A Study on Nonparametric Inference Approaches for Stochastic Point Processes and Their Reliability Applications

Dissertation submitted in partial fulfillment for the degree of Doctor of Engineering

Yasuhiro Saito

Under the supervision of Professor Tadashi Dohi

Dependable Systems Laboratory, Department of Information Engineering, Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima, Japan

March 2016
Abstract

Stochastic point processes can be used as a powerful tool to describe stochastic behaviors of cumulative number of events occurred as time goes by. The occurrence of failures in repairable systems and the detection of faults in software testing are modeled by representative stochastic point processes. Non-homogeneous Poisson process (NHPP) is well known as the simplest but most useful method for modeling such phenomena.

Stochastic point processes are characterized by a conditional intensity function or the corresponding cumulative intensity function which is called mean value function especially for an NHPP. By assuming whether we can know the intensity function (or the corresponding cumulative intensity function) or not, two types of statistical inference approaches are considered. If the intensity function is known in advance, the model with the parametric intensity function is called parametric model. On the other hand, if the intensity function is unknown completely, it is called nonparametric model.

In this thesis, we mainly consider nonparametric estimation methods for stochastic point processes which include NHPP and a more generalized stochastic point process called the trend renewal process. In details, we discuss several nonparametric approaches for two different research areas; preventive maintenance scheduling problem of repairable systems and software reliability assessment.

In Chapter 2 and Chapter 3, we focus on parametric and nonparametric estimation methods for a periodic replacement problem with minimal repair which is a representative preventive maintenance scheduling problem. By modeling the occurrence of failures in repairable systems with NHPPs, we obtain the optimal periodic replacement time and its corresponding long-run average cost per unit time. We also discuss not only point estimation but also interval estimation for the same problem by applying several bootstrap techniques. It is revealed which method is an appropriate one in the both viewpoints of point estimation and interval estimation, throughout our simulation experiments and real failure data analyses.

In Chapter 4 and Chapter 5, we pay our attention to the software reliability assessment. Since NHPP-based Software reliability models (SRMs) are widely
used for modeling the detection of software faults in software testing, a variety of nonparametric estimation methods for NHPP models are considered. On the other hand, we also use a more generalized stochastic point process including NHPP, which is called non-homogeneous gamma process. By comparing our proposed models with conventional ones, we show the utility of our nonparametric models.

In Chapter 6, we concern a software release problem (SRP) based on a nonparametric NHPP-based SRM, where the intensity function of an NHPP-based SRM is unknown. To our best knowledge, there have no research result on the optimal software release problems under the assumption that the knowledge on the underlying software fault-detection process is incomplete. We calculate the predictive confidence interval as well as the point estimate of the optimal software release time which minimizes the expected total software cost. Finally, we conclude the thesis with some remarks in Chapter 7.
Acknowledgements

First and foremost, I would like to extend my sincere gratitude to Professor Tadashi Dohi, the supervisor of my study, for his constant guidance, kind advice and continuous encouragement throughout the progress of this thesis. Without his guidance and advices, the completion of this thesis would not have been possible.

Also, my thanks go to Professor Hiroaki Mukaidani and Associate Professor Hiroyuki Okamura, for their useful suggestions, many invaluable comments and checking the manuscript. I would like to greatly acknowledge Professor Won Young Yun of Pusan National University, who give me many invaluable comments and suggestions for my study in terms of preventive maintenance scheduling problem. And I also wish to thank professors of Japan Coast Guard Academy for their understanding and warm support.

Finally, it is my special pleasure to acknowledge the hospitality and encouragement of the past and present members of the Dependable Systems Laboratory, Department of Information Engineering, Hiroshima University.
Contents

Abstract iii

Acknowledgements vii

1 Introduction 1
  1.1 Periodic Replacement Problem (PRP) with Minimal Repair (MR) 2
  1.2 Software Reliability Assessment 4
  1.3 Optimal Software Release Problem (SRP) 7
  1.4 Organization of Dissertation 10

2 Uncertainty Analysis for a Periodic Replacement Problem with Minimal Repair: Parametric Bootstrapping 13
  2.1 Periodic replacement problem with minimal repair 14
    2.1.1 Model description 15
    2.1.2 Estimation of model parameters with failure data 16
  2.2 Parametric bootstrap method 17
  2.3 Numerical examples 19
  2.4 Real-life example 23

3 Kernel-Based Nonparametric Estimation Methods for a Periodic Replacement Problem with Minimal Repair 31
  3.1 Periodic Replacement Problem with Minimal Repair 32
    3.1.1 Model Description 32
    3.1.2 Parametric Estimation Method 33
    3.1.3 Constrained Nonparametric Maximum Likelihood Estimate 34
    3.1.4 Kernel-based Approach 36
  3.2 Interval Estimation 38
3.2.1 Bootstrapping ........................................... 38
3.2.2 Construction of Confidence Region ................. 39
3.3 Intensity Estimation with Multiple Minimal Repair Data Sets . 40
3.4 Simulation Experiments ................................. 42
  3.4.1 Point Estimation ........................................ 42
  3.4.2 Statistical Properties of Estimators .................. 45
  3.4.3 Interval Estimation ..................................... 48
3.5 Field Data Analysis .................................... 50

4 Software Reliability Assessment via Nonparametric Maximum Likelihood Estimation 57
  4.1 Software Reliability Modeling .......................... 57
    4.1.1 NHPP-based SRMs .................................. 57
    4.1.2 Maximum Likelihood Estimation ...................... 58
  4.2 Nonparametric Estimation ............................... 59
    4.2.1 Nonparametric ML Estimation ....................... 59
    4.2.2 Alternative NPMLE ................................. 61
    4.2.3 Total Time on Test Plot ............................ 63
  4.3 Software Reliability Assessment for Discrete Time Case ........ 64
    4.3.1 Discrete Time Non-homogeneous Poisson Process .... 65
    4.3.2 Constrained Nonparametric Maximum Likelihood Estimation ...... 66
  4.4 Numerical Illustrations ................................ 67
    4.4.1 Goodness-of-Fit Performance ....................... 67
    4.4.2 Software Reliability Measures ....................... 70
    4.4.3 Numerical Examples with Software Fault Count Data .. 77

5 Robustness of Non-homogeneous Gamma Process-based Software Reliability Models 79
  5.1 Summary on NHPP-based SRMs .......................... 79
    5.1.1 Model Description ................................. 79
    5.1.2 Constrained Nonparametric ML Estimation [A3] ....... 83
  5.2 NHGP-Based Software Reliability Models ................ 84
    5.2.1 A Generalization of NHPPs ......................... 84
CONTENTS

5.2.2 Behavior of Expected Cumulative Number of Software Faults in NHGP-based SRMs ....... 87
5.2.3 Constrained Nonparametric ML Estimation of NHGP ... 88
5.3 Numerical Illustrations .............................................. 91

6 Optimal Software Release Decision Based on Nonparametric Inference Approach

6.1 NHPP-Based Software Reliability Modeling ........................................ 107
6.1.1 Model Description .................................................. 107
6.1.2 Nonparametric Point Estimation ....................................... 108
6.2 Nonparametric Inference ................................................. 110
6.3 Optimal Software Release Decision .......................................... 114
6.4 Numerical Illustrations .................................................. 116

7 Conclusions .............................................................. 123

7.1 Summary and Remarks .................................................. 123
7.2 Future Works .......................................................... 125

Appendix A Properties of Nonparametric Estimators

A.1 Convergence Property of Kernel Estimator [1] ......................... 127
   A.3.1 Property 1 ....................................................... 128
   A.3.2 Property 2 ....................................................... 129

Publication List of the Author ............................................. 140
Chapter 1

Introduction

Stochastic point processes can be used as a powerful tool to describe stochastic behaviors of cumulative number of events occurred as time goes by. The occurrence of failures in repairable systems or the detection of faults in software testing is modeled by representative stochastic point processes. Non-homogeneous Poisson process (NHPP) is well known as the simplest but most useful method for modeling such phenomena.

Stochastic point processes are characterized by a conditional intensity function or the corresponding cumulative intensity function which is called mean value function especially for an NHPP. By assuming whether we can know the intensity function (or the corresponding cumulative intensity function) or not, two types of statistical inference approaches are considered. If the intensity function is known in advance, the model with the parametric intensity function is called parametric model. On the other hand, if the intensity function is unknown completely, it is called nonparametric model.

In this thesis, we mainly consider nonparametric estimation methods for stochastic point processes which include NHPP and a more generalized stochastic point process called the trend renewal process. In details, we discuss several nonparametric approaches for two different research areas; preventive maintenance scheduling problem of repairable systems and software reliability assessment.
1.1 Periodic Replacement Problem (PRP) with Minimal Repair (MR)

The periodic replacement problem by Barlow and Proschan [4] is one of the simplest, but most important preventive maintenance scheduling problems. Since the seminal contribution by Barlow and Hunter [5], several authors extended the original model from various points of view [6], [7]. Boland [8] gave the optimal periodic replacement time in case where the minimal repair cost depends on the age of component, and showed necessary and sufficient conditions for the existence of an optimal periodic replacement time in the case where the failure rate is strictly increasing failure rate (IFR). Nakagawa [9] proposed generalized periodic replacement policies with minimal repair where preventive maintenances are performed sequentially. Recently, Okamura et al. [10] developed a dynamic programming algorithm to obtain the optimal periodic replacement time in Nakagawa [9] more effectively. In another view, Sheu [11] considered a different periodic replacement model where the cost of minimal repair depends on the number of minimal repairs and the component age. Sheu [12] also proposed another generalized periodic replacement problem with minimal repair in which the minimal repair cost is assumed to be composed of age-dependent random and deterministic parts. Park et al. [13] considered the situation where each preventive maintenance relieves stress temporarily and reduces the rate of system degradation, while the failure rate of the system remains monotonically increasing. Colosimo et al. [14] discussed a different periodic replacement problem with minimal repair in which two different types of failures could be observed for systems according to their causes.

In this way, though a number of variations have been studied in the literature, it is assumed there that the failure time distribution or equivalently minimal repair process is completely known. In other words, when the failure time distribution is unknown in advance, the analytical models in the literature cannot provide the optimal periodic replacement time. The commonly used technique to identify the failure time distribution or the minimal repair process is to assume any parametric model and to estimate model parameters from the underlying minimal repair data, by means of maximum likelihood (ML) estimation. In many papers, the power-law process [15] and Cox-Lewis process [16]
are frequently assumed without justification as the representative minimal repair processes. Once the minimal repair process is identified, the point estimates of the long-run average cost per unit time and the optimal periodic replacement time are derived as plug-in estimates with model parameters estimated by the ML method.

For a simple age replacement model [4], Arunkumar [17], Bergman [18], Ingram and Scheaffer [19] concerned nonparametric estimation of the optimal age replacement time under the assumption where the independent and identically distributed (i.i.d.) failure time data is available but its probability distribution is unknown. They proposed statistically consistent estimators of the optimal age replacement time as point estimates. Leger and Cleroux [20] developed a nonparametric confidence interval of the optimal age replacement time in the same situation, where a nonparametric bootstrap method is applied. It should be noted that the interval estimation of the optimal maintenance policy is more difficult in analysis but more useful in practice than the point estimation, because it can take account of uncertainty in estimation. Since the convergence of nonparametric estimators is in general slower than parametric estimators in many cases, it is worth mentioning that the parametric models can sometimes work better despite of the incomplete knowledge on failure time distribution. Recently, Tokumoto et al. [21] derived parametric confidence intervals of the optimal age replacement problem under Weibull failure time distribution by means of parametric bootstrapping. In a somewhat different situation, Croteau et al. [22] proposed nonparametric confidence intervals of the optimal periodic replacement time with minimal repair, based on several nonparametric point estimates of minimal repair process. However, they implicitly assume the situation where the i.i.d. failure time data are available, but not where the record of minimal repairs is available. On the other hand, Gilardoni et al. [23] considered another nonparametric bootstrap confidence interval for a periodic replacement problem with minimal repair, when the minimal repair process or its superposition is observed. Their nonparametric approach is based on the total time on test transform [18] and a nonparametric maximum likelihood estimation (NPMLE) of non-homogeneous Poisson process by Boswell [24].

An alternative technique, which is called the kernel-based method, can be
applied to periodic replacement problem with minimal repair. Diggle and Marron [25] proved the equivalence of smoothing parameter selectors in probability density function estimation from i.i.d. sample and the intensity function estimation from the minimal repair data, in the framework of kernel-based techniques. In fact, the kernel-based approach is quite useful to improve the convergence of nonparametric estimators [26]. Gilardoni and Colosimo [27] applied the kernel-based estimation method to obtain the point estimate of the optimal periodic replacement time with several sets of minimal repair data, and obtained the nonparametric bootstrap confidence intervals.

In this thesis, we consider two situations in which the failure time distribution or equivalently minimal repair process is completely known or unknown. At first, we aim to derive the probability distributions of statistical estimators of optimal periodic replacement time using parametric bootstrap methods in a periodic replacement problem with minimal repair, and calculate the higher moments and the two-sided confidence intervals of these estimators. Next, we consider nonparametric estimation methods for a periodic replacement problem with minimal repair, where the expected cumulative number of failures (minimal repairs) is unknown. We mainly focus on the kernel based estimation methods with single minimal repair data. Furthermore, we use the well-known Gaussian kernel function, and apply two cross-validation methods for bandwidth estimation with integrated least squares error criterion [25] and log likelihood function criterion [28]. For constructing the bootstrap confidence interval of the optimal periodic replacement time, we apply three replication techniques for bootstrap samples; simulation-based technique and resampling-based techniques by Cowling et al. [2]. The above combinations on techniques are compared with the existing methods by Boswell [24] and Gilardoni et al. [23]. It is revealed which method is an appropriate one in the both viewpoints of point estimation and interval estimation, throughout our simulation experiments.

1.2 Software Reliability Assessment

Software reliability is a still challenging issue because almost all computer-based systems are controlled by software. Especially, quantification of software reliability is quite important from the standpoint of product liability. Since the
quantitative software reliability is defined as the probability that software failures caused by software faults do not occur for a given period of time, software reliability model (SRM) has been extensively studied in both software engineering and reliability engineering community. In fact, during the last four decades, a large number of SRMs have been proposed in the literature, and some of them have been used to assess software reliability and to control quantitatively software testing [29],[30]. Among them, SRMs based on non-homogeneous Poisson processes (NHPPs) have been extensively used for describing the stochastic behavior of the number of detected faults, from their tractability and goodness-of-fit performance. Achcar et al. [31], Goel and Okumoto [32], Goel [33], Gokhale and Trivedi [34], Ohba [35], Ohishi et al. [36], Okamura et al. [37], Yamada et al. [38], Zhao and Xie [39], among others, are well-known as representative NHPP-based SRMs. These NHPP-based SRMs can be classified into parametric models, where the mean value function or the intensity function characterizing NHPP-based SRMs is known in advance. More precisely, since the parametric SRMs depend on the statistical property of fault-detection time, the choice of NHPP-based SRMs is equivalent to choosing the fault-detection time distribution. However, the lesson learned from a number of empirical researches reported during the last four decades suggests that the best parametric SRM does not exist, which can fit every type of software fault data. This fact means that the best parametric NHPP-based SRM has to be selected carefully from many candidates by checking their goodness-of-fit performance in each software development project.

From the above motivation, the nonparametric SRMs without specifying the fault-detection time distribution have been considered by some authors. This is really a non-trivial issue because the NHPP-based SRM has to be characterized under the incomplete knowledge on software fault-detection time distribution. For unknown software intensity function or mean value function of an NHPP, Sofer and Miller [40] proposed a piecewise-linear interpolation estimator with breakpoints for the mean value function, and defined its slope as an estimate of the software intensity function. Gandy and Jensen [41], Burghout et al. [42], Wang et al. [43], Dharmasena et al. [44], Kaneishi and Dohi [45] independently proposed several kernel-based approaches to estimate the software inten-
sity function. However, the above all nonparametric NHPP-based SRMs are not consistent to the common maximum likelihood principle. Strictly speaking, it may be meaningless in some cases to compare the parametric NHPP-based SRMs with maximum likelihood estimates with the nonparametric NHPP-based SRMs under the different estimation principle.

On the other hand, it is well known that Poisson processes, even though they were non-homogeneous, do not allow for the variance to be adjusted independently of the mean value function, and fail to have the so-called overdispersion, which is the presence of greater variability (statistical dispersion) in data sets than would be expected based on given NHPP-based SRMs and is a very common feature in reliability analysis. All the representative NHPP-based SRMs [31],[32],[33],[34],[35],[36],[38],[39],[46],[47],[48] possess the above property and have some limitation. Apart from the selection of parametric and nonparametric estimation techniques in the common NHPP-based SRMs, it is also interesting to focus on more generalized SRMs as another modeling framework with overdispersion. Ishii and Dohi [49] developed non-homogeneous gamma process (NHGP)-based SRMs by employing an interesting idea by Berman [50]. The NHGP can be characterized as a modulated gamma renewal process by a trend function, which is equivalent to the mean value function of an NHPP. For instance, the NHGP is an extension of NHPPs by generalizing the exponentially distributed inter-renewal time to the gamma distributed one. Ishii and Dohi [49] implicitly assumed only a few types of trend functions and do not select the best trend function from possible candidates. In the life data analysis, the trend-renewal process (TRP) by Lindqvist et al. [51] is becoming a standard to analyze the complex failure count phenomena. It involves the NHGP as a special case, and overlaps a large amount of the so-called general repair model [52]. Although it is possible to develop different models from NHGP-based SRMs from the standpoint of TRP modeling framework, we pay our attention to evaluate the potential applicability and robustness of the NHGP-based SRMs [49].

In this thesis, we consider nonparametric approaches for NHPP-based SRMs and NHGP-based SRMs, respectively. At first, we develop alternative nonparametric approaches for NHPP-based SRMs along the similar line to the classical parametric approach. More specifically, we consider two nonparametric maxi-
1.3 Optimal Software Release Problem (SRP)

Apart from the software reliability assessment, the main concern in software process management by practitioners is to find when to stop software testing and to release software to the market or users. The problem to determine the optimal timing to stop software testing is called the optimal software release problem (SRP). In fact, there is a well-recognized tradeoff relationship in software costs. If the length of software test is much shorter, then the total testing cost can be reduced but the larger debugging cost after releasing software may occur in operational phase, because the debugging cost in operational phase is much more expensive than that in testing phase. Conversely, the longer testing period may result higher software reliability, but leads to increase of testing cost. Thus, it is important to find an appropriate software release time taking account of the expected total software cost. Okumoto and Goel [55] derived the optimal software release time such that the software reliability attains a certain requirement level, for their NHPP-based software reliability model (SRM). Alternative way introduced in [55] is to stop software testing so as to minimize the expected total software cost, by taking account of the tradeoff relationship. Since the seminal
work, many authors consider a number of cost-based software release problems under different model assumptions and optimization criteria. Koch and Kubat [56] discussed the similar problem under the reliability criterion to [55] with a different SRM. Yamada and Osaki [57,58] formulated the optimal software release problems by taking account of cost and reliability requirement simultaneously for NHPP-based SRMs. Hou et al. [59,60] obtained the optimal release policies with scheduled delivery time for the so-called hypergeometric distribution-based SRM. Dohi et al. [61,62] focused on a graphical feature of the cost-based software release problem and propose an estimation framework of the optimal software release time via artificial neural networks and autoregressive models. Pham and Zhang [63] proposed a somewhat different software cost model with warranty and risk costs. Xie and Yang [64] examined an effect of imperfect debugging in a software release problem. Yang et al. [65] also studied a different software cost model from [64] and investigated the uncertainty in software release problem. Sgarbossa and Pham [66] formulated a more complex software release problem under random field environments and reliability. In this way, considerable attentions have been paid for the derivation of optimal software release policies with different cost criteria and SRMs.

It should be noted that almost all works in past concern the point estimation of the optimal software release time and never take account uncertainty of the optimal policy into consideration. In other words, if the statistical estimation error occurs, then the resulting policy may not be able to provide the optimal software release decision appropriately. The pioneering work is made by Zhao and Xie [67], where the authors considered approximately the interval estimation in two software release problems. Recently, Xie et al. [68] generalized their result and characterized the risk-based optimal software release policies. Okamura et al. [69] also formulated the exactly same problems within the Bayesian estimation framework and derived analytically the credible interval of the optimal software release times without approximations. Tokumoto and Dohi [70] applied a parametric bootstrapping [71] to obtain the two-sided confidence interval of the optimal software release time numerically. As another research direction in software reliability, some authors concern nonparametric estimation of software fault-detection process based on the kernel-based methodology and
1.3. OPTIMAL SOFTWARE RELEASE PROBLEM (SRP)

the wavelet-shrinkage estimations. However, such methodologies never work well for the optimal software release problems, because one needs to predict the number of software failures occurred in the long-term future, after releasing software to the market or users. More precisely, it is not so easy to formulate the software reliability function or the expected total software cost under the nonparametric assumption. To our best knowledge, there have no research result on the optimal software release problems under the assumption that the knowledge on the underlying software fault-detection process is uncompleted.

We consider a challenging issue on nonparametric optimal software release problem, where the intensity function of an NHPP-based SRM is unknown. Under such incomplete knowledge on software fault-detection process, the main difficulty here is that the statistical estimates of intensity function are functions of only fault-detection time data in the above nonparametric estimation methods. We focus on a different statistical inference paradigm and propose to use an alternative nonparametric approach by Sofer and Miller [40]. The most significant contribution of their work is that it can provide the upper and lower predictive limits of the cumulative number of software faults even after releasing the software. Since the long-term prediction is definitely needed to formulate the common optimal software release problem, we apply Sofer and Miller approach [40] to estimate the expected total software cost. In addition, we calculate the predictive confidence interval as well as the point estimate of the optimal software release time which minimizes the expected total software cost. For this purpose, we apply a nonparametric bootstrap method used in Kaneishi and Dohi [45] to obtain the two-sided predictive confidence interval of the optimal software release time by minimizing the upper or lower limit of the expected total software cost. Since the resulting optimal stopping problem enables us to make the decision on when to stop or continue software testing, it is feasible to provide a realistic false alarm on the optimal release decision (note that almost all works in past [55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65] have not dealt with statistical estimation problems, so their release alarm might indicate an unrealistic time point such as the past point looked back from the current observation point). Since our method is based on a predictive approach in spite of its nonparametric nature, it is useful to make flexible decision making
CHAPTER 1. INTRODUCTION

on when to stop software testing under uncertainty.

1.4 Organization of Dissertation

This thesis is organized as follows:

In Chapter 2, we consider a statistical estimation problem for a periodic replacement problem with minimal repair and propose two parametric bootstrap methods. Especially, we concern two data analysis techniques: direct data analysis of the minimal repair data which obeys a non-homogeneous Poisson process and indirect data analysis after data transformation to a homogeneous Poisson process. Through simulation experiments, we investigate statistical features of the proposed parametric bootstrap methods. Also, we analyze the real minimal repair data to demonstrate the proposed methods in practice.

In Chapter 3, we consider nonparametric estimation methods for a periodic replacement problem with minimal repair, where the expected cumulative number of failures (minimal repairs) is unknown. To construct the confidence interval of an estimator of the optimal periodic replacement time which minimizes the long-run average cost per unit time, we apply two kernel-based bootstrap estimation methods and three replication techniques for bootstrap samples. In simulation experiments, we compare those results with the well-known constrained nonparametric maximum likelihood estimate (CNPMLE) and some parametric models. We also conduct the field data analysis based on an actual minimal repair data, and refer to an applicability of our methods.

In Chapter 4, we consider three nonparametric estimation methods for software reliability assessment without specifying the fault-detection time distribution, where the underlying stochastic process to describe software fault-counts in the system testing is given by a non-homogeneous Poisson process. Two of these are nonparametric estimation methods with fault detection time data. On the other hand, one of these is a nonparametric estimation method with software fault count data. The resulting data-driven methodologies can give the useful probabilistic information on the software reliability assessment under the incomplete knowledge on fault-detection time distribution. Throughout examples with real software fault data, it is shown that the proposed methods provide
1.4. ORGANIZATION OF DISSERTATION

more accurate estimation results than the common parametric approach.

In Chapter 5, we consider a software release decision to stop the software testing by minimizing the expected total software cost, under the assumption that the probability law of software fault-detection process is unknown. We focus on the nonparametric prediction method of a non-homogeneous Poisson process by Sofer and Miller [40] and apply it to the optimal software release problem. We calculate the predictive confidence interval as well as the point estimate of the optimal software release time. Since our method is based on a predictive approach in spite of its nonparametric nature, it is useful to make flexible decision making to stop software testing.

In Chapter 6, we extend non-homogeneous gamma process (NHGP)-based software reliability models (SRMs) by Ishii and Dohi [49] from both viewpoints of modeling and parameter estimation. We compare our nonparametric maximum likelihood estimation method with the parametric maximum likelihood estimation method. Since an NHGP involves an NHPP as a special case, it is shown that NHGP-based SRMs are much more robust than the common NHPP-based SRMs and that our nonparametric method can improve the goodness-of-fit performance of the conventional parametric one.

Finally, the thesis is concluded with some remarks and future directions in Chapter 7.
Chapter 2

Uncertainty Analysis for a Periodic Replacement Problem with Minimal Repair: Parametric Bootstrapping

In this chapter, we consider a statistical estimation problem for a periodic replacement problem (PRP) with minimal repair (MR) which is one of the most fundamental maintenance models in practice, and propose two parametric bootstrap methods which are categorized into simulation-based approach and resampling-based approach. Especially, we concern two data analysis techniques: direct data analysis of the minimal repair data which obeys a non-homogeneous Poisson process and indirect data analysis after data transformation to a homogeneous Poisson process. In details, we take place the high level statistical estimation of the optimal preventive maintenance time and its associated long-run average cost per unit time, and derive estimators of higher moments of the optimal maintenance policy, and its confidence interval. Then, the parametric bootstrap methods play a significant role. The proposed approach enables us the statistical decision making on the preventive maintenance planning under uncertainty. Through simulation experiments, we investigate statistical features of the proposed parametric bootstrap methods. Also, we analyze the real minimal repair data to demonstrate the proposed methods in practice.
CHAPTER 2. PARAMETRIC APPROACH FOR PRP WITH MR

2.1 Periodic replacement problem with minimal repair

Notation;

\( C(\tau) \): long-run average cost per unit time

\( C(\hat{\tau}^*) \): maximum likelihood (ML) point estimate of the long-run average cost per unit time

\( C_{(k)}(\hat{\tau}^*) \): \( k \)-th estimate of the the long-run average cost per unit time via bootstrap method

\( c_1 (> 0) \): cost of each minimal repair

\( c_2 (> c_1) \): cost of each periodic replacement

\( m \): number of replications via bootstrap method

\( \{ N(t), t \geq 0 \} \): cumulative number of failures experienced by time \( t \) (the total number of minimal repairs by time \( t \))

\( n \): number of failure data

\( T \): failure time of the component (non-negative random variable)

\( t_i \): occurrence time of \( i \)-th failure under minimal repair

\( \tau \): periodic replacement time

\( \tau^* \): optimal periodic replacement time

\( \hat{\tau}^* \): ML point estimate of the optimal periodic replacement time

\( \hat{\tau}_{(k)}^* \): \( k \)-th estimate of the optimal periodic replacement time via bootstrap method

\( t_{k,i}^* \): \( i \)-th failure time data at \( k \)-th (\( k = 1, 2, \cdots, m \)) replication

\( w_{k,i}^* \): \( i \)-th failure time interval data at \( k \)-th (\( k = 1, 2, \cdots, m \)) replication

\( \alpha \in (0,1) \): significance level

\( \beta (> 1) \): shape parameter of power law model
2.1. PERIODIC REPLACEMENT PROBLEM WITH MINIMAL REPAIR

\( \hat{\beta} \): ML estimate of the shape parameter

\( \eta (> 0) \): scale parameter of power law model

\( \hat{\eta} \): ML estimate of the scale parameter

\( \Lambda(t) \): expected cumulative number of failures occurred (equivalently the cumulative number of minimal repairs) by time \( t \)

\( \lambda(t) \): failure intensity function of component

2.1.1 Model description

We consider a periodic replacement problem with minimal repair for single unit system. Suppose that the failure time \( T \) has an absolutely continuous probability distribution function \( \Pr(T \leq t) = F(t) \) and a probability density function \( dF(t)/dt = f(t) \). In the periodic replacement problem with minimal repair, the component which fails before time \( \tau (> 0) \) is restored back to a working condition that is only as good as it was just before failure. In other words, after each failure, only minimal repair is made so that the failure rate remains undisturbed by repair. Also, the used component is replaced by a new one at time \( \tau \) preventively. Since the administrator need not to record the past replacement history in this way, it is easy to take place the preventive maintenance. On the other hand, even if the component is temporarily repaired before time \( \tau \), the same component is replaced by a new one at time \( \tau \). Therefore, there is an additional cost for the preventive maintenance. The expected total cost for one cycle can be represented by \( c_1 \Lambda(\tau) + c_2 \), where one cycle is the time length from the beginning of component operation to the end of preventive replacement. Therefore, the long-run average cost per unit time is given by

\[
C(\tau) = \frac{c_1 \Lambda(\tau) + c_2}{\tau},
\]

(2.1)

The problem here is to obtain the optimal periodic replacement time \( \tau^* \) minimizing Eq.(2.1). By differentiating the function \( C(\tau) \) with respect to \( \tau \) and setting it equal to zero, we have \( q(\tau^*) = 0 \) where

\[
q(\tau) = c_1 \left\{ \tau \lambda(\tau) \right\} \Lambda(\tau) \right\} + c_2
\]

(2.2)
and $\lambda(\tau) = d\Lambda(\tau)/d\tau$. Under the condition that the intensity function increases with respect to time ($i.e., d\lambda(t)/dt > 0$), if $q(\infty) > 0$, then there exists a unique and finite optimal periodic replacement time $\tau^* \ (0 < \tau^* < \infty)$.

### 2.1.2 Estimation of model parameters with failure data

The failure occurrence phenomenon with minimal repair is described by a non-homogeneous Poisson process (NHPP). It is well known that the stochastic point process $N(t)$ possesses the following properties:

- $N(0) = 0$,
- $\{N(t), t \geq 0\}$ has independent increments,
- $\Pr\{N(t + \Delta t) - N(t) \geq 2\} = o(\Delta t)$,
- $\Pr\{N(t + \Delta t) - N(t) = 1\} = \lambda(t)\Delta t + o(\Delta t)$,

where the function $o(\Delta t)$ is the higher term of $\Delta t$. The probability mass function of $N(t)$ is given by

$$\Pr(N(t) = n) = \frac{[\Lambda(t)]^n}{n!} \exp\{-\Lambda(t)\}, \quad n = 0, 1, 2, \cdots, \quad (2.3)$$

$$\Lambda(t) = \int_0^t \lambda(x)dx. \quad (2.4)$$

Here, the function $N(t)$ is called an NHPP having the mean value function $\Lambda(t)$ and intensity function $\lambda(t) = d\Lambda(t)/dt$.

Furthermore, we assume the power law model (see [15]):

$$\Lambda(t; \eta, \beta) = \log F(t) = \frac{t^\beta}{\eta}, \quad (2.5)$$

where $F(t) = 1$ $F(t)$ and $F(t) = 1$ $\exp\{ - (t/\eta)^\beta \}$. If $\beta > 1$, then we have $d\lambda(t)/dt > 0$ and $q(\infty) > 0$ in the power law model, so that the optimal periodic replacement time $\tau^*$ exists uniquely. We focus on only the plausible situation of $\beta > 1$.

The next step is to estimate the model parameters $(\eta, \beta)$. Suppose that $n$ failure time data $t_i \ (i = 1, 2, \cdots, n)$ under minimal repair, which obey the power law model, are available. The maximum likelihood (ML) estimates $(\hat{\eta}, \hat{\beta})$ are defined as the parameters which maximize the following log likelihood function:

$$\log L_{\text{NHPP}}(\eta, \beta | t_i) = \sum_{i=1}^n \log \lambda(t_i; \eta, \beta) \quad \Lambda(t_i; \eta, \beta). \quad (2.6)$$
2.2 Parameter Bootstrap Method

From the first order condition of optimality in Eq. (2.6), we calculate the ML estimates \((\hat{\eta}, \hat{\beta})\) by solving the following simultaneous equations:

\[
\hat{\beta} = \frac{n}{n} \log t_n = \frac{1}{n} \sum_{i=1}^{n} \log t_i, \tag{2.7}
\]

\[
\hat{\eta} = \left( \frac{\mu^{\beta}}{n} \right)^{\frac{1}{\beta}}. \tag{2.8}
\]

By substituting the resulting estimates \((\hat{\eta}, \hat{\beta})\) into Eq. (2.2), we obtain the ML plug-in point estimates \(\hat{\tau}^*\) and \(C(\hat{\tau}^*)\) of the optimal periodic replacement time \(\tau^*\) and its associated long-run average cost per unit time \(C(\tau^*)\).

2.2 Parametric bootstrap method

It may be useful to obtain the ML estimate \(\hat{\tau}^*\) of the optimal periodic replacement time. However, since the resulting estimate is calculated from a fixed sample of failure time data \(t_1, t_2, \ldots, t_n\), we cannot correspond to unknown failure patterns in the future, and cannot consider the uncertainty of the estimator as a random variable. For such a problem, it is well known that the interval estimation may work better to make the valid decision under uncertainty. Unfortunately, it is difficult to derive analytically the interval estimates of complex variables such as the optimal replacement time and the corresponding long-run average cost per unit time. Hence, we consider a statistical estimation problem with the parametric bootstrap method and derive the probability distribution of estimators such as the optimal periodic replacement time and the corresponding long-run average cost per unit time. The bootstrap method is the representative statistical approach to replicate different failure time data sets from original failure time (minimal repair time) data \(t_1, t_2, \ldots, t_n\) (see [72]). The generated failure time data sets are called the bootstrap samples. For the periodic replacement problem with minimal repair, we replicate \(m (> 0)\) bootstrap samples from original failure time (minimal repair time) data \(t_1, t_2, \ldots, t_n\) and obtain the \(m\) ML estimates of model parameters \(\eta\) and \(\beta\) with \(m\) bootstrap samples. In this chapter we propose two bootstrap methods: simulation-based method and re-sampling-based method to replicate the bootstrap samples. Furthermore, we propose two data analysis techniques: direct application of the minimal repair data which obey a non-homogeneous Poisson process and data transformation to a homogeneous
Poison process (HPP). To this end, we consider four bootstrap methods to replicate the bootstrap samples as follows.

**Method (i) Simulation-based method to NHPP data (NHPP-BS1):** With the ML estimates \( \hat{\eta} \) and \( \hat{\beta} \) calculated from the original failure time data under minimal repair \( t_1, t_2, \ldots, t_n \), we generate bootstrap samples \( t_{k,i}^* \) at \( k \)-th simulation which follow the power law model, where \( k = 1, 2, \ldots, m \) and \( i = 1, 2, \ldots, n \).

It is known that the first failure time \( t_{k,1}^* \) obeys the probability distribution function

\[
F(t_{k,1}^*) = 1 - \exp\left( - \left( \frac{\eta}{\beta} \log U_k(0,1) \right)^{\frac{1}{\beta}} \right),
\]

(2.9)

and the conditional probability distribution of the \( n \)-th failure time \( t_{k,j}^* \) \((j = 2, 3, \ldots, n)\) is given by

\[
F(t_{k,j}^* | t_{k,1}^*) = 1 - \exp\left( - \left( \frac{\eta}{\beta} \log U_k(0,1) + t_{k,1}^* \right) \right)^{\frac{1}{\beta}}.
\]

(2.10)

where \( U_k(0,1) \) is the uniform random numbers within \([0,1]\) (see [73]). Then we estimate \( m \) ML estimates \( (\hat{\eta}_k, \hat{\beta}_k) \) \((k = 1, 2, \ldots, m)\) based on the bootstrap samples generated from Eqs.(2.9) and (2.10).

**Method (ii) Re-sampling-based method to NHPP data (NHPP-BS2):** We sample \( n \) failure time data with replacement randomly from the original data \( t_1, t_2, \ldots, t_n \). We obtain \( m \) ML estimates \( (\hat{\eta}_k, \hat{\beta}_k) \) \((k = 1, 2, \ldots, m)\) with bootstrap samples \( t_{k,i}^* \) \((i = 1, 2, \ldots, n; k = 1, 2, \ldots, m)\) obtained by the re-sampling technique.

**Method (iii) Simulation-based method to HPP data (HPP-BS1):** First, we generate the pseudo failure time interval data \( w_{k,i}^* \) which obey the exponential distribution with parameter \( \lambda = 1 \) by using the inverse function of the exponential distribution function, where \( i = 1, 2, \ldots, n \) and \( k = 1, 2, \ldots, m \). Second, we transform the generated failure time interval data to the failure time data which follows an HPP. These data can be transformed to the failure time data sets \( t_{k,i}^* \) which follows the power law model with the inverse function of the mean value function and the ML estimates \( \hat{\eta} \) and \( \hat{\beta} \) calculated from the original data \( t_1, t_2, \ldots, t_n \). With these bootstrap samples \( t_{k,i}^* \), we obtain \( m \) ML estimates \( (\hat{\eta}_k, \hat{\beta}_k) \) \((k = 1, 2, \ldots, m)\).
2.3. NUMERICAL EXAMPLES

Method (iv) Re-sampling-based method to HPP data (HPP-BS2): First, we transform the original failure time data under minimal repair \( t_1, t_2, \ldots, t_n \) to the mean value function data (HPP data) with mean value function of the power law model and the ML estimates \( \hat{\eta} \) and \( \hat{\beta} \). After transforming these data to the time interval data, we randomly sample \( n \) failure time interval data with replacement. Then we retransform the failure time interval data \( w_{k,i} \) \( (i = 1, 2, \ldots, n; k = 1, 2, \ldots, m) \) to the failure time data sets \( t_{k,i} \) which follow the power law model. Finally, we obtain \( m \) ML estimates \( (\hat{\eta}_k, \hat{\beta}_k) \) \( (k = 1, 2, \ldots, m) \) with these bootstrap samples.

From Eq.(2.2) and \( m \) ML estimates \( (\hat{\eta}_k, \hat{\beta}_k) \) \( (k = 1, 2, \ldots, m) \) derived in the above four methods, we calculate \( m \) ML estimates of the optimal periodic replacement time \( \tau^* \). Furthermore, we calculate the corresponding ML estimates of the minimum long-run average cost per unit time \( C(\tau^*) \) in Eq.(2.1) and derive the estimator distributions of \( \tau^* \) and \( C(\tau^*) \). Let \( \hat{\tau}^*_k \) be each estimate of the optimal periodic replacement time \( \tau^* \) which is calculated by using bootstrap samples, where \( k = 1, 2, \ldots, m \) and \( \hat{\tau}^*_k \) satisfy the condition \( \hat{\tau}^*_1 \geq \hat{\tau}^*_2 \geq \cdots \geq \hat{\tau}^*_m \). Also, let \( C_k(\hat{\tau}^*) \) \( (C_1(\hat{\tau}^*) \geq C_2(\hat{\tau}^*) \geq \cdots \geq C_m(\hat{\tau}^*)) \) be the estimate of minimum long-run average cost per unit time \( C(\tau^*) \). We label \( \hat{\tau}^*_{(m/2)} \) and \( \hat{\tau}^* = (\sum_{k=1}^{m} \hat{\tau}^*_k) / m \) as the BS-median and the BS-mean, respectively. Furthermore, the two-sided 100(1 - \( \alpha \)% confidence interval is given by \( \left[ \hat{\tau}^*_{(\alpha/2)} \right. \hat{\tau}^*_{(1 - \alpha/2)}] \), where \( \alpha \in (0, 1) \) represents the significance level. The variance \( V \), skewness \( S \) and kurtosis \( K \) of estimators of the optimal periodic replacement time are defined by the following equations:

\[
V_{\tau^*} = \frac{\sum_{k=1}^{m} \left( \hat{\tau}^*_k - \hat{\tau}^* \right)^2}{m - 1} \quad (2.11)
\]

\[
S_{\tau^*} = \frac{\sum_{k=1}^{m} \left( \hat{\tau}^*_k - \hat{\tau}^* \right)^3}{m V_{\tau^*}^{\frac{3}{2}}} \quad (2.12)
\]

\[
K_{\tau^*} = \frac{\sum_{k=1}^{m} \left( \hat{\tau}^*_k - \hat{\tau}^* \right)^4}{m V_{\tau^*}^2} \quad (2.13)
\]

2.3 Numerical examples

Suppose the power law model with \( (\beta, \eta) = (3, 0.2) \) and generate the original failure time data under minimal repair in our simulation experiment. The cost
parameters are given by $c_1 = 1$ and $c_2 = 1,000$. In this case, the real solution of the optimal periodic replacement time and the corresponding long-run average cost per unit time are calculated as $\tau^* = 1.587$ and $C(\tau^*) = 945$, respectively. We apply four bootstrap methods in Section 2.2 to generate $m = 10,000$ bootstrap samples. We obtain 10,000 model parameters with ML estimation, and obtain 10,000 estimates of the optimal periodic replacement time and its corresponding long-run average cost per unit time. Based on the estimator distributions, we derive variance, skewness and kurtosis of the optimal periodic replacement time and its corresponding long-run average cost per unit time. For the significance level $\alpha = 0.05$, we derive 95% two-sided confidence intervals of the optimal periodic replacement time and the corresponding long-run average cost per unit time. It is well known that if the skewness is closed to 0.0 and the kurtosis is closed to 3.0, then we can regard the estimator distribution as normal distribution approximately.

To investigate the effect of the number of original data $n$, we change the number of original data from $n = 5$ to $n = 245$ by 5 and calculate the BS estimates (BS-mean and BS-median), high moments (variance, skewness, kurtosis) and 95% two-sided confidence intervals (CI). Tables 2.1-2.2 show estimation results of the optimal periodic replacement time and the corresponding long-run average cost per unit time with four bootstrap methods. From these tables, it is observed that the variance of probability distribution of estimators reduces as the number of original data increases in all bootstrap methods. Also it can be seen that the skewness and kurtosis of minimum long-run average cost per unit time in NHPP-BS1, HPP-BS1 and HPP-BS2 cases are close to 0.0 and 3.0, respectively, even in cases with small sample sizes. In NHPP-BS2 case, the estimator distribution for the minimum long-run average cost per unit time cannot be approximated by the normal distribution. On the other hand, looking at the optimal periodic replacement time, the skewness and kurtosis tend to be greater than 0.0 and 3.0, respectively in all cases.

Table 2.3 presents the estimation of normal approximation which is applied with BS-mean and variance of estimators of the minimum long-run average cost per unit time. We compare the length of 95% confidence interval in the case where the derived estimator distribution is approximated by normal distribution
2.3. NUMERICAL EXAMPLES

with that of 95% confidence interval derived with our estimator distributions. From this table, we can know that the differences of two 95% confidence intervals get smaller as the number of original data increases. Furthermore, the differences on two 95% confidence intervals from NHPP-BS2 method are most remarkable since the values of skewness and kurtosis are little bit far from values in the normal distribution in NHPP-BS2 case corresponding to other cases.

Table 2.1: The results of optimal periodic replacement time with four bootstrap methods.

<table>
<thead>
<tr>
<th>n</th>
<th>MLE</th>
<th>Mean</th>
<th>$V_{T_T}$</th>
<th>$S_{T_T}$</th>
<th>$K_{T_T}$</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NHPP BS1</td>
<td>50</td>
<td>1.533</td>
<td>1.522</td>
<td>0.080</td>
<td>1.219</td>
<td>6.122</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1.608</td>
<td>1.601</td>
<td>0.033</td>
<td>0.681</td>
<td>3.832</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>1.672</td>
<td>1.671</td>
<td>0.023</td>
<td>0.653</td>
<td>3.979</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>1.566</td>
<td>1.565</td>
<td>0.009</td>
<td>0.437</td>
<td>3.351</td>
</tr>
<tr>
<td>BS2</td>
<td>50</td>
<td>1.533</td>
<td>1.445</td>
<td>0.052</td>
<td>0.683</td>
<td>3.765</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1.608</td>
<td>1.590</td>
<td>0.029</td>
<td>0.721</td>
<td>4.032</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>1.672</td>
<td>1.677</td>
<td>0.020</td>
<td>0.639</td>
<td>3.789</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>1.566</td>
<td>1.565</td>
<td>0.007</td>
<td>0.541</td>
<td>3.680</td>
</tr>
<tr>
<td>HPP BS1</td>
<td>50</td>
<td>1.533</td>
<td>1.527</td>
<td>0.083</td>
<td>1.205</td>
<td>5.880</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1.608</td>
<td>1.605</td>
<td>0.033</td>
<td>0.716</td>
<td>3.848</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>1.672</td>
<td>1.670</td>
<td>0.023</td>
<td>0.651</td>
<td>3.762</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>1.566</td>
<td>1.563</td>
<td>0.009</td>
<td>0.473</td>
<td>3.465</td>
</tr>
<tr>
<td>BS2</td>
<td>50</td>
<td>1.533</td>
<td>1.540</td>
<td>0.098</td>
<td>1.399</td>
<td>7.189</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1.608</td>
<td>1.597</td>
<td>0.027</td>
<td>0.734</td>
<td>4.059</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>1.672</td>
<td>1.670</td>
<td>0.020</td>
<td>0.579</td>
<td>3.555</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>1.566</td>
<td>1.566</td>
<td>0.009</td>
<td>0.426</td>
<td>3.424</td>
</tr>
</tbody>
</table>

Figures 2.1-2.2 show asymptotic behaviors of estimators of the optimal periodic replacement time and the corresponding long-run average cost per unit time in HPP-BS2 case. We also provide the similar results to other methods. From these figures, we can see that the confidence intervals get tighter as the sample size of original data $n$ increases. Furthermore, it is observed that the BS-mean and BS-median get closer to the ML estimate. But, even if the sample
Table 2.2: The results of minimum long-run average cost per unit time with four bootstrap methods.

<table>
<thead>
<tr>
<th>n</th>
<th>MLE</th>
<th>Mean</th>
<th>$V_{C^-}$</th>
<th>$S_{C^-}$</th>
<th>$K_{C^-}$</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>967</td>
<td>987</td>
<td>11634</td>
<td>-0.093</td>
<td>3.026</td>
<td>[771, 1190]</td>
</tr>
<tr>
<td>100</td>
<td>941</td>
<td>948</td>
<td>3679</td>
<td>-0.027</td>
<td>3.084</td>
<td>[827, 1066]</td>
</tr>
<tr>
<td>150</td>
<td>931</td>
<td>934</td>
<td>1970</td>
<td>-0.070</td>
<td>3.088</td>
<td>[845, 1019]</td>
</tr>
<tr>
<td>200</td>
<td>948</td>
<td>950</td>
<td>857</td>
<td>-0.035</td>
<td>2.925</td>
<td>[892, 1006]</td>
</tr>
<tr>
<td>50</td>
<td>967</td>
<td>1020</td>
<td>9854</td>
<td>0.095</td>
<td>2.771</td>
<td>[833, 1220]</td>
</tr>
<tr>
<td>100</td>
<td>941</td>
<td>952</td>
<td>2646</td>
<td>-0.230</td>
<td>3.080</td>
<td>[845, 1048]</td>
</tr>
<tr>
<td>150</td>
<td>931</td>
<td>932</td>
<td>1249</td>
<td>-0.322</td>
<td>3.172</td>
<td>[857, 996]</td>
</tr>
<tr>
<td>200</td>
<td>948</td>
<td>950</td>
<td>358</td>
<td>-0.437</td>
<td>3.395</td>
<td>[909, 983]</td>
</tr>
<tr>
<td>50</td>
<td>967</td>
<td>985</td>
<td>11990</td>
<td>-0.065</td>
<td>3.110</td>
<td>[765, 1198]</td>
</tr>
<tr>
<td>100</td>
<td>941</td>
<td>947</td>
<td>3622</td>
<td>-0.086</td>
<td>2.971</td>
<td>[826, 1063]</td>
</tr>
<tr>
<td>150</td>
<td>931</td>
<td>934</td>
<td>1985</td>
<td>-0.113</td>
<td>3.075</td>
<td>[843, 1019]</td>
</tr>
<tr>
<td>200</td>
<td>948</td>
<td>950</td>
<td>865</td>
<td>-0.017</td>
<td>3.022</td>
<td>[893, 1008]</td>
</tr>
<tr>
<td>50</td>
<td>967</td>
<td>983</td>
<td>13375</td>
<td>-0.129</td>
<td>3.008</td>
<td>[751, 1202]</td>
</tr>
<tr>
<td>100</td>
<td>941</td>
<td>948</td>
<td>2944</td>
<td>-0.098</td>
<td>3.096</td>
<td>[839, 1053]</td>
</tr>
<tr>
<td>150</td>
<td>931</td>
<td>934</td>
<td>1779</td>
<td>-0.066</td>
<td>3.032</td>
<td>[848, 1016]</td>
</tr>
<tr>
<td>200</td>
<td>948</td>
<td>950</td>
<td>879</td>
<td>0.041</td>
<td>3.028</td>
<td>[892, 1008]</td>
</tr>
</tbody>
</table>
size of data \( n \) increases as large as possible, BS-mean and BS-median are not equal to the ML estimate. This fact means that we cannot get the consistent estimator from only one failure time data set \( t_1, t_2, \ldots, t_n \). It can be observed that BS-mean and BS-median give values closed to the real optimal solutions, but these values are not the exactly same value. Hence, we cannot conclude which bootstrap method in Section 2.2 is the best from the view point of point estimation. Figures 2.3-2.4 show the trend of the confidence interval in each bootstrap method. In these figures, the vertical axis represents the length of 95% two-sided confidence intervals of optimal periodic replacement time \( \tau_{[95]}^* \) and minimum long-run average cost per unit time \( C_{[95]}(\tau^*) \). From Fig.2.3, it is obviously known that the 95% confidence interval of optimal periodic replacement time is the narrowest in NHPP-BS2 case. On the other hand, it is not clear which method has the superiority with respect to the 95% confidence interval of minimum long-run average cost per unit time. However, we note that there are relatively many cases in which the 95% confidence interval in NHPP-BS2 case is the narrowest from Fig.2.4. Therefore, it can be concluded that the NHPP-BS2 is the best method since the 95% two-sided confidence interval of estimators is tightest.

2.4 Real-life example

The real failure data analysis is useful to give a reality to the mathematical modeling (see [74]). We also study a real life example to demonstrate how to use our proposed parametric bootstrap methods for the periodic replacement problem with minimal repair. Suppose that the old car user encounters a problem whether to perform the minimal repair for the diesel engine of his or her car, or to replace by the new one. Since the user wishes to minimize the expected cost for maintenance management of diesel engine, he or she makes a record of times of minimal repair in order to plan optimally preventive maintenance.

In this section, we consider the minimal repair data of a diesel engine which is shown in Meeker and Escobar [75] as the failure data of car engine \( n = 71, \) \( \text{MTBF} = 0.359 \) month). Figure 2.5 shows the trend of the minimal repair data. It is obviously shown that these data have the convex shape with respect to time. Therefore, it is easy to know that the failure rate gets higher as time goes
### Table 2.3: The results of minimum long-run average cost per unit time with four bootstrap methods.

<table>
<thead>
<tr>
<th>$n$</th>
<th>Mean</th>
<th>$V_C$-Estimator</th>
<th>Normal distribution approximation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NHPP BS1</td>
<td>50</td>
<td>986.9</td>
<td>11634</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>947.8</td>
<td>3679</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>933.8</td>
<td>1970</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>949.6</td>
<td>857</td>
</tr>
<tr>
<td>NHPP BS2</td>
<td>50</td>
<td>1020.2</td>
<td>9854</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>951.9</td>
<td>2646</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>931.5</td>
<td>1249</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>949.5</td>
<td>358</td>
</tr>
<tr>
<td>HPP BS1</td>
<td>50</td>
<td>985.5</td>
<td>11990</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>946.6</td>
<td>3622</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>934.3</td>
<td>1985</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>950.3</td>
<td>865</td>
</tr>
<tr>
<td>HPP BS2</td>
<td>50</td>
<td>983.1</td>
<td>13375</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>948.0</td>
<td>2944</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>933.8</td>
<td>1779</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>949.7</td>
<td>879</td>
</tr>
</tbody>
</table>
by. The ML estimates of model parameters from the given data are obtained as $(\hat{\beta}, \hat{\eta}) = (2.76, 5.45)$. We assume the minimal repair cost $c_1 = 1(\$)$ and the periodic replacement cost $c_2 = 1,000(\$)$. We use NHPP-BS2 method to obtain 10,000 bootstrap samples and derive the estimator distribution, because the NHPP-BS2 method gives the best confidence intervals of optimal replacement times in many numerical cases of Section 2.3. Table 2.4 gives the estimation result of the optimal periodic replacement time and its corresponding long-run average cost per unit time. From this table, we can see that BS-mean and BS-median take much closed values to ML estimate (MLE). Figures 2.6-2.7 show the estimator distributions of the optimal periodic replacement time and its corresponding long-run average cost per unit time. From these figures, it is seen that the confidence interval includes the BS-mean, BS-median and MLE. From the shape of distribution and the values of skewness and kurtosis, it is observed that the estimator distribution of optimal periodic replacement time is skew to left and that of minimum long-run average cost per unit time is skew.
Figure 2.2: Asymptotic estimates of minimum long-run average cost per unit time in HPP-BS2 case.

Figure 2.3: Asymptotic length of 95\% two-sided confidence interval of optimal periodic replacement time.
2.4. REAL-LIFE EXAMPLE

Figure 2.4: Asymptotic length of 95% two-sided confidence interval of minimum long-run average cost per unit time.

to right. Furthermore, we can know that the distribution of the estimator of the minimum long-run average cost per unit time is relatively close to the normal distribution. On the other hand, it is seen that the estimator distribution of the optimal periodic replacement time is very far from the normal distribution, and has the longer tail on the right-hand side.

When the car owner uses the information of estimator distributions in real world, he or she can check that the probability that estimators of optimal periodic replacement time falls in the range from 37.69 months to 92.78 months is given by 95%. Therefore, the owner can take more flexible decision making to perform the preventive replacement within the 95% confidence interval, even if there is any reason why he or she cannot do the preventive maintenance at time 54.20 month (point estimate). Furthermore, the upper and lower limits of 95% confidence interval of minimum long-run average cost per unit time are 21.26($) and 35.45($), respectively. Then the car owner can know the information as reference and anticipate that it will cost at least 21.26($) to maintain car engine. Also he or she may think that it is possible to save money for the worst case
i.e., 35.45($). In real world, even if we know the optimal replacement time for maintenance, there are several situations in which it is difficult to perform the preventive replacement as scheduled. Similarly, we sometimes want to know the cost amount for replacement from the viewpoint of budget. For such situations, the proposed bootstrap methods will give the useful information to users.

![Cumulative failure numbers of failure time data under minimal repair.](image)

Figure 2.5: Cumulative failure numbers of failure time data under minimal repair.

<table>
<thead>
<tr>
<th></th>
<th>MLE</th>
<th>Mean</th>
<th>Median</th>
<th>V</th>
<th>S</th>
<th>K</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau^*$</td>
<td>54.20</td>
<td>56.26</td>
<td>53.36</td>
<td>202.99</td>
<td>1.582</td>
<td>7.222</td>
<td>[37.69, 92.78]</td>
</tr>
<tr>
<td>$C(\tau^*)$</td>
<td>28.93</td>
<td>29.00</td>
<td>29.20</td>
<td>13.22</td>
<td>-0.316</td>
<td>2.929</td>
<td>[21.26, 35.45]</td>
</tr>
</tbody>
</table>

Table 2.4: Estimation results in NHPP-BS2 case.
2.4. REAL-LIFE EXAMPLE

Figure 2.6: The probability distribution of estimator of optimal periodic replacement time (NHPP-BS2 case).

Figure 2.7: The probability distribution of estimator of minimum long-run average cost per unit time (NHPP-BS2 case).
Chapter 3

Kernel-Based Nonparametric Estimation Methods for a Periodic Replacement Problem with Minimal Repair

In this chapter, we consider nonparametric estimation methods for a periodic replacement problem (PRP) with minimal repair (MR), where the expected cumulative number of failures (minimal repairs) is unknown. To estimate the optimal periodic replacement time under incomplete knowledge on the failure time distribution, we apply two kernel-based bootstrap estimation methods and three replication techniques for bootstrap samples, and construct the confidence interval of an estimator of the optimal periodic replacement time which minimizes the long-run average cost per unit time. In simulation experiments, we compare those results with the well-known constrained nonparametric maximum likelihood estimate (CNPMLE) and some parametric models. We also conduct the field data analysis based on an actual minimal repair data, and refer to an applicability of our methods.
3.1 Periodic Replacement Problem with Minimal Repair

3.1.1 Model Description

In a repairable system, the system components are allowed to experience more than one failure throughout their life. After each failure, we have to perform a repair action to return the failed component state to the normal condition. Usually, such an activity restores only damaged part of the failure component back to a working condition that is only as good as it was just before the failure. This kind of repair is called minimal repair [4]. On the other hand, we plan to do a maintenance action called periodic replacement in advance and replace the used component by new one at pre-specified time. To model the behavior of cumulative number of minimal repairs, it is well known that a non-homogeneous Poisson process (NHPP) can be used [76].

The stochastic process \( \{N(t), t \geq 0\} \) which satisfies the following conditions is called NHPP:

- \( N(0) = 0 \),
- \( \{N(t), t \geq 0\} \) has independent increments,
- \( \Pr\{N(t + \Delta t) - N(t) \geq 2\} = o(\Delta t) \),
- \( \Pr\{N(t + \Delta t) - N(t) = 1\} = \lambda(t)\Delta t + o(\Delta t) \),

where \( o(\Delta t) \) is the higher term of \( \Delta t \). The probability mass function (p.m.f.) of the NHPP is given by

\[
\Pr\{N(t) = n\} = \frac{(\Lambda(t))^n}{n!} \exp\{-\Lambda(t)\}, \quad (3.1)
\]

\[
\Lambda(t) = \int_0^t \lambda(x) dx, \quad (3.2)
\]

where the function \( \Lambda(t) \) is called the intensity function and \( \Lambda(t) (= \text{E}[N(t)]) \) is called the mean value function which represents the expected cumulative number of minimal repairs.

Let \( c_1 (> 0) \) and \( c_2 (> 0) \) be the fixed costs of the periodic replacement and minimal repair, respectively. Then the long-run average cost per unit time is formulated by

\[
C(\tau) = \frac{c_1 \Lambda(\tau) + c_2}{\tau} = \frac{c_1 \int_0^\tau \lambda(t) dt + c_2}{\tau}, \quad (3.3)
\]
3.1. PERIODIC REPLACEMENT PROBLEM WITH MINIMAL REPAIR

where \( \tau(> 0) \) denotes the periodic replacement time (decision variable). Then, the problem is to derive \( \tau^* = \{ \tau > 0; \arg \min C(\tau) \} \). Differentiating Eq.(3.3) with respect to \( \tau \) and setting it to zero imply the first-order condition of optimality:

\[
\tau \lambda(\tau) \int_0^\tau \lambda(t)dt = \frac{c_2}{c_1}.
\] (3.4)

Hence if the intensity function \( \lambda(\tau) \) is unknown, it has to be estimated in either parametric or nonparametric way, according to the information on the intensity function, available from the past minimal repair record or history. If \( \lim_{\tau \to \infty} \tau \lambda(\tau) \int_0^\tau \lambda(t)dt > c_2/c_1 \) under the condition that the intensity function strictly increases with time, i.e., \( d\lambda(t)/dt > 0 \), then there exists a unique and finite optimal periodic replacement time \( \tau^* (0 < \tau^* < \infty) \) minimizing Eq.(3.3).

3.1.2 Parametric Estimation Method

In parametric method, we assume that the form of intensity function is completely known. Suppose that the failure-occurrence time data under minimal repair, which are the random variables, are given by \( 0 < T_1 \leq T_2 \leq \cdots \leq T_n \). That is, it is assumed that \( n \) failures (minimal repairs) occur by time \( t \) and the realizations of \( T_i \) (\( i = 1, 2, \ldots, n \)), say, \( t_i \) are observed, where \( t_n \leq t \). We regard the pair \((t_i, i)\) as a realization of the underlying NHPP. When the failure time distribution \( F(t) \) is given with unknown parameter (vector) \( \theta \), the mean value function of an NHPP is given by

\[
\Lambda(t; \theta) = \log \hat{F}(t; \theta).
\] (3.5)

The next step is to estimate the model parameter \( \theta \) from the count data \((t_i, i)\) (\( i = 1, 2, \ldots, n \)). The ML estimate \( \hat{\theta} \) is defined as the parameter which maximizes the following log likelihood function:

\[
\log L_{\text{NHPP}}(\theta | t_i) = \sum_{i=1}^n \log \lambda(t_i; \theta) \quad \Lambda(t_n; \theta).
\] (3.6)

By substituting the intensity function \( \hat{\lambda}(t; \theta) \) with the ML estimate \( \hat{\theta} \) into Eq.(3.4), we obtain the ML-based plug-in point estimates, \( \hat{\tau}^* \) and \( C(\hat{\tau}^*) \), of the optimal periodic replacement time \( \tau^* \) and its associated long-run average cost per unit time \( C(\tau^*) \), respectively.
Consider the case where the failure time distribution $F(t)$ and the intensity function $\lambda(t)$ are completely unknown. In this case, the common method of maximum likelihood does not work. In a fashion similar to the parametric case, suppose that the failure-occurrence (minimal repair) time data are given by random variables $0 < T_1 \leq T_2 \leq \cdots \leq T_n$ with realizations $0 < t_1 \leq t_2 \leq \cdots \leq t_n$. Without any loss of generality, we normalize the random variable $X_i = T_i/T_n \in (0, 1]$ with realizations $x_i = t_i/t_n \in (0, 1]$ for $i = 1, 2, \ldots, n$, dividing by the maximum length. If the number of failure time data is large enough, it may be appropriate to assume $x_n = 1$ as $t_n \approx \infty$. In this situation, the most intuitive but simplest method to estimate the intensity function of an NHPP is to use a piecewise-linear interpolation. For the $n$ minimal repair time data, $\chi = (x_i, i) (i = 1, 2, \ldots, n)$, define the following step-function estimate with breakpoints $x_i$ of the mean value function:

$$\hat{\Lambda}(x | \chi) = i + \frac{x - x_i}{x_{i+1} - x_i}, \quad x_i < x \leq x_{i+1}; \ i = 0, 1, \ldots, n_1. \quad (3.7)$$

By plotting $n$ failure points and connecting them by line segments, we can obtain the resulting estimate of the mean value function in Eq.(3.7). This is called the natural estimator because the mean squares error with the underlying minimal repair time data is always zero. Then, the associated natural estimate of intensity function is defined as each slope of mean value function by

$$\hat{\lambda}(x | \chi) = \frac{1}{x_i}, \quad x_i < x \leq x_{i+1}; \ x_0 = 0. \quad (3.8)$$

However, it is worth mentioning that the natural estimator in Eq.(3.8) does not work well for the generalization ability, because it can fit to only the past observation (training data) but cannot predict the unknown (future) pattern data. In addition, the natural estimator in Eq.(3.8) tends to fluctuate everywhere with big noise, and does not provide stable estimation results at all. Boswell [24] introduces the idea on isotonic estimation and gives a constrained nonparametric maximum likelihood estimate (CNPMLE). Suppose that the intensity function $\lambda(x)$ is non-decreasing with respect to time $x$, i.e. the mean value function is non-decreasing and convex in time. The likelihood function is
3.1. PERIODIC REPLACEMENT PROBLEM WITH MINIMAL REPAIR

defined as a function of unknown intensity functions at respective time points by

\[
L(\lambda(x_i), i = 1, 2, \ldots, n \mid \chi) = \exp\left\{ \int_0^{x_n} \lambda(x \mid \chi)dx \right\} \prod_{i=1}^{n} \lambda(x_i \mid \chi). \tag{3.9}
\]

Then the nonparametric maximum likelihood estimation with Eq.(3.9) is formulated as the following variational problem with respect to \(\lambda(\cdot)\):

\[
\arg \max_{\lambda(x_i), i=1,2,\ldots,n} L(\lambda(x_i), i = 1, 2, \ldots, n \mid \chi). \tag{3.10}
\]

Substituting Eq.(3.8) into Eq.(3.10), Boswell [24] solves it as a min-max solution and gives

\[
\hat{\lambda}(x_i \mid \chi) = \max_{1 \leq s \leq i} \min_{i \leq k \leq n} \frac{k}{s}, \tag{3.11}
\]

where \(x_0 = 0\) and \(x_n = 1\). It can be checked easily that Eq.(3.11) leads to an upper bound of \(L(\lambda(x_i), i = 1, 2, \ldots, n \mid \chi)\) with arbitrary points \(x_i (i = 1, 2, \ldots, n)\), by substituting Eq.(3.11) to Eq.(3.9).

The resulting estimator in Eq.(3.11) is somewhat smoother than Eq.(3.8) but is still discontinuous. An advantage for the CNPMLE is that the computation cost is quite low. The following is a simple algorithm to calculate the CNPMLE of an NHPP.

- Set \(h = 1\) and \(i_1 = 1\);
- Repeat until \(i_{h+1} = n\):
  - Set \(i_{h+1}\) to be the index \(i\) which minimizes the slopes between \((x_{i_h}, i_h + 1)\) and \((x_i, i - 1) (i = i_h + 1, \ldots, n)\);
  - The constrained nonparametric maximum likelihood estimate is then given by \(\hat{\lambda}(x \mid \chi) = (i_{j+1} - i_j)/(x_{i_{j+1}} - x_{i_j})\) whenever \(x_{i_j} < x \leq x_{i_{j+1}}\).

Since we assume that the intensity function is non-decreasing with respect to time, CNPMLE is regarded as a specific estimator which represents increasing intensity trend. Therefore, if the data has the increasing trend, then CNPMLE is expected to be useful. However, when the assumption on increasing intensity trend is violated, it may be less effective. The approach to estimate the optimal periodic replacement time with minimal repair by Gilardoni et al. [23] is based on the above CNPMLE. Next, we consider alternative nonparametric method to obtain nonparametric estimates of the optimal periodic replacement time.
3.1.4 Kernel-based Approach

Of our interest here is the derivation of absolutely continuous nonparametric estimators of the optimal periodic replacement time. For this purpose, we apply the kernel-based approach to estimate the intensity function of an NHPP. Define

\[ \hat{\lambda}(x | \chi) = \frac{1}{h} \sum_{i=1}^{n} K \left( \frac{x - x_i}{h} \right), \]  

(3.12)

where \( K(\cdot) \) denotes a kernel function and \( h \) is a positive constant, called smoothing parameter or bandwidth. It is well known that the choice of \( h \) is more sensitive rather than the choice of kernel function for the accuracy of \( \lambda(x) \). Therefore, we focus on only a well-known Gaussian kernel function:

\[ K(x) = \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{x^2}{2} \right). \]  

(3.13)

Roughly speaking, the kernel-based method approximates the intensity function with a superposition of kernel functions with location parameter at each minimal repair time. The main reason why absolutely continuous nonparametric estimator is preferred is that the natural estimator in Eq.(3.8) and the CNPMLE in Eq.(3.11) are functions of \( x_i \) (i = 1, 2, ..., n). This means that an estimate of the optimal periodic replacement time minimizing the long-run average cost per unit time has to be selected from the past minimal repair time. In other words, the approach by Gilardoni et al. [23] may work better for a sufficiently large number of minimal repair data, but will not in the other case.

In the kernel-based approach, we consider two estimation methods for bandwidth \( h \). The promising approach is the least-squares cross-validation (LSCV) method in Diggle and Marron [25] by minimizing the relevant integrated least squares error. On the other hand, Guan [28] determines the bandwidth by applying the log-likelihood cross-validation (LLCV) method. Suppose that the minimal repair time data are divided into training data and validation data. In the situation where \( n \) minimal repair time data are given, it is possible to make \( n \) training data sets with \( n - 1 \) failure data, by leaving out one of each \( i \) th \((i = 1, 2, ..., n)\) data from the original failure time data \( x_i \). The integrated least squares error of the intensity function \( \hat{\lambda}(x) \) in LSCV method is defined by

\[ \text{ISE}(h) = \int_0^1 \{ \hat{\lambda}(x | \chi) - \lambda(x | \chi) \}^2 dx. \]  

(3.14)
3.1. PERIODIC REPLACEMENT PROBLEM WITH MINIMAL REPAIR

After removing independent terms of $h$ in Eq.(3.14), it can be checked that the optimal bandwidth minimizing $\text{ISE}(h)$ is equal to $h$ minimizing the following function:

$$\text{CV}(h) = \int_0^1 \hat{\lambda}(x \mid \chi)^2 dx - 2 \sum_{j=1}^n \hat{\lambda}_{h,j}(x_i \mid \chi),$$

where

$$\hat{\lambda}_{h,j}(x \mid \chi) = \frac{1}{h} \sum_{i=1,i\neq j}^n K\left(\frac{x - x_i}{h}\right).$$

On the other hand, in LLCV method with the same $n$ training data sets, the problem is to maximize the log-likelihood function with unknown intensity function. Hence, it is reduced to obtain the optimal bandwidth $h$ maximizing

$$\ln L(h \mid \chi) = \sum_{k=1}^n \sum_{j=1}^n \ln \hat{\lambda}_{h,i}(x_k \mid \chi) \sum_{j=1}^n \hat{\lambda}_{h,i}(x_n \mid \chi),$$

where

$$\hat{\lambda}_{h,i}(x \mid \chi) = \int_0^x \hat{\lambda}_{h,i}(t \mid \chi) dt,$$

$$\hat{\lambda}_{h,i}(x \mid \chi) = \frac{1}{h} \sum_{i=1,i\neq j}^n K\left(\frac{x - x_i}{h}\right).$$

It is noted that the above kernel estimates of intensity function are absolutely continuous, but may fluctuate everywhere similar to the natural estimate. Nevertheless, the resulting estimates can be expected to be much smoother than that. Although we just focus on the Gaussian kernel function in this chapter, it is possible to apply several kinds of kernel function such as biweight kernel function and Epanechnikov kernel function [26]. An adaptive choice of bandwidth is another issue, i.e. $h = h(i) (i = 1, 2, \ldots n)$. However, we will not consider the improvement of kernel-based approach hereafter in this chapter, because our main concern here is to investigate an applicability of the kernel-based approach in the optimal periodic replacement problem with minimal repair. Substituting the resulting kernel estimates of intensity function into Eq.(3.4), the optimal periodic replacement time can be obtained.
3.2 Interval Estimation

3.2.1 Bootstrapping

Given a single set of failure (minimal repair) time data, some nonparametric methods can be applied to estimate the intensity function. A point estimate of the optimal periodic replacement time, $\hat{\tau}^*$, is obtained by minimizing the long-run average cost per unit time in Eq.(3.3) or solving the nonlinear equation in Eq.(3.4). The natural estimate and CNPMLE do not always guarantee the existence of the optimal solution satisfying Eq.(3.4), because the candidate of the estimate has to be chosen from the minimal repair data or its subset. In the kernel-based approach, on the other hand, the optimal periodic replacement time $\tau^*$ is estimated as an arbitrary real number if exits, or is given by $\tau^* = t_n \ (\sim \infty)$.

It is known that the point estimation does not take account of the uncertainty of estimator distribution. This is a serious problem, especially for nonparametric estimation. In the situation where only single time-series data on minimal repair process is given, the estimation must be done with only one sample data, and any serious estimation error may occur. Since the biased estimation cannot be avoided for single time-series analysis in principle, it is useful to obtain the confidence intervals of the optimal periodic replacement time and its minimum long-run average cost per unit time. Unfortunately, since these are very complex estimators, the estimator distributions in neither parametric nor nonparametric method are available analytically. In Chapter 2, we apply the bootstrap methods to construct the confidence regions of the optimal periodic replacement time and its associated long-run average cost per unit time. In the previous discussion, we assume a parametric form of intensity function and apply the ML method to estimate the model parameters. In both of parametric and nonparametric methods, three bootstrap (BS) methods proposed by Cowling et al. [2] can be applied. Although Gilardoni et al. [23] use the other replication techniques for BS samples under the CNPMLE, we focus on the above three methods by Cowling et al. [2] in the following:

(i) **BS Method 1 (Simulation-based approach):** Based on an arbitrary estimator of the intensity function, $\hat{\lambda}(x \mid \chi)$, which is estimated with $n$ min-
3.2. INTERVAL ESTIMATION

imal repair time data \( x_i \) \((i = 1, 2, \ldots, n)\), we generate the pseudo random time sequence \( X^*_k,i \) at \( k \)-th simulation. For the simulation of an NHPP, we use the well-known thinning algorithm [73] to generate the random variates. Note that \( X^*_k,i \) is always less than unity. More precisely, suppose that the intensity function is bounded from the above and takes the maximum value \( \lambda(0 \leq x \leq 1) \). Then, it can be shown that \( X^*_{k,1}, X^*_{k,2}, \ldots \) \((k = 1, 2, \ldots, m)\) follows an NHPP with intensity function \( \lambda \), and that the \( i \)-th minimal repair time \( X^*_{k,i} \) having intensity function \( \lambda(x_k,i | \chi) \) is independent of the other minimal repair times. The resulting BS samples, \( x^*_k,i \) and \( \hat{\lambda}_k(x | \chi) \) \((i = 1, 2, \ldots; k = 1, 2, \ldots, m)\), are called BS1 in this chapter, where \( \hat{\lambda}_k(x | \chi) \) denotes an estimator of intensity function based on \( x^*_k,i \).

(ii) BS Method 2 (Resampling-based approach): By resampling exactly \( n \) minimal repair time data with replacement from the underlying data \( x_i \) \((i = 1, 2, \ldots, n)\), we obtain another BS sample \( x^*_k,i \) and \( \hat{\lambda}^*_k(x | \chi) \) \((i = 1, 2, \ldots; k = 1, 2, \ldots, m)\).

(iii) BS Method 3 (Resampling-based approach): The third method is almost similar to BS2. Instead of \( n \) in BS2, we use the Poisson distributed (pseudo) random number with mean \( n \) as the number of resampling.

Let \( x^*_k,i \) denote the \( i \)-th pseudo normalized failure (minimal repair) time data (realization of \( X^*_k,i \)) at \( k \)-th \((k = 1, 2, \ldots, m)\) replication, where \( m \) is the number of replications. For \( m \) estimates of intensity function, we obtain \( m \) estimates of the optimal periodic replacement time by minimizing Eq.(3.3) or solving Eq.(3.4). Since these can be regarded as the i.i.d. BS sample of the optimal periodic replacement time \( \tilde{\tau}^*_k \) \((k = 1, 2, \ldots, m)\), it is possible to get the empirical distribution of the estimator of the optimal periodic replacement time and its finite moments. The case of the minimum long-run average cost per unit time can be considered in the similar way.

3.2.2 Construction of Confidence Region

Based on the empirical distribution of the estimator of the optimal periodic replacement time, we derive the confidence intervals of the optimal periodic
replacement time and its corresponding long-run average cost per unit time.

Let $\hat{\tau}_k^*$ be the $k$-th order statistics of the estimates $\hat{\tau}_k^*$ ($k = 1, 2, \ldots, m$), i.e. $0 = \hat{\tau}_{(0)}^* < \hat{\tau}_{(1)}^* \leq \cdots \leq \hat{\tau}_{(m)}^*$. For the long-run average cost per unit time, let $C_k(\hat{\tau}^*) (C_{(1)}(\hat{\tau}^*) \leq C_{(2)}(\hat{\tau}^*) \leq \cdots \leq C_{(m)}(\hat{\tau}^*))$ be the $k$-th estimate of the minimum long-run average cost $C(\hat{\tau}^*)$. From these, the BS-median and BS-mean can be derived as $\tilde{\tau}_{(m/2)}$ and $\bar{\tau}^* = \sum_{k=1}^{m} \hat{\tau}_k^*/m, \check{C}(\hat{\tau}^*) = \sum_{k=1}^{m} C_k(\hat{\tau}^*)/m$. The higher moments of the estimator of the optimal periodic replacement time are easily calculated. For instance, the variance $V_{\tau^*}$, skewness $S_{\tau^*}$ and kurtosis $K_{\tau^*}$ of the estimator of optimal periodic replacement time are also given by

\begin{align}
V_{\tau^*} &= \frac{\sum_{k=1}^{m} (\hat{\tau}_k^*)^2}{m}, \\
S_{\tau^*} &= \frac{\sum_{k=1}^{m} (\hat{\tau}_k^*)^3}{m V_{\tau^*}^{2/3}}, \\
K_{\tau^*} &= \frac{\sum_{k=1}^{m} (\hat{\tau}_k^*)^4}{m V_{\tau^*}^{2}}.
\end{align}

Finally, we calculate the two-sided $100(1-\alpha)\%$ confidence intervals by

$$
\left[ \hat{\tau}_{(m(\alpha/2))}, \hat{\tau}_{(m(1-\alpha/2))} \right], \quad \left[ C_{(m(\alpha/2))}(\hat{\tau}^*), C_{(m(1-\alpha/2))}(\hat{\tau}^*) \right]$$

with significance level $\alpha \in (0, 1)$. Since $m$ can be taken arbitrarily, it is not so difficult to take $m$ such that $m(\alpha/2)$ and $m(1-\alpha/2)$ become integers.

### 3.3 Intensity Estimation with Multiple Minimal Repair Data Sets

In the previous section, we discuss approaches for estimation of intensity function and construction of confidence region with single minimal repair data, since the main purpose of this chapter is to analyze the single field data of minimal repair. However, these approaches can be extended easily for multiple minimal repair data sets. Let $T_{j,i}$ ($j = 1, 2, \cdots, l; i = 1, 2, \cdots, n$) denote the random variable of $i$th failure-occurrence time data at $j$th Poisson process with realizations $t_{j,i}$, and let $T^* = \max_{1 \leq j \leq l} T_{j,n}$ with realization $t^*$. Then, the normalized random variable can be represented by $X_{j,i} = T_{j,i}/T^* \in (0, 1]$ with realizations $x_{j,i} = t_{j,i}/t^* \in (0, 1]$. Arrange the normalized failure-occurrence time data into a single ordered sample (0 =) $x_0 < \cdots < x_f (= 1)$. 
3.3. INTENSITY ESTIMATION WITH MULTIPLE MINIMAL REPAIR DATA SETS

Following the idea by Zielinski et al. [77], the CNPMLE with multiple minimal data sets are obtained by

\[
\hat{\lambda}(x) = \begin{cases} 
0, & 0 \leq x < x_1, \\
\hat{\lambda}_i, & x_i \leq x < x_{i+1}; \ i = 1, 2, \ldots, f \end{cases} 
\]  
(3.24)

where

\[
\hat{\lambda}_i = \max_{1 \leq s \leq i} \min_{i \leq k \leq f} \frac{\sum_{j=s}^{k} M_j}{\sum_{j=s}^{k} \Delta_j}, 
\]  
(3.25)

\[
\Delta_i = \sum_{j=1}^{i} \max \left\{0, \min \{x_n, x_i\}, (x_{i+1} - x_i)\right\} 
\]  
(3.26)

and \(m_i\) means the number of failures occurred at time \(x_i\).

For the kernel intensity estimation, define

\[
\hat{\lambda}_j(x \mid \chi) = \frac{1}{h} \sum_{j=1}^{n} K \left(\frac{x - x_{j,i}}{h}\right), 
\]  
(3.27)

where \(K(x)\) is already defined in Eq.(3.13). Then the CV\((h)\) in Eq.(3.15) of LSCV method can be rewritten as

\[
CV(h) = \sum_{j=1}^{l} \left(\int_{0}^{1} \hat{\lambda}_j(x \mid \chi)^2 dx \right)^{2} \frac{\sum_{p=1}^{n} \hat{\lambda}_{j,h,p}(x_{j,i} \mid \chi)}{2 \sum_{p=1}^{n} \hat{\lambda}_{j,h,p}(x_{j,n} \mid \chi)}, 
\]  
(3.28)

where

\[
\hat{\lambda}_{j,h,p}(x \mid \chi) = \frac{1}{h} \sum_{i=1, i \neq p}^{n} K \left(\frac{x - x_{j,p}}{h}\right). 
\]  
(3.29)

Furthermore, the \(\ln L(h \mid \chi)\) in Eq.(3.17) can be also rewritten as

\[
\ln L(h \mid \chi) = \sum_{j=1}^{l} \sum_{k=1}^{n} \sum_{p=1}^{n} \ln \hat{\lambda}_{j,h,i}(x_{j,k} \mid \chi) + \sum_{p=1}^{n} \hat{\lambda}_{j,h,i}(x_{j,n} \mid \chi), 
\]  
(3.30)

where

\[
\hat{\lambda}_{j,h,i}(x \mid \chi) = \int_{0}^{x} \hat{\lambda}_{j,h,i}(t \mid \chi) dt, 
\]  
(3.31)

\[
\hat{\lambda}_{j,h,i}(x \mid \chi) = \frac{1}{h} \sum_{i=1, i \neq p}^{n} K \left(\frac{x - x_{j,i}}{h}\right). 
\]  
(3.32)

After estimating the optimal bandwidth with Eq.(3.28) or Eq.(3.30), the kernel intensity function with multiple minimal repair data sets is obtained by [78]

\[
\hat{\lambda}(x \mid \chi) = \frac{1}{h} \sum_{j=1}^{l} \sum_{i=1}^{n} K \left(\frac{x - x_{j,i}}{h}\right) \frac{1}{\sum_{j=1}^{l} I(x \leq x_{j,n})}, 
\]  
(3.33)
where \( I(x \leq x_{j,n}) (j = 1, 2, \cdots, l) \) means indicator function. Based on the above methods, the estimators of the intensity function can be obtained for the multiple data sets case. Once the estimates of intensity function are obtained, we can get the optimal periodic replacement time with Eq.(3.4). Therefore, it is possible to handle the several sets of minimal repair data and construct the confidence region in the similar way.

3.4 Simulation Experiments

To investigate the accuracy of our methods, Monte Carlo simulation is conducted. Here, we also focus on the single minimal repair data, since the single field data of minimal repair is analyzed in the latter part. Suppose that the (unknown) minimal repair process follows a power law NHPP model, \( \lambda(t) = (\beta/\eta)(t/\eta)^{\beta - 1} \), with model parameters \((\beta, \eta) = (3.2, 0.23)\). We generate the original failure (minimal repair) time data as the pseudo random number by the thinning algorithm in Subsection 3.1.3. The cost parameters are fixed as \( c_1 = 1, c_2 = 30 \). In this case the optimal periodic replacement time and its minimum long-run average cost are given by \( \tau^* = 0.52 \) and \( C(\tau^*) = 83.9 \), respectively.

3.4.1 Point Estimation

For \( n \) minimal repair time data, we estimate the underlying NHPP in both of parametric and nonparametric methods and estimate the optimal periodic replacement time and its minimum long-run average cost per unit time. In the parametric approach, we assume two parametric models; Power law (PL) NHPP model, \( \lambda(t) = (\beta/\eta)(t/\eta)^{\beta - 1} \) and Cox-Lewis (CL) NHPP model, \( \lambda(t) = \exp\{\alpha + \beta t\} \). Since \( n \) may not be sufficiently large, it is almost impossible to select the correct model (power law model) by mean of the ML estimation. In the nonparametric approach, we compare the kernel-based approaches having LLCV and LSCV with the CNPMLE.

Tables 3.1-3.2 summarize the point estimation results of the optimal periodic replacement time and the minimum long-run average cost per unit time, where \( n \) represents the number of minimal repairs. For varying \( n \) from 10 to 190 by
Table 3.1: Point estimates of the optimal periodic replacement time.

<table>
<thead>
<tr>
<th>$n$</th>
<th>LLCV</th>
<th>LSCV</th>
<th>CNPMLE</th>
<th>PL</th>
<th>CL</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.52</td>
<td>0.56</td>
</tr>
<tr>
<td>120</td>
<td>0.50</td>
<td>0.51</td>
<td>0.50</td>
<td>0.52</td>
<td>0.58</td>
</tr>
<tr>
<td>180</td>
<td>0.50</td>
<td>0.51</td>
<td>0.50</td>
<td>0.52</td>
<td>0.59</td>
</tr>
</tbody>
</table>

Table 3.2: Point estimates of the minimum long-run average cost per unit time.

<table>
<thead>
<tr>
<th>$n$</th>
<th>LLCV</th>
<th>LSCV</th>
<th>CNPMLE</th>
<th>PL</th>
<th>CL</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>83.6</td>
<td>89.5</td>
<td>83.3</td>
<td>85.4</td>
<td>85.7</td>
</tr>
<tr>
<td>120</td>
<td>83.5</td>
<td>86.2</td>
<td>83.3</td>
<td>85.3</td>
<td>88.9</td>
</tr>
<tr>
<td>180</td>
<td>83.4</td>
<td>87.8</td>
<td>83.3</td>
<td>85.5</td>
<td>93.2</td>
</tr>
</tbody>
</table>

10, we examine the above estimates. Though we omit to show all the results for brevity, it can be seen that the miss-selection of parametric model (Cox-Lewis model) leads to the worst result, but the other methods can provide very close results to the real optimal solution $\tau^* = 0.52$. On the other hand, looking at Table 3.2, the nonparametric model with LSCV and the parametric models fail to get the good estimates of the minimum long-run average cost per unit time. It is evident that two nonparametric methods, kernel-based approach with LLCV and CNPMLE, show the similar estimation results. The reason why the estimation result of the long-run average cost per unit time by the power law model is biased is as follows: The minimum long-run average cost per unit time is very sensitive to the optimal periodic replacement time $\tau^*$. For $n = 60, 120, 180$, more precise estimates by the power law model are $\hat{\tau}^* = 0.5156, 0.5197, 0.5191$, while the real optimal solution is 0.52. These differences lead to the deviation from $C(\tau^*) = 83.9$. It can be also seen that Cox-Lewis model tends to make the larger difference from the real optimal periodic replacement time as the number of minimal repair time data increases. Next, the question arises; why does CNPMLE work well similar to the kernel-based approach with LLCV? The answer for this question is that the original minimal repair time data fortunately includes very closed data 0.50 to the optimal solution $\tau^* = 0.52$. If this is not the case even with $n = 60$, CNPMLE cannot provide such a nice estimate from all
the minimal repair time points. Finally, from the point estimation results, the kernel-based approach with LLCV and possibly CNPMLE can be recommended to estimate the optimal periodic replacement time.

In order to show counter examples for the CNPMLE, we estimate the long-run average cost per unit time at arbitrary periodic replacement time. Apart from the optimization, it seems to be interesting to examine the estimation accuracy of respective methods at different periodic replacement candidates. Table 3.3 presents the estimation results of long-run average cost per unit time when we perform periodic replacement at arbitrary time \( x \) in the case of \( n = 180 \). In the table, "True" means the estimate based on the power law model with model parameters \((\beta, \eta) = (3.2, 0.23)\). We calculate the estimates of long-run average cost per unit time from \( x = 0.05 \) to \( x = 1.00 \) by 0.05. If the estimation result is close to the theoretical value, we can say that the estimator has the good accuracy performance at different periodic replacement times. The mean squares errors between "True" and the other methods are calculated in all 20 points from \( x = 0.05 \) to \( x = 1.00 \). From this result, it is seen that the kernel-based approach with LLCV and the natural estimator have the similar good accuracy performance. On the other hand, the kernel-based approach with LSCV and CNPMLE tend to give different estimates, compared with above two methods.

Table 3.3: Estimation accuracy of respective methods for arbitrary periodic replacement time \( x \).

<table>
<thead>
<tr>
<th>( x )</th>
<th>True</th>
<th>LLCV</th>
<th>LSCV</th>
<th>CNPMLE</th>
<th>PL</th>
<th>CL</th>
<th>Natural</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>109.4</td>
<td>108.7</td>
<td>107.0</td>
<td>102.5</td>
<td>110.4</td>
<td>126.1</td>
<td>108.8</td>
</tr>
<tr>
<td>0.50</td>
<td>85.3</td>
<td>88.3</td>
<td>90.2</td>
<td>86.2</td>
<td>87.0</td>
<td>93.2</td>
<td>88.6</td>
</tr>
<tr>
<td>0.75</td>
<td>117.9</td>
<td>118.9</td>
<td>118.9</td>
<td>114.5</td>
<td>117.9</td>
<td>112.4</td>
<td>119.5</td>
</tr>
<tr>
<td>1.00</td>
<td>183.5</td>
<td>177.9</td>
<td>170.6</td>
<td>177.5</td>
<td>178.3</td>
<td>178.3</td>
<td>178.3</td>
</tr>
<tr>
<td>MSE</td>
<td>0.0</td>
<td>13.4</td>
<td>19.4</td>
<td>17.8</td>
<td>3.53</td>
<td>140.8</td>
<td>14.1</td>
</tr>
</tbody>
</table>
3.4.2 Statistical Properties of Estimators

Based on three replication techniques for BS samples, we generate \( m = 2,000 \) BS samples and obtain \( m \) estimates of \( \hat{\tau}^* \) and \( C(\hat{\tau}^*) \). Tables 3.4-3.5 present the BS estimation results of the optimal periodic replacement time and its corresponding minimum long-run average cost per unit time when the number of minimal repair data is 180. From Table 3.4, it can be shown that the kernel-based estimation with LLCV gives closest values to the real optimal solution with respect to the BS-mean and BS-median in nonparametric cases. Also it can be checked that the CNPMLE gives the almost similar results with LLCV. Dissimilar to the point estimation, the kernel-based approach with LSCV gives relatively nice BS-median and BS-mean with small variance. By checking the skewness (\( \approx 0 \)) and kurtosis (\( \approx 3 \)), we can verify the resulting estimator distributions can be approximated by the normal distributions. Especially, when BS2 is used in nonparametric estimation, the estimator distributions of the minimum long-run average cost per unit time can be approximated by the normal distributions.

In parametric models, it is worth noting that the BS-mean and BS-median with the power law model can give the minimum variance on the optimal periodic replacement time. By averaging the estimation error, these statistics approach to the real (but unknown) optimal solutions with \( n = 180 \) and \( m = 2,000 \). It is also found that the miss-selection of parametric model (Cox-Lewis model) also gives small variance on the estimator of the optimal periodic replacement time, but the deviation of the BS-mean and BS-median from the real optimal solution is rather large. From the view point of point estimation, it can be concluded that the kernel-based approach with LLCV shows the accurate estimation results with smaller variance on the BS-mean and BS-median. Although these results in Tables 3.4-3.5 are based on \( n = 180 \), even when the number of underlying minimal repair data increases, the BS estimates (BS-mean and BS-median) can give close values to the real optimal solutions in all nonparametric cases. In parametric model, the power law model gives the minimum variances in all cases, but the BS-mean and BS-median in this model tend to overestimate the minimum long-run average cost per unit time. It is evident to know the Cox-Lewis model fails to obtain the accurate estimates.
Table 3.4: Statistics of estimators of the optimal periodic replacement time.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Method</th>
<th>Mean</th>
<th>Median</th>
<th>$V_{1-e}$</th>
<th>$S_{1-e}$</th>
<th>$K_{1-e}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLCV</td>
<td>BS1</td>
<td>0.52</td>
<td>0.51</td>
<td>0.005</td>
<td>-0.581</td>
<td>3.664</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>0.52</td>
<td>0.51</td>
<td>0.004</td>
<td>-0.709</td>
<td>3.884</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>0.52</td>
<td>0.51</td>
<td>0.005</td>
<td>-0.630</td>
<td>3.692</td>
</tr>
<tr>
<td>LSCV</td>
<td>BS1</td>
<td>0.51</td>
<td>0.50</td>
<td>0.002</td>
<td>0.343</td>
<td>3.694</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>0.51</td>
<td>0.51</td>
<td>0.002</td>
<td>-0.117</td>
<td>4.451</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>0.51</td>
<td>0.51</td>
<td>0.002</td>
<td>-0.072</td>
<td>5.236</td>
</tr>
<tr>
<td>CNPMLE</td>
<td>BS1</td>
<td>0.50</td>
<td>0.51</td>
<td>0.005</td>
<td>-0.123</td>
<td>2.615</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>0.52</td>
<td>0.51</td>
<td>0.005</td>
<td>-0.735</td>
<td>3.805</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>0.52</td>
<td>0.51</td>
<td>0.005</td>
<td>-0.639</td>
<td>3.743</td>
</tr>
<tr>
<td>PL</td>
<td>BS1</td>
<td>0.52</td>
<td>0.52</td>
<td>0.000</td>
<td>0.116</td>
<td>2.954</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>0.52</td>
<td>0.52</td>
<td>0.000</td>
<td>0.401</td>
<td>3.133</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>0.52</td>
<td>0.52</td>
<td>0.000</td>
<td>0.169</td>
<td>3.065</td>
</tr>
<tr>
<td>CL</td>
<td>BS1</td>
<td>0.59</td>
<td>0.59</td>
<td>0.000</td>
<td>0.288</td>
<td>3.121</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>0.59</td>
<td>0.59</td>
<td>0.000</td>
<td>0.871</td>
<td>3.927</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>0.59</td>
<td>0.59</td>
<td>0.000</td>
<td>0.124</td>
<td>2.986</td>
</tr>
</tbody>
</table>
### Table 3.5: Statistics of estimators of the minimum long-run average cost per unit time.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Method</th>
<th>Mean</th>
<th>Median</th>
<th>( V_{C*} )</th>
<th>( S_{C*} )</th>
<th>( K_{C*} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLCV</td>
<td>BS1</td>
<td>81.1</td>
<td>81.1</td>
<td>38.55</td>
<td>-0.026</td>
<td>2.665</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>81.4</td>
<td>81.4</td>
<td>34.99</td>
<td>0.026</td>
<td>2.870</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>81.2</td>
<td>81.3</td>
<td>38.77</td>
<td>0.024</td>
<td>2.739</td>
</tr>
<tr>
<td>LSCV</td>
<td>BS1</td>
<td>88.2</td>
<td>87.9</td>
<td>44.93</td>
<td>0.276</td>
<td>3.077</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>87.2</td>
<td>86.8</td>
<td>38.31</td>
<td>0.179</td>
<td>2.952</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>87.0</td>
<td>86.8</td>
<td>43.55</td>
<td>0.192</td>
<td>2.949</td>
</tr>
<tr>
<td>CNPMLE</td>
<td>BS1</td>
<td>79.1</td>
<td>79.1</td>
<td>33.08</td>
<td>0.009</td>
<td>2.785</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>81.1</td>
<td>81.3</td>
<td>33.82</td>
<td>0.027</td>
<td>2.942</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>81.0</td>
<td>80.9</td>
<td>37.80</td>
<td>0.032</td>
<td>2.798</td>
</tr>
<tr>
<td>PL</td>
<td>BS1</td>
<td>85.2</td>
<td>85.1</td>
<td>29.60</td>
<td>0.202</td>
<td>2.994</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>85.6</td>
<td>85.4</td>
<td>24.69</td>
<td>0.145</td>
<td>2.992</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>85.4</td>
<td>85.2</td>
<td>29.08</td>
<td>0.179</td>
<td>3.027</td>
</tr>
<tr>
<td>CL</td>
<td>BS1</td>
<td>93.0</td>
<td>93.0</td>
<td>48.66</td>
<td>0.146</td>
<td>2.925</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>93.2</td>
<td>93.0</td>
<td>30.36</td>
<td>0.134</td>
<td>2.994</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>93.0</td>
<td>92.7</td>
<td>40.30</td>
<td>0.204</td>
<td>3.030</td>
</tr>
</tbody>
</table>
3.4.3 Interval Estimation

We derive the two-sided 95% confidence intervals of the optimal periodic replacement time and the minimum long-run average cost per unit time with parametric and nonparametric methods. Figures 3.1-3.3 show the two-sided 95% confidence intervals of optimal periodic replacement time in the case of \( n = 180 \) via three bootstrap methods, where the length of each box plot denotes the two-sided 50% confidence intervals. First we can find that the confidence intervals for nonparametric methods are longer than those for parametric models. This is because the nonparametric methods are subject to much more uncertainty. Among nonparametric approaches, the kernel-based approach with LSCV gives tighter confidence regions. Second, it is seen that the BS-median for LLCV with BS1, BS2, BS3 and for CNPMLE with BS2 and BS3 are included within the box range, but are almost lower levels in 50% confidence intervals. Furthermore, when BS2 is used in both nonparametric and parametric methods, it tends to give tighter confidence region than BS1 and BS3.

![Figure 3.1: Estimation results with respective methods via BS 1.](image)

Figures 3.4-3.8 illustrate asymptotic behaviors of the two-sided 95% confidence regions using kernel-based approach with LLCV and LSCV, CNPMLE and two parametric models, where only BS1 is used for BS replication. If the
3.4. SIMULATION EXPERIMENTS

Figure 3.2: Estimation results with respective methods via BS 2.

Figure 3.3: Estimation results with respective methods via BS 3.
ML estimation works well, the confidence intervals in Fig. 3.7 can be considered as an actual confidence region or be rather closed. From this result, if the number of underlying minimal repair time data is greater than 60, the confidence interval is very stable and keeps its width within an almost constant range. It can be shown in Fig. 3.8 that the miss-selection of parametric model leads to incorrect confidence region which does not involve the optimal solution. In three nonparametric estimation methods, respective confidence intervals have wider regions than the parametric models. As mentioned above, this is because the degree of freedom in nonparametric models is larger than that in parametric models, so that the estimation under incomplete knowledge on the minimal repair process faces more uncertain situation.

The confidence intervals based on kernel-based estimation with LLCV and CNPMLE tend to be stable even if the number of original data is rather small. On the other hand, when we apply the kernel-based estimation with LSCV, the resulting interval estimation involves fluctuated noise compared with other two nonparametric methods. In the case with small sample size, LSCV does not work to get the accurate confidence regions, and fails to obtain the solutions satisfying Eq.(3.4). But, as the number of minimal repair time data becomes greater than 70, the BS-mean and BS-median in the kernel-based approach with LSCV gives closer value to the real optimal periodic replacement time. Finally, from both points of view in point estimation and interval estimation, we recommend to apply the kernel-based estimation with LLCV.

3.5 Field Data Analysis

We give a simple example of the field data analysis. Let $c_1 = 1$ and $c_2 = 30$ be the minimal repair cost and the periodic replacement cost, respectively. Meeker and Escobar [75] report the minimal repair data of a diesel engine where the number of failures (minimal repairs) $n = 71$ and MTBF= 0.359hr if these data are regarded as i.i.d. samples. In accordance with our recommendation through simulation experiments, we apply the kernel-based estimation with LLCV and the CNPMLE with $m = 2,000$ bootstrap sample. Tables 3.6-3.7 present the estimation results of the optimal periodic replacement time and its corresponding long-run average cost per unit time. In both cases, both BS-mean and
3.5. FIELD DATA ANALYSIS

Figure 3.4: Asymptotic behavior of the estimate on the optimal periodic replacement time with LLCV.

Figure 3.5: Asymptotic behavior of the estimate on the optimal periodic replacement time with LSCV.
Figure 3.6: Asymptotic behavior of the estimate on the optimal periodic replacement time with CNPMLE.

Figure 3.7: Asymptotic behavior of the estimate on the optimal periodic replacement time with Power law model.
Figure 3.8: Asymptotic behavior of the estimate on the optimal periodic replacement time with Cox-Lewis model.

BS-median take closed values from each other. In the optimal periodic replacement time estimation, BS3 in the kernel estimation with LLCV and BS2 in the CNPMLE provide the tightest confidence intervals. On the other hand, in the minimum long-run average cost per unit time estimation, BS2 in the kernel-based method with LLCV gives the tightest bound. Figure 3.9 depicts the estimator distribution of the minimum long-run average cost per unit time with LLCV in the case of BS2. From this result, we can know that both BS-mean and BS-median take closed values to the point estimation with original minimal repair time data, where the estimator distribution of the minimum long-run average cost per unit time can be approximated by a normal distribution.
CHAPTER 3. NONPARAMETRIC APPROACH FOR PRP WITH MR

Figure 3.9: Estimator distribution of the minimum long-run average cost per unit time.

Table 3.6: Real data analysis of the optimal periodic replacement time.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Method</th>
<th>Mean</th>
<th>Median</th>
<th>$V_{\tau}$</th>
<th>$S_{\tau}$</th>
<th>$K_{\tau}$</th>
<th>95%CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLCV</td>
<td>BS1</td>
<td>18.7</td>
<td>19.0</td>
<td>0.78</td>
<td>-0.981</td>
<td>3.751</td>
<td>[16.4, 19.9]</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>18.7</td>
<td>19.0</td>
<td>0.73</td>
<td>-1.084</td>
<td>3.900</td>
<td>[16.4, 19.9]</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>18.7</td>
<td>19.0</td>
<td>0.72</td>
<td>-0.974</td>
<td>3.856</td>
<td>[17.3, 19.9]</td>
</tr>
<tr>
<td>CNPMLE</td>
<td>BS1</td>
<td>18.8</td>
<td>19.0</td>
<td>0.79</td>
<td>-0.725</td>
<td>4.520</td>
<td>[16.8, 20.4]</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>18.8</td>
<td>19.1</td>
<td>0.75</td>
<td>-1.153</td>
<td>4.274</td>
<td>[16.5, 19.9]</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>18.8</td>
<td>19.1</td>
<td>0.83</td>
<td>-0.934</td>
<td>3.910</td>
<td>[16.5, 20.1]</td>
</tr>
</tbody>
</table>
3.5. FIELD DATA ANALYSIS

Table 3.7: Real data analysis of the minimum long-run average cost per unit time.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Method</th>
<th>Mean</th>
<th>Median</th>
<th>$V_C^*$</th>
<th>$S_C^*$</th>
<th>$K_C^*$</th>
<th>95%CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLCV</td>
<td>BS1</td>
<td>2.46</td>
<td>2.43</td>
<td>0.05</td>
<td>0.302</td>
<td>3.000</td>
<td>[2.07, 2.90]</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>2.45</td>
<td>2.44</td>
<td>0.03</td>
<td>0.118</td>
<td>3.096</td>
<td>[2.11, 2.80]</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>2.44</td>
<td>2.42</td>
<td>0.04</td>
<td>0.246</td>
<td>3.145</td>
<td>[2.04, 2.89]</td>
</tr>
<tr>
<td>CNPML</td>
<td>BS1</td>
<td>2.41</td>
<td>2.40</td>
<td>0.04</td>
<td>0.328</td>
<td>3.271</td>
<td>[2.03, 2.85]</td>
</tr>
<tr>
<td></td>
<td>BS2</td>
<td>2.44</td>
<td>2.45</td>
<td>0.04</td>
<td>0.073</td>
<td>2.976</td>
<td>[2.09, 2.82]</td>
</tr>
<tr>
<td></td>
<td>BS3</td>
<td>2.44</td>
<td>2.44</td>
<td>0.04</td>
<td>0.349</td>
<td>3.283</td>
<td>[2.05, 2.88]</td>
</tr>
</tbody>
</table>
Chapter 4

Software Reliability Assessment via Nonparametric Maximum Likelihood Estimation

In this chapter we consider three nonparametric estimation methods for software reliability assessment without specifying the fault-detection time distribution, where the underlying stochastic process to describe software fault-counts in the system testing is given by a non-homogeneous Poisson process. Two of these are based on the fault detection time data, which are introduced in Section 4.1 and 4.2. On the other hand, one of these is based on the software fault count data, which is argued in Section 4.3. The resulting data-driven methodologies can give the useful probabilistic information on the software reliability assessment under the incomplete knowledge on fault-detection time distribution. Throughout examples with real software fault data, it is shown that the proposed methods provide more accurate estimation results than the common parametric approach.

4.1 Software Reliability Modeling

4.1.1 NHPP-based SRMs

Let \( N(t) \) be the cumulative number of software faults detected by time \( t \). Suppose that the number of remaining software faults before software testing is
CHAPTER 4. NPMLES FOR NHPP-BASED SRMS

given by \( N_0 (\geq 0) \) and that the fault-detection time of each software fault is independent and identically distributed continuous random variable having the probability distribution function \( F(t; \boldsymbol{\theta}) \) with parameter (vector) \( \boldsymbol{\theta} \). Then the probability that \( n \) software faults are detected by time \( t \) is given by the binomial distribution:

\[
\Pr \{ N(t) = n \} = \binom{N_0}{n} F(t; \boldsymbol{\theta})^n (1 - F(t; \boldsymbol{\theta}))^{N_0 - n}.
\]  

(4.1)

Furthermore, if the initial number of software faults remaining in the software, \( N_0 \), follows the Poisson distribution with mean \( \omega (> 0) \), then the probability that the total number of software faults detected by time \( t \) equals \( n \) is given by

\[
\Pr \{ N(t) = n \} = \frac{\omega F(t; \boldsymbol{\theta})^n}{n!} \exp \{ -\omega F(t; \boldsymbol{\theta}) \}.
\]

(4.2)

It corresponds to the probability mass function of a non-homogeneous Poisson process (NHPP) with mean value function \( \Lambda(t; \xi) = \omega F(t; \boldsymbol{\theta}) \), where \( \xi = (\omega, \boldsymbol{\theta}) \). Hence, the NHPP-based software reliability model (SRM) can be characterized by the fault-detection time distribution \( F(t; \boldsymbol{\theta}) \). Define the failure rate of the probability distribution function \( F(t; \boldsymbol{\theta}) \) by

\[
r(t; \boldsymbol{\theta}) = \frac{f(t; \boldsymbol{\theta})}{1 - F(t; \boldsymbol{\theta})},
\]

(4.3)

if the probability density function \( f(t; \boldsymbol{\theta}) = dF(t; \boldsymbol{\theta})/dt \) exists. From Eq.(4.3), it is easily seen that

\[
f(t; \boldsymbol{\theta}) = r(t; \boldsymbol{\theta}) \exp \left\{ \int_0^t r(x; \boldsymbol{\theta}) \, dx \right\},
\]

(4.4)

so that the intensity function of an NHPP, \( d\Lambda(t; \xi)/dt = \lambda(t; \xi) = \omega f(t; \boldsymbol{\theta}) \), is represented by

\[
\lambda(t; \xi) = \omega r(t; \boldsymbol{\theta}) \exp \left\{ \int_0^t r(x; \boldsymbol{\theta}) \, dx \right\}.
\]

(4.5)

### 4.1.2 Maximum Likelihood Estimation

The commonly used technique to estimate unknown parameters \( \xi = (\omega, \boldsymbol{\theta}) \) is the maximum likelihood (ML) estimation. Suppose that \( n \) (> 0) software fault-detection time data, \( t_i \) (\( i = 1, 2, \ldots, n \)), are observed during the testing phase.
t \in (0, T], where T (\geq t_n) is the observation point. Then the likelihood function of an NHPP is given by

\[ LF(t_1, \ldots, t_n, T) = \exp \left\{ \Lambda(T; \xi) \right\} \prod_{i=1}^{n} \lambda(t_i; \xi). \] (4.6)

Taking the natural logarithm of both sides of Eq.(4.6), we have the log-likelihood function:

\[ LLF(\xi | t_1, \ldots, t_n, T) = \sum_{i=1}^{n} \log \lambda(t_i; \xi) \Lambda(T; \xi). \] (4.7)

Then, the maximum likelihood estimates of parameters, \( \hat{\xi} \), is defined by

\[ LLF(\xi | t_1, \ldots, t_n, T) = \arg \max_{\xi} LLF(\xi | t_1, \ldots, t_n, T). \] (4.8)

Since the above log-likelihood function is a multi-modal nonlinear function for an arbitrary intensity function \( \lambda(t; \xi) \), it is known that the common nonlinear optimization algorithm such as Newton method does not often work well to obtain the global optimization solution. Then, the EM (Expectation-Maximization) algorithm [37] with a global convergence property can be applied to obtain the ML estimate \( \hat{\xi} \).

4.2 Nonparametric Estimation

4.2.1 Nonparametric ML Estimation

As mentioned in Section 4.1, it is worth noting that the best parametric NHPP-based SRM to fit every type of software-fault detection time data does not exist. This means that a careful selection of the software fault-detection time distribution \( F(t; \theta) \) is definitely needed to obtain the best fitted NHPP-based SRM to the underlying data. Okamura and Dohi [37] suggest that it is enough to consider eleven typical software fault-detection time distributions, which include exponential [32], gamma [38],[39], Pareto [46],[47], truncated normal [48], log-normal [31],[48], truncated logistics [35], log-logistics [34], truncated extreme-value maximum [36], log-extreme-value maximum [36], truncated extreme-value minimum [36], log-extreme-value minimum (Weibull) [33],[36] distributions. However, it cannot be guaranteed that one of the above eleven NHPP-based SRMs always outperforms the other models in the literature [29],[30]. In such an uncertain situation where the best mean value function or equivalently the best
fault-detection probability is unknown, it makes sense to consider nonparametric estimation of the NHPP-based SRM.

First, we introduce a nonparametric ML estimation developed by Boswell [24], where \( \Lambda(t; \xi) = \omega F(t; \theta) \) is not assumed at the moment. So we do not restrict the class of NHPP-based SRMs such as \( \Lambda(t; \xi) = \omega F(t; \theta) \). Suppose that the intensity function \( \lambda(t) \) is a non-negative monotone non-increasing function of \( t \in (0, T] \). For unknown intensity function \( \lambda(t) \), Boswell [24] considers the following variational problem:

\[
\arg \max_{\lambda(t)} \sum_{i=1}^{n} \log \lambda(t_i) \int_{0}^{T} \lambda(x)dx.
\]

(4.9)

Note that an increase in the above log-likelihood function can occur only by an increase in \( \lambda(\cdot) \) at the fault-detection time points. Boswell [24] proves that a solution in Eq.(4.9) must consist of step functions closed on the right with no jumps except at some of the event occurrence points. This fact leads to the idea on so-called isotonic estimation, and gives a constrained nonparametric maximum likelihood estimate (CNPMLE). For \( t_1 < t_2 < \ldots < t_n \leq T \), define the likelihood function as a function of unknown intensity function at each fault-detection time:

\[
LF(\lambda(t_i), i = 1, 2, \ldots, n) = \exp \left\{ \int_{0}^{T} \lambda(x)dx \prod_{i=1}^{n} \lambda(t_i) \right\}.
\]

(4.10)

Hence the problem can be reduced to the following finite dimensional problem:

\[
\arg \max_{\lambda(t_i, i = 1, 2, \ldots, n)} LF(\lambda(t_i), i = 1, 2, \ldots, n).
\]

(4.11)

Boswell [24] obtains the following min-max solution:

\[
\hat{\lambda}(t_j) = \min_{1 \leq j \leq n} \max_{k \leq j \leq n} \frac{k}{t_k} \frac{h}{t_h}.
\]

(4.12)

The resulting estimator is discontinuous, but constructs an upper bound of \( LF(\lambda(t_i), i = 1, 2, \ldots, n) \). Although we assume that the intensity function is non-increasing to represent the reliability growth phenomenon, this assumption can be relaxed. That is, if the intensity function is non-decreasing, then the resulting CNPMLE can be given by replacing min operator and max operator in Eq.(4.12) from each other. The CNPMLE has also an advantage that the
4.2. NONPARAMETRIC ESTIMATION

Computation cost is quite low, compared with the other representative nonparametric methods \cite{42,44,41,45,40,43}. We give a simple optimization code to derive the CNPMLE as follows.

- Set $h = 1$ and $i_1 = 1$;
- Repeat until $i_{h+1} = n$:
  - Set $i_{h+1}$ to be the index $i$ which minimizes the slopes between $(t_{i_h}, i_h - 1)$ and $(t_i, i - 1)$ ($i = i_h + 1, \ldots, n$);
- The CNPMLE is then given by $\hat{\lambda}(t) = (i_{j+1} - i_j)/(t_{i_{j+1}} - t_{i_j})$ whenever $t_{i_j} < t \leq t_{i_{j+1}}$.

4.2.2 Alternative NPMLE

The CNPMLE in Subsection 4.2.1 assumes the monotone property of intensity function. Especially, in the non-increasing case, the mean value function can be bounded. However, dissimilar to the parametric model in Eq.(4.2), it is impossible to estimate the initial number of software faults before testing, $\omega$, in the current modeling framework with a finite number of fault-detection time data. This will penalize for estimating several significant software reliability measures. In this subsection, we consider alternative NPMLE for the specific NHPP-based SRM with mean value function $\omega F(t)$. From Eq.(4.1), it is well-known that $t_1 < t_2 < \ldots < t_n$ can be considered as a general order statistics \cite{54} sampled from the probability distribution function $F(t)$. When $t_i$ ($i = 1, 2, \ldots, n$) are regarded as independent and identically distributed (i.i.d.) samples, the log-likelihood function of the i.i.d. sample is given by

$$LLF(r(t_i), i = 1, 2, \ldots, n) = \sum_{i=1}^{n} \log r(t_i) \sum_{i=1}^{n} \int_{0}^{t_i} r(x)dx. \quad (4.13)$$

To maximize the log-likelihood function in Eq.(4.13), Marshall and Proschan \cite{3} apply the similar idea to Boswell \cite{24} and formulate the following variational problem with respect to $r(\cdot)$:

$$\arg \max_{r(t_i), i = 1, 2, \ldots, n} LLF(r(t_i), i = 1, 2, \ldots, n). \quad (4.14)$$
The resulting NPMLE estimator is called the Marshall and Prochans estimator [3]. More specifically, if the fault-detection time distribution is IFR (Increasing Failure Rate), then the NPMLE is given by

\[
r(t) = \begin{cases} 
0, & 0 \leq t < t_1, \\
{r'}^{IFR}(t), & t_k \leq t < t_{k+1} \ (k = 1, \cdots, n - 1), \\
{r'}^{IFR}(t), & t = t_n, \\
\infty, & t_n < t,
\end{cases}
\]  

(4.15)

where

\[
{r'}^{IFR}(t) = \min_{v \geq k+1} \max_{u \leq k} \left[ \frac{v}{J(u,v)} \right],
\]  

(4.16)

\[
J(u,v) = \sum_{i=u+1}^{v} (n - i + 1)(t_i - t_{i-1}).
\]  

(4.17)

On the other hand, if the fault-detection time distribution is DFR (Decreasing Failure Rate), then we have the corresponding NPMLE by

\[
r(t) = \begin{cases} 
{r'}^{DFR}(t), & 0 \leq t \leq t_1, \\
{r'}^{DFR}(t), & t_k < t < t_{k+1} \ (k = 1, \cdots, n - 1), \\
0, & t_n < t,
\end{cases}
\]  

(4.18)

where

\[
{r'}^{DFR}(t) = \max_{v \geq k+1} \min_{u \leq k} \left[ \frac{v}{J(u,v)} \right].
\]  

(4.19)

From the above results, it is seen that the failure rate function \( r(\cdot) \) can be estimated from the monotone (IFR or DFR) property of the fault-detection time distribution.

In actual software development projects, it is known that \( S \)-shaped curve is frequently observed in terms of the cumulative number of software faults. This is caused by the non-decreasing \( S \)-shaped probability distribution function \( F(t) \) (see [33],[79],[38]). From the above observation, it seems to be interesting to consider the \( S \)-shaped pattern on the scaled TTT plot in the subsequent subsection. Define the change point \( t_m \) from DFR to IFR. Then we can define
4.2. NONPARAMETRIC ESTIMATION

the S-shaped failure rate by

\[ r(t) = \begin{cases} 
  r_0^{DFR}(t), & 0 \leq t \leq t_1, \\
  r_k^{DFR}(t), & t_k < t \leq t_{k+1} \ (k = 1, \ldots, m), \\
  r_k^{IFR}(t), & t_k < t \leq t_{k+1} \ (k = m, \ldots, n), \\
  \infty, & t_n < t. 
\end{cases} \]  

(4.20)

In other words, the change point \( t_m \) can be found from the observation \( t_1, t_2, \ldots, t_n \), but is not estimated in our problem (see [67],[80] for the change point estimation). In a fashion similar to Boswell’s estimator [24], the resulting failure rate estimator is discontinuous, but the software intensity function with these estimators is smoothed. We also give a simple algorithm to calculate the IFR failure rate (the case of DFR is similar).

- Set \( h = 1 \) and \( i_1 = 1 \);
- Repeat until \( i_{h+1} = n \):
  - Set \( i_{h+1} \) to be the index \( i \) which minimizes the value \( (i \ i_h) / J(i_h, i) \ (i = i_h + 1, \ldots, n) \);
  - Then the IFR estimator is given by \( \hat{r}(t) = (i_{j+1} \ i_j) / J(i_j, i_{j+1}) \) whenever \( t_{i_j} < t \leq t_{i_{j+1}} \).

Substituting the resulting estimator of failure rate to Eq.(4.5) yields an estimator of intensity function. Differentiating Eq.(4.7) with respect to \( \omega \) and equal to zero lead to following equation:

\[ \hat{\omega} = \frac{1}{n} \exp \left( \frac{\int_0^{t_n} r(t) \, dt}{F(t_n)} \right) = \frac{n}{F(t_n)}. \]  

(4.21)

Since the estimators in Eqs.(4.15),(4.18),(4.20) are NPMLEs of the failure rate, the above estimator in Eq.(4.21) maximizes the likelihood function in terms of \( \omega \). As the number of software fault-detection time data, \( n \), increases, then \( t_n \to \infty \) and \( F(t_n) \to 1 \), so that \( \hat{\omega} \) approaches to \( n \), from Eq.(4.21).

4.2.3 Total Time on Test Plot

In Subsection 4.2.2, we develop an NPMLE under the assumption that the mean value function is bounded, say, \( \Lambda(t) \leq \omega \). However, since this method is based
on the monotone property of failure rate, given the order statistics of software fault-detection time data, it is always needed to test the aging property of the underlying probability distribution function \( F(t) \). The simplest way is to apply the scaled total time on test (TTT) plot. The scaled TTT plot was introduced by Barlow and Campo [81] as a graphical tool for analyzing failure time data. For \( n \) software fault-detection time data, \( t_1, t_2, \ldots, t_n \), the total time on test statistics is defined by

\[
T_{TTT_i} = \sum_{j=1}^{i}(n - j + 1)(t_j - t_{j-1}), \quad (i = 1, \ldots, n). \tag{4.22}
\]

By normalizing Eq.(4.22), we can get the scaled TTT statistics:

\[
ST_i = \frac{T_{TTT_i}}{T_{TTT_n}}, \quad (i = 1, \ldots, n; \ ST_0 = 0). \tag{4.23}
\]

Define the empirical distribution \( F_n(t) \) for the software fault-detection time data, \( t_1, t_2, \ldots, t_n \), by

\[
F_n(t) = \begin{cases} 
\frac{i}{n}, & t_i \leq t < t_{i+1}, \quad (i = 0, \ldots, n - 1; t_0 = 0), \\
1, & t_n \leq t.
\end{cases} \tag{4.24}
\]

By plotting \((i/n, ST_i)\) in the \([0, 1] \times [0, 1]\) plane and connecting them by line segments, we can obtain the scaled TTT plot. If the resulting scaled TTT plot always exceeds the 45-degree line, it can be judged that the fault-detection time distribution is IFR. On the other hand, if the scaled TTT plot is under the 45-degree line, then the fault-detection time is DFR. If the failure rate changes from DFR to IFR, then the scaled TTT plot becomes an \( S \)-shaped curve across the 45-degree line. In this way, we can check the monotone property of the failure rate by means of the scaled TTT plot, and can estimate the intensity function with Eqs.(4.5) and (4.20).

### 4.3 Software Reliability Assessment for Discrete Time Case

In this section, an isotonic estimation of software fault intensity function for software fault count data is provided under the incomplete knowledge on the mean value function in an NHPP-based software reliability model, where the
4.3. SOFTWARE RELIABILITY ASSESSMENT FOR DISCRETE TIME CASE

basic idea comes from the nonparametric maximum likelihood estimation and the min-max principle.

4.3.1 Discrete Time Non-homogeneous Poisson Process

Suppose that the system test of software starts at time \( i = 0 \). Let \( N_i \) be the cumulative number of software faults detected by testing time \( i (= 1, 2, \cdots, k) \). Then, it is said that the discrete time stochastic process \( \{N_i; i = 1, 2, \cdots\} \) is a discrete time non-homogeneous Poisson process (DNHPP) if the probability mass function (p.m.f.) at discrete time \( i \) is represented by

\[
\Pr(N_i = n | N_0 = 0) = \frac{(\Lambda_i)^n}{n!} \exp(-\Lambda_i),
\]

where \( \Lambda_i \) is called mean value function of DNHPP, and represents the expected cumulative number of software faults detected by time \( i \), i.e. \( \Lambda_i = \mathbb{E}[N_i] \). Furthermore, the function \( \lambda_i = \Lambda_i - \Lambda_{i-1} \) \((i = 1, 2, \cdots; \Lambda_0 = 0)\) is called the discrete software fault intensity function, and denotes the expected number of software faults at time \( i \), that is \( \lambda_i = \mathbb{E}[N_i] - \mathbb{E}[N_{i-1}] \). Since the DNHPP \( \{N_i; i = 1, 2, \cdots\} \) is characterized by only the mean value function \( \Lambda_i \) or equivalently software fault intensity function \( \lambda_i \), we can express various software faults detection patterns by modeling the mean value function or fault intensity function.

Suppose that the cumulative number of software faults, \( n_i \), detected by time \( i (= 1, 2, \cdots, k) \) are available. If the parametric form of mean value function of DNHPP is known, the problem is to estimate model parameters \( \theta \) involved in the mean value function \( \Lambda_i, \theta \). In general, the maximum likelihood estimation method is a commonly used technique to estimate model parameters. The likelihood function of DNHPP is given as the function of model parameters \( \theta \):

\[
LF(\theta) = \exp \left\{ \Lambda_{n, \theta} \times \prod_{i=1}^{k} \frac{(\Lambda_i, \theta)}{(n_i - n_{i-1})!} \right\},
\]

where \( \Lambda_{0, \theta} = 0 \) and \( n_0 = 0 \). By deriving the model parameters \( \theta \) which maximize Eq.(4.26), we can obtain the maximum likelihood estimate \( \hat{\theta} \) of model parameters.
4.3.2 Constrained Nonparametric Maximum Likelihood Estimation

Next consider the case where the form of mean value function is completely unknown. In this case, the maximum likelihood estimation method mentioned above cannot be applied any longer. If the cumulative number of software faults $n_i$ detected by time $i$ (i.e., $i = 1, 2, \ldots, k$) are given, it is intuitive to represent the mean value function by the actual number of software faults detected by discrete time $i$, i.e., $\hat{m}_i = n_i$. Then, an estimate of software fault intensity function $\hat{\lambda}_i$ is given by $\hat{\lambda}_i = \hat{m}_i \left( \hat{m}_{i-1} = n_{i-1} \right)$, where $n_0 = 0$. Although the mean squares error with the underlying software fault count data is always zero, say, $\hat{m}_i = n_i$, the resulting estimate $\hat{\lambda}_i$ is naive and does not work well for the generalization ability, because it tends to have big noise and fluctuates everywhere.

Therefore, we introduce the similar but somewhat different constrained nonparametric maximum likelihood estimation (CNPMLE) from Boswell [24] as alternative smoothed estimate. Imposing the restriction on software fault intensity function, Boswell [24] shows that the maximization of log-likelihood for non-decreasing intensity functions can be attained with software fault detection time data. On the other hand, we assume that the software fault intensity function is non-increasing in time $i$ to represent the software reliability growth phenomenon. Define the log-likelihood function for the unknown intensity function $\lambda_i$ (i.e., $i = 1, 2, \ldots, k$) by

$$LLF(\lambda_1, \lambda_2, \cdots, \lambda_k) = \sum_{i=1}^{k} \left( (n_i - n_{i-1} \log(\lambda_i) - \lambda_i \log\{ (n_i - n_{i-1})! \} \right).$$  

(4.27)

Under the condition that the software fault intensity function is non-increasing in time $i$, the nonparametric maximum likelihood estimation problem with Eq.(4.27) can be reduced to the following variational problem with respect to $\lambda_i$;

$$\arg \max_{\lambda_i \geq \lambda_2 \geq \cdots \geq \lambda_k} LLF(\lambda_1, \lambda_2, \cdots, \lambda_k).$$  

(4.28)

The CNPMLE of software fault intensity function with software fault count data is given by

$$\hat{\lambda}_i = \min_{0 \leq k \leq i} \max_{1 \leq j \leq k} \frac{n_j}{n_k} \left( n_{i-k} \right),$$  

(4.29)
where $n_0 = 0$. It can be checked easily that CNPMLE given in Eq.(4.29) maximizes $LLF(\lambda_i, i = 1, 2, \cdots, k)$ in Eq.(4.27) for an arbitrary non-increasing piecewise linear intensity function $\lambda_i$ ($i = 1, 2, \cdots, k$) (see [77]).

In general, the computation cost is quite expensive to find nonparametric estimators of mean value function, when the kernel approach is taken [41], [42], [43], [44]. We give a computation algorithm to find the CNPMLE of software fault intensity function as follows.

- Set $h = 0$, $i_0 = 0$ and $n_0 = 0$;
- Repeat until $i_{h+1} = k$:
  - Set $i_{h+1}$ to be the index $i$ which maximizes the slopes between $(i_h, n_{i_h})$ and $(i, n_i)$ ($i = i_h + 1, \cdots, k$);
  - CNPMLE is given by $\hat{\lambda}_i = (n_{i_{h+1}} - n_{i_h})/(i_{h+1} - i_h)$ at time $i = i_h + 1, \cdots, i_{h+1}$ and set $h = h + 1$.

From the above algorithm it is easy to obtain the CNPMLE of software fault intensity function with fault count data. Once the intensity function is estimated, we can get the CNPMLE of mean value function by $\hat{\Lambda}_i = \sum_{j=1}^{i} \hat{\lambda}_j$.

### 4.4 Numerical Illustrations

In this section, we consider numerical examples with fault-detection time data and continuous time non-homogeneous Poisson process in Subsection 4.4.1 and 4.4.2. On the other hand, we discuss numerical examples with software fault count data and discrete time non-homogeneous Poisson process in Subsection 4.4.3.

#### 4.4.1 Goodness-of-Fit Performance

We compare the goodness-of-fit performances between parametric and non-parametric NHPP-based SRMs. For the comparison purpose, we apply eleven parametric NHPP-based SRMs; exponential (exp), gamma (gamma), Pareto (pareto), truncated normal (tnorm), log-normal (lnorm), truncated logistics (tlogis), log-logistics (llogis), truncated extreme-value maximum (txvmax), log-extreme-value maximum (lxvmax), truncated extreme-value minimum (txvmin), log-extreme-value minimum (lxvmin) distributions, which are implemented in
Table 4.1: Data sets.

<table>
<thead>
<tr>
<th>n</th>
<th>Trend</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>136</td>
<td>S-shape lxvmin</td>
</tr>
<tr>
<td>DS2</td>
<td>86</td>
<td>IFR exp</td>
</tr>
<tr>
<td>DS3</td>
<td>54</td>
<td>S-shape lxvmax</td>
</tr>
<tr>
<td>DS4</td>
<td>129</td>
<td>S-shape lxvmax</td>
</tr>
<tr>
<td>DS5</td>
<td>397</td>
<td>DFR lnorm</td>
</tr>
<tr>
<td>DS6</td>
<td>104</td>
<td>S-shape lxvmax</td>
</tr>
<tr>
<td>DS7</td>
<td>21</td>
<td>S-shape exp</td>
</tr>
<tr>
<td>DS8</td>
<td>41</td>
<td>S-shape lxvmax</td>
</tr>
<tr>
<td>DS9</td>
<td>278</td>
<td>S-shape gamma</td>
</tr>
<tr>
<td>DS10</td>
<td>197</td>
<td>IFR txvmin</td>
</tr>
<tr>
<td>DS11</td>
<td>207</td>
<td>S-shape exp</td>
</tr>
</tbody>
</table>

SRATS (Software Reliability Assessment Tool on Spreadsheet) [37]. We also use eleven software fault-detection time data sets; DS1 ∼ DS11, which are observed in actual software testing phases. Table 4.1 summarizes the data sets used for analysis, where \( n \) denotes the total number of software faults detected in testing, “Trend” means the scaled TTT plotting results, and “Best” implies the best parametric NHPP-based SRM among eleven models in terms of the minimum Akaike information criterion. In Figs.4.1-4.3, we give three examples to check the trend of failure rate with scaled TTT plot for DS1, DS5 and DS10, respectively. Since the resulting curve crosses 45-degree line in Fig.4.1, we can know that DS1 has the \( S \)-shaped trend. On the other hand, DS5 (DS10) shows the almost DFR (strict IFR) trend in Fig.4.2 (Fig.4.3). We choose these three data sets which have three specific trends (IFR, DFR and \( S \)-shape).

Figures 4.4-4.6 depict the estimation results on mean value function (cumulative number of software faults) with four nonparametric NHPP-based SRMs and the best parametric SRM for DS1, DS5 and DS10, respectively. From these results, it can be seen that (i) the best parametric NHPP-based SRM can catch the average trend of cumulative number of software faults, but fails to follow the microscopic behavior, (ii) the CNPMLE always overestimates the cumulative
number of software faults and can construct an upper bound of the underlying cumulative data. (iii) the NPMLE based on the failure rate estimator is better fitted if the trend is correctly estimated. So, the $S$-shaped failure rate gives the best performance for DS1, but IFR and DFR cases overestimate and underestimate the cumulative number of software faults. The above observations seem to be quite reasonable, because (i) the parametric NHPP-based SRM strongly depends on the shape of selected probability distribution function, (ii) the CNPMLE is a pessimistic estimator because the estimator consists of the maximizer of Eq.(4.11), (iii) taking account of the trend in past observation, the NPMLE based on the selection of failure rate can estimate the detailed behavior of the cumulative number of software faults. Especially, it is interesting to see that the DFR estimator is always greater than the IFR estimator when the testing time is relatively short. The difference between these two estimators becomes smaller in the last phase of software test.

To compare the goodness-of-fit performances quantitatively, we calculate the mean squared error (MSE) and maximum log-likelihood (MLL) as criteria for the goodness-of-fit performance:

\[
\text{MSE} = \frac{\sqrt{\sum_{i=1}^{n} \left( \frac{\Lambda(t_i)}{n} - \hat{\lambda}(t_i) \right)^2}}{n},
\]

\[
\text{MLL} = \sum_{i=1}^{n} \log \lambda(t_i) \Lambda(t_n).
\]

Tables 4.2-4.3 represent the goodness-of-fit performance results. Looking at MSE, the NPMLE with $S$-shaped failure rate gives the minimum MSE among three monotone properties; IFR, DFR and $S$-shaped. We can see that the $S$-shaped NPMLE shows smaller MSE in almost all cases, even compared with CNPMLE. In comparison between $S$-shaped NPMLE and the best parametric NHPP-based SRM, it can be found that the NPMLE has the better goodness-of-fit performance in the sense of minimization of MSE in all cases, except DS3 and DS11. Even in the cases of DS3 and DS11, the difference on MSE between NPMLE and the best parametric model is very small. On the other hand, focusing on the results with MLL, all the NPMLEs including the CNPMLE show the better goodness-of-fit performance than the best parametric NHPP-based SRMs in all data sets. Since the degree of freedom for NPMLE is greater than the parametric NHPP-based SRMs, this result will satisfy our intuitions. How-
ever, the practical advantage is that the model selection is not needed to check the goodness-of-fit performance before estimating software reliability measures. This attractive property enables us to understand the detailed behavior of the software fault-detection process more accurately. It can be considered that the model which has better goodness-of-fit performance enables us to obtain the software reliability measures more accurately.

<table>
<thead>
<tr>
<th></th>
<th>IFR</th>
<th>DFR</th>
<th>S-shape</th>
<th>CNPMLE</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>0.701</td>
<td>0.520</td>
<td>0.145</td>
<td>0.196</td>
<td>0.222</td>
</tr>
<tr>
<td>DS2</td>
<td>0.200</td>
<td>0.678</td>
<td>0.152</td>
<td>0.154</td>
<td>0.220</td>
</tr>
<tr>
<td>DS3</td>
<td>0.575</td>
<td>0.380</td>
<td>0.198</td>
<td>0.249</td>
<td>0.192</td>
</tr>
<tr>
<td>DS4</td>
<td>1.181</td>
<td>0.637</td>
<td>0.430</td>
<td>0.417</td>
<td>0.534</td>
</tr>
<tr>
<td>DS5</td>
<td>2.116</td>
<td>0.764</td>
<td>0.540</td>
<td>0.374</td>
<td>0.814</td>
</tr>
<tr>
<td>DS6</td>
<td>0.694</td>
<td>0.703</td>
<td>0.194</td>
<td>0.206</td>
<td>0.322</td>
</tr>
<tr>
<td>DS7</td>
<td>0.466</td>
<td>0.496</td>
<td>0.176</td>
<td>0.269</td>
<td>0.408</td>
</tr>
<tr>
<td>DS8</td>
<td>0.731</td>
<td>0.531</td>
<td>0.384</td>
<td>0.434</td>
<td>0.390</td>
</tr>
<tr>
<td>DS9</td>
<td>0.360</td>
<td>2.138</td>
<td>0.256</td>
<td>0.624</td>
<td>0.630</td>
</tr>
<tr>
<td>DS10</td>
<td>0.291</td>
<td>2.098</td>
<td>0.263</td>
<td>0.443</td>
<td>0.281</td>
</tr>
<tr>
<td>DS11</td>
<td>0.296</td>
<td>1.262</td>
<td>0.236</td>
<td>0.284</td>
<td>0.228</td>
</tr>
</tbody>
</table>

### 4.4.2 Software Reliability Measures

Next, we evaluate some software reliability measures with the CNPMLE and eleven parametric NHPP-based SRMs. Note that the CNPMLE is not suitable for estimating the residual number of software faults and the fault-free probability, which is the probability that there is no fault remaining in software product, because these depend on the initial number of software faults before testing, \( \omega \). For an arbitrary time \( t \) and the fault-detection time distribution \( F(t) \), we define the residual number of software faults \( R(t) \) and fault-free probability at time \( t \)
4.4. NUMERICAL ILLUSTRATIONS

Figure 4.1: Scaled TTT plot with DS1.

Figure 4.2: Scaled TTT plot with DS5.

Figure 4.3: Scaled TTT plot with DS10.
Figure 4.4: Behavior of mean value functions with DS1.

Figure 4.5: Behavior of mean value functions with DS5.

Figure 4.6: Behavior of mean value functions with DS10.
Table 4.3: Goodness-of-fit performance (MLL).

<table>
<thead>
<tr>
<th></th>
<th>IFR</th>
<th>DFR</th>
<th>S-shape</th>
<th>Constrained</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>-961.6</td>
<td>-963.9</td>
<td>-958.6</td>
<td>-953.4</td>
<td>-966.1</td>
</tr>
<tr>
<td>DS2</td>
<td>-672.3</td>
<td>-689.7</td>
<td>-681.1</td>
<td>-682.4</td>
<td>-686.5</td>
</tr>
<tr>
<td>DS3</td>
<td>-437.2</td>
<td>-445.5</td>
<td>-441.3</td>
<td>-440.1</td>
<td>-445.3</td>
</tr>
<tr>
<td>DS4</td>
<td>-900.2</td>
<td>-910.0</td>
<td>-891.4</td>
<td>-897.0</td>
<td>-908.0</td>
</tr>
<tr>
<td>DS5</td>
<td>-2367.7</td>
<td>-2341.8</td>
<td>-2325.5</td>
<td>-2317.9</td>
<td>-2355.4</td>
</tr>
<tr>
<td>DS6</td>
<td>-588.6</td>
<td>-600.9</td>
<td>-585.0</td>
<td>-590.2</td>
<td>-599.1</td>
</tr>
<tr>
<td>DS7</td>
<td>-87.3</td>
<td>-97.9</td>
<td>-90.8</td>
<td>-93.6</td>
<td>-96.8</td>
</tr>
<tr>
<td>DS8</td>
<td>-482.5</td>
<td>-498.1</td>
<td>-491.6</td>
<td>-489.6</td>
<td>-501.1</td>
</tr>
<tr>
<td>DS9</td>
<td>-3616.8</td>
<td>-3662.2</td>
<td>-3613.7</td>
<td>-3628.0</td>
<td>-3651.4</td>
</tr>
<tr>
<td>DS10</td>
<td>-2611.4</td>
<td>-2668.4</td>
<td>-2625.4</td>
<td>-2638.4</td>
<td>-2645.3</td>
</tr>
<tr>
<td>DS11</td>
<td>-1077.3</td>
<td>-1099.4</td>
<td>-1076.4</td>
<td>-1082.5</td>
<td>-1093.5</td>
</tr>
</tbody>
</table>

by

\[ R(t) = \hat{\omega}(1 \ F(t)), \] (4.32)

\[ FFP(t) = \exp(\hat{\omega}(1 \ F(t))), \] (4.33)

respectively, where \( \hat{F}(t) \) is an estimate of the fault-detection time distribution.

Tables 4.4-4.6 present the estimation results on the software reliability measures with DS1-DS11, where residual number of software faults and fault-free probability are estimated at the last fault-detection time \( t_n \). Looking at the results on initial number of software faults and residual number of software faults, it can be seen that the NPMLE estimates the initial number of software faults closely to the total number of software fault data, \( n \), and the residual number of software faults much smaller, which is almost zero. The IFR estimator and S-shaped estimator show the very similar estimation results. On the other hand, DFR estimator tends to give a little bit larger estimation than the other two NPMLEs. The fault-free probability based on the NPMLE, especially in IFR estimator, tends to take the higher value. On the other hand, it is seen that the fault-free probability tends to take lower value when one applies the best parametric NHPP-based SRMs. Furthermore, we calculate two reliability
Table 4.4: Initial number of software faults at time $t = 0$.

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
</tr>
<tr>
<td>DS1</td>
</tr>
<tr>
<td>DS2</td>
</tr>
<tr>
<td>DS3</td>
</tr>
<tr>
<td>DS4</td>
</tr>
<tr>
<td>DS5</td>
</tr>
<tr>
<td>DS6</td>
</tr>
<tr>
<td>DS7</td>
</tr>
<tr>
<td>DS8</td>
</tr>
<tr>
<td>DS9</td>
</tr>
<tr>
<td>DS10</td>
</tr>
<tr>
<td>DS11</td>
</tr>
</tbody>
</table>

Table 4.5: Residual number of software faults at time $t = t_n$.

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IFR</td>
</tr>
<tr>
<td>DS1</td>
</tr>
<tr>
<td>DS2</td>
</tr>
<tr>
<td>DS3</td>
</tr>
<tr>
<td>DS4</td>
</tr>
<tr>
<td>DS5</td>
</tr>
<tr>
<td>DS6</td>
</tr>
<tr>
<td>DS7</td>
</tr>
<tr>
<td>DS8</td>
</tr>
<tr>
<td>DS9</td>
</tr>
<tr>
<td>DS10</td>
</tr>
<tr>
<td>DS11</td>
</tr>
</tbody>
</table>
Table 4.6: Fault-free probability at time $t = t_n$.

<table>
<thead>
<tr>
<th></th>
<th>IFR</th>
<th>DFR</th>
<th>S-shape</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>7.16E-01</td>
<td>3.38E-03</td>
<td>6.64E-01</td>
<td>9.61E-17</td>
</tr>
<tr>
<td>DS2</td>
<td>7.00E-01</td>
<td>1.61E-03</td>
<td>7.01E-01</td>
<td>4.76E-10</td>
</tr>
<tr>
<td>DS3</td>
<td>6.81E-01</td>
<td>6.49E-02</td>
<td>6.44E-01</td>
<td>3.77E-74</td>
</tr>
<tr>
<td>DS4</td>
<td>7.59E-01</td>
<td>2.65E-03</td>
<td>6.73E-01</td>
<td>2.21E-20</td>
</tr>
<tr>
<td>DS5</td>
<td>8.81E-01</td>
<td>5.51E-06</td>
<td>7.39E-01</td>
<td>4.71E-19</td>
</tr>
<tr>
<td>DS6</td>
<td>7.06E-01</td>
<td>3.38E-04</td>
<td>6.47E-01</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>DS7</td>
<td>6.49E-01</td>
<td>6.03E-02</td>
<td>6.06E-01</td>
<td>2.19E-09</td>
</tr>
<tr>
<td>DS8</td>
<td>9.33E-01</td>
<td>7.78E-02</td>
<td>9.06E-01</td>
<td>4.23E-24</td>
</tr>
<tr>
<td>DS9</td>
<td>8.04E-01</td>
<td>1.22E-12</td>
<td>8.03E-01</td>
<td>9.43E-87</td>
</tr>
<tr>
<td>DS10</td>
<td>7.76E-01</td>
<td>1.73E-11</td>
<td>7.77E-01</td>
<td>1.07E-08</td>
</tr>
<tr>
<td>DS11</td>
<td>7.23E-01</td>
<td>2.08E-07</td>
<td>7.23E-01</td>
<td>3.26E-22</td>
</tr>
</tbody>
</table>

measures which are related to the fault-detection time. The mean time between software failures (MTBF) is one of the most useful measures to represent the frequency of occurrence of software failures in operation. As MTBF is longer, the frequency of fault-detection becomes smaller, and the software reliability grows as well. Strictly speaking, since a unique MTBF cannot be defined for NHPP-based SRMs, the cumulative mean time between failures $MTBF_C(t)$ and instantaneous mean time between failures $MTBF_I(t)$ are sometimes used as alternative measures:

\[
    MTBF_C(t) = \frac{t}{\Lambda(t)} \quad (4.34)
\]

\[
    MTBF_I(t) = \frac{1}{\Lambda(t)} \quad (4.35)
\]

The cumulative MTBF represents the inverse of the expected number of software faults detected per unit time, and the instantaneous MTBF indicates the momentary mean time to detect a software fault at arbitrary time $t$.

Tables 4.7-4.8 present the estimation results on cumulative MTBF and instantaneous MTBF with DS1-DS11, where $t = t_n$ in Eqs.(4.34) and (4.35). On the cumulative MTBF, it is seen that the results are almost similar regardless of the selected models. On the other hand, the results of instantaneous MTBF
Table 4.7: Cumulative MTBF at time $t = t_n$.

<table>
<thead>
<tr>
<th></th>
<th>IFR</th>
<th>DFR</th>
<th>$S$-shape</th>
<th>Constrained</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>652</td>
<td>652</td>
<td>652</td>
<td>652</td>
<td>652</td>
</tr>
<tr>
<td>DS2</td>
<td>1193</td>
<td>1193</td>
<td>1193</td>
<td>1193</td>
<td>1193</td>
</tr>
<tr>
<td>DS3</td>
<td>2013</td>
<td>2013</td>
<td>2013</td>
<td>2013</td>
<td>2014</td>
</tr>
<tr>
<td>DS4</td>
<td>690</td>
<td>690</td>
<td>690</td>
<td>690</td>
<td>690</td>
</tr>
<tr>
<td>DS5</td>
<td>274</td>
<td>274</td>
<td>274</td>
<td>274</td>
<td>274</td>
</tr>
<tr>
<td>DS6</td>
<td>148</td>
<td>148</td>
<td>148</td>
<td>148</td>
<td>148</td>
</tr>
<tr>
<td>DS7</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>DS8</td>
<td>105185</td>
<td>105185</td>
<td>105185</td>
<td>105185</td>
<td>105227</td>
</tr>
<tr>
<td>DS9</td>
<td>197602</td>
<td>197602</td>
<td>197602</td>
<td>197602</td>
<td>197404</td>
</tr>
<tr>
<td>DS10</td>
<td>255009</td>
<td>255009</td>
<td>255009</td>
<td>255009</td>
<td>254919</td>
</tr>
<tr>
<td>DS11</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 4.8: Instantaneous MTBF at time $t = t_n$.

<table>
<thead>
<tr>
<th></th>
<th>IFR</th>
<th>DFR</th>
<th>$S$-shape</th>
<th>Constrained</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>11307</td>
<td>5155</td>
<td>9221</td>
<td>4116</td>
<td>2303</td>
</tr>
<tr>
<td>DS2</td>
<td>7316</td>
<td>6022</td>
<td>7363</td>
<td>3902</td>
<td>2967</td>
</tr>
<tr>
<td>DS3</td>
<td>20566</td>
<td>14323</td>
<td>17975</td>
<td>7899</td>
<td>6115</td>
</tr>
<tr>
<td>DS4</td>
<td>276</td>
<td>5551</td>
<td>192</td>
<td>2571</td>
<td>3752</td>
</tr>
<tr>
<td>DS5</td>
<td>4855</td>
<td>3064</td>
<td>2028</td>
<td>1629</td>
<td>2255</td>
</tr>
<tr>
<td>DS6</td>
<td>313</td>
<td>803</td>
<td>250</td>
<td>289</td>
<td>364</td>
</tr>
<tr>
<td>DS7</td>
<td>35</td>
<td>139</td>
<td>30</td>
<td>55</td>
<td>55</td>
</tr>
<tr>
<td>DS8</td>
<td>5.1E+06</td>
<td>6.9E+05</td>
<td>3.6E+06</td>
<td>1.2E+06</td>
<td>3.5E+05</td>
</tr>
<tr>
<td>DS9</td>
<td>9.3E+05</td>
<td>8.5E+05</td>
<td>9.3E+05</td>
<td>4.4E+05</td>
<td>3.4E+05</td>
</tr>
<tr>
<td>DS10</td>
<td>3.6E+06</td>
<td>9.3E+05</td>
<td>3.6E+06</td>
<td>1.4E+06</td>
<td>5.3E+05</td>
</tr>
<tr>
<td>DS11</td>
<td>663</td>
<td>409</td>
<td>663</td>
<td>332</td>
<td>205</td>
</tr>
</tbody>
</table>
show the quite different values when one applies different estimation methods. If the best fitted model is the most reliable, the instantaneous MTBF with the $S$-shaped failure rate may be realistic to quantify the momentary mean time to detect a software fault.

### 4.4.3 Numerical Examples with Software Fault Count Data

We derive the CNPMLE of mean value function with three real software fault count data. We name these data as DS1, DS2 and DS3, respectively. DS1 (DS2, DS3) consists of $k = 181$ ($41, 112$) fault count data, where $n_{181} = 224$ ($n_{41} = 351$, $n_{112} = 183$). For these data sets, we apply a discretized parametric (exponential [32]) NHPP and compare with our CNPMLE in terms of the goodness-of-fit performance. Figure 4.7 shows the estimation results on mean value function (cumulative number of software faults) with CNPMLE and exponential NHPP for DS2. From this figure, we can see that the CNPMLE always overestimates the cumulative number of software faults and can construct an upper bound of the underlying cumulative data. Next, we calculate the mean squares error (MSE) and the maximum log-likelihood (MLL) as criteria on the goodness-of-fit performances.

$$\text{MSE} = \sqrt{\frac{\sum_{i=1}^{k} \left( \hat{\Lambda}_i - n_i \right)^2}{k}}, \quad (4.36)$$

$$\text{MLL} = \sum_{i=1}^{k} (n_i - n_{i-1}) \log(\hat{\lambda}_i) - A_n \sum_{i=1}^{k} \log \{ (n_i - n_{i-1})! \}. \quad (4.37)$$

Table 4.9 presents the results of goodness-of-fit performance. It can be seen that the CNPMLE shows the better goodness-of-fit performance on MLL than the parametric NHPP in all data sets. Focusing on the results under MSE, we can see that the CNPMLE gives the almost similar value or smaller MSE than that of parametric NHPP.
Figure 4.7: Behavior of mean value function and real data with DS2.

Table 4.9: Comparison of Goodness-of-fit performances.

<table>
<thead>
<tr>
<th>DS</th>
<th>MLL</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CNPMLE</td>
<td>Exponential</td>
</tr>
<tr>
<td>1</td>
<td>-285.904</td>
<td>-348.224</td>
</tr>
<tr>
<td>2</td>
<td>-119.270</td>
<td>-160.749</td>
</tr>
<tr>
<td>3</td>
<td>-217.969</td>
<td>-220.581</td>
</tr>
</tbody>
</table>
Chapter 5

Robustness of Non-homogeneous Gamma Process-based Software Reliability Models

In this chapter we extend non-homogeneous gamma process (NHGP)-based software reliability models (SRMs) by Ishii and Dohi (2008) from both view points of modeling and parameter estimation. In modeling, we generalize the underlying NHGP-based SRMs to those for eleven kinds of trend function, which can characterize a variety of software fault-detection patterns. In parameter estimation, we develop a nonparametric maximum likelihood estimation method without the complete knowledge on trend functions, and compare it with the parametric maximum likelihood estimation method. Since an NHGP involves a non-homogeneous Poisson processes (NHPPs) as the simplest case, it is shown that NHGP-based SRMs are much more robust than the common NHPP-based SRMs and that our nonparametric method can improve the goodness-of-fit performance of the conventional parametric one.

5.1 Summary on NHPP-based SRMs

5.1.1 Model Description

Let \( \{N(t), t \geq 0\} \) be the cumulative number of software faults detected by time \( t \) in software testing and be a stochastic point process satisfying:
(i) \( N(0) = 0 \)

(ii) \( \{ N(t), t \geq 0 \} \) has independent increment

(iii) \( \Pr\{ N(t + \Delta t) - N(t) \geq 2 \} = o(\Delta t) \)

(iv) \( \Pr\{ N(t + \Delta t) - N(t) = 1 \} = \lambda(t)\Delta t + o(\Delta t) \),

where \( o(\Delta t) \) is the higher term of \( \Delta t \). Then the point process is said the non-homogeneous Poisson process (NHPP) with failure intensity function \( \lambda(t) \), and the probability that the total number of software faults detected by time \( t \) equals \( n \) is given by

\[
\Pr\{ N(t) = n \} = \frac{(\Lambda(t))^n}{n!} \exp\{ -\Lambda(t) \}, \quad (5.1)
\]

where

\[
\Lambda(t) = \mathbb{E}[N(t)] = \int_0^t \lambda(x)dx, \quad (5.2)
\]

is called the mean value function of the NHPP which represents the expected cumulative number of software faults detected by time \( t \).

In the traditional software reliability modeling based on NHPPs, the main research issue was to determine the failure intensity function \( \lambda(t) \), or equivalently the mean value function \( \Lambda(t) \) so as to fit the software fault count data. If the number of residual faults before software testing is expected to be finite, \( i.e. \), \( \lim_{t \to \infty} \Lambda(t) = a \) (\( a > 0 \)), the mean value function can be represented as the product of expected number of residual faults \( a \) and software fault-detection time distribution \( F(t; \theta) \), where \( \theta \) is model parameter (vector). For a given mean value function or equivalently software-fault detection time distribution, the next step is to estimate the model parameters included in the mean value function

Let \( \xi = (a, \theta) \) be the model parameters in an SRM, \( i.e. \), the mean value function is given by \( \Lambda(t; \xi) = \int_0^t \lambda(x; \xi)dx = aF(t; \theta) \). A commonly used technique for parameter estimation of NHPP-based SRMs is the maximum likelihood estimation. Let \( t_i \) \((i = 1, 2, \ldots)\) denote the time sequence to detect the \( i \)-th software fault (software fault-detection time) and be measured by CPU time in the software testing. Suppose that \( n \) \((n > 0)\) software fault-detection time data \( t_i \) \((i = 1, 2, \ldots, n; t_0 = 0)\) are observed during the testing phase \((0, T] \), where
5.1. SUMMARY ON NHPP-BASED SRMS

\( T (\geq t_n) \) is the observation point. From the independent increment of NHPP, the likelihood function is given by

\[
LF(\xi | t_1, \ldots, t_n, T) = \left[ \prod_{i=1}^{n} \lambda(t_i; \xi) \exp \left\{ \int_{t_{i-1}}^{t_i} \lambda(x; \xi) dx \right\} \right] \times \exp \left\{ \int_{t_n}^{T} \lambda(x; \xi) dx \right\} = \exp[\Lambda(T; \xi)] \prod_{i=1}^{n} \lambda(t_i; \xi).
\]

(5.3)

Taking the logarithm, we have the log-likelihood function given by

\[
LLF(\xi | t_1, \ldots, t_n, T) = \sum_{i=1}^{n} \log \lambda(t_i; \xi) \Lambda(T; \xi).
\]

(5.4)

The problem is to derive the estimate which maximizes \( LLF(\xi) \). That is, the maximum likelihood (ML) estimate of parameters \( \hat{\xi} \) is given by a solution of

\[
LLF(\hat{\xi} | t_1, \ldots, t_n, T) = \arg \max_{\xi} LLF(\xi | t_1, \ldots, t_n, T).
\]

(5.5)

This problem can be also reduced to solve the simultaneous likelihood equations:

\[
\frac{dLLF(\xi)}{d\xi} = 0.
\]

(5.6)

Since the above log-likelihood function in Eq.(5.5) is a multi-modal nonlinear function for an arbitrary failure intensity function \( \lambda(t; \xi) \), one needs any nonlinear optimization algorithm such as Newton’s method to obtain the ML estimate \( \hat{\xi} \). It is well known that the best parametric NHPP-based SRM to fit every type of software fault-detection time data does not exist [29],[30]. That is, we have to select the mean value function (trend) \( \Lambda(t; \xi) \) or the corresponding software fault-detection time distribution \( F(t; \theta) \) carefully in order to obtain the best fitted NHPP-based SRM to the underlying data. Okamura and Dohi [37] suggest that it is enough to consider eleven typical software fault-detection time distributions, which include exponential [32], gamma [38],[39], Pareto [46],[47], truncated normal [48], log-normal [31],[48], truncated logistics [35], log-logistics [34], truncated extreme-value maximum [36], log-extreme-value maximum [36], truncated extreme-value minimum [36], log-extreme-value minimum (Weibull) [33],[36] distributions. Table 5.1 presents eleven mean value (trend) functions and software fault-detection time distributions used here.
### Table 5.1: Mean value functions.

<table>
<thead>
<tr>
<th>Models</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential dist. (exp) [32]</td>
<td>$\Lambda(t) = aF(t)$, $F(t) = 1 - e^{-bt}$</td>
</tr>
<tr>
<td>Gamma dist. (gamma) [38],[39]</td>
<td>$\Lambda(t) = aF(t)$, $F(t) = \int_0^t e^{bs} \frac{1 - e^{-cs}}{b} ds$</td>
</tr>
<tr>
<td>Pareto dist. (pareto) [46],[47]</td>
<td>$\Lambda(t) = aF(t)$, $F(t) = \left(\frac{c}{t+c}\right)^b$</td>
</tr>
<tr>
<td>Truncated normal dist. (tnorm) [48]</td>
<td>$\Lambda(t) = a\frac{F(t)}{F(0)}$, $F(t) = \frac{1}{\sqrt{2\pi b}} \int_0^t e^{-\frac{(s-t)^2}{2b^2}} ds$</td>
</tr>
<tr>
<td>Log-normal dist. (lnorm) [31],[48]</td>
<td>$\Lambda(t) = aF(\log t)$, $F(t) = \frac{1}{\sqrt{2\pi b}} \int_0^t e^{-\frac{(\log s-c)^2}{2b^2}} ds$</td>
</tr>
<tr>
<td>Truncated logistic dist. (tlogis) [35]</td>
<td>$\Lambda(t) = a\frac{F(t)}{F(0)}$, $F(t) = \frac{1}{1+e^{-\frac{c}{b}t}}$</td>
</tr>
<tr>
<td>Log-logistic dist. (llogis) [34]</td>
<td>$\Lambda(t) = aF(\log t)$, $F(t) = \frac{1}{1+e^{-\frac{c}{b}t}}$</td>
</tr>
<tr>
<td>Truncated extreme-value max dist. (txvmax) [36]</td>
<td>$\Lambda(t) = a\frac{F(t)}{F(0)}$, $F(t) = e^{-\frac{c}{b}t}$</td>
</tr>
<tr>
<td>Log-extreme-value max dist. (lxvmax) [36]</td>
<td>$\Lambda(t) = aF(\log t)$, $F(t) = e^{-\frac{c}{b}t}$</td>
</tr>
<tr>
<td>Truncated extreme-value min dist. (txvmin) [36]</td>
<td>$\Lambda(t) = a\frac{F(t)}{F(0)}$, $F(t) = e^{-\frac{c}{b}t}$</td>
</tr>
<tr>
<td>Log-extreme-value min dist. (lxvmin) [33],[36]</td>
<td>$\Lambda(t) = a(1 - F(\log t))$, $F(t) = e^{-\frac{c}{b}t}$</td>
</tr>
</tbody>
</table>
5.1.2 Constrained Nonparametric ML Estimation [A3]

Boswell [24] introduces a nonparametric ML estimation, where $\Lambda(t; \xi) = a F(t; \theta)$ is not assumed. So, he does not restrict the class of NHPP such as $\Lambda(t; \xi) = a F(t; \theta)$, but, instead, assumes that the failure intensity function $\lambda(t)$ is a non-negative monotone non-increasing function of $t \in (0, T]$. For unknown failure intensity function $\lambda(t)\), Boswell [24] considers the following variational problem:

$$\arg \max_{\lambda(t)} \left[ \sum_{i=1}^{n} \log \lambda(t_i) \int_0^T \lambda(x) dx \right]. \quad (5.7)$$

Note that an increase in the above log-likelihood function can occur only by an increase in $\lambda(t)$ at the fault-detection time points. Boswell [24] proves that a solution in Eq.(5.7) must consist of step functions closed on the right with no jumps except at some of the event occurrence points. This fact leads to the idea on the so-called isotonic estimation, and gives a constrained nonparametric maximum likelihood estimate (constrained NPMLE). For $t_1 < t_2 < \ldots < t_n \leq T$, define the likelihood function as a function of unknown failure intensity function at each fault-detection time:

$$LF(\lambda(t_i), i = 1, 2, \ldots, n) = \exp \left\{ \int_0^T \lambda(x) dx \right\} \prod_{i=1}^{n} \lambda(t_i). \quad (5.8)$$

Taking the logarithm, the log-likelihood function is given by

$$LLF(\lambda(t_i), i = 1, 2, \ldots, n) = \sum_{i=1}^{n} \log \lambda(t_i) \int_0^T \lambda(x) dx. \quad (5.9)$$

Hence the problem can be reduced to the following finite dimensional problem:

$$\arg \max_{\lambda(t_i), i = 1, 2, \ldots, n} LLF(\lambda(t_i), i = 1, 2, \ldots, n). \quad (5.10)$$

For this problem, it turns out that the following min-max solution is optimal [53],[24]:

$$\hat{\lambda}(t_j) = \min_{1 \leq h \leq j \leq k \leq n} \max_{l_k} \frac{k}{t_h} \frac{h}{t_k}. \quad (5.11)$$

The resulting estimator is discontinuous, but constructs an upper bound of $LLF(\lambda(t_i), i = 1, 2, \ldots, n)$. Although we assume that the failure intensity function is non-increasing to represent the reliability growth phenomenon, this assumption can be relaxed. That is, if the failure intensity function is non-decreasing, then the resulting constrained NPMLE consists of step functions
closed on the left with no jumps except at some of event occurrence points, and
can be given by replacing min operator and max operator in Eq.(5.11) from
each other. Another advantage of the constrained NPMLE is the low com-
putation cost compared with the other representative nonparametric methods
[42],[44],[41],[45],[40],[43]. A simple optimization code to derive the constrained
NPMLE is given as follows.

\begin{itemize}
  \item Set $h = 1$ and $i_1 = 1$;
  \item Repeat until $i_{h+1} = n$:
    \begin{itemize}
      \item Set $i_{h+1}$ to be the index $i$ which minimizes the slopes between $(t_{i_h}, i_h)$ and $(t_i, i)$ ($i = i_h + 1, \ldots, n)$;
      \item The constrained NPMLE is then given by $\hat{\lambda}(t) = (i_{j+1} \ i_j)/(t_{i_{j+1}} \ t_{i_j})$
    \end{itemize}
  \end{itemize}

whenever $t_{i_j} < t \leq t_{i_{j+1}}$.

\section{NHGP-Based Software Reliability Models}

\subsection{A Generalization of NHPPs}

Next, we introduce non-homogeneous gamma process (NHGP)-based SRMs in
the sense of Berman [50]. As well known, the NHPP possesses the time-scale
transform property, i.e., the random variable $(T_i; \xi)$ ($i = 0, 1, \ldots, n$) can be
regarded as a homogeneous Poisson process (HPP) with unit failure intensity
$\lambda(t; \xi) = 1$ for all $t \geq 0$, where $T_i$ are the software fault-detection times (random
variables). This means that $\Lambda(T_{i+1}; \xi)$ and $\Lambda(T_i; \xi)$ are independent and identi-
cally distributed random variables having the exponential distribution with unit
scale parameter. It holds even under the trend function is bounded. Berman
[50] considers a situation where there are $\kappa > 0$ NHPPs with same failure
intensity function and where the observed software fault-detection epoch cor-
responds to every successive $\kappa$th event of the underlying NHPP. Let $\kappa$ be an
unknown integer-valued parameter which has to be estimated from the soft-
ware fault-detection time data. For example, when $\kappa = 2$, the software fault is
detected at every second event. Thus software whose faults are detected and
removed is in better condition than would be before the detection of their faults,
because two other events must occur in order to observe the next software fault.
occurrence. That is, when $\kappa > 1$, the software is improved in better condition just before the removal of software faults, and the larger $\kappa$ indicates the larger improvement effect. On the other hand, if $\kappa < 1$, then the software is in worse condition just after the detection of software faults. From the definition, the random variables $\Lambda(T_{i+1}; \xi) - \Lambda(T_i; \xi)$ are regarded as the Stieltjes convolution of $\kappa$ exponential distributions with unit scale parameter, and obeys the gamma probability density function with shape parameter $\kappa$ and unit scale parameter:

$$g(t) = \frac{t^{\kappa-1} \exp(-t)}{\Gamma(\kappa)},$$

where $\Gamma(\cdot)$ denotes the standard gamma function. In other words, $\Lambda(T_{i+1}; \xi) - \Lambda(T_i; \xi)$ follow a renewal process having the gamma renewal density $g(t)$ with shape parameter $\kappa$ and unit scale parameter (see Figure 5.1).

The NHGP sequence $T_i$ is given by taking the inverse time-scale transform with $\Lambda(t; \xi)$. It is worth mentioning that the intensity function of an NHGP as a stochastic point process is no longer deterministic except for $\kappa = 1$. Therefore, when $\kappa \neq 1$, the intensity function of the NHGP may be stochastic and depends on the past history of events. This is usually called the intensity process for a stochastic point process (e.g. see [82]). For instance, let \{\$\zeta(t \mid \mathcal{F}_t)$, $t \geq 0$\} be the intensity process at time $t$ for a point process \{\$N(t), t \geq 0$\}, where $\mathcal{F}_t$ denotes the history of $N(t)$ up to, but not including time $t$. When $\kappa = 1$, it is evident that the intensity process $\zeta(t \mid \mathcal{F}_t) = \lambda(t; \xi)$ is deterministic. To avoid confusion, we call $\lambda(t)$ the failure intensity parameter in the latter discussion, and distinguish $\lambda(t)$ from intensity process $\zeta(t \mid \mathcal{F}_t)$ for the general NHGP. Furthermore, we call $\Lambda(t; \xi) = \int_0^t \lambda(x; \xi) dx$ the trend function for NHGP which
is equivalent to the mean value function for NHPP when $\kappa = 1$. It turns out [51] that the mean value function of NHGP $E[N(t)]$ does not equal $\Lambda(t; \xi)$ when $\kappa \neq 1$.

Suppose that $n$ software fault-detection time data $t_i$ ($i = 1, 2, \ldots, n$) observed from the time $t = 0$ to the end point $t = T$ are available. Then, the likelihood function of the NHGP-based SRM is given by

$$LF(\xi, \kappa) = \left[ \prod_{i=1}^{n} \lambda(t_i; \xi) \left\{ \Lambda(t_i; \xi) \quad \Lambda(t_{i-1}; \xi) \right\}^{\kappa-1} \right] \times \exp\left\{ \Lambda(T; \xi) / (\kappa)^n \right\}.$$  (5.13)

Taking the logarithm of Eq.(5.13), we have the log-likelihood function of the NHGP-based SRM:

$$LLF(\xi, \kappa) = \sum_{i=1}^{n} \left[ \log \lambda(t_i; \xi) + (\kappa - 1) \log \left\{ \Lambda(t_i; \xi) / \Lambda(t_{i-1}; \xi) \right\} \right] n \log (\kappa),$$  (5.14)

where $t_0 = \Lambda(t_0; \xi) = 0$ without any loss of generality. Similar to the NHPP-based SRMs, once we select the failure intensity parameter $\lambda(t; \xi)$, or equivalently trend function $\Lambda(t; \xi)$, we can estimate model parameters by maximizing the log-likelihood function in Eq.(5.14).

After one fault correction, the software system would thus be better than it was just before the fault detection. When $\kappa = 1$, Eq.(5.13) is reduced to Eq.(5.3) as a special case. Therefore, we can see that the NHPP is a special case of an NHGP. On the other hand, the software system would not be as same as it was just before the fault detection when $\kappa \neq 1$. Furthermore, when the failure intensity parameter is given by a constant, i.e., $\lambda(t; \xi) = \rho$, the NHGP is equivalent to the renewal process with the gamma distributed inter-renewal time with shape parameter $\kappa$ and scale parameter $\rho$ (gamma renewal process). Then the software system in this case would be as bad as old before testing and would not behave like reliability growth. In other words, the NHGP is a non-stationary stochastic point process characterized by the failure intensity parameter $\lambda(t; \xi)$ or the trend function $\Lambda(t; \xi)$, but can represent two different aging properties in the same model, called minimal repair property ($\kappa = 1$) and major repair property ($\lambda(t; \xi) = \rho$) in the common reliability engineering (see
5.2. NHGP-BASED SOFTWARE RELIABILITY MODELS

5.2.2 Behavior of Expected Cumulative Number of Software Faults in NHGP-based SRMs

As mentioned before, the NHGP is known as a generalized stochastic point process including the NHPP in the case of $\kappa = 1$. However, dissimilar to NHPP, we have not known well the probabilistic property of an NHGP as a counting process. For example, the cumulative number of software faults in the NHGP-based SRM $\mathbb{E}[N(t)]$ cannot be represented analytically and does not equal $\Lambda(t; \xi)$ anymore. The simplest way to obtain the expected cumulative number of software faults in NHGP-based SRMs is applying the elementary renewal theorem. In the existing work in Bandyopadhyay and Sen [83], the expected cumulative number of events in the NHGP can be derived approximately as follows:

$$
\mathbb{E}[N(t)] \approx \frac{\Lambda(t; \xi)}{\kappa} + \frac{1}{2\kappa} \kappa.
$$

(5.15)
For a sufficiently large $t$, it is known that Eq.(5.15) does work well from the analogy of the usual renewal process. It is obvious that the above formula is exactly consistent with the mean value function of an NHPP in the case of $\kappa = 1$.

For validation on the accuracy of Eq.(5.15), we apply the Monte Carlo simulation to investigate the behavior of the expected cumulative number of software faults in NHGP-based SRMs. First, we generate the pseudo random number $z_i$ ($i = 1, 2, \cdots, n$) following the gamma distribution with shape parameter $\kappa$ and unit scale parameter, and obtain $w_i = \sum_{j=1}^{i} z_i$. Transforming $w_i$ with the inverse of a trend function, we get $s_i = \Lambda^{-1}(w_i)$ which are samples from an NHGP. We repeat this procedure $m$ times to obtain $m$ sets of sample path which follow the NHGP. Finally, taking the average of these sample paths at an arbitrary time $t$, the expected cumulative number of software faults at time $t$ can be calculated numerically. For a sufficiently large number of $m$, this simulated mean value function is close to the actual mean value for the NHGP from the law of large numbers.

### 5.2.3 Constrained Nonparametric ML Estimation of NHGP

Bartoszynski et al. [53] extend the Boswell’s constrained NPMLE for an NHPP and derive a non-increasing NPMLE of $\lambda(t)$ in the situation where multiple sample paths of NHPP can be observed. We further extend this approach to the NHGP-based SRMs. Similar to the NHPP-based SRMs[A3], we seek the estimates which maximize the log-likelihood function given in Eq.(5.14) under the condition that $\lambda(t)$ belongs to the class of nonnegative, non-increasing functions on $(0, T]$. The optimal $\lambda(t)$ must also consist of step functions closed on the right with no jumps except at some of the fault-detection time points.

To see this, suppose that $\tilde{\lambda}(t)$ is any non-increasing function. Let $q_i = \int_{t_{i-1}}^{t_i} \tilde{\lambda}(x)dx$ for any fixed $i = 1, 2, \cdots, n + 1$, where we define $t_{n+1} = T$. If we choose $\lambda(t) = q_i/(t_i - t_{i-1})$ so as to be constant in $t \in (t_{i-1}, t_i)$, then $\int_{t_{i-1}}^{t_i} \lambda(x)dx = q_i$ holds. Therefore, substituting $\tilde{\lambda}(t)$ to Eq.(5.14) instead of $\lambda(t)$, only the term $\log \lambda(t_i)$ is changed to $\log \tilde{\lambda}(t_i)$. Since $\tilde{\lambda}(t)$ is a non-increasing function, it is seen that $\lambda(t_i) \geq \tilde{\lambda}(t_i)$ and the equality holds if $\lambda(t) = \tilde{\lambda}(t)$ for $t \in (t_{i-1}, t_i)$. So, unless $\tilde{\lambda}(t)$ is not constant on every interval of $(t_{i-1}, t_i]$, it is
5.2. NHGP-BASED SOFTWARE RELIABILITY MODELS

impossible to increase Eq.(5.14) without violating the non-increasing assumption.

The method of obtaining a non-increasing NPMLE of \( \lambda(t) \) in NHGP-based SRMs is very similar to that in NHPP-based SRMs. Let \( \lambda_i = \lambda(t_i) \) and \( x_i = t_i - t_{i-1} \) \((i = 1, 2, \ldots, n+1)\), with \( t_{n+1} = T \). Looking at Eq.(5.14), it is obvious that the optimum value of \( \lambda(t) \) is zero for \( t > t_n \), i.e., \( \lambda_{n+1} = 0 \), otherwise, only the term \( \Lambda(t) \) increases and the LLF decreases consequently. Since \( \Lambda(t_i) = \Lambda(t_{i-1}) \) can be represented by \( \lambda_i x_i \), the problem of maximizing Eq.(5.13) is simplified to the problem of maximizing

\[
LF(\lambda_1, \lambda_2, \ldots, \lambda_n, \kappa) = \prod_{i=1}^{n} \lambda_i [\lambda_i x_i]^{\kappa-1} \exp\left(\sum_{i=1}^{n} \lambda_i x_i / (\kappa)^n\right) 
\]

subject to \( \lambda_1 \geq \cdots \geq \lambda_n \geq 0 \). This is equivalent to maximize

\[
LLF(\lambda_1, \lambda_2, \ldots, \lambda_n, \kappa) = \sum_{i=1}^{n} \log \lambda_i + (\kappa - 1) \sum_{i=1}^{n} \log \lambda_i x_i - \sum_{i=1}^{n} \lambda_i x_i n \log (\kappa). 
\]

Unfortunately, it would not be easy to obtain analytically the ML estimates of both \( \lambda(t) \) and \( \kappa \) simultaneously. However, it might be possible to obtain the ML estimates by an iteration technique. That is, we start from an initial guess of the integer-valued parameter \( \kappa \), and maximize the log-likelihood function with the given parameter to obtain an ML estimate of \( \lambda(t) \). Next we estimate the parameter \( \kappa \) which maximizes the log-likelihood function using the estimate of \( \lambda(t) \) again. The similar iteration will continue until convergence. For a given \( \kappa \), we obtain ML estimates of \( \lambda_i \) by solving the following maximization problem:

\[
\max_{\lambda_1, \lambda_2, \ldots, \lambda_n} \sum_{i=1}^{n} \left\{ \kappa \log \lambda_i - \lambda_i x_i \right\}
\]

subject to \( \lambda_1 \geq \cdots \geq \lambda_n \geq 0 \). The solution to this problem is given by

\[
\lambda_1 = \lambda_2 = \cdots = \lambda_{k_1} = \max_{1 \leq t \leq n} \frac{\kappa t}{\sum_{j=1}^{k_1} x_j}
\]

\[
\lambda_{k_1+1} = \lambda_{k_1+2} = \cdots = \lambda_{k_2} = \max_{k_1+1 \leq t \leq n} \frac{\kappa (t - k_1)}{\sum_{j=k_1+1}^{k_2} x_j}
\]

\[
\lambda_{k_2+1} = \lambda_{k_2+2} = \cdots = \lambda_{k_3} = \max_{k_2+1 \leq t \leq n} \frac{\kappa (t - k_2)}{\sum_{j=k_2+1}^{k_3} x_j}
\]

\[
\vdots
\]
CHAPTER 5. NPMLE FOR NHGP-BASED SRMS

where \( k_1 \) is the value of \( t \) which satisfies Eq.(5.19), \( k_2 (> k_1) \) is the value of \( t \) which satisfies Eq.(5.20), \( k_3 (> k_2) \) is the value of \( t \) which satisfies Eq.(5.21), etc. Continue the above steps until \( k_l = n \). If there are more than one \( t \) giving the same maximum value, the biggest one should be chosen. Even if not, the solution can be achieved.

Next, consider the problem of maximizing Eq.(5.17) when \( \kappa \) is unknown. The iteration scheme is given as follows: Let \( \kappa^{(1)} \) be some initial guess of parameter \( \kappa \). Then use Eqs.(5.19)-(5.21) to find \( \lambda^{(1)}_1, \lambda^{(1)}_2, \cdots, \lambda^{(1)}_n \). This results the first estimate of \( \lambda(t) \). Since the process \( \Lambda(T_i) = \Lambda(T_{i-1}) \) \((i = 1, 2, \cdots, n)\) can be regarded as a renewal process from the NHGP assumption, we can find an updated estimate of \( \kappa^{(2)} \). Note that this is a renewal process with independent and identically distributed inter-renewal times. Since it is the gamma distribution with unit scale parameter (i.e., \( g(t) = t^{\kappa-1} e^{-t/\kappa} \)), the profile likelihood function becomes

\[
LF(\kappa) = \prod_{i=1}^{n} g(\Lambda(t_i) \Lambda(t_{i-1})). \tag{5.22}
\]

Define \( y_i^{(1)} = \sum_{j=1}^{i} \lambda^{(1)}_j x_j \) \((y_0^{(1)} = 0)\) using \( \lambda^{(1)}_i \) \((i = 1, 2, \cdots, n)\). Then the profile log-likelihood function for unknown \( \kappa \) can be written by

\[
LLF(\kappa) = \sum_{i=1}^{n} \log g(y_i^{(1)} y_i^{(1)}) = (\kappa - 1) \sum_{i=1}^{n} \log \lambda^{(1)}_i x_i \sum_{i=1}^{n} \lambda^{(1)}_i x_i \ n \log \kappa. \tag{5.23}
\]

The difference between Eq.(5.17) and Eq.(5.23) is only the constant term \( \sum_{i=1}^{n} \log \lambda^{(1)}_i \). A new estimate \( \kappa^{(2)} \) is then obtained by maximizing Eq.(5.23) numerically with respect to \( \kappa \). Then use \( \kappa^{(2)} \) to obtain an updated estimate \( \lambda^{(2)}_i \) \((i = 1, 2, \cdots, n)\) of \( \lambda(t) \) with Eqs.(5.19)-(5.21). We use \( \lambda^{(2)} \) to find the estimate of \( \kappa^{(3)} \) and continue the similar iterations until the difference between two successive estimates of \( \kappa \) is smaller than a given tolerance level \( \epsilon \). That is, when \( \kappa^{(i+1)} - \kappa^{(i)} \leq \epsilon \) holds, the estimation procedure ends with the final estimate \( \kappa^{(i+1)} \).
5.3 Numerical Illustrations

Our main purpose is the examination of goodness-of-fit performance of NHGP-based SRMs quantitatively. To compare the results, we also apply the well-known NHPP-based SRMs and compare the goodness-of-fit performances between NHGP-based SRMs and NHPP-based SRMs with an actual dataset which consists of $n = 54$ software fault-detection time data observed in a real software testing phase [84]. We identify the best SRM by calculating the maximum log-likelihood (MLL), Akaike information criterion (AIC), Bayesian information criterion (BIC) and mean squares error (MSE) in each observation point (10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 100%), where

$$\text{AIC} = 2 \times \text{MLL} + 2\theta, \quad (5.24)$$

$$\text{BIC} = 2 \times \text{MLL} + \theta \ln(n), \quad (5.25)$$

$$\text{MSE} = \sqrt{\frac{\sum_{i=1}^{n}(E[N(t)] - \hat{t})^2}{n}}, \quad (5.26)$$

and $\theta$ is the model dimension (number of free parameters), $n$ is the sample size (number of data). In NHPP-based SRMs, we calculate MSE with mean value function $\Lambda(t; \hat{\xi}) (= E[N(t)])$ directly. In NHGP-based SRMs, on the other hand, we apply the approximate method in Eq.(5.15) and the Monte Carlo simulation, where 500 samples are generated to obtain the $E[N(t; \hat{\xi})]$ for NHGP-based SRMs.

First, we compare the goodness-of-fit performance of parametric NHGP-based SRMs with that of parametric NHPP-based SRMs. We apply eleven trend functions (exp, gamma, pareto, tnorm, lnorm, tlogis, llogis, txvmax, lxvmax, txvmin, lxvmin) in Table 5.1 for both NHGP-based SRMS and NHPP-based SRMs. As mentioned before, we focus on the integer-valued $\kappa$ for NHGP-based SRMs to give their physical interpretations. Tables 5.2-5.5 present the calculation results of MLL and MSE with eleven parametric NHPP-based SRMs and eleven parametric NHGP-based SRMs. From these results, it can be seen that the NHGP-based SRMs can provide the higher goodness-of-fit performance than NHPP-based SRMs from the view point of MLL maximization and MSE minimization. In the latter testing phase it can be observed that NHGP-based SRMs show the exactly same goodness-of-fit performance to NHPP-based SRMs. This is because NHPP-based SRMs are special cases of NHGP-based SRMs and the
latter approaches to the former as the software test progresses. More specifically, we give the calculation results on AIC and BIC when the exponential trend function is assumed for NHPP-based SRM and NHGP-based SRM in Tables 5.6 and 5.7, respectively. At the first look, since AIC and BIC in 70%-100% observation points for the NHPP-based SRM are smaller than those for the NHGP-based SRM, it may be considered that the NHPP-based SRM outperforms the NHGP-based SRM. However, this observation will convince us, because the NHGP-based SRM can be reduced to the NHPP-based SRM by choosing $\kappa = 1$. In the case where the parameter $\kappa$ is redundant, the NHGP-based SRM shows the exactly same goodness-of-fit performance, and the model selection based on AIC and BIC has no significant meaning. In other words, the model selection in our generalized framework depends on the selection of $\kappa$ in addition to the trend function. The lesson learned from the comparative study is the adjustment of $\kappa$, so that in the initial and middle testing phases, the shape parameter $\kappa$ rather influences to the goodness-of-fit performance, but in the last testing phase it works similar to the common NHPP-based SRM. This is the similar insight to the previous result in [49] but is clarified in detail through our numerical experiment conducted in this chapter.
Table 5.2: Goodness-of-fit performance on MLL with NHPP-based SRMs.

<table>
<thead>
<tr>
<th>time</th>
<th>exp</th>
<th>gamma</th>
<th>pareto</th>
<th>tnorm</th>
<th>lnorm</th>
<th>tlogis</th>
<th>llogis</th>
<th>txvmax</th>
<th>lxvmax</th>
<th>txvmin</th>
<th>lxvmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>15.390</td>
<td>15.471</td>
<td>15.648</td>
<td>15.690</td>
<td>15.667</td>
<td>15.628</td>
<td>15.659</td>
<td>18.313</td>
<td>15.534</td>
<td>15.390</td>
<td>15.513</td>
</tr>
<tr>
<td>90%</td>
<td>2.772</td>
<td>4.728</td>
<td>5.461</td>
<td>2.733</td>
<td>5.660</td>
<td>4.329</td>
<td>5.131</td>
<td>3.967</td>
<td>5.824</td>
<td>2.772</td>
<td>4.942</td>
</tr>
</tbody>
</table>
Table 5.3: Goodness-of-fit performance on MLL with NHGP-based SRMs.

<table>
<thead>
<tr>
<th>time</th>
<th>exp</th>
<th>gamma</th>
<th>pareto</th>
<th>tnorm</th>
<th>lnorm</th>
<th>llogis</th>
<th>tnorm</th>
<th>lnorm</th>
<th>llogis</th>
<th>txvmax</th>
<th>lxvmax</th>
<th>txvmin</th>
<th>lxvmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>90%</td>
<td>2.772</td>
<td>4.728</td>
<td>5.461</td>
<td>2.733</td>
<td>5.660</td>
<td>4.329</td>
<td>5.131</td>
<td>3.967</td>
<td>5.824</td>
<td>2.772</td>
<td>4.942</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 5.4: Goodness-of-fit performance on MSE with NHPP-based SRMs.

<table>
<thead>
<tr>
<th>time</th>
<th>exp</th>
<th>gamma</th>
<th>pareto</th>
<th>tnorm</th>
<th>hnorm</th>
<th>tlogis</th>
<th>llogis</th>
<th>txvmax</th>
<th>lxvmax</th>
<th>txvmin</th>
<th>lxvmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>0.093</td>
<td>0.255</td>
<td>0.093</td>
<td>0.318</td>
<td>0.219</td>
<td>0.319</td>
<td>0.274</td>
<td>0.304</td>
<td>0.204</td>
<td>0.316</td>
<td>0.274</td>
</tr>
<tr>
<td>20%</td>
<td>0.218</td>
<td>0.221</td>
<td>0.200</td>
<td>0.219</td>
<td>0.189</td>
<td>0.209</td>
<td>0.211</td>
<td>0.218</td>
<td>0.178</td>
<td>0.218</td>
<td>0.219</td>
</tr>
<tr>
<td>30%</td>
<td>0.179</td>
<td>0.179</td>
<td>0.159</td>
<td>0.177</td>
<td>0.157</td>
<td>0.166</td>
<td>0.168</td>
<td>0.179</td>
<td>0.150</td>
<td>0.179</td>
<td>0.177</td>
</tr>
<tr>
<td>40%</td>
<td>0.184</td>
<td>0.168</td>
<td>0.139</td>
<td>0.191</td>
<td>0.137</td>
<td>0.158</td>
<td>0.153</td>
<td>0.185</td>
<td>0.129</td>
<td>0.185</td>
<td>0.163</td>
</tr>
<tr>
<td>50%</td>
<td>0.195</td>
<td>0.165</td>
<td>0.144</td>
<td>0.201</td>
<td>0.138</td>
<td>0.167</td>
<td>0.154</td>
<td>0.195</td>
<td>0.133</td>
<td>0.196</td>
<td>0.160</td>
</tr>
<tr>
<td>60%</td>
<td>0.188</td>
<td>0.156</td>
<td>0.137</td>
<td>0.234</td>
<td>0.164</td>
<td>0.143</td>
<td>0.142</td>
<td>0.398</td>
<td>0.202</td>
<td>0.188</td>
<td>0.146</td>
</tr>
<tr>
<td>70%</td>
<td>0.455</td>
<td>0.336</td>
<td>0.222</td>
<td>0.488</td>
<td>0.246</td>
<td>0.268</td>
<td>0.259</td>
<td>0.485</td>
<td>0.233</td>
<td>0.455</td>
<td>0.303</td>
</tr>
<tr>
<td>80%</td>
<td>0.447</td>
<td>0.293</td>
<td>0.226</td>
<td>0.461</td>
<td>0.231</td>
<td>0.288</td>
<td>0.249</td>
<td>0.564</td>
<td>0.226</td>
<td>0.448</td>
<td>0.272</td>
</tr>
<tr>
<td>90%</td>
<td>0.528</td>
<td>0.302</td>
<td>0.239</td>
<td>0.545</td>
<td>0.237</td>
<td>0.354</td>
<td>0.260</td>
<td>0.556</td>
<td>0.225</td>
<td>0.528</td>
<td>0.277</td>
</tr>
<tr>
<td>100%</td>
<td>0.552</td>
<td>0.273</td>
<td>0.207</td>
<td>0.569</td>
<td>0.205</td>
<td>0.353</td>
<td>0.229</td>
<td>0.588</td>
<td>0.194</td>
<td>0.552</td>
<td>0.246</td>
</tr>
</tbody>
</table>
Table 5.5: Goodness-of-fit performance on MSE with NHGP-based SRMs.

<table>
<thead>
<tr>
<th>time</th>
<th>exp</th>
<th>gamma</th>
<th>pareto</th>
<th>tnorm</th>
<th>lnorm</th>
<th>tlogis</th>
<th>llogis</th>
<th>txvmax</th>
<th>lxvmax</th>
<th>txvmin</th>
<th>lxvmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>0.306</td>
<td>0.282</td>
<td>0.285</td>
<td>0.297</td>
<td>0.272</td>
<td>0.309</td>
<td>0.278</td>
<td>0.302</td>
<td>0.284</td>
<td>0.309</td>
<td>0.271</td>
</tr>
<tr>
<td>20%</td>
<td>0.208</td>
<td>0.176</td>
<td>0.161</td>
<td>0.214</td>
<td>0.172</td>
<td>0.186</td>
<td>0.176</td>
<td>0.209</td>
<td>0.169</td>
<td>0.206</td>
<td>0.178</td>
</tr>
<tr>
<td>30%</td>
<td>0.186</td>
<td>0.177</td>
<td>0.166</td>
<td>0.211</td>
<td>0.158</td>
<td>0.171</td>
<td>0.175</td>
<td>0.192</td>
<td>0.179</td>
<td>0.185</td>
<td>0.167</td>
</tr>
<tr>
<td>40%</td>
<td>0.180</td>
<td>0.153</td>
<td>0.139</td>
<td>0.197</td>
<td>0.129</td>
<td>0.149</td>
<td>0.134</td>
<td>0.180</td>
<td>0.125</td>
<td>0.184</td>
<td>0.130</td>
</tr>
<tr>
<td>50%</td>
<td>0.189</td>
<td>0.140</td>
<td>0.132</td>
<td>0.205</td>
<td>0.133</td>
<td>0.167</td>
<td>0.134</td>
<td>0.350</td>
<td>0.123</td>
<td>0.191</td>
<td>0.129</td>
</tr>
<tr>
<td>60%</td>
<td>0.195</td>
<td>0.134</td>
<td>0.147</td>
<td>0.230</td>
<td>0.188</td>
<td>0.136</td>
<td>0.148</td>
<td>0.182</td>
<td>0.222</td>
<td>0.167</td>
<td>0.142</td>
</tr>
<tr>
<td>70%</td>
<td>0.443</td>
<td>0.320</td>
<td>0.200</td>
<td>0.488</td>
<td>0.231</td>
<td>0.256</td>
<td>0.241</td>
<td>0.408</td>
<td>0.229</td>
<td>0.450</td>
<td>0.285</td>
</tr>
<tr>
<td>80%</td>
<td>0.447</td>
<td>0.293</td>
<td>0.226</td>
<td>0.461</td>
<td>0.231</td>
<td>0.288</td>
<td>0.249</td>
<td>0.564</td>
<td>0.226</td>
<td>0.448</td>
<td>0.272</td>
</tr>
<tr>
<td>90%</td>
<td>0.528</td>
<td>0.302</td>
<td>0.239</td>
<td>0.545</td>
<td>0.237</td>
<td>0.354</td>
<td>0.260</td>
<td>0.556</td>
<td>0.225</td>
<td>0.528</td>
<td>0.277</td>
</tr>
<tr>
<td>100%</td>
<td>0.552</td>
<td>0.273</td>
<td>0.207</td>
<td>0.569</td>
<td>0.205</td>
<td>0.353</td>
<td>0.229</td>
<td>0.588</td>
<td>0.194</td>
<td>0.552</td>
<td>0.246</td>
</tr>
</tbody>
</table>
5.3. NUMERICAL ILLUSTRATIONS

Table 5.6: Goodness-of-fit performance with exponential NHPP-based SRMs.

<table>
<thead>
<tr>
<th>time</th>
<th>MLL</th>
<th>AIC</th>
<th>BIC</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>6.737</td>
<td>-9.474</td>
<td>-10.701</td>
<td>0.093</td>
</tr>
<tr>
<td>20%</td>
<td>13.082</td>
<td>-22.164</td>
<td>-21.559</td>
<td>0.218</td>
</tr>
<tr>
<td>30%</td>
<td>16.843</td>
<td>-29.686</td>
<td>-28.270</td>
<td>0.179</td>
</tr>
<tr>
<td>40%</td>
<td>20.260</td>
<td>-36.519</td>
<td>-34.430</td>
<td>0.184</td>
</tr>
<tr>
<td>50%</td>
<td>22.633</td>
<td>-41.266</td>
<td>-38.749</td>
<td>0.195</td>
</tr>
<tr>
<td>60%</td>
<td>15.390</td>
<td>-26.780</td>
<td>-23.849</td>
<td>0.188</td>
</tr>
<tr>
<td>70%</td>
<td>12.576</td>
<td>-21.151</td>
<td>-17.930</td>
<td>0.455</td>
</tr>
<tr>
<td>80%</td>
<td>7.528</td>
<td>-11.055</td>
<td>-7.533</td>
<td>0.447</td>
</tr>
<tr>
<td>90%</td>
<td>2.772</td>
<td>-1.545</td>
<td>2.198</td>
<td>0.528</td>
</tr>
<tr>
<td>100%</td>
<td>-6.912</td>
<td>17.825</td>
<td>21.803</td>
<td>0.552</td>
</tr>
</tbody>
</table>

Next, we evaluate the nonparametric maximum likelihood estimation. Table 5.8 presents the comparison of MLL and MSE in both NHPP-based SRMs and NHGP-based SRMs with constrained NPMLEs, respectively. Note that the degree of freedom in the nonparametric approach is larger than that in the parametric approach, so that it is impossible to evaluate AIC and BIC in the nonparametric maximum likelihood estimation. Similar to the comparison in Tables 5.6 and 5.7, even though the nonparametric maximum likelihood estimation method is applied, it can be seen that the NHGP-based SRM outperforms the NHPP-based SRM in the initial and middle testing phases (from 10% to 80% observation points) and that both SRMs give the exactly same MLL values in the latter testing phase. On the other hand, focusing on MSE, it is found that NHGP-based SRM gives the smaller MSE in the early testing phase (from 10% to 80% observation points).

In Table 5.9, we compare the results of parametric NHGP-based SRMs and the nonparametric NHGP-based SRM, where the best SRM among eleven trend functions is selected. From this result, we can see that the NHGP-based SRM with CNPMLE shows the smaller MLL in almost all cases except for 30% observation point. On the other hand, it can be checked that the NHGP-based SRM with CNPMLE provides the smaller MSE in initial and middle phases (from
Table 5.7: Goodness-of-fit performance with exponential NHGP-based SRMs.

<table>
<thead>
<tr>
<th>time</th>
<th>MLL</th>
<th>AIC</th>
<th>BIC</th>
<th>MSE</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>16.730</td>
<td>-27.460</td>
<td>-28.631</td>
<td>0.306</td>
<td>384</td>
</tr>
<tr>
<td>20%</td>
<td>23.201</td>
<td>-40.402</td>
<td>-39.494</td>
<td>0.208</td>
<td>17</td>
</tr>
<tr>
<td>30%</td>
<td>26.656</td>
<td>-47.312</td>
<td>-44.995</td>
<td>0.186</td>
<td>8</td>
</tr>
<tr>
<td>40%</td>
<td>26.491</td>
<td>-46.982</td>
<td>-43.848</td>
<td>0.180</td>
<td>3</td>
</tr>
<tr>
<td>50%</td>
<td>30.369</td>
<td>-54.737</td>
<td>-50.850</td>
<td>0.189</td>
<td>3</td>
</tr>
<tr>
<td>60%</td>
<td>24.111</td>
<td>-42.221</td>
<td>-37.824</td>
<td>0.195</td>
<td>3</td>
</tr>
<tr>
<td>70%</td>
<td>13.807</td>
<td>-21.614</td>
<td>-16.781</td>
<td>0.443</td>
<td>2</td>
</tr>
<tr>
<td>80%</td>
<td>7.528</td>
<td>-9.055</td>
<td>-3.772</td>
<td>0.447</td>
<td>1</td>
</tr>
<tr>
<td>90%</td>
<td>2.772</td>
<td>0.455</td>
<td>6.069</td>
<td>0.528</td>
<td>1</td>
</tr>
<tr>
<td>100%</td>
<td>-6.912</td>
<td>19.825</td>
<td>25.792</td>
<td>0.552</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.8: Goodness-of-fit performance with nonparametric maximum likelihood estimation.

<table>
<thead>
<tr>
<th>time</th>
<th>MLL NHPP</th>
<th>MSE NHPP</th>
<th>MLL NHGP</th>
<th>MSE NHGP</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>8.314</td>
<td>0.000</td>
<td>45.433</td>
<td>0.201</td>
</tr>
<tr>
<td>20%</td>
<td>13.577</td>
<td>0.051</td>
<td>28.988</td>
<td>0.144</td>
</tr>
<tr>
<td>30%</td>
<td>19.379</td>
<td>0.123</td>
<td>24.759</td>
<td>0.097</td>
</tr>
<tr>
<td>40%</td>
<td>21.531</td>
<td>0.101</td>
<td>29.867</td>
<td>0.080</td>
</tr>
<tr>
<td>50%</td>
<td>23.386</td>
<td>0.094</td>
<td>34.651</td>
<td>0.075</td>
</tr>
<tr>
<td>60%</td>
<td>18.598</td>
<td>0.083</td>
<td>31.576</td>
<td>0.062</td>
</tr>
<tr>
<td>70%</td>
<td>17.889</td>
<td>0.167</td>
<td>23.388</td>
<td>0.141</td>
</tr>
<tr>
<td>80%</td>
<td>15.604</td>
<td>0.299</td>
<td>18.508</td>
<td>0.249</td>
</tr>
<tr>
<td>90%</td>
<td>10.729</td>
<td>0.279</td>
<td>10.729</td>
<td>0.279</td>
</tr>
<tr>
<td>100%</td>
<td>2.065</td>
<td>0.249</td>
<td>2.065</td>
<td>0.249</td>
</tr>
</tbody>
</table>
5.3. NUMERICAL ILLUSTRATIONS

Table 5.9: Comparison of Best parametric models with CNPMLE in NHGP-based SRMs.

<table>
<thead>
<tr>
<th>time</th>
<th>MLL</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best trend</td>
<td>CNPMLE</td>
</tr>
<tr>
<td>10%</td>
<td>18.286 (lxvmax)</td>
<td>45.433</td>
</tr>
<tr>
<td>20%</td>
<td>25.141 (lxvmax)</td>
<td>28.988</td>
</tr>
<tr>
<td>30%</td>
<td>27.367 (lxvmax)</td>
<td>24.759</td>
</tr>
<tr>
<td>40%</td>
<td>27.319 (lxvmax)</td>
<td>29.867</td>
</tr>
<tr>
<td>50%</td>
<td>31.742 (lxvmax)</td>
<td>34.651</td>
</tr>
<tr>
<td>60%</td>
<td>24.682 (lxvmin)</td>
<td>31.576</td>
</tr>
<tr>
<td>70%</td>
<td>17.998 (lxvmax)</td>
<td>23.388</td>
</tr>
<tr>
<td>80%</td>
<td>12.996 (lxvmax)</td>
<td>18.508</td>
</tr>
<tr>
<td>90%</td>
<td>5.824 (lxvmax)</td>
<td>10.729</td>
</tr>
<tr>
<td>100%</td>
<td>-3.148 (lxvmax)</td>
<td>2.065</td>
</tr>
</tbody>
</table>
Finally, we illustrate the estimation behavior of the cumulative number of software faults detected at 20%, 60%, 80% and 100% observation points in Figs. 5.3-5.14. In Figs. 5.3-5.6, we apply the parametric NHPP-based SRM with a trend function ($lxv_{max}$) and nonparametric NHPP-based SRM. On the other hand, we depict the cumulative number of software faults with parametric NHGP-based SRM with a trend function ($lxv_{max}$) and nonparametric NHGP-based SRM in Figs. 5.7-5.10, where the approximation formula in Eq.(5.15) is applied. In Figs. 5.11-5.14, we have the similar plots to Figs. 5.7-5.10 when the Monte Carlo simulation is applied. In Figs. 5.3-5.14, we find that all SRMs carefully selected in respective methods show the quite similar tendency, especially in the last testing phase. Comparing Figs. 5.7-5.10 with Figs. 5.11-5.14, the estimated curves are almost identical. Our numerical calculation indicates that the approximation formula in Eq.(5.15) works very well even in the early testing phase with a relatively small testing time. Surprisingly, this is true even if the testing time $t$ is rather small (20% observation point). This result enables us to apply the approximate formula in Eq.(5.15) to calculate the expected cumulative number of software faults in NHGP-based SRMs.
5.3. NUMERICAL ILLUSTRATIONS

Figure 5.3: Estimating the cumulative number of software faults with NHPP-based SRMs (20% observation point).

Figure 5.4: Estimating the cumulative number of software faults with NHPP-based SRMs (60% observation point).
CHAPTER 5. NPMLE FOR NHGP-BASED SRMS

Figure 5.5: Estimating the cumulative number of software faults with NHPP-based SRMs (80% observation point).

Figure 5.6: Estimating the cumulative number of software faults with NHPP-based SRMs (100% observation point).
Figure 5.7: Estimating the cumulative number of software faults with NHGP-based SRMs: analytical approach (20% observation point).

Figure 5.8: Estimating the cumulative number of software faults with NHGP-based SRMs: analytical approach (60% observation point).
CHAPTER 5. NPMLE FOR NHGP-BASED SRMS

Figure 5.9: Estimating the cumulative number of software faults with NHGP-based SRMs: analytical approach (80% observation point).

Figure 5.10: Estimating the cumulative number of software faults with NHPP-based SRMs: analytical approach (100% observation point).
Figure 5.11: Estimating the cumulative number of software faults with NHGP-based SRMs: analytical approach (20% observation point).

Figure 5.12: Estimating the cumulative number of software faults with NHGP-based SRMs: analytical approach (60% observation point).
Figure 5.13: Estimating the cumulative number of software faults with NHGP-based SRMs: analytical approach (80% observation point).

Figure 5.14: Estimating the cumulative number of software faults with NHPP-based SRMs: analytical approach (100% observation point).
Chapter 6

Optimal Software Release Decision Based on Nonparametric Inference Approach

Even in estimating software reliability with any nonparametric method, it is indeed needed to predict the number of software failures which may occur in the future, after releasing software to the market or users. In this chapter we consider a software release decision on when to stop the software testing by minimizing the expected total software cost, under the assumption that the probability law of software fault-detection process is unknown. We focus on the nonparametric prediction method of a non-homogeneous Poisson process (NHPP) by Sofer and Miller [40] and apply it to the optimal software release problem. We calculate the predictive confidence interval as well as the point estimate of the optimal software release time. Since our method is based on a predictive approach in spite of its nonparametric nature, it is useful to make flexible decision making to determine the optimal software release timing.

6.1 NHPP-Based Software Reliability Modeling

6.1.1 Model Description

Suppose that the system test of a software product starts at time $t = 0$. Let $\{N(t), t \geq 0\}$ be the cumulative number of software faults detected by time $t$ and
be a stochastic (non-decreasing) counting process. In particular it is said that
\( N(t) \) is a non-homogeneous Poisson process (NHPP) if the following conditions hold:

- \( N(0) = 0 \),
- \( \{N(t), t \geq 0\} \) has independent increments,
- \( \Pr\{N(t + \Delta t) - N(t) \geq 2\} = o(\Delta t) \),
- \( \Pr\{N(t + \Delta t) - N(t) = 1\} = \lambda(t; \theta)\Delta t + o(\Delta t) \),

where \( \lambda(t; \theta) \) is the intensity function of an NHPP and denotes the instantaneous fault-detection rate per each fault. In the above definition, \( \theta \) is the model parameter (vector) included in the intensity function, and \( o(\Delta t) \) is the higher term of \( \Delta t \). Then the probability mass function (p.m.f.) of the NHPP is given by

\[
\Pr\{N(t) = n\} = \frac{[\Lambda(t; \theta)]^n}{n!} e^{-\Lambda(t; \theta)},
\]

\[
\Lambda(t; \theta) = \int_0^t \lambda(x; \theta)dx,
\]

where the function \( \Lambda(t; \theta) = E[N(t)] \) is called the mean value function and indicates the expected cumulative number of software faults detected by time \( t \). Hence, if the mean value function \( \Lambda(t; \theta) \) or the intensity function \( \lambda(t; \theta) \) is specified, the identification problem of an NHPP is reduced to a statistical estimation problem of unknown model parameter \( \theta \). In this way, when the parametric form of the mean value function or the intensity function is given, the resulting NHPP-based SRMs are called the parametric NHPP-based SRMs. Okamura and Dohi [48] summarize the most typical parametric NHPP-based based on eleven mean value functions and give a parameter estimation tool.

6.1.2 Nonparametric Point Estimation

If the parametric form of the mean value function or the intensity function is unknown, the identification problem of an NHPP becomes more difficult. Here we overview a nonparametric prediction approach by Sofer and Miller [40]. Consider a plausible case where the intensity function \( \lambda(t) \) is completely unknown. This kind of situation may occur when a new software product is
developed and the fault-detection patterns cannot be known from the past experience. The most intuitive but simplest method to estimate the intensity function is a piecewise-linear interpolation. For the fault-detection time data \( \chi_n = \{t_1, \ldots, t_n\} \) and the observation point \( T (\geq t_n) \), we define a naive estimator \( \hat{\lambda}(t | \chi) = 1/(t_i - t_{i-1}) \) \( (t_{i-1} < t \leq t_i; \ i = 1, 2, \ldots, n, \ t_0 = 0) \). Sofer and Miller [40] apply the following step-function estimate with breakpoints \( t_i \):

\[
\hat{\lambda}(t | \chi_n) = \begin{cases} 
  i + (t - t_i)/(t_{i+1} - t_i); & t_i < t \leq t_{i+1}, \\
  n + \delta(t - t_n)/(T - t_n); & t_n \leq t \leq T,
\end{cases} \tag{6.3}
\]

where \( \delta \in [0, 1] \) is an arbitrary turning parameter. The resulting estimate of the mean value function in Eq.(6.3) can be obtained by plotting \( n \) failure points and connecting them by line segments. Since it is common that only one time-series sample \( t_i \) \( (i = 0, 1, 2, \ldots, n) \), is available in each software testing, this estimator seems to be the straightforward but the most natural one of the cumulative number of software faults, because it is always possible to reduce the mean squares error from the underlying software fault-detection time data to zero. However, since such a nonparametric model possesses the same degree of freedom as the number of fault-detection time data, the resulting estimate of the intensity function is discontinuous everywhere and tends to fluctuate with large noise. This leads to the so-called overfitting problem, so that the model can fit only the training (observed) data with higher accuracy but cannot predict the unknown pattern in future. Almost all nonparametric NHPP-based SRMs [42],[45],[85] are trapped into this overfitting problem.

Apart from the traditional nonparametric approach in the above works, Sofer and Miller [40] propose a smoothing technique of the intensity function, by using a quadratic programming, for the purpose of prediction. For convenience, we divide the time interval \( [0, T] \) into \( k \) intervals of equal length \( \theta = T/k \), and define an alternative time scale \( s_i = i\theta \) \( (i = 1, 2, \ldots, k) \). In each discrete time \( s_i \), define an estimate \( \hat{\lambda}(s_i | \chi_n) = \hat{m}_i \) \( (\chi_n = (t_1, \ldots, t_n); \ i = 1, 2, \ldots, k) \) in Eq.(6.3) and its first order difference:

\[
\dot{r}_i = \{\hat{m}_i \quad \hat{m}_{i-1}\} / \theta. \tag{6.4}
\]
More generally, we define the order $j$ backward difference operator $\Delta^j$ by

$$\Delta^j r_i = \Delta^j 1 r_i \Delta^j 1 r_{i-1}, \quad j = 2, 3, \ldots,$$

(6.5)

where $\Delta^0 r_i = r_i$ and $\Delta^1 r_i = r_i - r_{i-1}$. The sequence $r_i (i = 1, 2, \ldots, k)$ is said to be completely monotone if

$$(1)^j \Delta^j r_i \geq 0, \quad j + 1 \leq i, \quad j = 0, 1, \ldots$$

(6.6)

The above property seems to be somewhat restrictive in analysis but is satisfied in almost all NHPP-based SRMs [30]. Since the estimate of intensity function at each discrete time $s_i (i = 1, \ldots, k)$ is given by $(\hat{r}_1, \ldots, \hat{r}_k)$ from Eq.(6.4), the next step is to smooth it under some criterion. Sofer and Miller [40] formulate the smoothing procedure by the following quadratic programming:

$$\min_{m_i} : \sum_{i=1}^{k} \omega_i (m_i - \hat{m}_i)^2$$

subject to:

$$(1)^{d+1} \Delta^d m_i \geq 0$$

$$(1)^{j+1} \Delta^j m_{k+1} \geq 0$$

$$d \leq i \leq k + l, \quad 1 \leq j \leq d - 1$$

$$m_k = n + \delta, \quad m_0 = 0, \quad k > 0.$$  

(6.7)

In Eq.(6.7), $\omega_i (i = 1, 2, \ldots, k)$ are weight parameters and $l$ is an arbitrary integer ($l = 0$ at the moment). Since two parameters $\delta$ and $\omega_i$ are arbitrary, $\delta = \omega_i = 1$ may be a plausible choice for simplify the analysis. Finally, the resulting estimates $(m_1, \ldots, m_k)$ satisfy the completely monotone property with order $d$, which can be adjusted to guarantee the smoothness and accuracy for the goodness-of-fit performance.

### 6.2 Nonparametric Inference

In Subsection 6.1.2, we gave an overview of a nonparametric point estimation of the intensity function under completely monotone property. Based on the fault-detection time experienced in past, we want to predict the future value of the intensity function or the mean value function at the observation point $T$. In parametric modeling, the prediction is easily done by substituting estimated model parameter $\theta$ into the time evolution $\lambda(t; \theta)$. In nonparametric model,
6.2. NONPARAMETRIC INFERENCE

however, since \( \hat{m}_i \), \( m_i \), and \( \hat{r}_i \) are the functions of data, it is difficult to represent
these functions as any time evolitional functions at an arbitrary time greater
than the present observation point \( s_k \). Barghout et al. [42] restrict their attention
to a one-stage look ahead prediction on the next software fault-detection
time. Kaneishi and Dohi [45] also apply the weighted kernel function to get a
short-term prediction problem. Since the optimal software release problems are
essentially based on the long-term prediction of an NHPP, however, the above
two methods do not work well. To our best knowledge, the long-term prediction
by Sofer and Miller [40] is an applicable and unique solution to predict the future
values of intensity function and mean value function under the nonparametric
assumption.

Consider the completely monotone sequence of an intensity function with or-
der d by \( (r_1, \ldots, r_k) \). For an arbitrary prediction point \( l \), the sequence \( (r_{k+1}, \ldots, r_{k+l}) \) is defined to be a feasible completely monotone extrapolation of order d
for the past observation, if the sequence \( (r_1, \ldots, r_{k+l}) \) is completely monotone
up to order d so as to satisfy

\[
(1)^d \Delta^d r_i \geq 0, \quad d + 1 \leq i \leq k + l, \quad (6.8)
\]

\[
(1)^j \Delta^j r_{k+l} \geq 0, \quad 0 \leq j \leq d - 1. \quad (6.9)
\]

The above extrapolations result a lower bound for all feasible extrapolations of
order d, if any other such extrapolations \( (\Sigma_{k+1}, \ldots, \Sigma_{k+l}) \) satisfy \( \Sigma_{k+i} \leq r_{k+i} \) for
\( i = 1, 2, \ldots, l \). Similarly, it constitutes an upper envelop if \( r_{k+i} \geq r_{k+i} \) for all
\( i \). Sofer and Miller [40] derive conditions for the existence for such upper and
lower envelops for the completely monotone extrapolations.

In what follows, we summarize the results by Sofer and Miller [40] in short.
For \( d = 1 \) and \( d = 2 \), the sequence \( (r_1, \ldots, r_k) \) can be extrapolated into the
future by letting \( r_{k+i} = r_k \) (\( i = 1, \ldots, l \)). This extrapolation is clearly the
upper envelop for all completely monotone extrapolations of \( d = 1 \) and \( d = 2 \). For \( d = 1 \), the extrapolation \( r_{k+i} = 0 \) is a lower envelop for all isotone
extrapolations. For \( d = 2 \), let \( (r_1, \ldots, r_k) \) denote a feasible solution to Eqs.(6.8)
and (6.9) with \( l = 0 \). Then, for an arbitrary \( l > 0 \), the extrapolation

\[
\Sigma_{k+i} = \begin{cases} 
  r_k + i \Delta^1 r_k; & i = 1, 2, \ldots, p_r \\
  0; & i = p_r + 1, p_r + 2, \ldots, l 
\end{cases} \quad (6.10)
\]
is a lower envelope for all feasible extrapolations of $d = 2$ to $(r_1, \ldots, r_k)$, where

$$p_r = \begin{cases} \text{gilb}(r_k / \Delta^1 r_k); & \Delta^1 r_k < 0 \\ l; & \Delta^1 r_k = 0. \end{cases} \tag{6.11}$$

Throughout this chapter, the function gilb($\cdot$) means the greatest integer lower bound. For $d = 3$, a solution $(r_1, \ldots, r_k)$ satisfying Eq.(6.8) and Eq.(6.9) with $l = 0$ can be extrapolated to a solution $(r_1, \ldots, r_{k+1})$ which satisfies Eq.(6.8) and Eq.(6.9) with $l > 0$ if and only if

$$r_k + j \Delta^1 r_k + j(j + 1) \Delta^2 r_k / 2 \geq 0, \ j = 1, 2, \ldots, l. \tag{6.12}$$

Then, the upper and lower envelopes of all feasible extrapolations for $d = 3$ are given by

$$\tilde{r}_{k+i} = \begin{cases} r_k + i \Delta^1 r_k + i(i + 1) \Delta^2 r_k / 2; & i = 1, 2, \ldots, q_r \\ r_k + i \Delta^1 r_k; & i = q_r + 1, q_r + 2, \ldots, l \end{cases} \tag{6.13}$$

and

$$\bar{r}_{k+i} = \begin{cases} r_k + i \Delta^1 r_k; & i = 1, 2, \ldots, p_r, \ p_r \geq l \\ \tilde{r}_{k+i}; & i = 1, 2, \ldots, l, \ p_r \leq l, \end{cases} \tag{6.14}$$

where

$$q_r = \begin{cases} \text{gilb}(\Delta^1 r_k / \Delta^2 r_k); & \Delta^2 r_k > 0 \\ l; & \Delta^2 r_k = 0, \end{cases} \tag{6.15}$$

$$u = \min \{l, 1 + \text{gilb}(2r_k / \Delta^1 r_k)\}. \tag{6.16}$$

In a fashion similar to the intensity function, we can derive the envelopes for prediction for the mean value function. Let $(m_1, \ldots, m_k)$ be a sequence of the mean value function with $d = 4$ satisfying the constraints in Eq.(6.7). The sequence $(m_{k+1}, \ldots, m_{k+l})$ is defined to be a feasible extrapolation for $(m_1, \ldots, m_k)$ with $d = 4$ satisfying the above conditions. This extrapolation constitutes upper and lower bounds for all feasible extrapolations of $d = 4$, if
any other such extrapolations \((\tilde{m}_{k+1}, \ldots, \tilde{m}_{k+l})\) and \((\hat{m}_{k+1}, \ldots, \hat{m}_{k+l})\) satisfy \(\tilde{m}_{k+i} \geq m_{k+i}\) and \(\hat{m}_{k+i} \leq m_{k+i}\) \((i = 1, \ldots, l)\), respectively. Consider the constraints in Eq. (6.7) with \(d = 4\) and fixed \(l > 0\). A solution \((m_1, \ldots, m_k)\) satisfying Eq. (6.7) with \(l = 0\) can be extrapolated to a solution \((m_1, \ldots, m_{k+l})\) which satisfies Eq. (6.7) with \(l > 0\) if and only if

\[
\Delta^1 m_k + j \Delta^2 m_k + j(j + 1) \Delta^3 m_k / 2 \geq 0, \quad j = 1, 2, \ldots, i.
\] (6.18)

Then, the upper and lower envelopes of all feasible extrapolations for \(d = 4\) are given by

\[
\tilde{m}_{k+i} = \begin{cases} 
  m_k + i \Delta^1 m_k + i(i + 1) \Delta^2 m_k / 2 \\
  + i(i + 1)(i + 2) \Delta^3 m_k / 6; \\
  i = 1, 2, \ldots, q_m \\
  m_k + q_m + (i - q_m) \alpha; \\
  i = q_m + 1, q_m + 2, \ldots, l 
\end{cases}
\] (6.19)

and

\[
\hat{m}_{k+i} = \begin{cases} 
  m_k + i \Delta^1 m_k + i(i + 1) \Delta^2 m_k / 2; \\
  i = 1, 2, \ldots, p_m; \quad p_m \geq l \\
  \hat{m}_{k+i}; \quad i = 1, 2, \ldots, l; \quad p_m \leq l, 
\end{cases}
\] (6.20)

respectively, where

\[
\tilde{m}_{k+i} = \begin{cases} 
  m_k + i \Delta^1 m_k + i(i + 1) \Delta^2 m_k / 2 \\
  i(i + 1)(i + 2)(\Delta^1 m_k \\
  + u \Delta^2 m_k) / 3u(u + 1); \\
  i = 1, 2, \ldots, u \\
  m_k + u; \quad i = u + 1, \ldots, l, 
\end{cases}
\] (6.21)

with

\[
p_m = \begin{cases} 
  \text{gilb}(\Delta^1 m_k / \Delta^2 m_k); \quad \Delta^2 m_k < 0 \\
  l; \quad \Delta^2 m_k = 0, 
\end{cases}
\] (6.22)

\[
q_m = \begin{cases} 
  \text{gilb}(\Delta^2 m_k / \Delta^3 m_k); \quad \Delta^3 m_k > 0 \\
  l; \quad \Delta^3 m_k = 0, 
\end{cases}
\] (6.23)

\[
\alpha = \Delta^1 m_k + q_m \Delta^2 m_k + q_m(q_m + 1) \Delta^3 m_k / 2, 
\] (6.24)

\[
u = \min \{l, 1 + \text{gilb}(2\Delta^1 m_k / \Delta^2 m_k)\}.
\] (6.25)
6.3 Optimal Software Release Decision

Suppose that the system test starts at $t = 0$ and terminates at $t = t_0$. Let $T_L$ be the software lifetime or the upper limit of the software warranty period, where the time length $(t_0, T_L]$ denotes the operational period of software and the time length $(0, t_0]$ is the testing period. Without loss of generality, it is assumed that $t_0 \leq \tilde{t}_0$, where $\tilde{t}_0 (\geq 0)$ is the upper limit of the software testing time and the delivery schedule of software product. When the fault-detection time data $\chi_n = \{t_1, \ldots, t_n\}$ are observed at time $T (0 < T \leq t_0)$, the model parameter $\theta$ is estimated with $\chi_n$ in parametric NHPP-based SRMs. Define the following cost components:

- $c_0 (> 0)$: testing cost per unit system testing time,
- $c_1 (> 0)$: removal cost per fault in system testing phase,
- $c_2 (> c_1)$: removal cost per fault in operational phase.

Based on the above cost parameters, the expected total software cost is given by

$$C(t_0; T; \theta) = c_0 t_0 + c_1 \left\{ n + \Lambda(t_0; \theta) \Lambda(T; \theta) \right\}$$

$$+ c_2 \left\{ \Lambda(T_L; \theta) n \left[ \Lambda(t_0; \theta) \Lambda(T; \theta) \right] \right\}.$$  \hspace{1cm} (6.26)

Note that the above formulation is somewhat different from the past references [55],[70],[68],[67], because the observation point (decision point) $T (\geq 0)$ is explicitly involved in the formulation. At an arbitrary observation point $T$, we wish to know when to stop software testing. Substituting $n = 0$ and $\Lambda(T; \theta) = 0$ in Eq.(6.26), the problem is to minimize the expected total software cost $C(t_0; 0; \theta)$. However, it is evident that the optimal software release time $t^*_0$ minimizing $C(t_0; 0; \theta)$ is independent of $T$ and may be smaller than $T$.

In addition the parameter $\theta$ also depends on the observation point $T$. In our formulation, differentiating $C(t_0; T; \theta)$ with respect to $t_0$ yields $dC(t_0; T; \theta)/dt_0 = dC(t_0; 0; \theta)/dt_0 = c_0 (c_2 - c_1) \lambda(t_0; \theta)$ and $d^2C(t_0; T; \theta)/dt_0^2 = d^2C(t_0; 0; \theta)/dt_0^2 = (c_2 - c_1) \lambda'(t_0; \theta)$. If the mean value function is strictly increasing and convex in $t_0$, i.e. $d^2C(t_0; T; \theta)/dt_0^2 > 0$, the function $C(t_0; T; \theta)$ is strictly convex. Hence, if $c_0 (c_2 - c_1) \lambda(T; \theta) < 0$ and $c_0 (c_2 - c_1) \lambda(\tilde{t}_0; \theta) > 0$, there exists
6.3. OPTIMAL SOFTWARE RELEASE DECISION

a unique optimal software release time $t_0^*$ ($T < t_0^* < l_0$), otherwise, $t_0^* = l_0$ or $t_0^* = T$. That is, if the degree of complete monotonicity is more than 2, it is obvious that the optimal software release time exists for discrete time case. Furthermore, $t_0^* = T$ implies that software test should stop immediately at the observation point, and $t_0^* = l_0$ does that software test should continue as long as possible. Of course, since stopping at $t_0^*$ leads to continue testing until time $t_0^*$, it is possible to update the release timing at each decision (observation) point before time $t_0^*$, because the estimation result of $\theta$ may change at each decision point.

In our nonparametric scheme, we formulate the optimal software release time in discrete setting. Suppose that $(m_1, \ldots, m_k)$ are observed at discrete time $s_i = i\theta$ ($i = 1, 2, \ldots, k$). We predict at time $s_k$ the future behavior of mean value function $(m_{k+1}, m_{k+2}, \ldots, m_{k+l})$, where $l \geq 0$ denotes an arbitrary prediction point. From the results by Sofer and Miller [40], we get the upper and lower predictions of mean value function: $(\bar{m}_{k+1}, \bar{m}_{k+2}, \ldots, \bar{m}_{k+l})$ and $(\underline{m}_{k+1}, \underline{m}_{k+2}, \ldots, \underline{m}_{k+l})$, which correspond to optimistic and pessimistic predictions of the cumulative number of software faults at $s_{k+1}, s_{k+2}, \ldots, s_{k+l}$. Then, we formulate the following two optimization problems, $\min_{l_0} \bar{C}(l_0; s_k)$ and $\min_{l_0} \underline{C}(l_0; s_k)$, where

$$\bar{C}(l_0; s_k) = c_0 s_{k+l_0} + c_1 \left\{ n + \bar{m}_{k+l_0} \bar{m}_k \right\} + c_2 \left\{ \bar{m}_L n \left[ \bar{m}_{k+l_0} \bar{m}_k \right] \right\},$$  \hspace{1cm} (6.27)

and

$$\underline{C}(l_0; s_k) = c_0 s_{k+l_0} + c_1 \left\{ n + \underline{m}_{k+l_0} \underline{m}_k \right\} + c_2 \left\{ \underline{m}_L n \left[ \underline{m}_{k+l_0} \underline{m}_k \right] \right\}. \hspace{1cm} (6.28)$$

In Eqs.(6.27) and (6.28), $\bar{m}_L$ and $\underline{m}_L$ are predictions of the total number of software faults during the software life cycle in the sense of optimistic and pessimistic predictions. The problem is to seek $l_0^* = \{l_0 = 0, 1, \ldots, \bar{l}; \min_{l_0} \bar{C}(l_0; s_k)\}$ and $l_0^* = \{l_0 = 0, 1, \ldots, \bar{l}; \min_{l_0} \underline{C}(l_0; s_k)\}$ with the delivery time limit $s_{k+l}$ ($l = 0, 1, \ldots$). Since these optimization problems are finite dimension problems, it is relatively easy to enumerate all the possible $l_0^*$ and $l_0^*$. 
6.4 Numerical Illustrations

We give numerical examples to predict the optimal software release time based on the nonparametric inference approach, where two data sets, DS1 and DS2, are used for analysis [29]. DS1 consists of 397 software fault-detection time data with the maximum value 108,840 (min). The cost parameters are given by $c_0 = 53$, $c_1 = 50000$ and $c_2 = 100000$. On the other hand, the data analysis with DS2 treats 207 software fault-detection time data with the maximum value 16,656 (sec), where $c_0 = 8$, $c_1 = 1000$ and $c_2 = 2000$ are assumed. Cost parameters are set to avoid the trivial case where both optimal solutions of nonparametric and parametric models are $l_0 = l$ or $l_0 = k$. Figures 6.1 and 6.2 illustrate the behavior of cumulative number of software faults detected in the testing phase in DS1 and DS2, respectively. In the analysis, we set the upper limit of release time as $t_0 = 108,840$ and $t_0 = 16,656$ for DS1 and DS2, respectively. Also, we assume that the free warranty period (the end point of software life cycle) of each software depends on the size of software. Therefore, the free warranty period is given by 60 times length of each testing period in both cases with DS1 and DS2. Here we compare our nonparametric method with the common method based on the best parametric NHPP model out of eleven candidates [48], where the degree of complete monotonicity in the nonparametric method is given by $d = 4$. From the definition of the complete monotonicity, upper and lower predictions of mean value function with $d = 2$ always give the predictive optimal software release time as $l_0^* = l$ or $l_0^* = k$. Also, the upper and lower limits of mean value function of $d = 3$ shows quite similar results with that of $d = 4$. Therefore, we omit these cases for brevity. Here, the best parametric NHPP model is the model which shows the highest goodness of fit performance in the meaning of maximum log likelihood for the underlying data. Actually, these best parametric NHPP models are not equivalent to the models which give the closest predictive optimal software release time to the real optimal software release time. In each software development project, we estimate the optimal software release time with the whole data (i.e., 100%) by minimizing the expected total software cost [61],[62]. More specifically, the real optimal software release time in DS1 and DS2 are calculated as $t_0^* = 82,098$ and $t_0^* = 12,839$, respectively, which could minimize the both expected total software costs with
optimistic and pessimistic predictions in Eqs. (6.27) and (6.28). Tables 6.1 and 6.2 present the predictive optimal software release time at each observation point (60%, 70%, 80% and 90% point of upper limit of release time $t_0$) with the upper predictions of mean value function, where the relative errors between the predictive software release time and the real optimal solution are shown for easy understanding. From Tables 6.1 and 6.2, the nonparametric method by Sofer and Miller [40] provides the smaller relative error than the conventional parametric NHPP-based SRM, which is the best model among eleven candidates [48], where “lnorm”, “exp” and “lxvmin” indicate the lognormal, exponential and log-extreme-value models, respectively. Furthermore, it can be seen that relative error gets smaller as the data increases in the case of 60% and 70% observation points. This means that the predictive optimal software release time can be updated by adding software fault-detection time data. Since 80% and 90% observation points have already exceeded the real optimal release time, the predictive optimal software release time of these two cases is equivalent to the observation point. Therefore, it can be said that our nonparametric estimation method works well to judge when to stop the software testing.

Next we concern the interval estimation of the optimal software release time.
CHAPTER 6. SRP BASED ON NONPARAMETRIC APPROACH

Cumulative number of software faults

![Plot of cumulative number of software faults vs. time](image)

Figure 6.2: DS2.

<table>
<thead>
<tr>
<th>Time</th>
<th>Parametric Error</th>
<th>Nonparametric Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>65304 (lnorm) 20%</td>
<td>91426 11%</td>
</tr>
<tr>
<td>70%</td>
<td>76188 (lnorm) 7%</td>
<td>79453 3%</td>
</tr>
<tr>
<td>80%</td>
<td>87072 (lnorm) 6%</td>
<td>87072 6%</td>
</tr>
<tr>
<td>90%</td>
<td>97956 (lnorm) 19%</td>
<td>97956 19%</td>
</tr>
</tbody>
</table>

Table 6.1: Prediction of optimal software release time (DS1).

The (1 - α)100% two-sided predictive confidence intervals (PCIs) are calculated by the simulation-based bootstrap method [71],[70], where bootstrap samples are produced by thinning algorithm and the number of bootstrap samples is fixed as 1,000. That is, we generate bootstrap samples for each observation point. For each bootstrap sample, we derive upper and lower predictions of mean value function to derive the predictive optimal software release time. In Tables 6.3 and 6.4, we present the bootstrap mean, bootstrap median, bootstrap standard deviation (s.d.) of estimators on the optimal software release time and the (1 - α)100% two-sided predictive intervals for varying α = 0.01, 0.05, 0.10, 0.20, 0.30 with upper predictions of mean value function, where d = 4 is assumed. Since
### Table 6.2: Prediction of optimal software release time (DS2).

<table>
<thead>
<tr>
<th>time</th>
<th>parametric</th>
<th>error</th>
<th>nonparametric</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>10963 (exp)</td>
<td>15%</td>
<td>10993</td>
<td>14%</td>
</tr>
<tr>
<td>70%</td>
<td>12092 (exp)</td>
<td>6%</td>
<td>13491</td>
<td>5%</td>
</tr>
<tr>
<td>80%</td>
<td>13325 (lxvmin)</td>
<td>4%</td>
<td>13325</td>
<td>4%</td>
</tr>
<tr>
<td>90%</td>
<td>14990 (exp)</td>
<td>17%</td>
<td>14990</td>
<td>17%</td>
</tr>
</tbody>
</table>

even in the interval prediction case, no remarkable differences between the upper and lower predictions in Eqs. (6.27) and (6.28) cannot be recognized, we omit to show results of lower case here. In the case where the prediction point is smaller than the real optimal solution (60% and 70% points of whole data), the resulting predictive interval of the optimal release time can involve the real optimal solution. So, in the plausible cases, our method works well to give the predictive bounds. Looking at the first and second moments, and median of estimators, it can be seen that the bootstrap standard deviation decreases as software testing progresses. Furthermore, we can see that the length of predictive intervals becomes narrow, as the significance level $\alpha$ increases.

On the bootstrap point estimation, in almost all cases except 60% observation point, the bootstrap median is more closed to the real optimal solution than the bootstrap mean in DS1 and DS2.
Table 6.3: Predictive confidence interval of optimal software release time based on upper limit (DS1).

<table>
<thead>
<tr>
<th>time</th>
<th>mean</th>
<th>median</th>
<th>s.d.</th>
<th>$\alpha$</th>
<th>PCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>91414</td>
<td>93602</td>
<td>16849</td>
<td>0.3</td>
<td>[69658, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[66392, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[65304, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[65304, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[65304, 108840]</td>
</tr>
<tr>
<td>70%</td>
<td>85536</td>
<td>79453</td>
<td>12025</td>
<td>0.3</td>
<td>[76188, 105575]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[76188, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[76188, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[76188, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[76188, 108840]</td>
</tr>
<tr>
<td>80%</td>
<td>96241</td>
<td>91426</td>
<td>9714</td>
<td>0.3</td>
<td>[87072, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[87072, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[87072, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[87072, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[87072, 108840]</td>
</tr>
<tr>
<td>90%</td>
<td>98817</td>
<td>97956</td>
<td>2766</td>
<td>0.3</td>
<td>[97956, 97956]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[97956, 99044]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[97956, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[97956, 108840]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[97956, 108840]</td>
</tr>
</tbody>
</table>
### Table 6.4: Predictive confidence interval of optimal software release time based on upper limit (DS2).

<table>
<thead>
<tr>
<th>time</th>
<th>mean</th>
<th>median</th>
<th>s.d.</th>
<th>α</th>
<th>PCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>12659</td>
<td>11409</td>
<td>2797</td>
<td>0.3</td>
<td>[9994, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[9994, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[9994, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[9994, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[9994, 16656]</td>
</tr>
<tr>
<td>70%</td>
<td>14090</td>
<td>13575</td>
<td>2252</td>
<td>0.3</td>
<td>[11659, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[11659, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[11659, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[11659, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[11659, 16656]</td>
</tr>
<tr>
<td>80%</td>
<td>14627</td>
<td>13491</td>
<td>1506</td>
<td>0.3</td>
<td>[13325, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[13325, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[13325, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[13325, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[13325, 16656]</td>
</tr>
<tr>
<td>90%</td>
<td>15112</td>
<td>14990</td>
<td>411</td>
<td>0.3</td>
<td>[14990, 14990]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[14990, 14990]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[14990, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[14990, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[14990, 16656]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>time</th>
<th>mean</th>
<th>median</th>
<th>s.d.</th>
<th>α</th>
<th>PCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>12659</td>
<td>11409</td>
<td>2797</td>
<td>0.3</td>
<td>[9994, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[9994, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[9994, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[9994, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[9994, 16656]</td>
</tr>
<tr>
<td>70%</td>
<td>14090</td>
<td>13575</td>
<td>2252</td>
<td>0.3</td>
<td>[11659, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[11659, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[11659, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[11659, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[11659, 16656]</td>
</tr>
<tr>
<td>80%</td>
<td>14627</td>
<td>13491</td>
<td>1506</td>
<td>0.3</td>
<td>[13325, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[13325, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[13325, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[13325, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[13325, 16656]</td>
</tr>
<tr>
<td>90%</td>
<td>15112</td>
<td>14990</td>
<td>411</td>
<td>0.3</td>
<td>[14990, 14990]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2</td>
<td>[14990, 14990]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>[14990, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.05</td>
<td>[14990, 16656]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>[14990, 16656]</td>
</tr>
</tbody>
</table>
Chapter 7

Conclusions

7.1 Summary and Remarks

In Chapter 2, we have proposed two bootstrap methods which were categorized into simulation-based approach and re-sampling-based approach. We have also applied the proposed parametric bootstrap methods to derive the probability distributions of statistical estimators of the optimal periodic replacement time and the corresponding minimum expected cost in a periodic replacement problem with minimal repair. Furthermore, we have calculated the higher moments and two-sided confidence intervals, as well as the mean and median. As a result, it has been shown that the confidence intervals could be derived with bootstrap methods for the periodic replacement problem with minimal repair. We have also investigated several statistical properties with respect to various estimators through simulation experiments and real data analysis. It could be confirmed that the point estimate was included between the 95% two-sided confidence intervals from the results of simulation experiment. On the other hand, we have shown that the user could utilize the useful information in order to determine the periodic replacement time by using the proposed parametric bootstrap methods in real example.

In Chapter 3, we have considered a periodic replacement problem with minimal repair which minimizes the long-run average cost per unit time, and have estimated the optimal periodic replacement time under the incomplete knowledge on minimal repair process, via nonparametric estimation methods. We have proposed the kernel-based approaches and their application to the interval
CHAPTER 7. CONCLUSIONS

estimation with bootstrapping. Throughout simulation experiments, it has been shown that the kernel-based estimation with LLCV provided accurate point estimation and stable confidence intervals of the optimal periodic replacement time and its associated long-run average cost. The propose methods have also been applied to the analysis of the field data.

In Chapter 4, we have applied two nonparametric maximum likelihood estimators (NPMLEs) to the NHPP-based SRMs. The main feature of constrained NPMLE is to estimate the intensity function so as to construct an upper bound of the likelihood function. The NPMLE with failure rate function is to estimate directly the failure rate of fault-detection time distribution. Our numerical experiments with actual software fault-detection time data have suggested that the NPMLE by checking the monotone property of the failure rate showed the best performances in terms of the maximum likelihood. We have also shown the S-shaped estimator was more suitable to estimate several software reliability measures for almost all software fault-detection time data.

In Chapter 5, we have developed NHGP-based SRMs with trend functions and the nonparametric maximum likelihood estimation. Throughout numerical examples with a real software fault data, we have shown that the NHGP-based SRMs could provide the better goodness-of-fit performances in the initial to middle testing phases than the NHPP-based SRMs in terms of maximum log likelihood, and could be consistent with the NHPP-based SRMs in the latter testing phase. Especially, it has been clarified that the NHGP-based SRMs with constrained NPMLE could give the larger maximum log likelihood. These observations imply that the relatively simple SRM such as NHGP-based SRM can improve the goodness-of-fit performance in the earlier testing phase by adjusting the shape parameter $\kappa$, and that the nonparametric maximum likelihood estimation is also useful to get more accurate fitting result even if the trend function is unknown.

In Chapter 6, we have formulated a nonparametric optimal software release problem under the assumption that the mean value function of an NHPP-based SRM is unknown and derived both of point and interval predictions of the optimal software release time which minimizes the expected total software cost. We have given illustrative examples with two real software fault data to derive
estimates of the optimal software release time and performed the sensitivity analysis on the observation point and significance level.

7.2 Future Works

For the same preventive maintenance problem in Chapter 3, we will investigate the dependence of the bandwidth selection method on the point and interval estimations. For instance, any adaptive approach to estimate time-dependent bandwidth will be useful to improve the estimation accuracy. We will also apply the present techniques to the other maintenance problems. For instance, a block replacement problem [22] is another challenging issue because the renewal function has to be handled in the estimation framework.

For the software reliability assessment in Chapter 4, we will focus on the interval estimation of other software reliability measures (see [54],[86],[39]). For this purpose, we intend to apply the nonparametric bootstrap method which is a representative statistical approach to replicate the original software fault-detection time data (e.g. see [45]).

Also, we will investigate a trend renewal process for the software fault count data with different renewal property and trend functions.
Appendix A

Properties of Nonparametric Estimators

A.1 Convergence Property of Kernel Estimator

The non-homogeneous Poisson process (NHPP) model can be considered as a simple multiplicative intensity model which is a specific form of Aalen’s multiplicative intensity model [87]. Suppose that $x_1, x_2, \ldots, x_n$ are observations from an NHPP with intensity

$$\lambda_c(x) = c \alpha(x), \quad x \in [0,1],$$  \hspace{1cm} (A.1)

where $c$ is a positive constant, and $\alpha(x)$ is an unknown nonnegative deterministic function with $\int_0^1 \alpha(x)dx = 1$. The kernel intensity estimator of $\lambda(x)$ is defined by Eq.(3.12).

To consider convergence properties for kernel intensity estimator, we require the values of $c$ which come from the sequence of positive real numbers $\{c_s\}_{s=1}^{\infty}$ such that $c/s \to \tau$ for some constant $\tau > 0$ as $s \to \infty$. For the sequence of intensity functions $\lambda_{c_s}(x) = c_s \alpha(x)$, we construct a corresponding sequence of kernel estimators $\hat{\lambda}_s(x)$. Each bandwidth $h$ of these kernel estimators is dependent on $s$. From Ramlau-Hansen [78], it is known that $\hat{\lambda}_s(x)$ is uniformly consistent and asymptotically normal as $c_s \to \infty$, $h \to 0$ and $hc_s \to \infty$. 
A.2 Asymptotic Property of Bootstrap Methods [2]

Assume that the asymptotic model \( \lambda(x) = c \alpha(x) \) again, where the smooth function \( \alpha(x) \) is held and the scalar parameter \( c \geq 1 \) diverges. We consider the asymptotic theory describing the distribution of the stochastic process \( \hat{\lambda}(x) \), and its bootstrap samples \( \hat{\lambda}^*(x) \). Let us derive \( \hat{\lambda}^*(x) \) by any one of three different bootstrap approaches developed in Chapter 3, and write \( E'[\cdot] \) for expectation conditional on \( \chi = (x_1, x_2, \ldots, x_n) \).

Assume that \( K(\cdot) \) is compactly supported, so that \( K'(\cdot) \) exists and satisfies a Lipschitz condition of order 1. Suppose that \( 0 < \inf_x \alpha(x) \leq \sup_x \alpha(x) < \infty \) and \( \sup_x |\alpha'(x)| < \infty \), and that for some \( \rho > 0, c^{(3/10)+\rho} \leq h \leq c^{-\rho} \). Then we may write

\[
\hat{\lambda}(x) \quad E[\hat{\lambda}(x)] = \sqrt{\frac{c}{h}} \left( \sqrt{\alpha(x)} U_n(x) + R_{1n}(x) \right), \quad (A.2)
\]

\[
\hat{\lambda}^*(x) \quad E'[\hat{\lambda}^*(x)] = \sqrt{\frac{c}{h}} \left( \sqrt{\alpha(x)} U_n(x)^* + R_{1n}^*(x) \right), \quad (A.3)
\]

where \( U_n(x) \), and \( U_n^*(x) \) conditional on \( \chi \), are stationary Gaussian processes on \((0,1)\) with identical distribution, zero means, and covariances

\[
cov\{U_n(x_1), U_n(x_2)\} = \int K(y)K \left\{ y + \frac{x_1}{h}, \frac{x_2}{h} \right\} dy, \quad (A.4)
\]

and \( R_{1n}(x) \) and \( R_{1n}^*(x) \) are functions satisfying

\[
E \left[ \sup_{\| x \| \leq 1} |R_{1n}(x)|^k + \sup_{\| x \| \leq 1} \left| R_{1n}^*(x) \right|^k \right] = O(c^{-k\sigma}) \text{ as } c \to \infty \quad (A.5)
\]

for some \( \sigma > 0 \), and all \( k \geq 1 \) and \( \epsilon \in (0,1/2) \).

From this argument, it is revealed that \( \hat{\lambda}(x) \quad E(\hat{\lambda}(x)) \) is asymptotically normally distributed with zero mean and variance \( h^{-1} \lambda(x) \int K^2(x)dx \), and that the asymptotic distribution is estimated consistently by the bootstrap approach.


A.3.1 Property 1

If true failure rate function \( \tilde{r}(t) \) is increasing and continuous on \([a,b]\), then

\[
\lim_{n \to \infty} \sum_{t \in [a,b]} |r(t) - \tilde{r}(t)| = 0, \quad (A.6)
\]
A.3. PROPERTIES OF MARSHALL AND PROSCHAN ESTIMATOR

with probability one, where the Marshall and Proschan estimator $r(t)$ is defined by Eq.(4.15).

A.3.2 Property 2

For the true failure rate function $\tilde{r}(t)$ and any real number $\beta$,

$$
\int_{-\infty}^{\beta} \{r_n(t)\, \tilde{r}(t)\}^2 dF_n(t) \geq \int_{-\infty}^{\beta} \{r(t)\, \tilde{r}(t)\}^2 dF_n(t)
+ \int_{-\infty}^{\beta} \{r_n(t)\, r(t)\}^2 dF_n(t), \quad (A.7)
$$

where

$$
r_n(t) = \begin{cases} 
0, & 0 \leq t < t_1, \\
(n-j)(t_{k+1} - t_k)^{1}, & t_k \leq t < t_{k+1} \ (k = 1, \cdots, n-1), \\
\infty, & t_n < t, 
\end{cases} \quad (A.8)
$$

and $F_n(t)$ represents empirical distribution for $t_i \ (i = 1, 2, \cdots, n)$. From Eq.(A.7), it is revealed that $r(t)$ is closer to $\tilde{r}(t)$ than $r_n(t)$. 
Bibliography


BIBLIOGRAPHY


[84] Data analysis center for software, the software reliability dataset. URL http://www.dacs.dtic.mil/databases/sled/swrel.shtml/.


Publication List of the Author


