\{1-p(s)\}^n p(s)^{N-n}\right]\left\{\frac{n!}{\prod_{j=1}^{s-1} n(j)!}\prod_{j=1}^{s-1}\gamma(j)^{n(j)}\right\} \quad (4.14)$$

It is clear from the above equation that only the first part of the likelihood depends on N and the second term consists of *s-1* nuisance parameters. It has been shown that only the first part of the likelihood should be used for inference about N (Reid 1995). For Bayesian inference we need to assign joint prior $\pi(N, p(s), \gamma)$ and an independent prior $\pi(N)\pi(p(s),\gamma)$ is commonly used (Basu 2003). If we can further assume that $\pi(p(s),\gamma) = \pi(p(s))\pi(\gamma)$ then it can be easily calculated that the inference of N only depends on the first part of the likelihood (Eqn. 4.14). If $\{p(1),.....p(s)\}$ are assumed to have a Dirichlet distribution $(\kappa\alpha)$ prior with mean $\alpha = \{\alpha(1),....\alpha(s))$ where $\sum\alpha(r) = 1$ and concentration parameter κ then the required condition of a priori independence of $p(s)$ and γ is satisfied. Here α represents the supervisor's belief about the chance of detecting an error (for example chance of detection

may be 35%). The concentration parameter κ represents the strength of the supervisor's belief.

Let us assume a Dirichlet prior on $\{p(1),.....p(s)\}$, then (as discussed before) the inference for N will only depend on first part of likelihood in Eqn. 4.14. Further, $p(s)$ can also be integrated out and the marginal posterior for N is given by,

$$\pi(N \mid data) \propto \pi(N)N!\Gamma(N-n+\kappa\alpha(s))/\{N-n)!\Gamma(N+\kappa)\} \qquad (4.15)$$

The marginal posterior satisfies the following recursion relation (George *et al.*, 1992),

$$\frac{\pi(N+1 \mid data)}{\pi(N \mid data)} = \frac{(N+1)(N-n+\kappa\alpha(s))}{(N+1-n)(N+\kappa)} \frac{\pi(N+1)}{\pi(N)} \quad , \qquad (4.16)$$

which can be used to obtain the complete posterior distribution of N up to a constant of proportionality.

In the above equation the choice of prior $\pi(N)$ is very important. According to Basu (2003) the marginal posterior $\pi(N \mid data)$ is proper if,

$$\sum (N-n+\kappa)^{-\kappa\{1-\alpha(s)\}} \pi(N) < \infty .$$

According to this result,

$$1-\alpha(s) > 0 \text{ and, } \kappa\{1-\alpha(s)\} > 1 \qquad (4.17)$$

So the range of values for κ and α should satisfy the above equations.

To evaluate the model extensive simulations were performed by varying different parameters like κ, α, number of inspectors, number of defects etc. In the next chapter we talk about the research method and evaluation criteria in detail.

Chapter 5

Research Method and Evaluation Criteria

The research method used to simulate the model under different sets of parameters and evaluation criteria used to study the performance of the estimator is explained in this chapter. Also discussed are various factors affecting the performance of the estimator like the number of inspectors, number of defects and the degree of dependence among the inspectors.

5.1 Factors Affecting the Performance of the Estimator

There are different factors that can have a strong impact on the performance of an estimator namely, the number of inspectors, the number of defects and their degree of difficulty and the degree of dependence among the inspectors.

5.1.1 Number of Inspectors and their Abilities

The first important factor to be kept in mind while applying the DCET is the number of inspectors and their abilities. More inspectors mean more error detections and hence a better estimate, but employing a large number

of inspectors is not feasible for practical and economical reasons. Each additional inspector costs a lot both in terms of money and time.

In studies dealing with the biological application of capture-recapture models, use of five trapping occasions (equivalent to five inspectors in software engineering) is suggested, though a number of 7 or 10 was found more appropriate (Otis *et al.* 1978, Briand *et al.* 2000).

In software inspection researches, the reported number of inspectors varies. According to Eick *et al.* (1992) inspections at AT&T can involve a number of inspectors that can range up to twelve. Briand *et al.* (2000) have investigated the accuracy for inspections involving two to six inspectors. Their research recommended that capture-recapture models ought not to be used with less than four inspectors unless shown to work in a particular environment. El Emam and Laitenberger (2001) report on the evaluation of capture-recapture models with two inspectors.

There is no benchmark for the number of inspectors to be used for inspection, hence we use 2 to 4 inspectors for our simulations.

Apart from the number of inspectors the other factor that affects the performance of an estimator is the defect detection abilities of the

inspectors. For example a team of 5 novices would certainly do worse than a team of 2-3 experts. Hence, it is more useful to employ a small number of experts than to employ a large number of novices. Therefore, we vary the inspector abilities from 0.1 (novice) to 0.9 (expert).

5.1.2 Number of Defects and their degree of Difficulty

The number of defects present in a piece of software can have a serious impact on the accuracy of the estimate. If the number of defects is low then the number of detection would be less thereby affecting the accuracy of the result. On the other hand if the number of defects is high then there will be more number of detections and hence a more accurate estimate.

The effect of number of defects is not considered openly in software inspection literature. The simulations conducted by Otis *et al.* (1978) used the number of defects that ranged from 100 to 800. Eick *et al.* (1992) present the results of inspections with the number of defects ranging approximately from 15 to 200. In their later work they present the data matrix of an inspection meeting with a document containing 47 defects. El Emam and Laitenberger (2001) performed simulations with 30 defects with the inspection team of two inspectors. In the biological framework of

capture-recapture studies the population is much higher, ranging to several hundred or even thousands of animals (White *et al.* 1982).

In practice, the number of defects that seriously affect a software, i.e. major defect is low and most of the CR models do not work well when the number of defects is low (El Emam *et al.* 1997, El Emam *et al.* 1998, El Emam *et al.*, 2000, Gupta 2003). A robust model should be able to estimate accurately under this condition. Our focus is to test models that work well with low number of defects and in our research we have considered a small defect population ranging from 10 to 30. Along with the number of defects the other factor affecting the accuracy of the estimate is the degree of difficulty of the defect (see Section 1.2). Since we are interested in major defects, a degree of difficulty of 0.1 (very hard to find) and 0.4 (moderately difficult to find) is used in the simulations.

5.1.3 Dependence Among Inspectors

As stated earlier, in our research we want to study the impact of dependence among inspectors on the accuracy of the estimate. Software inspection literatures have not addressed the issue of dependence so far and they assume that the inspectors work independently. However this assumption is

not necessarily true all the time, since inspectors can share their findings during informal or formal talks and hence can affect the accuracy of the result. In our study we use two levels of dependence, 0.2 (weak correlation) and 0.4 (moderate correlation) for the simulations.

5.2 Evaluation Criteria

We use the evaluation criteria used by El Emam *et al.* (2000) and Gupta (2003) to evaluate the performance of the models.

5.2.1 Bias, Failure and Dispersion

For each model median relative error (med (RE)) is calculated which is the median of the 1000 simulations. The med (RE) gives an indication of the bias of a model. RE is defined as follows:

$$RE = \frac{\hat{N} - N}{N} \tag{5.1}$$

where, $\hat{N}$ is the estimated number of defects and N denotes the actual number of defects.

Another quantity that was used to evaluate the estimators was the failure rate. This occurs, for example, due to divisions by zero. The last value used was the inter-quartile range (IQR) of the relative error. The IQR and the

presence of extreme outliers were used as a measure of dispersion in relative error (i.e., whether the extent of over/underestimates is consistent). For both the med (RE) and the IQR calculations, case wise deletion of missing values was performed. Missing values occurred when an estimator fails to provide an estimate.

5.2.2 Decision Accuracy

CR models are used to make a binary reinspection decision, i.e. reinspect or not reinspect. For controlling inspections, this decision would be based on whether the effectiveness of the inspection is above a specified threshold. The effectiveness threshold is set to ensure a high quality inspection that ensures that the most detectable defects have been detected in the software artefact. Since actual effectiveness is not known CR estimate are used to calculate the estimated effectiveness.

Let Q_p denote the threshold effectiveness set by the organization, then the decision can be stated in terms of the following inequality (Gupta, 2003):

$$Q_p \leq \frac{D}{\hat{N}} \tag{5.2}$$

where, $\frac{D}{\hat{N}}$ is the estimated inspection effectiveness. The artefact is passed on to the next phase if this inequality is satisfied. If it is not satisfied, then the artefact should be reinspected. The whole decision for controlling inspection effectiveness across many inspections can be defined as follows (El Emam *et al.* 2000, Gupta, 2003):

$$\hat{\lambda} = \begin{cases} 1, \hat{N} \leq \frac{D}{Q_p} \\ 0, \hat{N} > \frac{D}{Q_p} \end{cases} \tag{5.3}$$

where, $\hat{\lambda}$ is the decision based on the CR model, and is one (pass) if the estimated effectiveness is higher than or equal to a certain threshold, and zero (reinspect) if it is lower than the threshold.

To evaluate decision accuracy, the calculated decision based on the estimates, i.e. $\hat{\lambda}$ can be compared with the decision that would be made if the CR model was perfectly accurate (i.e., always made the correct decision), which is denoted by λ:

$$\lambda = \begin{cases} 1, N \leq \frac{D}{Q_p} \\ 0, N > \frac{D}{Q_p} \end{cases} \tag{5.4}$$

The results of an evaluation study over M inspections can be placed in a confusion matrix as shown in Table 5.1.

Table 5.1: Notation for a confusion matrix showing the decision of a CR model.

		$\hat{\lambda}$ = 0	$\hat{\lambda}$ = 1	
λ	0	m_{11}	m_{12}	M_{1+}
	1	m_{21}	m_{22}	M_{2+}
		M_{+1}	M_{+2}	M

The value of M is 1000, which is the number of simulation runs. Here, m_{11} is the number of times both $\hat{\lambda}$ and λ give the decision to reinspect the artefact, whereas m_{22} is the number of times both give the decision to pass the document. The decision accuracy in terms of the proportion of correct decisions that would be made using the estimates can be defined as:

$$\text{Decision Accuracy} = \text{DA} = \frac{m_{11} + m_{22}}{M} \tag{5.5}$$

Though this definition of decision accuracy seems perfect, it does not take into account the improvement due to the use of the CR model estimates. Reinspections are rarely performed in practice, hence, the "no reinspection" decision can be considered as the default one. Consider if 90% of the time the default decision is the correct one and the use of CR model estimates also results in achieving the correct decision 90% of the time, then one does not

gain anything from using the CR model estimates. Therefore, it is also necessary to consider the default decision.

El Emam *et al.* (2000) and Gupta (2003) used Relative Decision Accuracy (RDA) to get the improvement over the default decision which is defined as follows

$$RDA = DA - A_d \tag{5.6}$$

where A_d is the accuracy obtained when using the default decision, which in our case is always pass. A_d can be defined with respect to the confusion matrix as follows:

Table 5.2: Notation for a confusion matrix with the default decision.

		Default Decision: 0	Default Decision: 1	
λ	0	0	m_{12}	M_{1+}
	1	0	m_{22}	M_{2+}
		0	M_{+2}	M

Therefore,

$$A_d = \frac{m_{22}}{M} \tag{5.7}$$

Relative decision accuracy indicates how much better a CR model estimate is beyond the default decision-making criteria. It is positive if the CR model

decision is better, zero if they are the same and negative if the CR model decision is worse than the default decision.

Also, El Emam *et al.* (2000) and Gupta, (2003) mention two values of the threshold, 0.57 and 0.7 for the evaluation of the models and used those two values for Q_P during simulation. In a study (Briand *et al.*, 1998) it was found that the average effectiveness of code inspections in practice was 0.57, and the most likely value was 0.7 and hence Gupta (2003), used these two values. The lower threshold intended to ensure "above average" defect detection effectiveness, and the higher threshold is intended to ensure "best in class" effectiveness.

5.3 Simulations

This section describes the methodology used to carry out the simulations. In our model the posterior distribution for N is given by Eqn. 4.15 and it was very difficult to get the posterior estimate using the traditional integration method. Hence, we used Gibbs Sampling (George *et al.* 1992) to get the estimate for N.

5.3.1 Gibbs Sampling

The Gibbs sampler is a technique for generating random variables from a (marginal) distribution indirectly, without having to calculate the density (George *et al.* 1992). Most of the applications of Gibbs sampler have been in Bayesian models and it reduces the amount of work required to do complicated calculations. For example, suppose we are given a joint density $f(x, y_1, ..., y_p)$ and are interested in calculating the characteristics of the marginal density,

$$f(x) = \int ..... \int f(x, y_1, ... y_p) dy_1 ..... dy_p \tag{5.8}$$

such as the mean or variance. The traditional approach would be to calculate $f(x)$ directly and use to get the desired characteristic. However, there are many cases in which the integrations in Eqn. 5.8 can be very difficult to perform. In such cases the Gibbs sampler provides an alternative method to obtain $f(x)$. For a large variety of priors $\pi(N, p)$, the Gibbs sampler gives an approximate marginal distribution of N, namely $\pi(N \mid D)$ by simulated sampling from the conditional distributions (Gelfand *et al.* 1990 and Casella *et al.* 1992),

$$\pi(N \mid p, D), \quad \pi(p \mid N, D) \tag{5.9}$$

A Gibbs sequence,

$$N^{(0)}, p^{(0)}, N^{(1)}, p^{(1)}, \ldots\ldots, \tag{5.10}$$

can be generated starting with an initial value $N^{(0)}$ for N, where,

$$N^{(k)} \sim \pi(N \mid p^{(k-1)}, D) \text{ and } p^{(k)} \sim \pi(p \mid N^{(k)}, D).$$

To apply the Gibbs sampler we have to first derive the conditional distributions for the likelihood in Eqn. 4.14. As already explained in Section 4.4.2, N and p are a priori independent and, therefore, following Castledine (1981) the posterior conditionals for Eqn. 4.14 can be written as,

$$\pi(N \mid p, D) \propto \frac{N!}{(N-n)!}\left\{\prod_{i=1}^{k}(1-p_i)\right\}^{N} \pi(N), \tag{5.11}$$

$$\pi(p \mid N, D) \propto \left\{\prod_{i=1}^{k} p_i^{n_i}(1-p_i)^{N-n_i}\right\}\pi(p) \tag{5.12}$$

The next step now is to specify the prior distribution for N and p and calculate the conditional posteriors (Eqn. 5.11 and Eqn. 5.12). We used Jeffreys prior (George *et al.* 1992) $\pi(N) = 1/N$ for N and the conditional posterior of N is the negative binomial (see Eqn. 5.13) with parameters $n-1$ and $1-\pi(1-p_i)$.

For simulation of $\{p^{(k)}\}$, assuming independent beta prior $Be(a,b)$ for p_i's, the conditional posterior of p_i is $Be(n_i+a, N-n_i+b)$ (see Eqn. 5.14). Therefore,

$$\pi(N \mid p, D) = \frac{\Gamma(n-1+N)}{N!\Gamma(n-1)} \{1-\pi(1-p_i)\}^{n-1} \{1-(1-\pi(1-p_i))\}^{N} \tag{5.13}$$

$$\pi(p \mid N, D) = \frac{\Gamma(N+a+b)}{\Gamma(n_i+a)\Gamma(N-n_i+b)} p^{n_i+a-1}(1-p)^{N-n_i+b-1} \tag{5.14}$$

Using above conditional distributions the value N can be calculated by applying Gibbs Sampling.

5.3.2 Selection of Bayesian Parameters

The most important factor in a Bayesian method is the choice of the prior distribution. In our research the prior distribution used is the Dirichlet distribution $(\kappa\alpha)$ described by κ and α (see Section 4.4.2). As stated earlier we use Gibbs sampling to obtain the estimate for N and the conditional distributions are given by Eqn. 5.13 and Eqn. 5.14.

This distribution is described by a and b parameters which are determined by the prior mean $E(p)$ and the standard deviation σ_p, (see Eqn. 4.12).

Since we do not have any knowledge of the prior, we try to determine these parameters from the data matrix described in Section 3.6. We define the mean as n_0/N. We allow our estimated number to deviate as much as up to $\pm 30\,\%$ from the actual defect population. Thus, the prior mean can now be written as

$$E(p) = \frac{n_0}{N - Er \times N} \tag{5.15}$$

where, offset from the actual mean, $Er = 0,\ \pm 0.1,\ \pm 0.2,\ \pm 0.3$.

We chose values of 0.025, 0.05, 0.075, 0.1 and 0.2 for standard deviation. For each value of the standard deviation, we had seven values of E (p) as given above. Now, a and b can be calculated from $E(p)$ and σ_p as follows,

$$a = E(p)\left(\frac{E(p)}{\text{var}(p)}[1 - E(p)] - 1\right) \tag{5.16}$$

$$b = a\frac{1 - E(p)}{E(p)} \tag{5.17}$$

κ and α are derived from a and b using the following formula,

$$a = \kappa\alpha \tag{5.18}$$

$$b = \kappa(1 - \alpha) \tag{5.19}$$

5.3.3 Generating the Data Matrix

We used two values of correlation (dependence), i.e. 0.2 (weak correlation) and 0.4 (moderately strong correlation) and used the method described by Basu (2003) to generate the data matrix. A multivariate normal distribution is used to get $Z_i = (Z_{i1},.....Z_{iR})^T$.

Multivariate Normal Distribution

A random vector $X = [X_1,....X_N]$ follows a multivariate normal distribution, also sometimes called a multivariate Gaussian distribution, if it satisfies the following equivalent conditions:

- every linear combination $Y = a_1X_1 + ... + a_NX_N$ is normally distributed

- there is a random vector $Z = [Z_1,....Z_M]$, whose components are independent standard normal random variables, a vector $\mu = [\mu_1,....\mu_N]$ and an $N \times M$ matrix A such that

$$X = AZ + \mu \tag{5.20}$$

- there is a vector $\mu = [\mu_1,....\mu_N]$ and a symmetric, covariance matrix Σ (NxN matrix) such that X has density

$$f_X(x_1,....x_N) = \frac{1}{(2\pi)^{N/2}\,|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \qquad (5.21)$$

At first, we generated the covariance matrix Σ using the following relation,

$$\Sigma = (1-\rho)I + \rho E \qquad (5.22)$$

where, I is the identity matrix and each entry of E is 1. ρ represents the correlation and as stated before we use two values of correlation 0.2 and 0.4.

We used Cholesky decomposition to get A (see Eqn. 5.20) from the covariance matrix Σ as follows,

$$\left.\begin{aligned} A_{i,j} &= \frac{1}{A_{j,j}}\left(\Sigma_{i,j} - \sum_{k=1}^{j-1} A_{i,k}A_{j,k}\right), \text{ for } i<j \\ A_{i,i} &= \sqrt{\Sigma_{i,i} - \sum_{k=1}^{i-1} A_{i,k}^2} \end{aligned}\right\} \qquad (5.23)$$

The matrix X in Eqn. 5.20 is used to denote the degree of difficulty of each defect. Now Z can be generated as,

$$Z = A^{-1}(X-\mu) \qquad (5.24)$$

Next, we assign $X_{ir} = 1$, if $\Phi(Z_{ir}) \le p_r$, and = 0 otherwise where $\Phi(\bullet)$ is the normal distribution function and $p_1,....,p_R$ are the inspector detection capabilities.

5.3.4 Study Points

We consider the following sets of variables for our simulations: the number of difficult defects, the probability of a defect being found, number of inspectors and their defect detection capability. We performed simulations with the population size of 10, 20 and 30 difficult defects with detection probabilities of 0.1 (very difficult to detect) and 0.4 (moderately difficult). We used 2, 3 and 4 inspectors for the simulations and we also considered for the simulations the general defect detection effectiveness of the inspectors themselves (i.e., their ability to detect defects). The last variable that we used was the correlation factor as described in Section 5.3.3.

For 2-inspector simulations, the inspector abilities were denoted as P_x and P_y for inspector X and Y respectively. These abilities are defined as follows:

- $\{P_x = 0.1, P_y = 0.9\}$: one with low capability and the other with high capability
- $\{P_x = 0.25, P_y = 0.75\}$: one with low capability and other with relatively high capability
- $\{P_x = 0.4, P_y = 0.6\}$: similar capabilities
- $\{P_x = 0.3, P_y = 0.3\}$: both relatively low capabilities
- $\{P_x = 0.8, P_y = 0.8\}$: both relatively high capabilities
- $\{P_x = 0.5, P_y = 0.5\}$: both average capabilities

For 3-inspector simulations, the inspector abilities were denoted as P_x, P_y and P_z for inspector X, Y and Z respectively. These abilities are defined as follows:

- $\{P_x = 0.1, P_y = 0.5, P_z = 0.9\}$
- $\{P_x = 0.25, P_y = 0.5, P_z = 0.75\}$
- $\{P_x = 0.4, P_y = 0.5, P_z = 0.6\}$
- $\{P_x = 0.3, P_y = 0.5, P_z = 0.6\}$
- $\{P_x = 0.8, P_y = 0.5, P_z = 0.8\}$
- $\{P_x = 0.5, P_y = 0.5, P_z = 0.5\}$

For 4-inspector simulations, the inspector abilities were denoted as P_x, P_y, P_z and P_t for inspector X, Y, Z and T respectively. These abilities are defined as follows:

- $\{P_x = 0.1, P_y = 0.4, P_z = 0.6, P_t = 0.9\}$
- $\{P_x = 0.1, P_y = 0.1, P_z = 0.9, P_t = 0.9\}$
- $\{P_x = 0.5, P_y = 0.5, P_z = 0.9, P_t = 0.9\}$
- $\{P_x = 0.9, P_y = 0.9, P_z = 0.9, P_t = 0.9\}$
- $\{P_x = 0.1, P_y = 0.1, P_z = 0.1, P_t = 0.1\}$
- $\{P_x = 0.5, P_y = 0.5, P_z = 0.5, P_t = 0.5\}$
- $\{P_x = 0.5, P_y = 0.5, P_z = 0.5, P_t = 0.9\}$
- $\{P_x = 0.1, P_y = 0.5, P_z = 0.5, P_t = 0.5\}$

- $\{P_x = 0.1, P_y = 0.1, P_z = 0.1, P_t = 0.9\}$
- $\{P_x = 0.1, P_y = 0.9, P_z = 0.9, P_t = 0.9\}$

Study points for each type of simulation were constructed by combining the values of the above mentioned variables. For each study point 1000 inspections were simulated. We selected the above values for our study points due to the following reasons. Briand *et al.* (2000) suggested that, for a reasonable performance of CR models the minimum number of inspectors is 4, and El Emam and Laitenberger (2001) concluded that selecting 2 inspectors is a reasonable choice. Also, a significant incremental gain in median relative error was not observed from 3 to 4 inspectors (Freimut 1997), we therefore selected 2, 3 and 4 inspectors for our simulations. We selected the range of defects as 10, 20 and 30 due to the reason mentioned in Section 4.1. Since we are dealing with only one type of defects, i.e. the difficult defects, which can have serious impact on the software quality, therefore we chose to vary the defects using two degrees of difficulty, 0.1 (very difficult to detect) and 0.4 (moderately difficult to detect). Finally, for the last variable, we chose the same inspector abilities as was considered by El Emam and Laitenberger (2001) for two inspectors. For 3 inspectors, we added an average ability inspector and likewise, for 4 inspectors, we formed the teams with low capability, average capability, and high capability and mixed capabilities inspectors.

Chapter 6

Simulation Results

The purpose of our simulations has been to study the effect of dependence among inspectors on the decision accuracy and compare it with the previous Bayesian model (Gupta 2003, Ananda 1997). As previously mentioned (Chapters 2, 3 and 4) we have used realistic scenarios of small numbers of rather difficult defects, which can have serious impact on the quality of the product. In this chapter, we present the results in terms of median relative error, IQR, number of failures, decision accuracy and relative decision accuracy for both the thresholds of 0.57 and 0.7 (see Section 5.2.2) for 2, 3 and 4 inspectors and the defect size of 10, 20 and 30. The above results are also presented with respect to two degrees of difficulties, 0.1 (extremely difficult defects) and 0.4 (moderately difficult defects); and two degrees of dependence, 0.2 (weak correlation) and 0.4 (moderate correlation) as was done by Basu and Ebrahimi (1998), Basu and Ebrahimi (2001), and Basu (2003).

6.1 Main Results

The main variables in our simulations were the number of inspectors and their abilities, number of defects and their degree of difficulty and the correlation factor. The detailed results are given in Appendix A. We present a summary of the results in this section.

We observed that the DA decreased as the standard deviation of the prior increased. We also noted that within a given standard deviation, the DA did not seem to change significantly with respect to the error (Er). It was also observed that the failure rate in each simulation increased when the abilities of the inspectors decreased and also when the defects became more difficult to find. Additionally, the DA increased with an increase in the dependence among inspectors. These general trends were observed throughout the entire simulations.

1. We first considered changing the number of inspectors while keeping all the other parameters fixed. The results for 10, 20 and 30 defects and the degree of difficulty of 0.1 and standard deviation of 0.025 are shown in Tables 6.1, 6.2 and 6.3. The DA for both the thresholds (0.57 and 0.7) increased as the number of inspectors increased. Here we chose the inspector abilities as 0.5 for all the inspectors, which means a moderate ability. The DA of 10 defects increased from 0.82 to 1.0 for a

threshold of 0.7 as the number of inspectors increased from 2 to 4. There was a similar increase in DA for a threshold of 0.57. This trend was observed for 20 and 30 defects as well (Fig. 6.1).

2. We obtained the second set of results by varying the ability of the inspectors. In Table 6.4, we show the DA for a case of 4 inspectors and 20 defects. The two groups of inspectors have been chosen to illustrate the extreme variations, i.e. a team of 4 experts and a team of 4 novices. The DA for the team of experts is significantly more than the DA of a team of novices, for both the thresholds used in our simulations (Fig. 6.2).
3. Next, we varied the number of defects while keeping the number and the ability of the inspectors fixed. Tables 6.5, 6.6 and 6.7 show the results of DA for 2, 3 and 4 inspectors respectively. The DA increased with increasing number of defects for a given number of inspectors. For example, the DA increases from 0.54 to 0.84 for a threshold of 0.7 for 10 defects. Similar increase in DA was observed for 0.57 thresholds as well. The same trend was observed for all the inspection teams.
4. Lastly, we varied the degree of correlation while keeping all the other parameters fixed. Tables 6.8, 6.9, 6.10 and 6.11 show the DA for 10 and 20 defects. The DA increased with the increase in the degree of correlation. Similar trend is observed for 3 and 4 inspectors also.

5. The most important result is that the DA of Dependent Bayesian CR model is remarkably higher than the Bayesian CR model (Gupta, 2003) under realistic conditions.

	2 Inspectors		3 Inspectors		4 Inspectors	
Er	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.91	0.97	1.00	1.00	1.00	1.00
0.2	0.77	0.90	0.99	1.00	1.00	1.00
0.1	0.89	0.95	0.99	1.00	1.00	1.00
0.0	0.82	0.95	0.98	1.00	1.00	1.00
-0.1	0.79	0.95	0.96	1.00	1.00	1.00
-0.2	0.87	0.95	1.00	1.00	1.00	1.00
-0.3	0.82	0.88	0.97	1.00	1.00	1.00

Table 6.1: DA for 10 Defects of 0.1 degree of difficulty and standard deviation of 0.025 and ρ =0.2

	2 Inspectors		3 Inspectors		4 Inspectors	
Er	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.93	0.98	1.00	1.00	1.00	1.00
0.2	0.90	0.99	1.00	1.00	1.00	1.00
0.1	0.87	0.98	0.99	1.00	1.00	1.00
0.0	0.90	1.00	1.00	1.00	1.00	1.00
-0.1	0.92	0.98	0.99	1.00	1.00	1.00
-0.2	0.79	0.97	1.00	1.00	1.00	1.00
-0.3	0.92	0.97	0.99	1.00	1.00	1.00

Table 6.2: DA for 20 Defects of 0.1 degree of difficulty and standard deviation of 0.025 and ρ =0.2

	2 Inspectors		3 Inspectors		4 Inspectors	
Er	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.87	1.00	1.00	1.00	1.00	1.00
0.2	0.91	0.99	1.00	1.00	1.00	1.00
0.1	0.93	0.99	1.00	1.00	1.00	1.00
0.0	0.91	0.99	1.00	1.00	1.00	1.00
-0.1	0.90	1.00	1.00	1.00	1.00	1.00
-0.2	0.91	1.00	1.00	1.00	1.00	1.00
-0.3	0.84	0.99	0.99	1.00	1.00	1.00

Table 6.3: DA for 30 Defects of 0.1 degree of difficulty and standard deviation of 0.025 and ρ =0.2

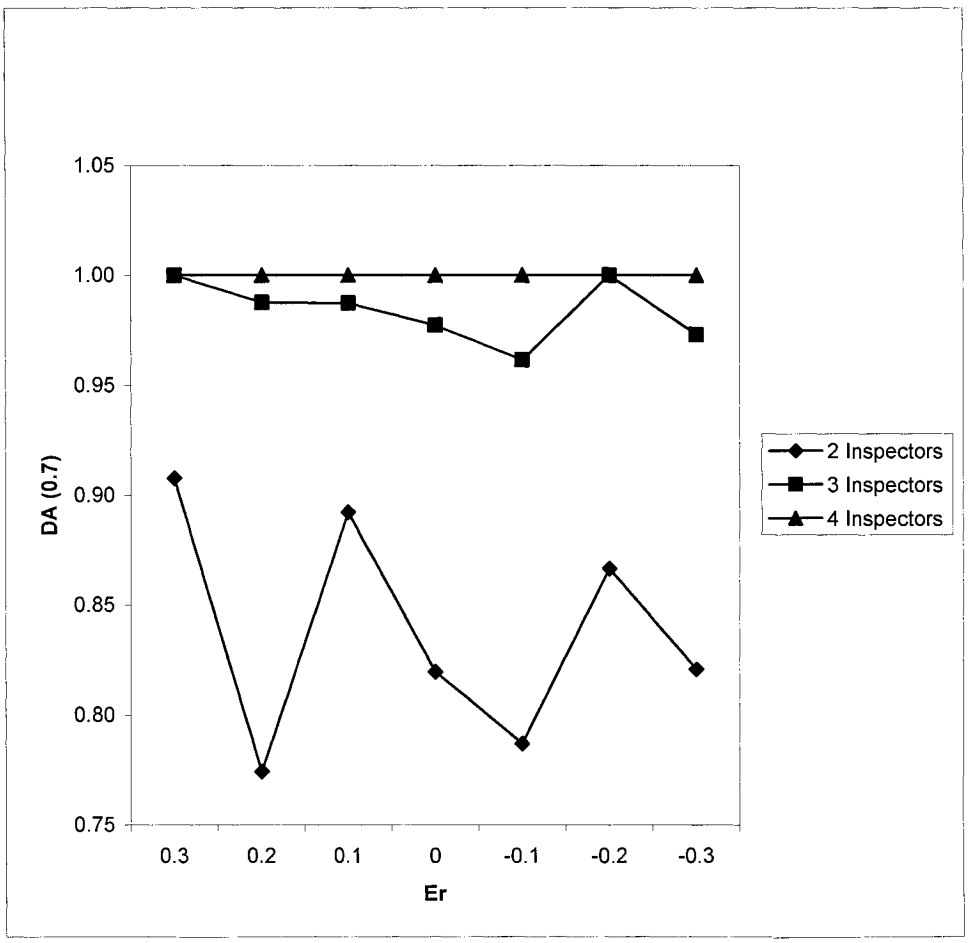

Fig. 6.1: DA for 10 defects of 0.1 degree of difficulty and standard deviation of 0.025 and ρ =0.2

	4 Inspectors Novices (0.1,0.1,0.1,0.1)		4 Inspectors Experts (0.9,0.9,0.9,0.9)	
Er	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.04	0.21	1.00	1.00
0.2	0.07	0.33	1.00	1.00
0.1	0.12	0.23	1.00	1.00
0	0.06	0.39	1.00	1.00
-0.1	0.04	0.29	1.00	1.00
-0.2	0.14	0.28	1.00	1.00
-0.3	0.07	0.22	1.00	1.00

Table 6.4: DA for a team of experts and a team of novices. The table shows 20 defects of 0.1 degree of difficulty and standard deviation of 0.025 and ρ =0.2.

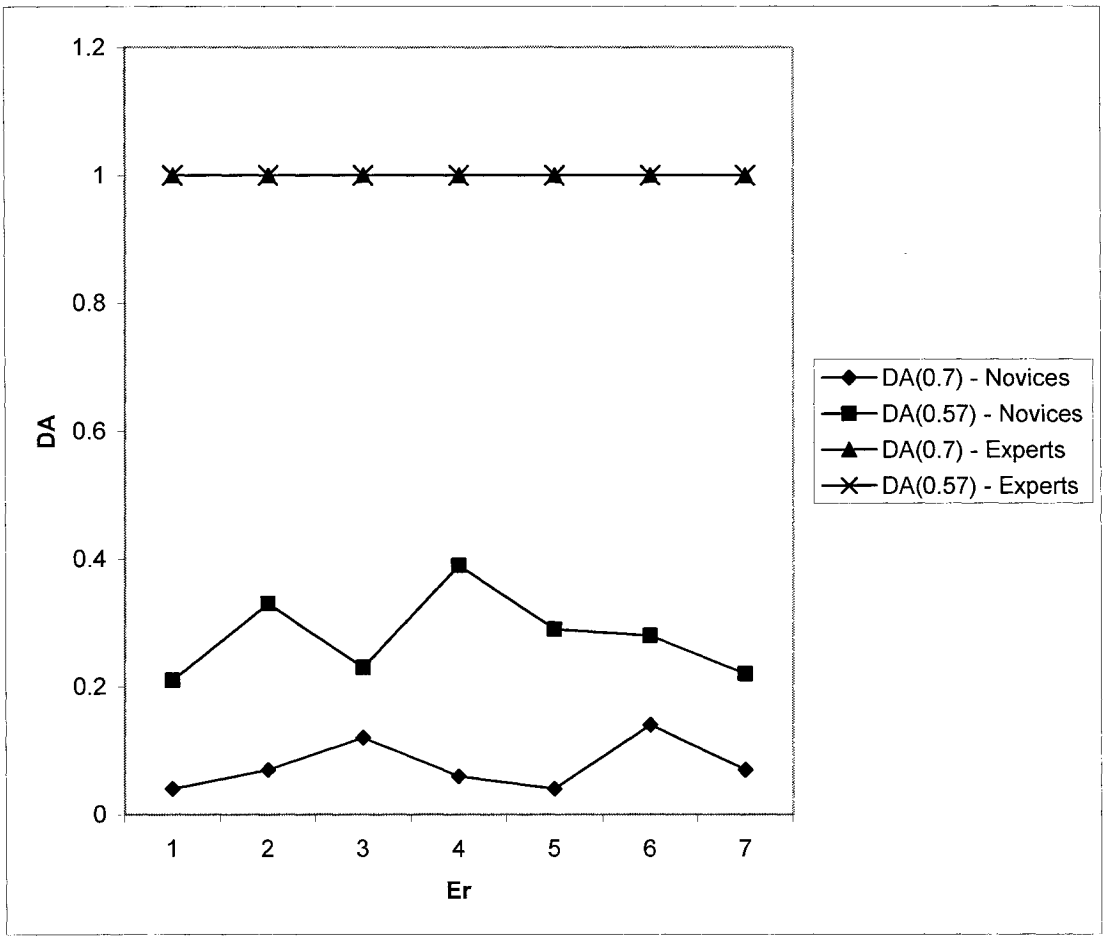

Fig. 6.2: DA for a team of experts (0.9, 0.9, 0.9, 0.9) and a team of novices (0.1, 0,1, 0.1, 0.1). The table shows 20 defects of 0.1 degree of difficulty and standard deviation of 0.025 and ρ =0.2.

	10 Defects		20 Defects		30 Defects	
Er	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.54	0.79	0.74	0.93	0.79	1.00
0.2	0.61	0.80	0.57	0.94	0.9	0.99
0.1	0.57	0.69	0.60	0.97	0.91	0.99
0	0.54	0.81	0.68	0.96	0.84	0.99
-0.1	0.59	0.74	0.66	0.96	0.84	0.99
-0.2	0.61	0.69	0.70	0.97	0.85	0.98
-0.3	0.61	0.81	0.74	0.97	0.8	0.97

Table 6.5: DA for 2 Inspectors with 0.1 degree of difficulty and standard deviation of 0.025. (Inspector abilities are 0.25, 0.75) and ρ =0.2.

Er	10 Defects		20 Defects		30 Defects	
	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.85	0.82	0.98	1.00	1.00	1.00
0.2	0.79	0.89	0.99	1.00	1.00	1.00
0.1	0.8	0.95	0.96	1.00	1.00	1.00
0	0.78	0.91	0.97	1.00	1.00	1.00
-0.1	0.82	0.96	0.91	1.00	1.00	1.00
-0.2	0.81	0.89	1.00	1.00	0.99	1.00
-0.3	0.85	0.87	0.99	1.00	1.00	1.00

Table 6.6: DA for 3 Inspectors with 0.1 degree of difficulty and standard deviation of 0.025. (Inspector abilities are 0.25, 0.50, and 0.75) and ρ =0.2.

Er	10 Defects		20 Defects		30 Defects	
	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.86	0.75	0.88	0.99	0.98	1.00
0.2	0.74	0.86	0.88	0.99	0.98	1.00
0.1	0.79	0.71	0.87	1.00	1.00	1.00
0.0	0.70	0.77	0.92	0.98	0.99	1.00
-0.1	0.85	0.78	0.87	0.99	0.99	1.00
-0.2	0.69	0.77	0.90	0.98	0.98	1.00
-0.3	0.71	0.85	0.94	0.98	0.99	1.00

Table 6.7: DA for 4 Inspectors with 0.1 degree of difficulty and standard deviation of 0.025. (Inspector abilities are 0.1, 0.6, 0.2 and 0.4) and ρ =0.2.

Er	2 Inspectors		3 Inspectors		4 Inspectors	
	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.91	0.97	1.00	1.00	1.00	1.00
0.2	0.77	0.90	0.99	1.00	1.00	1.00
0.1	0.89	0.95	0.99	1.00	1.00	1.00
0	0.82	0.95	0.98	1.00	1.00	1.00
-0.1	0.79	0.95	0.96	1.00	1.00	1.00
-0.2	0.87	0.95	1.00	1.00	1.00	1.00
-0.3	0.82	0.88	0.97	1.00	1.00	1.00

Table 6.8: DA for 10 defects with 0.1 degree of difficulty and standard deviation of 0.025. (Inspector abilities are 0.5) and ρ =0.2.

Er	2 Inspectors		3 Inspectors		4 Inspectors	
	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.88	0.97	1.00	1.00	1.00	1.00
0.2	0.85	0.96	0.97	1.00	1.00	1.00
0.1	0.88	0.97	0.99	1.00	1.00	1.00
0	0.84	0.97	1.00	1.00	1.00	1.00
-0.1	0.87	0.98	1.00	1.00	1.00	1.00
-0.2	0.88	0.96	1.00	1.00	1.00	1.00
-0.3	0.90	0.99	1.00	1.00	1.00	1.00

Table 6.9: DA for 10 defects with 0.1 degree of difficulty and standard deviation of 0.025. (Inspector abilities are 0.5) and ρ =0.4.

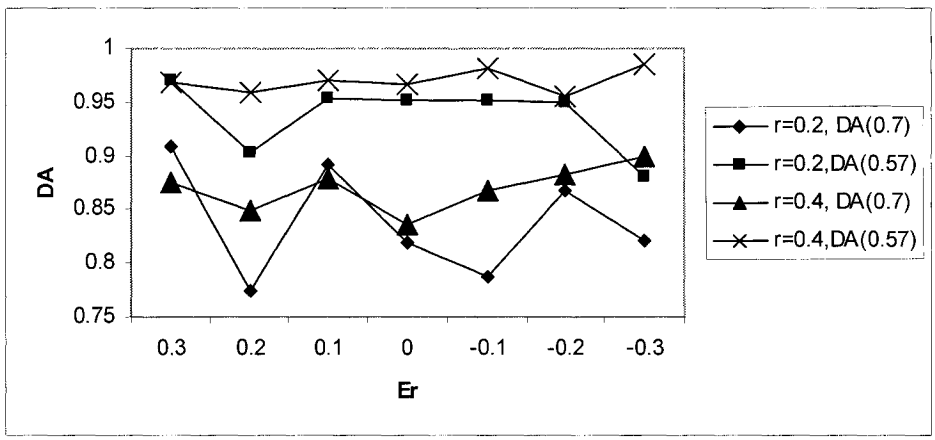

Fig. 6.3: Comparison for ρ =0.2 and 0.4, 10 defects and 0.1 degree of difficulty, 2 inspector and moderate inspector abilities (0.5) and SD = 0.025.

Er	2 Inspectors		3 Inspectors		4 Inspectors	
	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.93	0.98	1.00	1.00	1.00	1.00
0.2	0.90	0.99	1.00	1.00	1.00	1.00
0.1	0.87	0.98	0.99	1.00	1.00	1.00
0	0.90	1.00	1.00	1.00	1.00	1.00
-0.1	0.92	0.98	0.99	1.00	1.00	1.00
-0.2	0.79	0.97	1.00	1.00	1.00	1.00
-0.3	0.92	0.97	0.99	1.00	1.00	1.00

Table 6.10: DA for 20 defects with 0.1 degree of difficulty and standard deviation of 0.025. (Inspector abilities are 0.5) and ρ =0.2.

Er	2 Inspectors		3 Inspectors		4 Inspectors	
	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)	DA(0.7)	DA(0.57)
0.3	0.84	1.00	1.00	1.00	1.00	1.00
0.2	0.87	0.99	1.00	1.00	1.00	1.00
0.1	0.86	0.95	1.00	1.00	1.00	1.00
0	0.90	0.99	1.00	1.00	1.00	1.00
-0.1	0.91	1.00	1.00	1.00	1.00	1.00
-0.2	0.92	1.00	1.00	1.00	1.00	1.00
-0.3	0.84	0.99	1.00	1.00	1.00	1.00

Table 6.11: DA for 20 defects with 0.1 degree of difficulty and standard deviation of 0.025. (Inspector abilities are 0.5) and ρ =0.4.

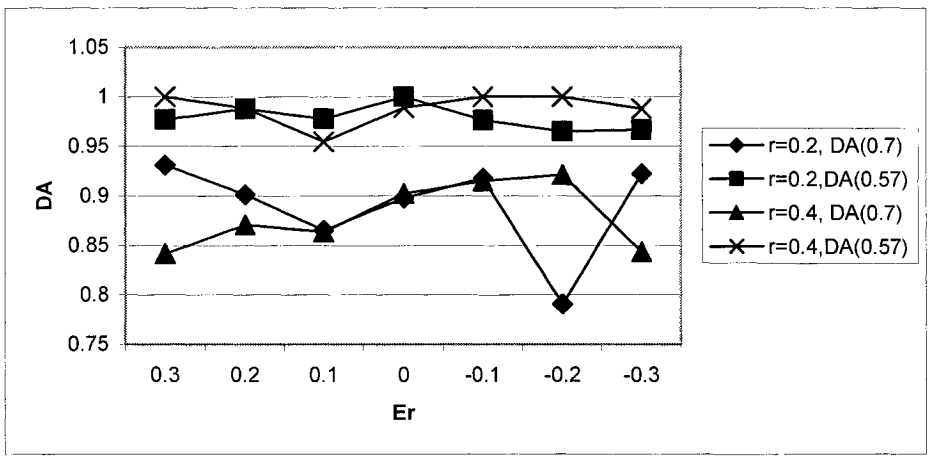

Fig. 6.4: Comparison for ρ =0.2 and 0.4, 20 defects and 0.1 degree of difficulty, 2 inspector and moderate inspector abilities (0.5, 0.5) and SD = 0.025.

6.2 Explanation of Results

We can understand the results and the behaviour of the Bayesian model in terms of the data matrix and the prior (defined in Sections 3.6 and 4.2 respectively). The product of the ability of the inspectors and the number of defects and their degree of difficulty determine the sparseness of the data matrix. The estimated population

depends on the values of n_i, a and b parameters of the Eqn. 5.13 and Eqn. 5.14. The parameter n_i determines the number of defects found by each inspector. It is clear from Eqn. 6.3 that if $n_i = 0$ (happens more frequently when data matrix is sparse), the estimated population depends more on a and b values, and hence the prior mean $E(p)$ and standard deviation σ_p. On the other hand when the data matrix is filled (happens when the ability of the inspectors is high and the defects are easier to find, as well as the defects being more in number) the estimated population depends more on parameter n_i. Therefore, in the case of a sparse data matrix, the estimated population depends entirely on the prior mean and standard deviation; hence it is very important to have an accurate value of the prior. This can be explained by taking the example of worst-case scenario when the data matrix is extremely sparse. This is the case with 2 inspectors of low capabilities and 10 defects of 0.1 degree of difficulty. Here, with standard deviation of 0.025 and all its seven values of $E(p)$, the DA remains almost constant, but when the data matrix gets filled up the DA no longer remains entirely dependent on the prior mean as can be seen with 4 inspectors of moderate abilities and 30 defects. The failure to estimate the population occurs when the data matrix is completely empty. This happens when the team consists of novices and the degree of difficulty of the defects is high.

Given the above explanation of the data matrix, we can easily understand all the results of the Bayesian model. The data matrix is sparse for low number of defects and gets filled with large number of defects as well as larger number of inspectors, thereby increasing the DA. For example, there is a 25% gain in the DA as the defect population increases from 10 to 20 defects, and from 20 to 30 defects for 2 inspectors and with 0.1 degree of difficulty (see Table 6.5) Also, DA is increased as the inspector abilities increase from a team of novices to a team of experts. Basically, for novices the matrix will be sparse and their subjective estimates will be wrong in the worst-case scenario.

The main focus of our research was to study the effect of dependence among inspectors. Tables 6.8, 6.9, 6.10 and 6.11 show the DA for the two values of ρ for 10 and 20 defects. It is clear from these tables that the DA of $\rho = 0.4$ is better then the lower value of $\rho = 0.2$. The reason for this increase can be understood from the parameter n_i. As the dependence is increased, the inspectors will find more errors and hence the data matrix gets filled up and therefore the estimate about the number of errors is better.

Chapter 7

Comparison Between the two Bayesian Models

In Chapter 4 we discussed two types of Bayesian models, i.e. Bayesian model with independent inspectors and Bayesian model with dependent inspectors. In this chapter we compare the two models based on their performance under similar simulation parameters and conditions.

7.1 DA of the two Bayesian Models

Figs. 7.1 to 7.8 show the comparison graphs for decision accuracy (DA) for the two Bayesian models. Figs. 7.1 and 7.2 show the DA (0.7 and 0.57 thresholds respectively) as a function of Er for 2, 3 and 4 inspectors for both Bayesian models for 10 defects and 0.1 degree of difficulty and ρ=0.2. Figs. 7.3 and 7.4 show the DA (0.7 and 0.57 thresholds respectively) for both Bayesian models for 10 defects and 0.1 degree of difficulty and ρ=0.4. Figs. 7.5 and 7.6 show the DA (0.7 and 0.57 thresholds respectively) as a function of Er for 2,3 and 4 inspectors for both Bayesian models for 20 defects and 0.1 degree of difficulty and ρ=0.2. Figs. 7.7 and 7.8 show the DA (0.7 and 0.57 thresholds respectively) for both Bayesian models for 20 defects and 0.1

degree of difficulty and ρ=0.4. We have chosen DA as a comparison parameter between the two models as we essentially use relative error as an input parameter (Er).

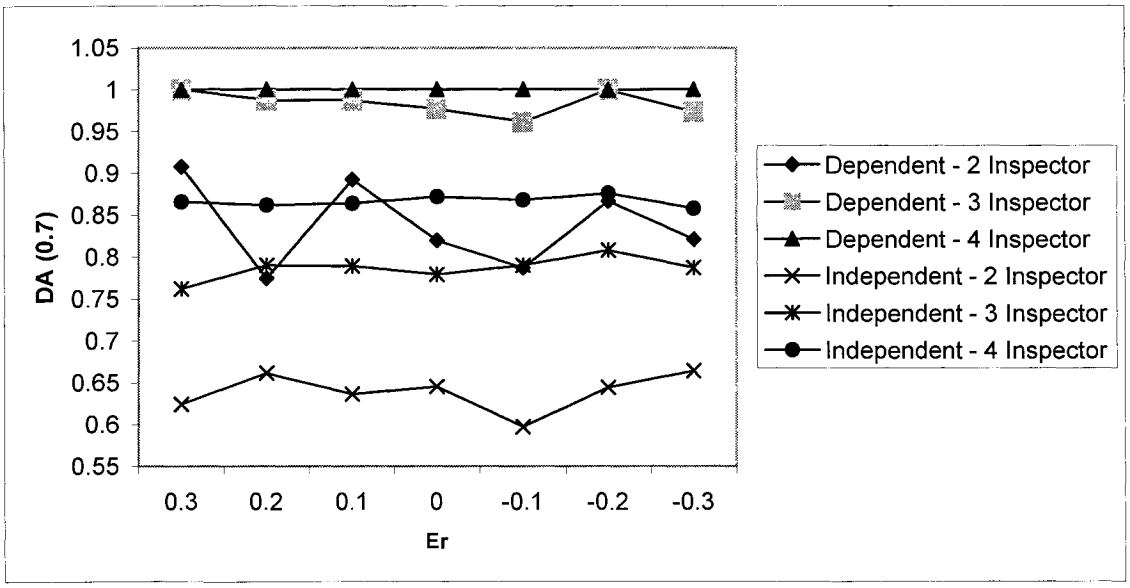

Fig. 7.1: Comparison of DA of 0.7 for the two Bayesian models for 10 defects and 0.1 degree of difficulty and ρ =0.2. Inspection team consists of moderate abilities, i.e. 0.5.

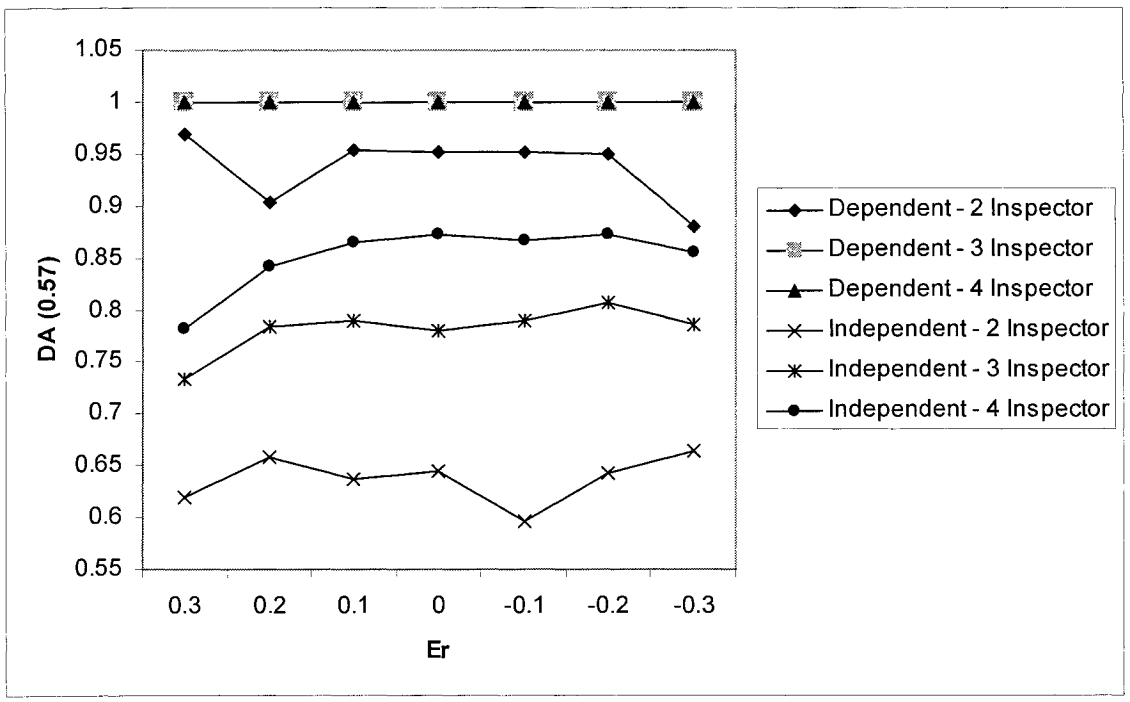

Fig. 7.2: Comparison of DA of 0.57 for the two Bayesian models for 10 defects and 0.1 degree of difficulty and ρ =0.2. Inspection team consists of moderate abilities, i.e. 0.5.

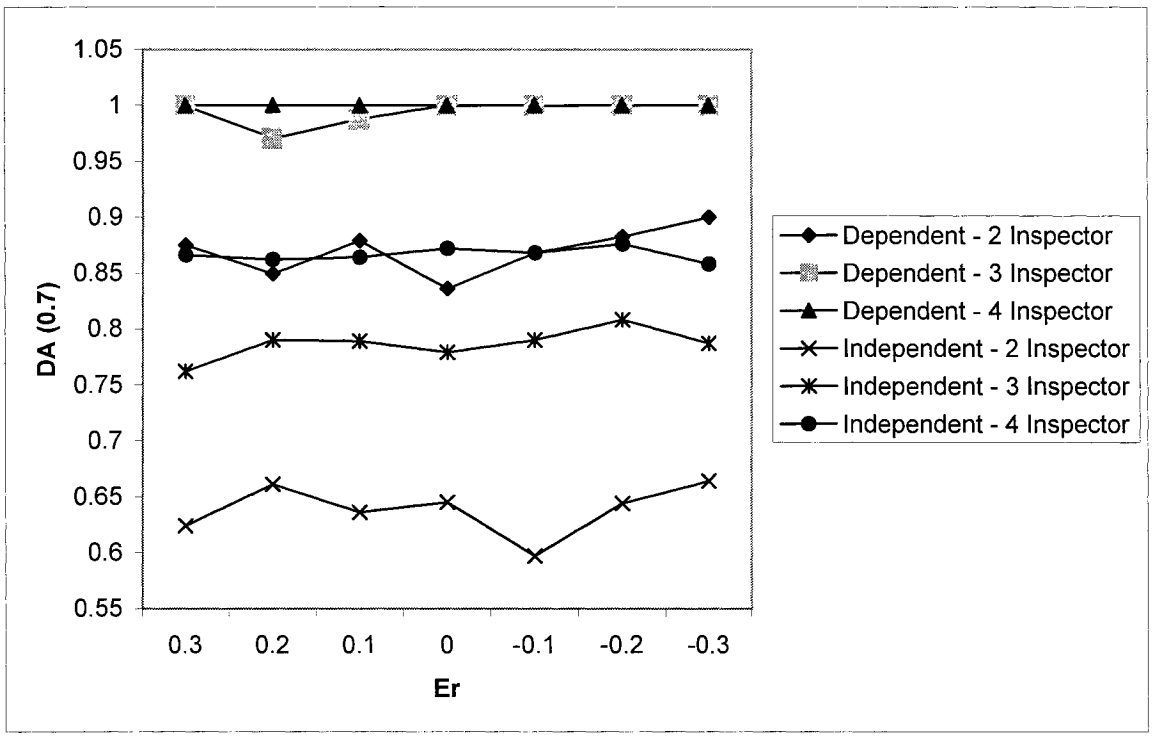

Fig. 7.3: Comparison of DA of 0.7 for the two Bayesian models for 10 defects and 0.1 degree of difficulty and ρ =0.4. Inspection team consists of moderate abilities, i.e. 0.5

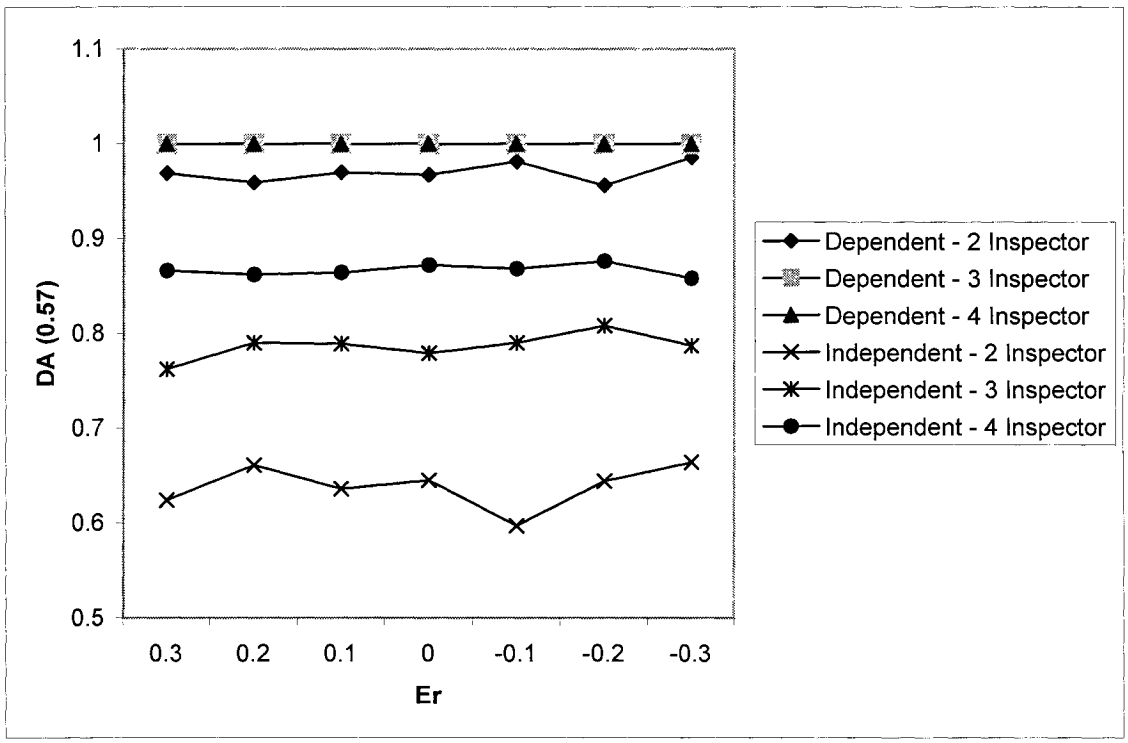

Fig. 7.4: Comparison of DA of 0.57 for the two Bayesian models for 10 defects and 0.1 degree of difficulty and ρ =0.4. Inspection team consists of moderate abilities, i.e. 0.5

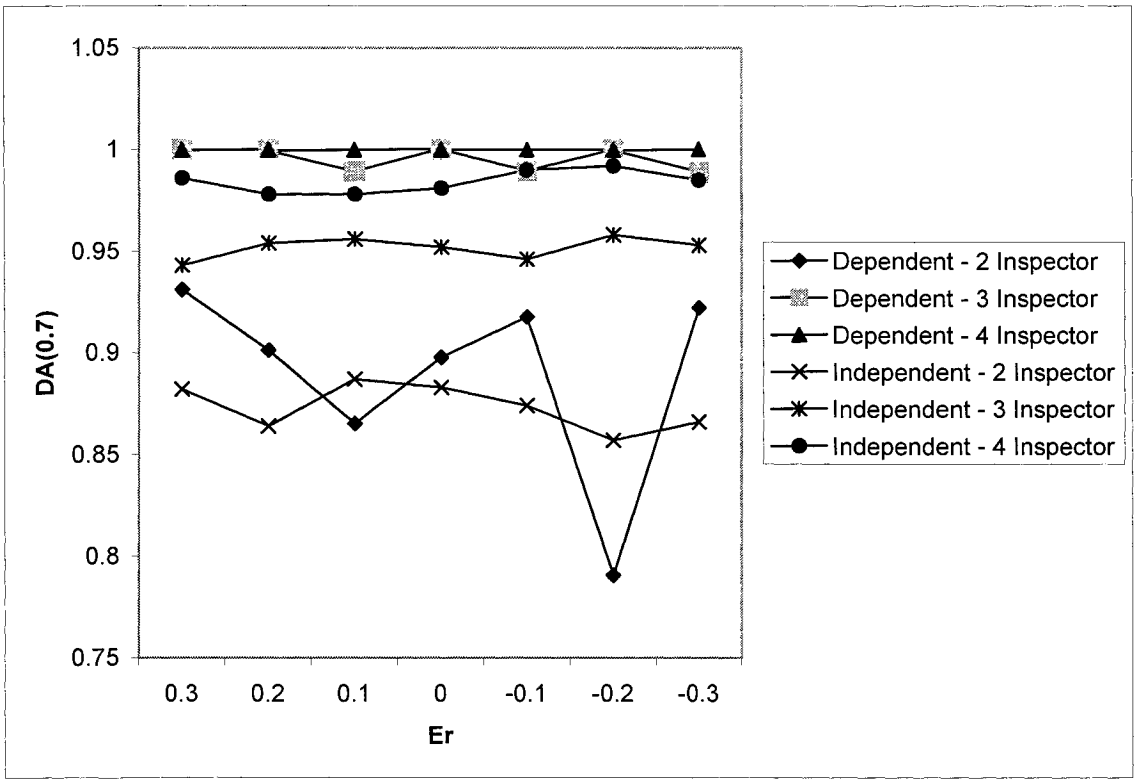

Fig. 7.5: Comparison of DA of 0.7 for the two Bayesian models for 20 defects and 0.1 degree of difficulty and ρ =0.2. Inspection team consists of moderate abilities, i.e. 0.5

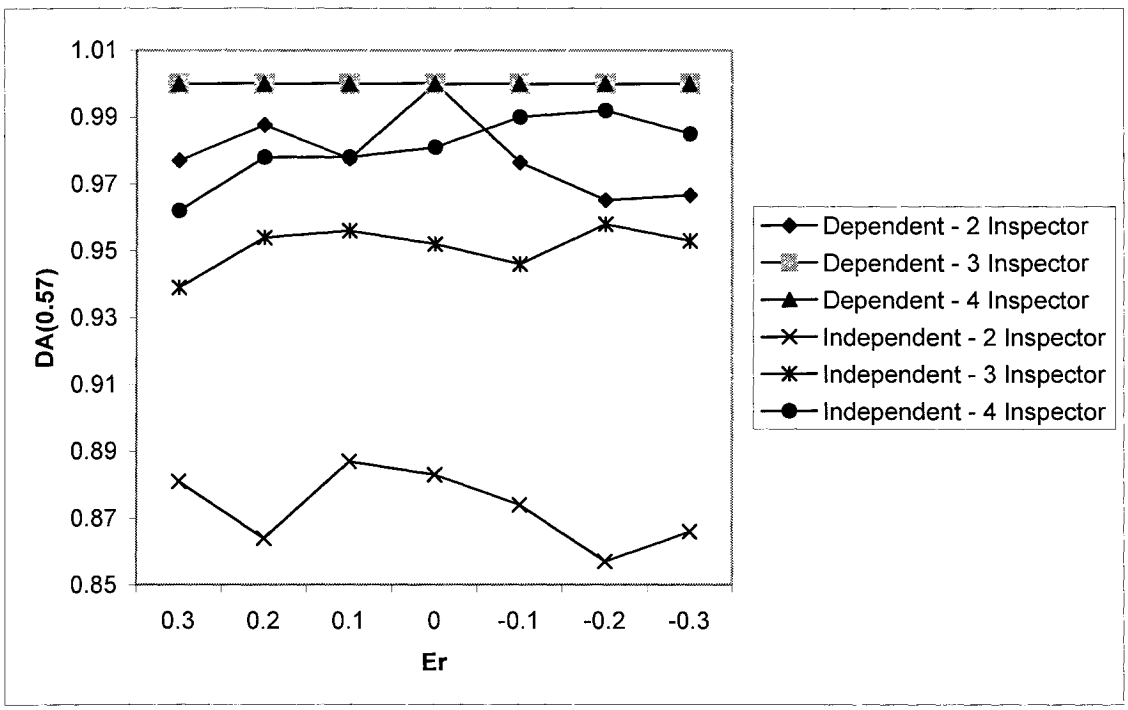

Fig. 7.6: Comparison of DA of 0.57 for the two Bayesian models for 20 defects and 0.1 degree of difficulty and ρ =0.2. Inspection team consists of moderate abilities, i.e. 0.5

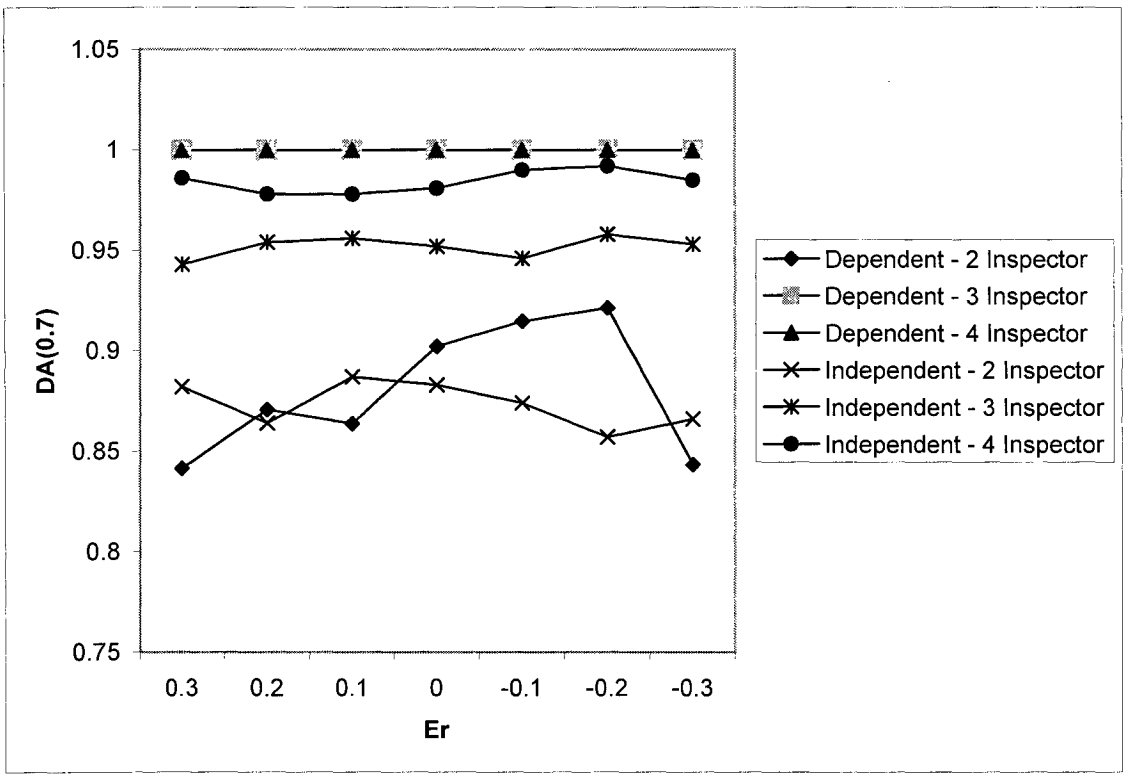

Fig. 7.7: Comparison of DA of 0.7 for the two Bayesian models for 20 defects and 0.1 degree of difficulty and ρ =0.4. Inspection team consists of moderate abilities, i.e. 0.5

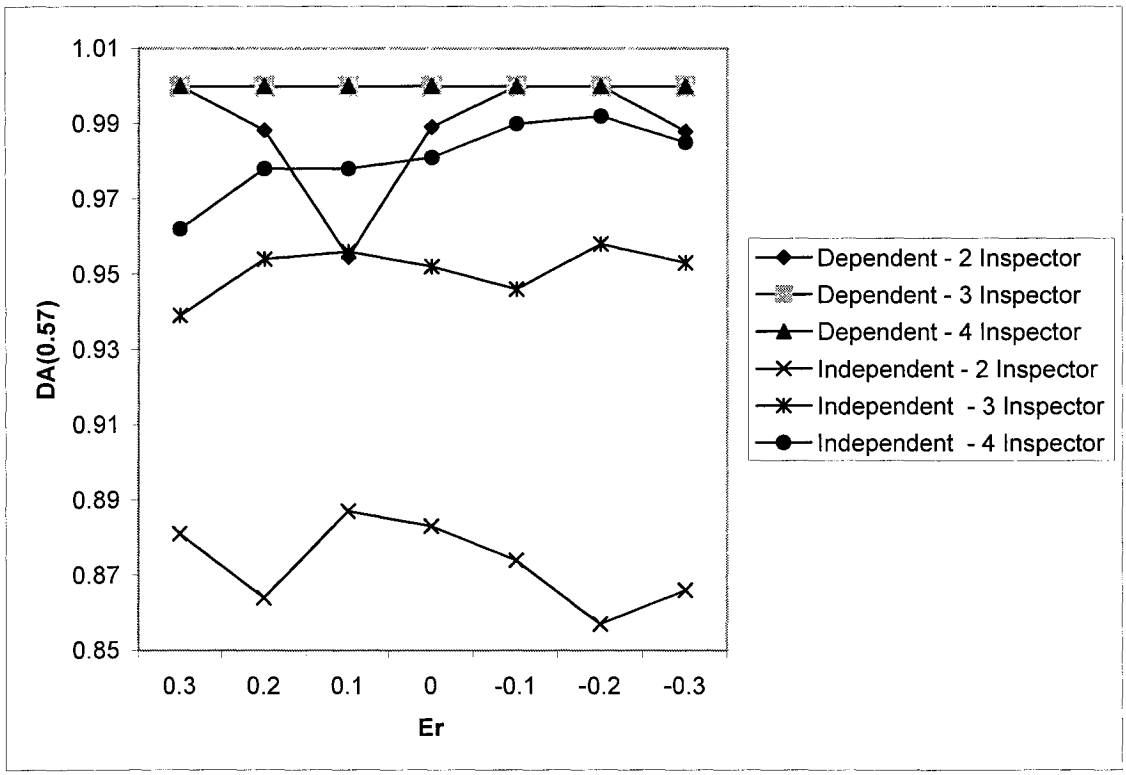

Fig. 7.8: Comparison of DA of 0.57 for the two Bayesian models for 20 defects and 0.1 degree of difficulty and ρ =0.4. Inspection team consists of moderate abilities, i.e. 0.5

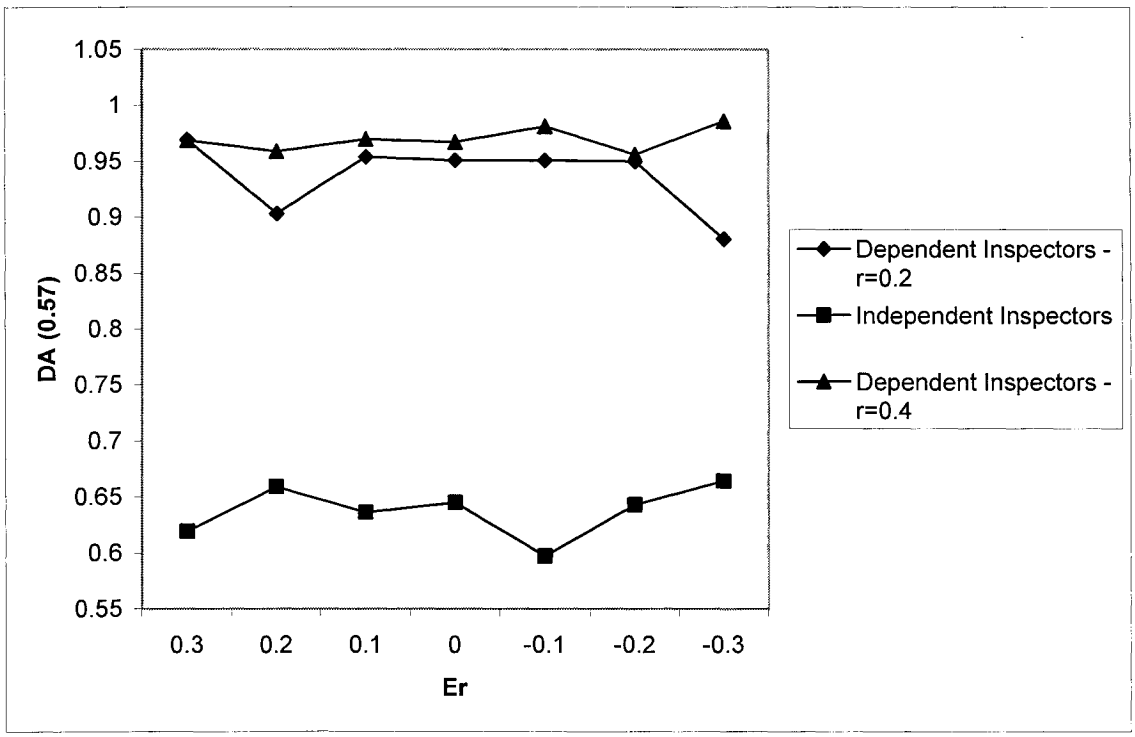

Fig. 7.9: Comparison of DA of 0.57 for the two Bayesian models for 10 defects and 0.1 degree of difficulty. Inspection team consists of moderate abilities, i.e. 0.5

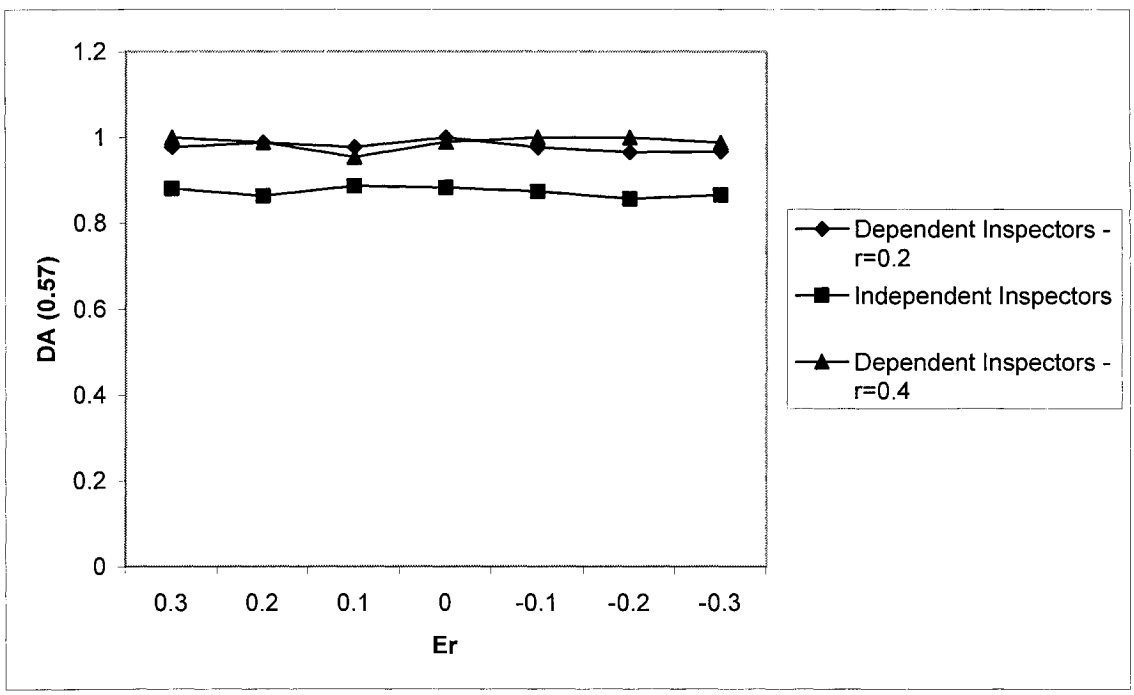

Fig. 7.10: Comparison of DA of 0.57 for the two Bayesian models for 20 defects and 0.1 degree of difficulty. Inspection team consists of moderate abilities, i.e. 0.5

7.2 Comparison Between the two Models

Figs. 7.1 to 7.8 show the comparison between the DA (0.57 and 0.7) for both the models. It can be seen from the figures that the performance of dependent Bayesian model is better than the independent Bayesian model. Fig. 7.2 shows that the performance of the dependent Bayesian model for 2 Inspectors is better than the independent Bayesian model for 4 inspectors. All the graphs show that the dependent inspector model outperforms the independent inspector on most of the study points. Figs. 7.9 and 7.10 show that increasing the dependence among inspectors further improves the decision accuracy. The reason for this improvement can be understood from the number of defects found by each inspector. As the dependence is increased, the inspectors will find more errors and hence the data matrix gets filled up and therefore the estimate about the number of errors are better. Another important fact that can be inferred from Figs. 7.1 to 7.8 is that the DA for dependent inspector model for 3 and 4 inspectors is similar, i.e. the same decision accuracy can be obtained by a team of 3 inspectors rather than having a team of 4 inspectors thereby saving the cost and time of an inspector.

7.3 Practical Application

A software inspection is a method by which defects are detected, eliminated and corrected by a team of inspectors during the development life cycle. The inspection

team consists of a moderator, designer, implementer, tester (see Section 2.1) and other inspectors. Entrance and exit criteria are used to determine if a product is ready to be inspected (entrance) and it successfully passed the inspection process (exit). Entrance criteria ensure that the product is mature enough to go through the inspection process. For example, entrance criteria for a code inspection are usually that the code has been compiled successfully and run through the automated standard checkers. Exit criteria are used mainly to assure that major defects found during the inspection process have been corrected.

The inspection process and its results are noted and documented by a set of forms. Some of the forms that could be used are (NASA 1993):

- Inspection Announcement, completed by the moderator that notifies participants of the inspection date, time, location, and other important information;
- Preparation Logs, completed by each inspector, that list defects found and time spent in preparation; and,
- An Inspection Defect List, completed by the recorder, to provide information on each defect.

In addition, the moderator should complete a detailed inspection report at the end of each inspection. Other useful forms include checklists that provide guidance for

inspectors in finding possible defects, and the Inspection Summary Report that summarizes data found during the inspection.

The software inspection stages have already been described in Section 2.1. In a typical inspection process the inspection team detects the errors and corrects them. Reinspection may be required when there are a large number of defects in the product (depends on the threshold, see Sections 2.2, and 5.2.2) or when one or more defects require extensive or complicated corrections. Reinspection allows the changes to the product to be reviewed by the entire team instead of just the moderator. The moderator and the team decide the necessity for reinspection at the end of the inspection meeting.

In an inspection process the inspector data is recorded in a form of a matrix called the data matrix (see Section 3.6). This data matrix serves as the input for all the estimation models. We also use this data matrix as an input for our model but have an additional input as the estimate by the most experienced inspector.

We use PERT-beta distribution to specify the estimate by the most experienced inspector. In this distribution the inspector specify three values, a (an optimistic estimate), m (a typical estimate) and b (a pessimistic estimate). These three values are used to get the a and b parameters of the prior beta distribution used in our model

(see Section 5.3). The conversion method is described in Appendix B. Once all the input data, i.e. data matrix and a and b parameters for prior, are available then Eqn. 5.13 and Eqn. 5.14 are used to get the defect estimate. Based on the threshold value and the defect estimate, a reinspection decision can be made using Eqn. 5.3.

Chapter 8

Conclusions and Future Work

The main focus of our research was to compare the performance of a Bayesian model with dependence among inspectors under a realistic scenario and compare it with the performance of the previous Bayesian model (Gupta 2003). We extended the work of Gupta (2003) and used the same set of parameters and variables to evaluate our model. Extensive Monte Carlo simulations were carried out for 2, 3 and 4 inspectors and defect population of 10, 20 and 30 with 0.1 and 0.4 degrees of difficulty and two degrees of dependence 0.2 and 0.4. We find that the decision accuracy improves with increase in the number of inspectors, and number of defects, when the defects become easier to find and also with an increase in the degree of dependence. We can understand the results and the behaviour of the Bayesian model in the context of the sparseness of the data matrix and the prior distribution. In the case of a sparse data matrix, the estimated population depends entirely on the prior mean and the standard deviation; hence in such a case it is very important to have an accurate value of the prior distribution. However, when the data matrix is not sparse, the estimation of the population of defects depends more on the likelihood function rather than on the prior distribution. Therefore, with a comparatively full data matrix, the likelihood

function gets better defined. As the prior moves away from the correct value, decision accuracy drops significantly. Since the reliability of the decision accuracy depends on the correctness of the prior distribution, particularly in the case of a sparse data matrix, it is rather essential to have a correct prior distribution. Also, with the introduction of dependence among inspectors, the number of defects detected increases, since an error detected by an inspector is more likely to be detected by other inspectors.

Comparing the independent inspector Bayesian model with the dependent inspector Bayesian model, it is observed that under most circumstances the decision accuracy values of the dependent inspector Bayesian model are higher than those for independent inspector Bayesian model. The decision accuracy, even for 4-inspector independent inspector Bayesian model in some cases is much less than 2, 3-inspector dependent inspector Bayesian model.

Future Work

The results for the dependent inspector Bayesian model are quite encouraging and show an improvement over the independent inspector Bayesian model (see Chapter 7). We used two levels of dependence among inspectors, i.e. 0.2 and 0.4. Further

research work can focus on the study of the effect of higher level of dependence among inspectors.

We assumed a close population of defects where the population remains constant throughout the inspection process (see Section 3.2.1). However, one can use different CR models to estimate the population (Seber 1982). Open population in software context could mean that correcting an error might lead to several more errors or it may automatically correct other errors. One can also consider supplementing the CR models studied in this work with some additional parameters such as size and complexity metrics of the software artefact.

Considering the success of the Bayesian model presented in this work, one can also design a tool for software inspections.

Moving away from purely Bayesian models, another avenue that is worthy of exploration is that of capture-recapture models for estimating endangered species. Such models are particularly good at dealing with a sparse data matrix, and therefore may provide additional value to making the reinspection decision.

REFERENCES

1. Ananda, M. M. A., "Bayesian methods for mark-resighting surveys", *Comm. in Statistics - Theory and Methods*, 26(3), 685 – 697, 1997.

2. Barnard, J., Emam, K. El. and Zubrow, D., "Using capture-recapture models for the reinspection decision," *Software Quality Professional*, vol 5, March 2003.

3. Basili, V. R., Green, S., Laitenberger, O., Laubile, F., Shull, F., Sorumgard, S. and Zelkowitz, M. V., "The empirical investigation of perspective-based reading", *Empirical Software Engineering*, 1(2), 133-164, 1996.

4. Basu, S. and Ebrahimi, N., "Estimating the number of undetected errors: Bayesian model selection", *Proc. of the International Symposium on Software Reliability Engineering*, pp 22-31, 1998.

5. Basu S. and Ebrahimi N., "Bayesian capture–recapture methods for error detection and estimation of population size: heterogeneity and dependence", *Biometrika*, 88(1): 269–279, 2001.

6. Basu S., "Bayesian inference for the number of undetected errors", 2003, Private Communication.

7. Biffl, S., "Estimating defect estimation models with major defects", *Journal of Systems and Software,* 65(2003), 13-29, 2001.

8. Biffl, S. and Grossman, W., "Evaluating the accuracy of defect estimation models based on inspection data from two inspection cycles", *Proc. of the 23rd International Conference on Software Engineering*, 145-154, 2001.

9. Briand, L., Emam, K. El. and Freimut, B., "A comparison and integration of capture recapture models and the detection profile method", *Proc. of the 9th International Symposium on Software Reliability Engineering*, 32-41, 1998(a).

10. Briand, L., Emam, K. El., Bomarius, F., "COBRA: A hybrid method for software cost estimation, benchmarking and risk assessment", *Proc. of the 20th International Conference on Software Engineering*, 390-399, 1998(b).

11. Briand, L., Emam, K. El. and Freimut, B., "A comprehensive evaluation of capture-recapture models for estimating software defect content", *IEEE Transactions on Software Engineering*, 26(6), 518-540, 2000.

12. Briand, L., Emam, K. El., Freimut, B. and Laitenberger, O., "Quantitative evaluation of capture recapture models to control software inspections", *Proc. of the 8th International Symposium on Software Reliability Engineering*, 234-244, 1997.

13. Briand, L., Freimut, B., Laitenberger, O., Ruhe, G., and Klein, B., "Quality assurance technologies for the euro conversion – industrial experience at Allianz life assurance", *in Proc. of Quality Week Europe*, 1998.

14. Burnham, K., "Estimation of population size in multiple capture recapture studies when capture probabilities vary among animals", *PhD theses*, Oregon State University, 1972.

15. Burnham, K., and Overton, W., "Estimation of the size of a closed population when capture probabilities vary among animals", *Biometrika*, 65, 625-633, 1978.

16. Casella, G., George, E.I., "Explaining the Gibbs sampler", *Am. Statistician*, 46, 167-174, 1992.

17. Castledine, B., "A Bayesian analysis of multiple recapture sampling for a closed population", *Biometrika*, 67, 197-210, 1981.

18. Chao, A., "Estimating the population size for capture-recapture data with unequal catchability", *Biometrics*, 43, 783-791, 1987.

19. Chao, A., "Estimating animal abundance with capture frequency data", *Journal of Wildlife Management*, 52, 295-300, 1988.

20. Chao, A., "Estimating population size for sparse data in capture-recapture experiments," *Biometrics*, 45, 427-438, 1989.

21. Chao, A., Lee, S. and Jeng, S., "Estimation of population size for capture-recapture data when capture probabilities vary by time and individual animal", *Biometrics*, 48, 201-216, 1992.

22. Darroch, J.N., "The multiple-recapture census: I. estimation of a closed population", *Biometrika*, 40, 343-359, 1958.

23. Doolan, E.P., "Experience with Fagan's inspection method," *Software-Practice and Experience*, 22(2), 173-182, 1992.

24. Ebenau, R.G., Strauss, S.H., "Software inspection process", *McGraw-Hill, New York*, 1994.

25. Eick, S.G, Loader, C.R., Long, M.D, Votta, L.G. and Vander Weil, S., "Estimating software fault content before coding", *Proc. of the 14th International Conference on Software Engineering*, 59-65, 1992.

26. Emam, K. El. and Laitenberger, O., "Evaluating capture-recapture models with two inspectors," *IEEE Transactions on Software Engineering*, 27, Sep 2001.

27. Emam, K. El., Laitenberger, O. and Harbich, T., "The application of subjective estimates of effectiveness to controlling software inspections", *Technical Report*, NRC/ERB-1060, 36 pages, NRC 43604, Oct 1999.

28. Fagan, M. E., "Advances in Software Inspections", *IEEE Transaction on Software Engineering*, 12(7), 744-751, 1986.

29. Fagan, M.E., "Design and code inspections to reduce errors in program development," *IBM Systems Journal*, 15(3), 182-211, 1976.

30. Freimut, B., "Capture-Recapture models to estimate software fault content", *Ph.D. Theses*, Universitaet Kaiserslautern, 1997.

31. Gelfand, A., Smith, A.F.M., "Sampling based approaches to calculating marginal densities" *J. Am. Statist. Assoc.* 85, 398-409, 1990.

32. George E.I. and Robert, C.P., "Capture-recapture estimation via Gibbs sampling", *Biometrika*, 79, 677-683, 1992.

33. Gilb, T. and Graham, D., "Software Inspection", *Addison-Wesley Publishing Company*, 1993.

34. Gupta, V., Patnaik, A. R., Emam, K. El. and Goel, N., "System for controlling software inspection", *CCECE 2003 -Canadian Conference on Electrical and Computer Engineering*, May 2003.

35. Gupta V.,"A study of estimation techniques of software defect content", *Masters Theses, School of Computer Science, Carleton University, Ottawa*, 2003.

36. Huber, J.J., "Trap response of confined cottontail populations", *J. Wildl. Manage.* 26(2): 177-185, 1962.

37. Miller, J., "Estimating the number of remaining defects after inspection", *International Software Engineering Research Network*, Technical Report ISERN-98-24, 1998.

38. Menkens, G. and Anderson, S., "Estimation of small mammal population size", *Ecology*, 69(6), 1952-1959, 1988.

39. NASA-GB-A302, "Software formal inspections guidebook", *National Aeronautics and Space Administration*, Washington, DC, 1993

40. Otis, D., Burnham, K., White, G. and Anderson, D., "Statistical inference from capture data on closed animal populations", *in Wildlife Monographs*, 62, 1-135, 1978.

41. O'Neill, D., "Issues in software inspection", *Software, IEEE Volume 14, Issue 1*, Jan.-Feb., 18 – 19, 1997.

42. Petersson, H., Thelin, T., Runeson, P., Wohlin, C., "Capture–recapture in software inspections after 10 years research—theory, evaluation and application", *The Journal of Systems and Software*, 72, 249–264, 2004.

43. Petersson, H. and Wohlin, C., "Evaluation of using capture-recapture methods in software review data", *Proc. of the 3rd International Conference on Empirical Assessment & Evaluation in Software Engineering*, 1999.

44. Reid, N., "The roles of conditioning on inference", *Statistical Science*, 10, 138-157, 1995.

45. Russell, G.W., "Experience with inspections in ultralarge-scale developments," *IEEE Software*, 8, 25-31, 1991.

46. Seber, G.A.F., "The Estimation of animal abundance and related parameters", *Charles Griffin & Company Ltd.*, London, 2nd edition, 1982.

47. Selby, R.W., "Evaluations of software technologies: testing, cleanroom, and metrics", *PhD-Theses*, Department of Computer Science, University of Maryland, 1985.

48. Swinebroad, J., "Net-shyness and wood thrush populations", *Bird-Banding* 33(3), 196-202, 1964.

49. Thelin, T. and Runeson, P., "Robust estimation of fault content with capture recapture and detection profile estimators", *in Proc. of the Conference of Empirical Assessment in Software Engineering*, 1999(a).

50. Thelin, T. and Runeson, P., "Capture-Recapture estimations for perspective-based reading- a simulated experiment ", *Proc. of the International Conference on Product Focused Software Process Improvement*, 182- 200, 1999(b).

51. Vander Weil, S. and Votta, L., "Assessing software designs using capture-recapture methods", *IEEE Transactions on Software Engineering,* 19(11), 1045-1054, 1993.

52. White, G.C., Anderson, D.R., Burnham, K.P. and Otis, D.L., "Capture-Recapture and removal methods for sampling closed population", *Technical report*, Los Alamos National Laboratory, 1982.

53. Wohlin, C. and Runeson, P., "Defect content estimation from review data", *Proc. of the 20th International Conference on Software Engineering*, 400-409, 1998.

54. Wohlin, C., Runeson, P. and Brantestam, J., "An experimental evaluation of capture recapture in software inspections", *In Software Testing, Verification and Reliability*, 5, 213-232, 1995.

55. Young, H., Neess J., and Emlen, J.T. Jr., "Heterogeneity of trap response in a population of house mice", *J. Wildl. Manage.* 16(2), 169-180, 1952.

56. http://www.cob.sjsu.edu/facstaff/davis_r/courses/QBAreader/PERTbetaSIM.html

Appendix A

This appendix gives partial simulation results[1] in the form of tables. The tables have been divided into four sections based on degree of difficulty of defects and degree of dependence.

Column heading used are:

SD is the standard deviation, Er is the offset from the actual mean (see Section 5.3.2), MRE is the median relative error, IQR is the inter quartile range for the error, Fail gives the number of times the model failed to estimate (out of 1000), DA(0.7) and DA(0.57) is the decision accuracy for the two thresholds (see Section 5.2.2), and RDA(0.7) and RDA(0.57) is the relative decision accuracy for the two thresholds.

1 Complete results take around 250 pages, here only the partial results are given.

TABLE FOR 0.1 DEGREE OF DIFFICULTY AND DEGREE OF DEPENDENCE 0.2

Table: D1-I2: 2 inspectors 10 defects 0.2 degree of dependence and 0.1 degree of difficulty

(0.5 SD	0.5) Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	-0.3	0.22	380	0.84	0	0.95	0
0.025	0.2	-0.2	0.24	260	0.81	0	0.91	0
0.025	0.1	-0.3	0.21	410	0.8	0	0.93	0
0.025	0	-0.3	0.25	350	0.78	0	0.89	0
0.025	-0.1	-0.2	0.23	220	0.81	0	0.91	0
0.025	-0.2	-0.2	0.24	260	0.81	0	0.93	0
0.025	-0.3	-0.2	0.21	300	0.86	0	0.96	0
0.05	0.3	-0.3	0.2	350	0.89	0	0.98	0
0.05	0.2	-0.2	0.18	350	0.86	0	0.98	0
0.05	0.1	-0.3	0.18	400	0.87	0	0.95	0
0.05	0	-0.2	0.21	320	0.85	0	0.9	0
0.05	-0.1	-0.4	0.26	380	0.73	0	0.82	0
0.05	-0.2	-0.3	0.2	390	0.87	0	0.93	0
0.05	-0.3	-0.3	0.2	420	0.84	0	0.97	0
0.075	0.3	-0.2	0.19	350	0.86	0	0.94	0
0.075	0.2	-0.2	0.22	300	0.8	0	0.96	0
0.075	0.1	-0.3	0.23	310	0.81	0	0.94	0
0.075	0	-0.3	0.23	330	0.91	0	0.97	0
0.075	-0.1	-0.3	0.22	340	0.83	0	0.97	0
0.075	-0.2	-0.3	0.2	340	0.83	0	0.95	0
0.075	-0.3	-0.3	0.22	350	0.74	0	0.92	0
0.1	0.3	-0.3	0.23	350	0.83	0	0.91	0
0.1	0.2	-0.2	0.19	320	0.87	0	0.93	0
0.1	0.1	-0.3	0.2	350	0.83	0	0.92	0
0.1	0	-0.3	0.22	370	0.84	0	0.92	0
0.1	-0.1	-0.3	0.18	380	0.77	0	0.95	0
0.1	-0.2	-0.3	0.19	330	0.82	0	0.94	0
0.1	-0.3	-0.3	0.19	380	0.89	0	0.95	0
0.2	0.3	-0.3	0.19	400	0.88	0	0.92	0
0.2	0.2	-0.2	0.22	290	0.8	0	0.92	0
0.2	0.1	-0.3	0.2	440	0.89	0	0.98	0
0.2	0	-0.3	0.17	380	0.92	0	1	0
0.2	-0.1	-0.4	0.18	420	0.79	0	0.9	0
0.2	-0.2	-0.3	0.2	380	0.79	0	0.94	0
0.2	-0.3	-0.3	0.23	340	0.8	0	0.94	0

Table: D1-I3: 3 inspectors 10 defects 0.2 degree of dependence and 0.1 degree of difficulty

(0.5	0.5	0.5)						
SD	Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	-0.1	0.11	190	1	0	1	0
0.025	0.2	0	0.1	200	1	0	1	0
0.025	0.1	-0.1	0.09	250	1	0	1	0
0.025	0	-0.1	0.13	240	0.99	0	1	0
0.025	-0.1	-0.1	0.12	230	0.99	0	1	0
0.025	-0.2	-0.1	0.08	310	1	0	1	0
0.025	-0.3	-0.1	0.1	240	1	0	1	0
0.05	0.3	-0.1	0.09	210	0.99	0	1	0
0.05	0.2	-0.1	0.1	270	0.96	0	1	0
0.05	0.1	-0.1	0.1	190	0.99	0	1	0
0.05	0	-0.1	0.1	190	1	0	1	0
0.05	-0.1	-0.1	0.09	170	0.99	0	1	0
0.05	-0.2	-0.1	0.12	210	1	0	1	0
0.05	-0.3	-0.1	0.11	210	1	0	1	0
0.075	0.3	-0.1	0.1	180	0.98	0	1	0
0.075	0.2	0	0.1	190	1	0	1	0
0.075	0.1	-0.1	0.1	230	0.99	0	1	0
0.075	0	-0.1	0.1	190	0.99	0	1	0
0.075	-0.1	-0.1	0.11	190	1	0	1	0
0.075	-0.2	-0.1	0.08	220	1	0	1	0
0.075	-0.3	0	0.08	180	1	0	1	0
0.1	0.3	-0.1	0.07	270	1	0	1	0
0.1	0.2	-0.1	0.09	170	0.98	0	1	0
0.1	0.1	-0.1	0.1	180	1	0	1	0
0.1	0	-0.1	0.09	210	0.99	0	1	0
0.1	-0.1	-0.1	0.1	280	0.99	0	1	0
0.1	-0.2	-0.1	0.08	220	1	0	1	0
0.1	-0.3	-0.1	0.09	240	0.99	0	1	0
0.2	0.3	-0.1	0.11	160	0.96	0	1	0
0.2	0.2	-0.1	0.1	210	0.99	0	1	0
0.2	0.1	-0.1	0.1	230	1	0	1	0
0.2	0	-0.1	0.11	250	1	0	1	0
0.2	-0.1	-0.1	0.1	170	1	0	1	0
0.2	-0.2	-0.1	0.08	180	0.99	0	1	0
0.2	-0.3	-0.1	0.08	180	1	0	1	0

Table: D1-I4: 4 inspectors 10 defects 0.2 degree of dependence and 0.1 degree of difficulty

(0.5	0.5	0.5	0.5)					
SD	Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	0	0.07	160	1	0	1	0
0.025	0.2	0	0.07	170	1	0	1	0
0.025	0.1	0	0.08	110	1	0	1	0
0.025	0	0	0.08	110	1	0	1	0
0.025	-0.1	0	0.07	120	1	0	1	0
0.025	-0.2	0	0.07	140	1	0	1	0
0.025	-0.3	0	0.09	140	1	0	1	0
0.05	0.3	0	0.07	150	0.99	0	1	0
0.05	0.2	0	0.06	180	1	0	1	0
0.05	0.1	0	0.09	140	1	0	1	0
0.05	0	0	0.07	110	1	0	1	0
0.05	-0.1	0	0.07	110	1	0	1	0
0.05	-0.2	0	0.05	180	1	0	1	0
0.05	-0.3	0	0.07	210	1	0	1	0
0.075	0.3	0	0.08	130	1	0	1	0
0.075	0.2	0	0.1	70	1	0	1	0
0.075	0.1	0	0.08	120	1	0	1	0
0.075	0	0	0.07	150	1	0	1	0
0.075	-0.1	0	0.06	150	1	0	1	0
0.075	-0.2	0	0.09	170	1	0	1	0
0.075	-0.3	0	0.06	130	1	0	1	0
0.1	0.3	0	0.09	140	1	0	1	0
0.1	0.2	0	0.08	120	1	0	1	0
0.1	0.1	0	0.08	140	1	0	1	0
0.1	0	0	0.08	130	1	0	1	0
0.1	-0.1	0	0.07	160	1	0	1	0
0.1	-0.2	0	0.09	100	1	0	1	0
0.1	-0.3	0	0.08	120	1	0	1	0
0.2	0.3	0	0.08	140	1	0	1	0
0.2	0.2	0	0.06	130	1	0	1	0
0.2	0.1	0	0.06	140	1	0	1	0
0.2	0	0	0.07	130	1	0	1	0
0.2	-0.1	0	0.07	180	1	0	1	0
0.2	-0.2	0	0.05	190	1	0	1	0
0.2	-0.3	0	0.09	110	1	0	1	0

TABLE FOR 0.1 DEGREE OF DIFFICULTY AND DEGREE OF DEPENDENCE 0.4

Table: D1-I2: 2 inspectors 10 defects 0.4 degree of dependence and 0.1 degree of difficulty

(0.5	0.5)							
SD	Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	-0.3	0.15	400	0.78	0	0.93	0
0.025	0.2	-0.2	0.19	310	0.91	0	0.99	0
0.025	0.1	-0.3	0.25	310	0.78	0	0.94	0
0.025	0	-0.3	0.18	430	0.91	0	0.98	0
0.025	-0.1	-0.3	0.22	410	0.83	0	0.95	0
0.025	-0.2	-0.3	0.2	380	0.85	0	0.97	0
0.025	-0.3	-0.2	0.2	290	0.89	0	1	0
0.05	0.3	-0.3	0.23	330	0.84	0	0.99	0
0.05	0.2	-0.3	0.18	430	0.86	0	0.93	0
0.05	0.1	-0.3	0.17	340	0.82	0	0.94	0
0.05	0	-0.3	0.22	320	0.85	0	0.93	0
0.05	-0.1	-0.3	0.23	350	0.78	0	0.94	0
0.05	-0.2	-0.3	0.19	350	0.88	0	0.95	0
0.05	-0.3	-0.3	0.17	340	0.85	0	0.94	0
0.075	0.3	-0.3	0.2	370	0.87	0	0.98	0
0.075	0.2	-0.2	0.19	330	0.87	0	0.94	0
0.075	0.1	-0.3	0.2	360	0.89	0	0.95	0
0.075	0	-0.3	0.18	380	0.82	0	0.94	0
0.075	-0.1	-0.3	0.21	400	0.83	0	0.97	0
0.075	-0.2	-0.3	0.2	370	0.83	0	0.95	0
0.075	-0.3	-0.3	0.22	350	0.78	0	0.92	0
0.1	0.3	-0.3	0.17	390	0.87	0	0.95	0
0.1	0.2	-0.3	0.19	410	0.8	0	0.9	0
0.1	0.1	-0.2	0.22	270	0.78	0	0.9	0
0.1	0	-0.3	0.2	350	0.88	0	1	0
0.1	-0.1	-0.3	0.22	310	0.86	0	0.96	0
0.1	-0.2	-0.3	0.18	390	0.89	0	1	0
0.1	-0.3	-0.3	0.16	450	0.89	0	0.93	0
0.2	0.3	-0.4	0.16	460	0.8	0	0.93	0
0.2	0.2	-0.3	0.13	420	0.86	0	0.98	0
0.2	0.1	-0.2	0.2	310	0.81	0	0.94	0
0.2	0	-0.3	0.17	370	0.84	0	0.97	0
0.2	-0.1	-0.2	0.18	330	0.91	0	0.96	0
0.2	-0.2	-0.2	0.22	300	0.87	0	0.97	0
0.2	-0.3	-0.3	0.18	390	0.87	0	0.93	0

Table: D1-I3: 3 inspectors 10 defects 0.4 degree of dependence and 0.1 degree of difficulty

(0.5	0.5	0.5)						
SD	Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	0	0.14	160	0.99	0	1	0
0.025	0.2	-0.1	0.1	200	1	0	1	0
0.025	0.1	-0.1	0.09	220	0.99	0	1	0
0.025	0	0	0.06	250	1	0	1	0
0.025	-0.1	-0.1	0.11	170	1	0	1	0
0.025	-0.2	-0.1	0.1	220	0.99	0	1	0
0.025	-0.3	-0.1	0.11	240	0.97	0	0.99	0
0.05	0.3	-0.1	0.1	190	1	0	1	0
0.05	0.2	0	0.07	210	1	0	1	0
0.05	0.1	0	0.09	150	1	0	1	0
0.05	0	-0.1	0.08	210	0.99	0	1	0
0.05	-0.1	-0.1	0.07	210	1	0	1	0
0.05	-0.2	-0.1	0.08	260	0.99	0	1	0
0.05	-0.3	-0.1	0.1	190	1	0	1	0
0.075	0.3	-0.1	0.1	260	1	0	1	0
0.075	0.2	-0.1	0.1	210	1	0	1	0
0.075	0.1	-0.1	0.09	250	1	0	1	0
0.075	0	-0.1	0.11	220	1	0	1	0
0.075	-0.1	0	0.09	150	0.99	0	1	0
0.075	-0.2	0	0.09	150	0.98	0	0.99	0
0.075	-0.3	0	0.1	220	1	0	1	0
0.1	0.3	-0.1	0.1	310	1	0	1	0
0.1	0.2	-0.1	0.11	230	0.97	0	1	0
0.1	0.1	0	0.1	180	1	0	1	0
0.1	0	0	0.09	210	1	0	1	0
0.1	-0.1	0	0.08	210	1	0	1	0
0.1	-0.2	-0.1	0.09	210	1	0	1	0
0.1	-0.3	-0.1	0.1	170	0.99	0	1	0
0.2	0.3	-0.1	0.09	220	0.96	0	0.99	0
0.2	0.2	-0.1	0.07	260	0.97	0	1	0
0.2	0.1	0	0.11	160	1	0	1	0
0.2	0	-0.1	0.1	230	1	0	1	0
0.2	-0.1	-0.1	0.09	250	0.97	0	1	0
0.2	-0.2	0	0.07	260	1	0	1	0
0.2	-0.3	-0.1	0.11	290	1	0	1	0

Table: D1-I4: 4 inspectors 10 defects 0.4 degree of dependence and 0.1 degree of difficulty

(0.5	0.5	0.5	0.5)					
SD	Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	0	0.07	110	1	0	1	0
0.025	0.2	0	0.06	170	1	0	1	0
0.025	0.1	0	0.08	120	1	0	1	0
0.025	0	0	0.06	130	1	0	1	0
0.025	-0.1	0	0.07	130	1	0	1	0
0.025	-0.2	0	0.06	160	1	0	1	0
0.025	-0.3	0	0.07	140	1	0	1	0
0.05	0.3	0	0.06	180	1	0	1	0
0.05	0.2	0	0.06	110	1	0	1	0
0.05	0.1	0	0.08	110	1	0	1	0
0.05	0	0	0.08	150	1	0	1	0
0.05	-0.1	0	0.07	110	1	0	1	0
0.05	-0.2	0	0.05	120	1	0	1	0
0.05	-0.3	0	0.06	120	1	0	1	0
0.075	0.3	0	0.07	150	1	0	1	0
0.075	0.2	0	0.08	110	1	0	1	0
0.075	0.1	0	0.07	150	1	0	1	0
0.075	0	0	0.08	110	1	0	1	0
0.075	-0.1	0	0.06	120	1	0	1	0
0.075	-0.2	0	0.07	100	1	0	1	0
0.075	-0.3	0	0.07	80	1	0	1	0
0.1	0.3	0	0.05	130	1	0	1	0
0.1	0.2	0	0.06	120	1	0	1	0
0.1	0.1	0	0.08	110	1	0	1	0
0.1	0	0	0.06	170	1	0	1	0
0.1	-0.1	0	0.06	90	1	0	1	0
0.1	-0.2	0	0.05	130	1	0	1	0
0.1	-0.3	0	0.06	110	1	0	1	0
0.2	0.3	0	0.07	110	1	0	1	0
0.2	0.2	0	0.06	90	1	0	1	0
0.2	0.1	0	0.06	120	1	0	1	0
0.2	0	0	0.07	80	1	0	1	0
0.2	-0.1	0	0.06	130	1	0	1	0
0.2	-0.2	0	0.06	160	1	0	1	0
0.2	-0.3	0	0.06	160	1	0	1	0

TABLE FOR 0.4 DEGREE OF DIFFICULTY AND DEGREE OF DEPENDENCE 0.2

Table: D1-I2: 2 inspectors 10 defects 0.2 degree of dependence and 0.4 degree of difficulty

(0.5	0.5)							
SD	Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	-0.2	0.21	0	0.82	0	0.95	0
0.025	0.2	-0.2	0.21	10	0.81	0	0.95	0
0.025	0.1	-0.1	0.18	20	0.84	0	0.97	0
0.025	0	-0.2	0.17	0	0.84	0	0.92	0
0.025	-0.1	-0.2	0.2	0	0.8	0	0.92	0
0.025	-0.2	-0.2	0.24	10	0.81	0	0.95	0
0.025	-0.3	-0.2	0.23	0	0.85	0	0.97	0
0.05	0.3	-0.1	0.21	0	0.9	0	0.96	0
0.05	0.2	-0.1	0.24	20	0.81	0	0.95	0
0.05	0.1	-0.2	0.2	20	0.87	0	0.95	0
0.05	0	-0.2	0.14	10	0.87	0	0.96	0
0.05	-0.1	-0.2	0.19	10	0.89	0	0.97	0
0.05	-0.2	-0.2	0.22	10	0.77	0	0.92	0
0.05	-0.3	-0.2	0.19	0	0.87	0	0.99	0
0.075	0.3	-0.2	0.2	20	0.84	0	0.93	0
0.075	0.2	-0.2	0.2	30	0.82	0	0.94	0
0.075	0.1	-0.1	0.17	10	0.9	0	0.97	0
0.075	0	-0.2	0.22	10	0.81	0	0.95	0
0.075	-0.1	-0.2	0.17	20	0.86	0	0.96	0
0.075	-0.2	-0.2	0.14	30	0.81	0	0.92	0
0.075	-0.3	-0.2	0.19	30	0.9	0	0.98	0
0.1	0.3	-0.1	0.2	0	0.89	0	0.96	0
0.1	0.2	-0.2	0.22	10	0.73	0	0.91	0
0.1	0.1	-0.2	0.18	20	0.88	0	0.95	0
0.1	0	-0.2	0.19	20	0.83	0	0.96	0
0.1	-0.1	-0.1	0.2	0	0.88	0	0.98	0
0.1	-0.2	-0.2	0.19	10	0.81	0	0.91	0
0.1	-0.3	-0.2	0.16	0	0.87	0	0.99	0
0.2	0.3	-0.2	0.17	0	0.84	0	0.97	0
0.2	0.2	-0.2	0.22	10	0.81	0	0.92	0
0.2	0.1	-0.2	0.21	10	0.87	0	0.97	0
0.2	0	-0.2	0.2	0	0.87	0	0.95	0
0.2	-0.1	-0.2	0.18	10	0.86	0	0.98	0
0.2	-0.2	-0.2	0.19	10	0.84	0	0.97	0
0.2	-0.3	-0.2	0.17	10	0.8	0	0.94	0

Table: D1-I3: 3 inspectors 10 defects 0.2 degree of dependence and 0.4 degree of difficulty

(0.5	0.5	0.5)						
SD	Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	0	0.13	0	1	0	1	0
0.025	0.2	0	0.15	0	0.99	0	1	0
0.025	0.1	0	0.14	0	0.97	0	1	0
0.025	0	0	0.15	0	1	0	1	0
0.025	-0.1	0	0.12	0	0.97	0	0.99	0
0.025	-0.2	-0.1	0.14	0	1	0	1	0
0.025	-0.3	0	0.12	0	0.99	0	1	0
0.05	0.3	0	0.13	10	0.99	0	1	0
0.05	0.2	0	0.13	0	0.98	0	1	0
0.05	0.1	0	0.14	20	0.97	0	1	0
0.05	0	0	0.12	0	0.99	0	1	0
0.05	-0.1	0	0.14	0	0.98	0	1	0
0.05	-0.2	0	0.14	0	0.99	0	1	0
0.05	-0.3	0	0.14	0	1	0	1	0
0.075	0.3	0	0.13	0	1	0	1	0
0.075	0.2	0	0.13	0	0.99	0	1	0
0.075	0.1	-0.1	0.15	0	1	0	1	0
0.075	0	0	0.13	0	0.98	0	0.99	0
0.075	-0.1	0	0.11	0	0.98	0	1	0
0.075	-0.2	0	0.09	0	0.99	0	1	0
0.075	-0.3	0	0.13	0	1	0	1	0
0.1	0.3	0	0.15	0	0.99	0	1	0
0.1	0.2	0	0.13	0	1	0	1	0
0.1	0.1	0	0.12	0	1	0	1	0
0.1	0	0	0.14	0	0.98	0	1	0
0.1	-0.1	0	0.16	0	1	0	1	0
0.1	-0.2	0	0.12	0	0.99	0	1	0
0.1	-0.3	0	0.12	0	0.98	0	1	0
0.2	0.3	0	0.15	0	0.99	0	1	0
0.2	0.2	0	0.14	0	1	0	1	0
0.2	0.1	0	0.15	0	0.99	0	1	0
0.2	0	0	0.14	0	0.98	0	1	0
0.2	-0.1	0	0.13	0	0.99	0	1	0
0.2	-0.2	0	0.16	0	0.98	0	0.99	0
0.2	-0.3	0	0.14	0	0.99	0	1	0

Table: D1-I4: 4 inspectors 10 defects 0.2 degree of dependence and 0.4 degree of difficulty

(0.5	0.5	0.5	0.5)					
SD	Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	0	0.09	10	1	0	1	0
0.025	0.2	0	0.09	0	1	0	1	0
0.025	0.1	0	0.11	0	1	0	1	0
0.025	0	0	0.09	0	1	0	1	0
0.025	-0.1	0	0.09	0	1	0	1	0
0.025	-0.2	0	0.1	0	1	0	1	0
0.025	-0.3	0	0.1	0	1	0	1	0
0.05	0.3	0	0.1	0	1	0	1	0
0.05	0.2	0	0.09	0	1	0	1	0
0.05	0.1	0	0.1	0	1	0	1	0
0.05	0	0	0.08	0	1	0	1	0
0.05	-0.1	0	0.1	0	1	0	1	0
0.05	-0.2	0	0.08	0	1	0	1	0
0.05	-0.3	0	0.09	0	1	0	1	0
0.075	0.3	0	0.09	0	1	0	1	0
0.075	0.2	0	0.06	0	1	0	1	0
0.075	0.1	0	0.08	0	1	0	1	0
0.075	0	0	0.09	0	1	0	1	0
0.075	-0.1	0	0.08	0	1	0	1	0
0.075	-0.2	0	0.09	0	1	0	1	0
0.075	-0.3	0	0.09	0	1	0	1	0
0.1	0.3	0	0.09	10	1	0	1	0
0.1	0.2	0	0.08	0	1	0	1	0
0.1	0.1	0	0.11	0	1	0	1	0
0.1	0	0	0.1	0	1	0	1	0
0.1	-0.1	0	0.11	0	1	0	1	0
0.1	-0.2	0	0.09	0	1	0	1	0
0.1	-0.3	0	0.11	0	0.99	0	1	0
0.2	0.3	0	0.11	0	1	0	1	0
0.2	0.2	0	0.1	0	1	0	1	0
0.2	0.1	0	0.12	0	1	0	1	0
0.2	0	0	0.08	0	1	0	1	0
0.2	-0.1	0	0.07	0	1	0	1	0
0.2	-0.2	0	0.08	0	1	0	1	0
0.2	-0.3	0	0.08	0	1	0	1	0

TABLE FOR 0.4 DEGREE OF DIFFICULTY AND DEGREE OF DEPENDENCE 0.4

Table: D1-I2: 2 inspectors 10 defects 0.4 degree of dependence and 0.4 degree of difficulty

(0.5	0.5)							
SD	Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	-0.1	0.21	0	0.89	0	0.97	0
0.025	0.2	-0.2	0.19	20	0.84	0	0.94	0
0.025	0.1	-0.1	0.16	10	0.92	0	0.97	0
0.025	0	-0.2	0.19	10	0.88	0	0.97	0
0.025	-0.1	-0.2	0.19	10	0.88	0	0.96	0
0.025	-0.2	-0.2	0.23	40	0.78	0	0.95	0
0.025	-0.3	-0.2	0.21	10	0.85	0	0.94	0
0.05	0.3	-0.2	0.18	10	0.88	0	0.97	0
0.05	0.2	-0.1	0.19	10	0.9	0	0.95	0
0.05	0.1	-0.2	0.18	0	0.8	0	0.96	0
0.05	0	-0.2	0.16	20	0.89	0	0.96	0
0.05	-0.1	-0.2	0.24	20	0.8	0	0.95	0
0.05	-0.2	-0.2	0.19	10	0.82	0	0.95	0
0.05	-0.3	-0.2	0.18	20	0.85	0	0.97	0
0.075	0.3	-0.2	0.14	10	0.91	0	0.97	0
0.075	0.2	-0.2	0.16	30	0.9	0	0.99	0
0.075	0.1	-0.1	0.19	0	0.87	0	0.96	0
0.075	0	-0.1	0.21	10	0.86	0	0.98	0
0.075	-0.1	-0.1	0.18	0	0.89	0	0.95	0
0.075	-0.2	-0.1	0.18	20	0.87	0	0.98	0
0.075	-0.3	-0.2	0.21	0	0.86	0	0.96	0
0.1	0.3	-0.1	0.17	0	0.89	0	0.96	0
0.1	0.2	-0.2	0.19	0	0.84	0	0.97	0
0.1	0.1	-0.1	0.16	10	0.85	0	0.94	0
0.1	0	-0.1	0.19	0	0.88	0	0.96	0
0.1	-0.1	-0.1	0.22	20	0.85	0	0.98	0
0.1	-0.2	-0.1	0.21	10	0.9	0	0.95	0
0.1	-0.3	-0.2	0.2	30	0.84	0	0.97	0
0.2	0.3	-0.2	0.16	10	0.87	0	0.97	0
0.2	0.2	-0.2	0.18	10	0.88	0	0.96	0
0.2	0.1	-0.2	0.23	10	0.81	0	0.97	0
0.2	0	-0.1	0.19	10	0.88	0	0.94	0
0.2	-0.1	-0.2	0.18	20	0.83	0	0.91	0
0.2	-0.2	-0.1	0.19	20	0.9	0	0.96	0
0.2	-0.3	-0.1	0.25	20	0.82	0	0.94	0

Table: D1-I3: 3 inspectors 10 defects 0.4 degree of dependence and 0.4 degree of difficulty

(0.5	0.5	0.5)						
SD	Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	0	0.16	0	0.98	0	0.99	0
0.025	0.2	0	0.12	0	0.98	0	1	0
0.025	0.1	0	0.12	0	1	0	1	0
0.025	0	0	0.11	0	1	0	1	0
0.025	-0.1	0	0.13	0	1	0	1	0
0.025	-0.2	0	0.13	0	1	0	1	0
0.025	-0.3	0	0.14	0	0.99	0	1	0
0.05	0.3	0	0.14	0	1	0	1	0
0.05	0.2	0	0.12	0	0.99	0	1	0
0.05	0.1	0	0.11	0	0.99	0	1	0
0.05	0	0	0.1	0	1	0	1	0
0.05	-0.1	0	0.12	10	1	0	1	0
0.05	-0.2	0	0.12	0	0.99	0	0.99	0
0.05	-0.3	0	0.12	0	1	0	1	0
0.075	0.3	0	0.17	0	1	0	1	0
0.075	0.2	0	0.14	0	1	0	1	0
0.075	0.1	0	0.11	0	1	0	1	0
0.075	0	0	0.13	0	0.99	0	1	0
0.075	-0.1	0	0.12	0	1	0	1	0
0.075	-0.2	0	0.13	0	1	0	1	0
0.075	-0.3	0	0.14	10	0.98	0	1	0
0.1	0.3	0	0.12	0	0.99	0	1	0
0.1	0.2	0	0.1	0	0.99	0	1	0
0.1	0.1	0	0.12	0	0.99	0	1	0
0.1	0	0	0.12	0	1	0	1	0
0.1	-0.1	0	0.12	0	1	0	1	0
0.1	-0.2	0	0.12	0	0.99	0	1	0
0.1	-0.3	0	0.13	0	0.98	0	1	0
0.2	0.3	0	0.12	0	1	0	1	0
0.2	0.2	0	0.16	0	0.96	0	0.99	0
0.2	0.1	0	0.11	0	0.99	0	1	0
0.2	0	0	0.12	0	0.99	0	1	0
0.2	-0.1	0	0.14	0	0.99	0	1	0
0.2	-0.2	0	0.13	20	1	0	1	0
0.2	-0.3	0	0.1	0	1	0	1	0

Table: D1-I4: 4 inspectors 10 defects 0.4 degree of dependence and 0.4 degree of difficulty

(0.5	0.5	0.5	0.5)					
SD	Er	MRE	IQR	Fail	DA(0.7)	RDA(0.7)	DA(0.57)	RDA(0.57)
0.025	0.3	0	0.07	0	1	0	1	0
0.025	0.2	0	0.06	0	1	0	1	0
0.025	0.1	0	0.11	0	1	0	1	0
0.025	0	0	0.09	0	1	0	1	0
0.025	-0.1	0	0.07	0	1	0	1	0
0.025	-0.2	0	0.09	0	1	0	1	0
0.025	-0.3	0	0.08	0	1	0	1	0
0.05	0.3	0	0.11	0	1	0	1	0
0.05	0.2	0	0.08	0	1	0	1	0
0.05	0.1	0	0.06	0	1	0	1	0
0.05	0	0	0.11	0	1	0	1	0
0.05	-0.1	0	0.08	0	1	0	1	0
0.05	-0.2	0	0.08	0	0.99	0	1	0
0.05	-0.3	0	0.06	0	1	0	1	0
0.075	0.3	0	0.07	0	1	0	1	0
0.075	0.2	0	0.07	0	1	0	1	0
0.075	0.1	0	0.08	0	1	0	1	0
0.075	0	0	0.07	0	1	0	1	0
0.075	-0.1	0	0.08	0	1	0	1	0
0.075	-0.2	0	0.09	0	1	0	1	0
0.075	-0.3	0	0.07	0	1	0	1	0
0.1	0.3	0	0.08	0	0.99	0	1	0
0.1	0.2	0	0.06	0	1	0	1	0
0.1	0.1	0	0.09	0	1	0	1	0
0.1	0	0	0.06	0	1	0	1	0
0.1	-0.1	0	0.07	0	1	0	1	0
0.1	-0.2	0	0.06	0	1	0	1	0
0.1	-0.3	0	0.09	0	1	0	1	0
0.2	0.3	0	0.06	0	1	0	1	0
0.2	0.2	0	0.06	0	1	0	1	0
0.2	0.1	0	0.09	0	1	0	1	0
0.2	0	0	0.05	0	1	0	1	0
0.2	-0.1	0	0.07	0	1	0	1	0
0.2	-0.2	0	0.08	0	0.99	0	1	0
0.2	-0.3	0	0.07	0	1	0	1	0

Appendix B

PERT – Beta Distribution

This appendix discusses the PERT Beta distribution and how an inspector's experience can be converted to a beta distribution.

Program Evaluation and Review Technique [PERT] was developed by the consulting firm of Booz, Allen & Hamilton in conjunction with the United States Navy in 1958 as a tool for coordinating the activities of over 11,00 contractors involved with the Polaris missile program .

The inspector's experience can be described using the following three values,

- a = an optimistic value (approximate number of defects)
- m = a typical value (most likely)
- b = a pessimistic estimate (minimum number of defects)

The statistical model upon which PERT is based is known as the *Beta* distribution. The *Beta* distribution looks and behaves much like the normal distribution when m is exactly centered between a and b.

PERT Approximation Formulas

The two statistics needed for statistical application are the mean time and the variance or standard deviation. The mean value formula is a weighted average of the three given values where the weight on the minimum and maximum value is one and the weight on the typical value is 4, thus mean is given by,

$$d = \frac{a + 4m + b}{6} \tag{B.1}$$

The variance formula is motivated by the fact that for the symmetric case, almost all of the probability distribution will be within 3 standard deviations of the mean, so that one-sixth of the range of the interval is a reasonable approximation for the standard deviation for the activity duration. Thus the variance is given by the equivalent formulas shown below:

$$\sigma^2 = \left(\frac{b - a}{6} \right)^2 \tag{B.2}$$

Beta Distribution Parameters for PERT

The general beta distribution has 4 parameters: minimum and maximum range limits (A to B), and two shape parameters referred to as α ("alpha") and β ("beta").

The mean and variance for the beta distribution are given by the following formulas:

$$\mu = E[X] = A + \left(\frac{\alpha}{\alpha+\beta}\right)(B-A) \tag{B.3}$$

and,

$$\sigma^2 = \left(\frac{\alpha}{\alpha+\beta}\right)\left(\frac{\beta}{\alpha+\beta}\right)\frac{(B-A)^2}{(\alpha+\beta+1)} \tag{B.4}$$

Note that the mean value is a weighted average of A and B such that when $0 < \alpha < \beta$ the mean is closer to A and the distribution is skewed to the right; whereas for $\alpha > \beta > 0$ the mean is closer to B and the distribution is skewed to the left. When $\alpha = \beta$ the mean is exactly half way between A and B and the distribution is symmetric around the mean.

Also note that for a given α / β ratio, the mean is constant and the variance of the distribution varies inversely with the absolute magnitude of $\alpha + \beta$. Thus by increasing α and β by proportionate amounts the standard deviation may be decreased while holding the mean constant; and conversely by decreasing α and β by proportionate amounts, the standard deviation may be increased while leaving the mean unchanged.

Thus the range of the beta distribution is determined by A and B, the mean is somewhere between A and B depending on the $\alpha/(\alpha+\beta)$ ratio, and the standard deviation is small or large depending on whether the absolute value of $\alpha+\beta$ is large or small.

Now in the PERT application we want to work these equations in the reverse order. For given end points A and B, we must solve for the appropriate α and β, given mean and variance values determined from the PERT approximation formulas (see Eqn. B.1 and B.2). This can be readily done by introducing the quantity,

$$p=\alpha/(\alpha+\beta).$$

From the formula for the mean it follows that $p=(\mu-A)/(B-A)$. Then, since $1-p=\beta/(\alpha+\beta)$, the variance formula implies that,

$$\alpha+\beta=\frac{p(1-p)(B-A)^2}{\sigma^2}-1 \qquad \text{(B.5)}$$

Then α is given by $p(\alpha+\beta)$ and β is given by $(1-p)(\alpha+\beta)$. The beta distribution obtained in this way is called a PERT-beta distribution since the (α,β) shape parameters are chosen to give mean and variance as given by the PERT approximation formulas.

Summarizing, letting A, B, μ, σ in the above be *a, b, d*, σ (Eqn. B.1 and B.2) it follows that,

$$\alpha = \left(\frac{d-a}{b-a}\right)\left(\frac{(d-a)(b-d)}{\sigma^2} - 1\right) \tag{B.6}$$

and,

$$\beta = \left(\frac{b-d}{d-a}\right)\alpha\,, \tag{B.7}$$

where the mean and variance terms d and σ^2 are given in terms of *a, m* and *b* by the PERT approximation formulas given by Eqn. B.1 and B.2.

With these parameter formulas in hand, the simulations can be performed with the desired beta distributions, i.e. with beta distributions having ranges and means and variances according to the PERT approximation.